

Peter Mueller · Sabu M. Thampi
Md. Zakirul Alam Bhuiyan · Ryan Ko
Robin Doss · Jose M. Alcaraz Calero (Eds.)

Communications in Computer and Information Science

625

Security in Computing and Communications

4th International Symposium, SSCC 2016
Jaipur, India, September 21–24, 2016
Proceedings

Communications in Computer and Information Science

625

Commenced Publication in 2007

Founding and Former Series Editors:

Alfredo Cuzzocrea, Dominik Ślęzak, and Xiaokang Yang

Editorial Board

Simone Diniz Junqueira Barbosa

*Pontifical Catholic University of Rio de Janeiro (PUC-Rio),
Rio de Janeiro, Brazil*

Phoebe Chen

La Trobe University, Melbourne, Australia

Xiaoyong Du

Renmin University of China, Beijing, China

Joaquim Filipe

Polytechnic Institute of Setúbal, Setúbal, Portugal

Orhun Kara

TÜBİTAK BİLGEM and Middle East Technical University, Ankara, Turkey

Igor Kotenko

*St. Petersburg Institute for Informatics and Automation of the Russian
Academy of Sciences, St. Petersburg, Russia*

Ting Liu

Harbin Institute of Technology (HIT), Harbin, China

Krishna M. Sivalingam

Indian Institute of Technology Madras, Chennai, India

Takashi Washio

Osaka University, Osaka, Japan

More information about this series at <http://www.springer.com/series/7899>

Peter Mueller · Sabu M. Thampi
Md. Zakirul Alam Bhuiyan · Ryan Ko
Robin Doss · Jose M. Alcaraz Calero (Eds.)

Security in Computing and Communications

4th International Symposium, SSCC 2016
Jaipur, India, September 21–24, 2016
Proceedings

Editors

Peter Mueller
IBM Zurich Research Laboratory
Rueschlikon
Switzerland

Sabu M. Thampi
Technology and Management
Indian Institute of Information Technology
and Management
Kerala
India

Md. Zakirul Alam Bhuiyan
Temple University
New York, NY
USA

Ryan Ko
Computer Science
University of Waikato
Hamilton
New Zealand

Robin Doss
Deakin University
Burwood, VIC
Australia

Jose M. Alcaraz Calero
University of the West of Scotland
Paisley, Glasgow
UK

ISSN 1865-0929 ISSN 1865-0937 (electronic)
Communications in Computer and Information Science
ISBN 978-981-10-2737-6 ISBN 978-981-10-2738-3 (eBook)
DOI 10.1007/978-981-10-2738-3

Library of Congress Control Number: 2016953330

© Springer Nature Singapore Pte Ltd. 2016

This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.

The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

The publisher, the authors and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, express or implied, with respect to the material contained herein or for any errors or omissions that may have been made.

Printed on acid-free paper

This Springer imprint is published by Springer Nature
The registered company is Springer Nature Singapore Pte Ltd.
The registered company address is: 152 Beach Road, #22-06/08 Gateway East, Singapore 189721, Singapore

Preface

The International Symposium on Security in Computing and Communications (SSCC) aims to provide the most relevant opportunity to bring together researchers and practitioners from both academia and industry to exchange their knowledge and discuss their research findings. The fourth edition, SSCC 2016 was hosted by the LNM Institute of Information Technology (LNMIIT), Jaipur (Rajasthan), India, during September 21–24, 2016. SSCC 2016 was co-located with the First International Conference on Applied Soft Computing and Communication Networks (ACN 2016).

In response to the call for papers, 136 submissions were sent to the symposium. These papers were evaluated on the basis of their significance, novelty, and technical quality. Each paper was reviewed by the members of the Program Committee and finally, 24 regular papers and 16 short papers were accepted.

There is a long list of people who volunteered their time and energy to put together the conference and who warrant acknowledgment. We would like to thank the authors of all the submitted papers, especially the accepted ones, and all the participants who made the symposium a successful event. Thanks to all members of the Technical Program Committee, and the external reviewers, for their hard work in evaluating and discussing papers.

We are grateful to the general chairs and members of the Steering Committee for their support. Our most sincere thanks go to all keynote and tutorial speakers who shared with us their expertise and knowledge. Special thanks to members of the Organizing Committee for their time and effort in organizing the conference.

We wish to express our thanks to Suvira Srivastav, Associate Editorial Director, Computer Science and Publishing Development and Publishing Editor of IAS, Springer India, for her help and cooperation. Finally, we thank Alfred Hofmann and his team at Springer for their excellent support in publishing the proceedings on time.

September 2016

Peter Mueller
Sabu M. Thampi
Md. Zakirul Alam Bhuiyan
Ryan Ko
Robin Doss
Jose M. Alcaraz Calero

Organization

Chief Patron

Lakshmi N. Mittal
(Chairman) LNMIIT, India

Patron

S.S. Gokhale (Director) LNMIIT, India

General Chairs

Peter Mueller IBM Zurich Research Laboratory, Switzerland
Sabu M. Thampi IIITM-Kerala, India

Steering Committee

Ngoc Thanh Nguyen Wroclaw University of Technology, Poland
Janusz Kacprzyk Polish Academy of Sciences, Poland
Sankar Kumar Pal Indian Statistical Institute, India
Hans-Jürgen Zimmermann RWTH Aachen University, Germany
Nikhil R. Pal Indian Statistical Institute, India
Sabu M. Thampi IIITM-K, India
Mario Koeppen Kyushu Institute of Technology, Japan
Michal Wozniak Wroclaw University, Poland
Zoran Bojkovic University of Belgrade, Serbia
Oge Marques Florida Atlantic University (FAU), USA
Ranjan Gangopadhyay LNMIIT Jaipur, India
Nabendu Chaki University of Calcutta, India
Abdenmour El Rhalibi Liverpool John Moores University, UK
Salah Bourennane Ecole Centrale Marseille, France
Selwyn Piramuthu University of Florida, USA
Peter Mueller IBM Zurich Research Laboratory, Switzerland
Robin Doss Deakin University, Australia
Md. Zakirul Alam Bhuiyan Temple University, USA
Axel Sikora University of Applied Sciences Offenburg, Germany
Ryan Ko University of Waikato, New Zealand
Sri Krishnan Ryerson University, Toronto, Canada
El-Sayed El-Alfy King Fahd University of Petroleum and Minerals,
Saudi Arabia

Junichi Suzuki	University of Massachusetts Boston, USA
Parag Kulkarni	iknowlation Research Labs Pvt. Ltd., and EKLaT Research, India
Narsi Bolloju	LNMIIT Jaipur, India
Sakthi Balan	LNMIIT Jaipur, India

Organizing Chairs

Raghuvir Tomar	LNMIIT, India
Ravi Prakash Gorthi	LNMIIT, India

Secretariat

Sandeep Saini	LNMIIT, India
Kusum Lata	LNMIIT, India
Subrat Dash	LNMIIT, India

Event Management Chair

Soumitra Debnath	LNMIIT, India
------------------	---------------

Publicity Co-chair

Santosh Shah	LNMIIT, India
--------------	---------------

Program Chairs

Md. Zakirul Alam Bhuiyan	Temple University, USA
Ryan Ko	University of Waikato, New Zealand
Robin Doss	School of Information Technology, Deakin University, Australia
Jose M. Alcaraz Calero	University of the West of Scotland, UK

Program Committee/Additional Reviewers

Alireza Abdollahpouri	University of Hamburg, Germany
Mohamad Kasim Abdul Jalil	Universiti Teknologi Malaysia (UTM), Malaysia
Mohammad Faiz Liew Abdullah	Universiti Tun Hussein Onn Malaysia (UTHM), Malaysia
Davide Adami	CNIT Pisa Research Unit, University of Pisa, Italy
Rachit Adhvaryu	Gujarat Technological University, India
Sasan Adibi	Deakin University, Australia
Deepak Aeloor	Shivaji University, India

Alok Aggarwal	JPIET, Meerut, India
S. Agrawal	Delhi Technological University (DTU) Formerly Delhi College of Engineering (DCE), India
Musheer Ahmad	Jamia Millia Islamia, New Delhi, India
Rizwan Ahmed	G.H. Rasoni College of Engineering, Nagpur, India
Maurizio Aiello	National Research Council, CNR-IEIIT, Italy
Syed Taqi Ali	NIT Kurukshetra, India
Saed Alrabaee	Concordia University, Canada
Anitha R.	Anna University, India
Ramalingam Anitha	PSG College of Technology, India
Wolfgang Apolinarski	Locoslab GmbH, Germany
Abdullahi Arabo	University of West England, UK
Knarig Arabshian	Hofstra University, USA
Claudio Ardagna	Università degli Studi di Milano, Italy
Yannis Askoxylakis	FORTH-ICS, Greece
Athira U.	IIITM-K, Trivandrum, India
Asrul Izam Azmi	Universiti Teknologi Malaysia, Malaysia
Ramesh Babu	DSCE, Bangalore, India
Amadou Bagayoko	Institut National Polytechnique Toulouse – ENSEEIH, France
Achimuthu Balu	Alagappa University, India
Nikolaos Bardis	Hellenic Military Academy, Greece
Carole Bassil	Lebanese University, Lebanon
Saumya Batham	ABES Engineering College, Ghaziabad, India
Ingmar Baumgart	Karlsruhe Institute of Technology (KIT), Germany
Yosra Ben Mustapha	Allianz, France
Salah Benbrahim	Ecole Polytechnique, Canada
Bruhadeshwar Bezawada	Koneru Lakshmaiah University, India
Pawan Bhandari	Intel, India
Bharat Bhargava	Purdue Unive, USA
Debojyoti Bhattacharya	Robert Bosch Engineering and Business Soluions Limited, India
Tapalina Bhattasali	University of Calcutta, India
Nidhal Bouaynaya	Rowan University, USA
Karima Boudaoud	University of Nice Sophia Antipolis, France
Christos Bouras	University of Patras CTI and P-Diophantus, Greece
Kai Bu	Zhejiang University, P.R. China
John Buford	Avaya Labs Research, USA
Christian Callegari	RaSS National Laboratory, CNIT, Italy
Enrico Cambiaso	National Research Council, CNR-IEIIT, Italy
Zhenfu Cao	Shanghai Jiao Tong University, P.R. China
Xianghui Cao	Southeast University, P.R. China
Xiaolin Chang	Beijing Jiaotong University, P.R. China
Chin-Chen Chang	Feng Chia University, Taiwan
Madhumita Chatterjee	Pillai Institute of Information Technology, India
Santanu Chatterjee	Research Center Imarat, India

Ankit Chaudhary	Truman State University, USA
Himanshu Chaurasiya	Amity School of Engineering and Technology, India
Mahesh Chavan	KIT's College of Engineering, India
Thomas Chen	City University London, UK
Yuanfang Chen	Guangdong University of Petrochemical Technology, P.R. China
Young-Long Chen	National Taichung University of Science and Technology, Taiwan
Xiaofei Cheng	Institute for Infocomm Research, Singapore
Chien-Fu Cheng	Tamkang University, Taiwan
Feng Cheng	University of Potsdam, Germany
Deepak Choudhary	LPU, India
Maxwell Christian	Gujarat Technological University, India
Chung-Hua Chu	National Taichung Institute of Technology, Taiwan
Phan Cong-Vinh	NTT University, Vietnam
Mauro Conti	University of Padua, Italy
Juan Corchado	Universidad de Salamanca, Spain
Joni Da Silva Fraga	UFSC, Brazil
Tasos Dagiuklas	London South Bank University, UK
Anil Dahiya	Manipal University Jaipur, India
Saad Darwish	University of Alexandria, Egypt
Anjana P. Das	College of Engineering, Trivandrum, India
Ashok Kumar Das	International Institute of Information Technology, Hyderabad, India
Pratish Datta	Indian Institute of Technology, Kharagpur, India
Sabrina De Capitani di Vimercati	Università degli Studi di Milano, Italy
Herbe Debar	Télécom SudParis, France
Deepthi P.P.	NIT Calicut, India
Abdelouahid Derhab	King Saud University, Saudi Arabia
Aaradhana Deshmukh	Aalborg University Denmark, India
Dipayan Dev	National Institute of Technology, Silchar, India
Yamuna Devi	VTU, Belagavi, India
Andreas Dewald	ERNW Research GmbH, Germany
Nilanjan Dey	West Bengal University of Technology, India
Vydeki Dharmar	Easwari Engineering College, India
Gianluca Dini	University of Pisa, Italy
Saurabh Dixit	Babu Banarsi Das University, Lucknow, India
Chawki Djeddi	University of Tebessa, Algeria
Nikunj Domadiya	S.V. National Institute of Technology, Surat (Gujarat), India
Jignesh Doshi	L.J. Institute of Management Studies, Ahmedabad, India
Manali Dubal	University of Pune, India
Ha Duyen Trung	Hanoi University of Science and Technology, Vietnam
Amit Dvir	Ariel University Center of Samaria, Israel

El-Sayed El-Alfy	King Fahd University of Petroleum and Minerals (KFUPM), Saudi Arabia
Chun-I Fan	National Sun Yat-sen University, Taiwan
Pingyi Fan	Tsinghua University, P.R. China
Eduardo Fernandez	Florida Atlantic University, USA
Ed' Wilson Ferreira	Federal Institute of Mato Grosso, Brazil
Mathias Fischer	University of Münster, Germany
Simon Fong	University of Macau, Macau, SAR China
Sara Foresti	Università degli Studi di Milano, Italy
Apostolos Fournaris	University of Patras, Greece
Thomas Gamer	ABB AG, Germany
Carlos Gañán	Delft University of Technology, The Netherlands
Dariusz Gasior	Wroclaw University of Technology, Poland
S. Geetha	VIT University, India
Geethakumari G	BITS-Pilani, Hyderabad Campus, India
Vadivelou Gnanapragasam	Pondicherry University, India
Kandaswamy Gokulnath	Anna University, India
Mario Goldenbaum	Princeton University, USA
Paulo Gondim	Universidade de Brasilia, Brazil
Gopinath V.	Sathyabama University, India
R. Goudar	Graphic Era University, India
Rahul Goyal	Amrita School of Engineering, India
Stefanos Gritzalis	University of the Aegean, Greece
Sghaier Guizani	Alfaisal University, Saudi Arabia
Manish Gupta	Hindustan Institute of Technology and Management, Agra, India
Ankur Gupta	Model Institute of Engineering and Technology, India
Hani Hamdan	SUPELEC, France
Ramesh Hansdah	Indian Institute of Science, Bangalore, India
E. Hari Krishna	KU College of Engineering and Technology, India
Harikrishnan K.	Amrita Vishwa Vidyapeetham, India
Houcine Hassan	Universidad Politecnica de Valencia, Spain
Wolfgang Hommel	Universität der Bundeswehr München, Germany
Tzung-Pei Hong	National University of Kaohsiung, Taiwan
Gwo-Jiun Horng	Southern Taiwan University of Science and Technology, Taiwan
Jiankun Hu	University of New South Wales, Australia
Chun-Chieh Huang	Industrial Technology Research Institute, Taiwan
Zequn Huang	University of Delaware, USA
Benoit Hudzia	SAP Research, UK
Mohamed Hussien	Misr University for Science and Technology, Egypt
Luigi Lo Iacono	Cologne University of Applied Sciences, Germany
Alfonso Iacovazzi	Singapore University of Technology and Design, Singapore
Athirai A. Irissappane	Nanyang Technological University, Singapore

Komal Balasubramanian	Sathyabama University, India
Priya Iyer	
Xiaoqi Jia	Institute of Information Engineering, Chinese Academy of Sciences, P.R. China
Jilna P.	National Institute of Technology, Calicut, India
Deveshkumar Jinwala	S.V. National Institute of Technology, India
Shreenivas Jog	Govt. College of Engineering Pune, University of Pune, India
Eduard Jorswieck	TU Dresden, Germany
Manisha Joshi	MGM College of Engineering, India
Jyothish G.	Centre for Computational Engineering and Networking, India
Kamatchi R.	Amity University, Mumbai, India
Georgios Kambourakis	University of the Aegean, Greece
S. Kami Makki	Lamar University, USA
Jayaprakash Kar	King Abdulaziz University, Saudi Arabia
Nirmalya Kar	National Institute of Technology Agartala, India
Sahil Kataria	Govt. Engineering College of Bikaner, India
Sokratis Katsikas	Norwegian University of Science and Technology, Norway
Kuljeet Kaur	TechNivarana Centre for Technical Learning, India
Al-Sakib Khan Pathan	Southeast University, Bangladesh
Muhammad Imran Khan	University of Toulouse, France
Harmeet Khanuja	University of Pune, India
Mahmoud Khasawneh	Concordia University, Canada
Gaurav Khatwani	Indian Institute of Management, Rohtak, India
Donghyun Kim	Kennesaw State University, USA
Kwangjo Kim	Korea Advanced Institute of Science and Technology, Korea
Kirubakaran R.	Kumaraguru College of Technology, India
Ravi Kodali	National Institute of Technology, Warangal, India
Grzegorz Kolaczek	Wroclaw University of Technology, Poland
Nikos Komninos	City University London, UK
Jerzy Konorski	Gdansk University of Technology, Poland
Zbigniew Kotulski	Warsaw University of Technology, Poland
Dimitrios Koukopoulos	University of Patras, Greece
Adrianna Koziarkiewicz-Hetmańska	Wrocław University of Technology, Poland
Deepa Krishnan	Pillai Institute of Information Technology, India
Dilip Krishnaswamy	IBM Research, India
Tarun Kumar	Govt. Engineering College of Bikaner, India
Adarsh Kumar	JIIT, Noida, India
Binod Kumar	JSPM's Jayawant Institute of Computer Applications, Pune, India
Raghvendra Kumar	LNCT Group of College, India
Sanjeev Kumar	University of Texas, RGV, USA

Jeril Kuriakose	St. John College of Engineering, India
Thomas Lagkas	The University of Sheffield International Faculty, CITY College, Greece
Yuan-Cheng Lai	Information Management, NTUST, Taiwan
Costas Lambrinouidakis	University of Piraeus, Greece
Peter Langendoerfer	IHP Microelectronics, Germany
Michael Lauer	Michael Lauer Information Technology, Germany
Ruidong Li	National Institute of Information and Communications Technology (NICT), Japan
Fengjun Li	The University of Kansas, USA
Jia-Chin Lin	National Central University, Taiwan
Yibei Ling	Telcordia Technologies, USA
Anyi Liu	Purdue University Fort Wayne, USA
Guilu Long	Tsinghua University, P.R. China
Pascal Lorenz	University of Haute Alsace, France
Rongxing Lu	Nanyang Technological University, Singapore
Yuan Luo	Shanghai Jiao Tong University, P.R. China
Wenjian Luo	University of Science and Technology of China, P.R. China
S.D. Madhu Kumar	National Institute of Technology Calicut, India
Parikshit Mahalle	Smt. Kashibai Navale College of Engineering, Pune, India
Eman Mahmoodi	Stevens Institute of Technology, USA
Soumyadev Maity	National Institute of Technology, Rourkela, India
Amrita Manjrekar	Shivaji University, India
Marius Marcu	Politehnica University of Timisoara, Romania
Sjouke Mauw	University of Luxembourg, Luxembourg
Chandan Mazumdar	Jadavpur University, India
Michael McGuire	University of Victoria, Canada
Prashant Menghal	MCEME, India
Philippe Merle	Inria Lille – Nord Europe, France
Nader Mir	San Jose State University, USA
Dheerendra Mishra	Indian Institute of Technology, Kharagpur, India
Punit Mittal	ABV-IIITM, India
Jose Molina	Universidad Carlos III de Madrid, Spain
Edward Moreno	UFS – Federal University of Sergipe, Brazil
Saswati Mukherjee	Anna University, India
Skanda Muthaiah	Trimble Navigation, USA
Nagaraja B.	Visvesvaraya Technological University, India
Aravind Nair	Amrita University, India
Ashok Nanda	B.V. Raju Institute of Technology, India
Geetika Narang	SIT College, India
Pratiksha Natani	DIAT, India
Quamar Niyaz	University of Toledo, USA
Andreas Noack	University of Applied Sciences Stralsund, Germany
Kenneth Nwizege	Ken Saro-Wiwa Polytechnic, Bori, Nigeria

Gabriele Oligeri	Third University of Rome, Italy
Alexis Olivereau	CEA, LIST, France
Jose A. Onieva	University of Malaga, Spain
Rolf Oppliger	eSECURITY Technologies, Switzerland
Michele Pagano	University of Pisa, Italy
Anala Pandit	Veermata Jijabai Technological Institute, India
Mauricio Papa	The University of Tulsa, USA
Gianluca Papaleo	National Research Council, CNR-IEIIT, Italy
Alex Pappachen James	Nazarbayev University, Kazakhstan
Stefano Paraboschi	University of Bergamo, Italy
Abhishek Parakh	University of Nebraska at Omaha, USA
Keyurkumar Patel	Australian Defence Force, Australia
Prakashgoud Patil	B.V. Bhoomaraddi College of Engineering and Technology, India
Vaibhav Patil	Shivaji University, India
Thomas Paul	TU Darmstadt, Germany
Vishnu Pandyala	Santa Clara University, USA
Antonio Pescapé	University of Naples Federico II, Italy
Dennis Pfisterer	University of Lübeck, Germany
Simon Pietro Romano	University of Naples Federico II, Italy
Anitha Pillai	Hindustan University, Chennai, India
Vincenzo Piuri	Università degli Studi di Milano, Italy
Geong-Sen Poh	MIMOS, Malaysia
Prabaharan Poornachandran	Amrita Vishwa Vidyapeetham, Amrita University, India
Digambar Povar	BITS Pilani Hyderabad, India
Sudir Prakash	VTU, India
Neeli Prasad	Center for TeleInfrastructure (CTIF), USA
Anand Prasad	NEC Corporation, Japan
Prathap C.	PESIT, Bangalore, India
Praveen K.	Amrita Vishwa Vidyapeetham, India
Guangzhi Qu	Oakland University, USA
Kester Quist-Aphetsi	University of Brest, France
Giuseppe Raffa	Intel Corporation, USA
Rajan A.	Tata Consultancy Services, India
Rajaram S.	TCE Madurai, India
Rajesh J.	VIT University, India
Venkat Raman	JNTU, India
Karthikeyan Ramasamy	Teclever Solutions Pvt. Ltd., India
Tarun Rao	Dayanand Sagar College of Engineering, India
Sherif Rashad	Florida Polytechnic University, USA
Nadana Ravishankar	B.S. Abdur Rahman University, India
Sanjay Rawat	IIIT Hyderabad, India
Slim Rekhis	University of Carthage, Tunisia
Remya Ajai A.S.	Amrita Vishwa Vidyapeetham, India
Eric Renault	Institut Mines-Telecom – Telecom SudParis, France

Vincent Roca	Inria Rhône-Alpes, France
Roberto Rojas-Cessa	New Jersey Institute of Technology, USA
Sohini Roy	Arizona State University, USA
Animesh Roy	Indian Institute of Engineering Science and Technology, Shibpur, India
Diptendu Sinha Roy	National Institute of Science and Technology, Berhampur, India
Giuseppe Ruggeri	University of Reggio Calabria, Italy
Antonio Ruiz-Martínez	University of Murcia, Spain
Jorge Sá Silva	University of Coimbra, Portugal
Amit Sachan	Samsung Research Institute-Noida, India
Navanath Saharia	Gheorghe Asachi Technical University of Iasi, Romania
Youssef Said	Tunisie Telecom, Tunisia
Bhavna Saini	Manipal University Jaipur, India
Samir Saklikar	Cisco, India
Panagiotis Sarigiannidis	University of Western Macedonia, Greece
Himangshu Sarma	University of Bremen, Germany
Mrudula Sarvabhatla	NBKR IST, India
Alka Sawlikar	Nagpur University, India
Rajat Saxena	Indian Institute of Technology, Indore, India
Ulrich Schoen	Self-employed, Germany
Jan Seedorf	HFT Stuttgart, Germany
Debajit Sensarma	University of Calcutta, India
Veerasley Senthil	Thiagarajar School of Management, India
Bartomeu Serra	Universitat de les Illes Balears, Spain
Pritam Shah	JIT, Bangalore, India
Jagruti Shah	Nagpur University, India
Hamid Sharif	University of Nebraska-Lincoln, USA
Shivani Sharma	ABES-EC, Ghaziabad, India
Yatendra Sharma	Arya College of Engineering Technology and Management, India
Anu Sharma	Chandigarh University, India
Amita Sharma	I. I. S. University, India
Aditi Sharma	MBM Engineering College Jodhpur, India
Sandip Shinde	Pune University, India
Preeti Shivach	Gurukul Kangri Vishwavidyalaya Haridwar, India
Rajeev Shrivastava	MPSIDC, India
Piyush Shukla	UIT RGPV, India
Sabrina Sicari	University of Insubria, Italy
Vikas Sihag	Central University of Rajasthan, India
Axel Sikora	University of Applied Sciences Offenburg, Germany
Sindhu M.	Amrita Vishwa Vidyapeetham, India
Rajiv Singh	Banasthali University, India
Kunwar Singh	NIT Trichy, India
Sarbjeet Singh	Panjab University, Chandigarh, India

Somayaji Siva Rama Krishnan	VIT University, India
Nicolas Sklavos	University of Patras, Greece
Vaclav Snasel	VSB-Technical University of Ostrava, FEECS, Czech Republic
Vikas Solanke	MSBTE, India
Karthik Srinivasan	Philips, India
Maicon Stihler	Federal Center for Technological Education – CEFET- MG, Brazil
Dimitrios Stratogiannis	National Technical University of Athens, Greece
Martin Strohmeier	University of Oxford, UK
Ravi Subban	Pondicherry University, Pondicherry, India
Muthukumar Subramanyam	Indian Institute of Information Technology, Tamilnadu, India
Sudarshan T.S.B.	Amrita Vishwa Vidyapeetham University, India
Suganya R.	Thiagarajar College of Engineering, India
Venkatasamy Sureshkumar	PSG College of Technology, India
Pawel Szalachowski	ETH Zurich, Switzerland
Cristina Alcaraz Tello	University of Malaga, Spain
Clement Temaneh-Nyah	University of Namibia, Namibia
Appala Naidu Tentu	CR Rao AIMSCS, India
Deepti Theng	G.H. Rasoni College of Engineering, India
Ciza Thomas	College of Engineering Trivandrum, India
Sandeep Thorat	Shivaji University, India
Padmaja Thurumella	K.L. University, Vaddeswaram, India
Harshvardhan Tiwari	SET, Jain University, India
Orazio Tomarchio	University of Catania, Italy
Zouheir Trabelsi	UAE University, United Arab Emirates
Hardik Upadhyay	GTU, India
Vallikannu R.	Hindustan University, Under UGC Act 3, India
Odelu Vanga	Indian Institute of Information Technology Chittoor, India
Anitha Varghese	ABB Research, India
Varsha Varute	Shivaji University, India
Emmanuel Varvarigos	University of Patras and Computer Technology Institute, Greece
Divya Vidyadharan	College of Engineering, Trivandrum, India
S Vijaykumar	6TH SENSE, An Advanced Research and Scientific Experiment Foundation, India
Vikram Raju R.	Manipal University Jaipur, India
Javier García Villalba	Universidad Complutense de Madrid, Spain
Vinod Chandra S.S.	University of Kerala, India
Chandra Vorugunti	Indian Institute of Information Technology, SriCity, India
Yong Wang	Dakota State University, USA
Qi Wang	University of the West of Scotland, UK

Shah Jahan Wani	University of Kashmir, India
Mohammad Wazid	IIIT, Hyderabad, India
Chih-Yu Wen	National Chung Hsing University, Taiwan
Andreas Wespi	IBM Zurich Research Laboratory, Switzerland
Carlos Becker Westphall	Federal University of Santa Catarina, Brazil
Setyawan Widyarto	Universiti Selangor, Malaysia
Marcus Wong	Huawei Technologies, USA
Bing Wu	Fayetteville State University, USA
Jun Wu	Shanghai Jiao Tong University, P.R. China
Liang Xiao	Xiamen University, P.R. China
Jiping Xiong	Zhejiang Normal University, P.R. China
Yurong Xu	Dartmouth College, USA
Kunjie Xu	Ericsson R&D, Silicon Valley, USA
Akihiro Yamamura	Akita University, Japan
Ping Yang	Binghamton University, USA
Chang Wu Yu	Chung Hua University, Taiwan
Meng Yu	University of Texas at San Antonio, USA
Dmitry Zaitsev	International Humanitarian University, Ukraine
Tao Zhang	Chinese University of Hong Kong, SAR China
Jianhong Zhang	North China University of Technology, P.R. China
Wuxiong Zhang	Shanghai Research Center for Wireless Communications, P.R. China
Haijun Zhang	The University of British Columbia, Canada
Mohamed Faten Zhani	École de Technologie Supérieure, University of Quebec, Canada
Kai Zheng, 2012 Labs	Huawei Technologies, P.R. China
Liang Zhou	Nanjing University of Posts and Telecommunications, P.R. China
Bo Zhu	Google Inc., Canada

Contents

Cryptosystems, Algorithms, Primitives

Computing Mod with a Variable Lookup Table	3
<i>Mark A. Will and Ryan K.L. Ko</i>	
Mutual Authentication Based on HECC for RFID Implant Systems.	18
<i>Asha Liza John and Sabu M. Thampi</i>	
On the Use of Asynchronous Cellular Automata in Symmetric-Key Cryptography	30
<i>Biswanath Sethi and Sukanta Das</i>	
A Random Key Generation Scheme Using Primitive Polynomials over GF(2).	42
<i>Inderjeet Singh and Alwyn R. Pais</i>	
Multi-factor Authentication Using Recursive XOR-Based Visual Cryptography in Online Voting System	52
<i>P. Sanyasi Naidu and Reena Kharat</i>	
Enhanced Image Based Authentication with Secure Key Exchange Mechanism Using ECC in Cloud	63
<i>Anurag Singh Tomar, Shashi Kant Shankar, Manmohan Sharma, and Aditya Bakshi</i>	
Differential Fault Analysis on Tiaoxin and AEGIS Family of Ciphers	74
<i>Prakash Dey, Raghvendra Singh Rohit, Santanu Sarkar, and Avishek Adhikari</i>	
A Comparison of Diffusion Properties of Salsa, ChaCha, and MCC Core. . . .	87
<i>Rajeev Sobti and G. Geetha</i>	
A Secure Keyword Ordered Multiuser Searchable Encryption Framework . . .	99
<i>Kulvaibhav Kaushik and Vijayaraghavan Varadharajan</i>	
Cryptographic Assessment of SSL/TLS Servers Popular in India.	112
<i>Prakhar Jain and K.K. Shukla</i>	
Key Identifications Using Hebbian Learning.	124
<i>Bhavya Ishaan Murmu, Anu Kumari, Manu Malkani, and Sanjeet Kumar</i>	

Security and Privacy in Networked Systems

An Automated Methodology for Secured User Allocation in Cloud 137
Srijita Basu, Anirban Sengupta, and Chandan Mazumdar

Provenance-Aware NoSQL Databases 152
Anu Mary Chacko, Munavar Fairooz, and S.D. Madhu Kumar

Efficient Key Management in IoT Using Mobile Aggregator 161
Sumit Saurabh, Alwyn R. Pais, and Sumanta Chatterjee

Cloud Resources Optimization for Air Pollution Monitoring Devices
 and Avoiding Post Pillar Problem 173
Parampreet Singh and Pankaj Deep Kaur

Credibility Assessment of Public Pages over Facebook 188
Himanshi Agrawal and Rishabh Kaushal

Elliptic Curve Based Secure Outsourced Computation in Multi-party
 Cloud Environment 199
V. Thangam and K. Chandrasekaran

Secure and Privacy Preserving Mobile Healthcare Data Exchange
 Using Cloud Service 213
Doyal Pal, Gobinda Senchury, and Praveenkumar Khethavath

Secure Certificateless Signature Scheme with Batch Verification from
 Bilinear Pairings 225
N.B. Gayathri and P. Vasudeva Reddy

System and Network Security

Security Requirements Elicitation and Modeling Authorizations 239
Rajat Goel, Mahesh Chandra Govil, and Girdhari Singh

Two Level Signature Based Authorization Model for Secure
 Data Warehouse 251
Anjana Gosain and Amar Arora

Nonlinear Tracking of Target Submarine Using Extended Kalman
 Filter (EKF) 258
S. Vikranth, P. Sudheesh, and M. Jayakumar

Tracking Inbound Enemy Missile for Interception from Target Aircraft
 Using Extended Kalman Filter 269
T.S. Gokkul Nath, P. Sudheesh, and M. Jayakumar

Steganography/Visual Cryptography/Image Forensics

A Secure One-Time Password Authentication Scheme Using Image Texture Features 283
Maitreya Maity, Dhiraj Manohar Dhane, Tushar Mungle, Rupak Chakraborty, Vasant Deokamble, and Chandan Chakraborty

Analyzing the Applicability of Bitsum Algorithm on LSB Steganography Technique 295
Bagga Amandeep and G. Geetha

Extreme Learning Machine for Semi-blind Grayscale Image Watermarking in DWT Domain. 305
Ankit Rajpal, Anurag Mishra, and Rajni Bala

A Passive Blind Approach for Image Splicing Detection Based on DWT and LBP Histograms 318
Mandeep Kaur and Savita Gupta

An Image Forensic Technique for Detection of Copy-Move Forgery in Digital Image 328
Ashwini Malviya and Siddharth Ladhake

Secure Authentication in Online Voting System Using Multiple Image Secret Sharing 336
P. Sanyasi Naidu and Reena Kharat

Applications Security

Touch and Track: An Anti-theft and Data Protection Technique for Smartphones. 347
Sohini Roy, Arvind Kumar Shah, and Uma Bhattacharya

Enhancement of Detecting Wicked Website Through Intelligent Methods. 358
Tarik A. Rashid and Salwa O. Mohamad

Prediction of Malicious Domains Using Smith Waterman Algorithm 369
B. Ashwini, Vijay Krishna Menon, and K.P. Soman

Outsource-Secured Calculation of Closest Pair of Points 377
Chandrasekhar Kuruba, Kethzi Gilbert, Prabhav Sidhaye, Gaurav Pareek, and Purushothama Byrapura Rangappa

Discovering Vulnerable Functions: A Code Similarity Based Approach 390
Aditya Chandran, Lokesh Jain, Sanjay Rawat, and Kannan Srinathan

Performance Analysis of Spectrum Sensing Algorithm Using Multiple Antenna in Cognitive Radio 403
Komal Pawar and Tanuja Dhope

Diagnosis of Multiple Stuck-at Faults Using Fault Element Graph with Reduced Power 414
T.S. Gokkul Nath, E.R. Midhila, Ashwin Swaminathan, Binitaa Lekshmi, and J.P. Anita

Intrusion Detection Using Improved Decision Tree Algorithm with Binary and Quad Split 427
Shubha Puthran and Ketan Shah

MalJs: Lexical, Structural and Behavioral Analysis of Malicious JavaScripts Using Ensemble Classifier 439
Surendran K., Prabakaran Poornachandran, Aravind Ashok Nair, Srinath N., Ysudhir Kumar, and Hrudya P.

SocialBot: Behavioral Analysis and Detection. 450
Madhuri Dewangan and Rishabh Kaushal

Vulnebdroid: Automated Vulnerability Score Calculator for Android Applications 461
Sugandha Gupta and Rishabh Kaushal

Author Index 473

Cryptosystems, Algorithms, Primitives

Computing Mod with a Variable Lookup Table

Mark A. Will^(✉) and Ryan K.L. Ko

Cyber Security Lab, The University of Waikato, Hamilton, New Zealand
{willm,ryan}@waikato.ac.nz

Abstract. Encryption algorithms are designed to be difficult to break without knowledge of the secrets or keys. To achieve this, the algorithms require the keys to be large, with some having a recommend size of 2048-bits or more. However most modern processors only support computation on 64-bits at a time. Therefore standard operations with large numbers are more complicated to implement. One operation that is particularly challenging to efficiently implement is modular reduction. In this paper we propose a highly-efficient algorithm for solving large modulo operations; it has several advantages over current approaches as it supports the use of a variable sized lookup table, has good spatial and temporal locality allowing data to be streamed, and only requires basic processor instructions. Our proposed algorithm is theoretically compared to widely used modular algorithms, and shows improvements over other algorithms using predefined lookup tables.

1 Introduction

Modular reduction, also known as the modulo or mod operation, is a value within Y , such that it is the remainder after Euclidean division of X by Y . This operation is heavily used in encryption algorithms [1–3], since it can “hide” values within large prime numbers, often called keys. Encryption algorithms also make use of the modulo operation because it involves more computation when compared to other operations like add. Meaning that computing the modulo operation is not as simple as just adding bits.

Modern Intel processors have implemented the modulo operation in the form of a single division instruction [4]. This instruction will return both the quotient and remainder in different registers. Therefore the modulo operation can be computed in a single clock cycle, even though the nature of the operation requires more computation than add. However this instruction only supports values up to 64-bits, and recommended key sizes for encryption algorithms can be higher than 2048-bits [5]. This is why it is challenging to compute the modulo operation, because processors cannot directly support these large numbers.

We propose an algorithm for modular reduction which has been designed so that it is hardware friendly. It only requires simple operations such as addition and subtraction, unlike some other solutions which use multiplication or division [6]. This allows for less costly custom hardware implementations in terms of area and power. While also making better use of the register inside processors.

Our proposed algorithm only requires chunks of data at once, starting at the uppermost bits. Meaning the data can be streamed into a processor or custom hardware. This gives it good spatial and temporal locality, and reduces cache misses. Another useful property is that it guarantees the amount of bits required for computation will that be of the modulo value Y . While allowing for an arbitrary bit length of the input value X , unlike the algorithm shown in [7]. This allows implementations to have fixed bus sizes, giving better performance.

Our proposed algorithm also makes use of a precomputed lookup table, which supports variable sizes. Therefore allowing it to be customised for its application, and so that it fits inside cache. This lookup table also has the unique property of being able to be used for an arbitrary number of bits, regardless of the lookup tables key size.

Existing algorithms will be described in Sect. 2, including the two most popular algorithms, Barrett's reduction and Montgomery's reduction. Then in Sect. 3.1 we will describe our proposed algorithm in detail, giving the theorems behind the algorithm, and their respective proofs. Implementation techniques and remarks are given in Sect. 4, including the use of lookup tables. Before comparisons and concluding marks are given in Sects. 5 and 6 respectively.

2 Previously Known Techniques

This section will cover the popular and related modulus algorithms. The first two reduction techniques, Barrett and Montgomery are the most commonly used. A less common technique, the Diminished Radix Algorithm is mentioned, before a newly proposed algorithm is also described.

2.1 The Barrett Reduction

The general idea behind the Barrett Reduction [6, 8] is

$$X \bmod Y \equiv X - \lfloor X/Y \rfloor Y$$

where X is divided by Y , floored (i.e. remove decimal places), and multiplied by Y . This value gives the closest multiple of Y to X , therefore we minus this value from X to give the modulus.

The actual equation proposed by Barrett [6] is modified so that

$$X \bmod Y \equiv X - \left\lfloor \frac{X}{\frac{b^{n-1}}{b^{n+1}}} u \right\rfloor Y$$

where $u = \left\lfloor \frac{b^{2n}}{Y} \right\rfloor$. This assumes that divisions by a power of b are computationally cheap, and that u is pre-computed. Therefore this is less expensive than the general equation. It can be improved further so that partially multi-precision multiplication are used when needed, which gives

$$X \bmod Y \approx (X \bmod b^{n+1} - (m\hat{q} \bmod b^{n+1})) \bmod b^{n+1}$$

where \hat{q} is an estimate of X/Y , resulting in the result being an estimate itself. Therefore a few subtractions of Y from the result can be required to give the correct modulo value.

2.2 The Montgomery Reduction

Montgomery Reduction [9] is a common algorithm used for modulus reduction. The unique property of this algorithm is that it does not compute the modulus directly, but instead the modulus multiplied by a constant. An overview of the algorithm is shown in Algorithm 1 [10], where Y is odd, X is limited to $0 \leq X < Y^2$, and K is the number of bits.

Algorithm 1. Reduction

Compute: $2^{-K}X \bmod Y$
 1: $x = X$
 2: **for** $k = 1$ to K **do**
 3: **if** x is odd **then**
 4: $x = x + Y$
 5: $x = x/2$
return x

Algorithm 2. Modification

Compute: $2^{-K}X \bmod Y$
 1: $x = X$
 2: **for** $k = 1$ to K **do**
 3: **if** the k^{th} bit is high **then**
 4: $x = x + 2^k Y$
 return $x/2^K$

Algorithm 1 is a simple interpretation of the Montgomery Reduction, with further improvements required to match the performance of the Barrett Reduction. This is because currently the algorithm requires $2K^2$ single precision shifts and K^2 additions [10]. A small improvement is shown in Algorithm 2, where K must now satisfy the condition $2^K > Y$.

Now the number of single precision shifts has been reduced to $K^2 + K$, however the divide in the return statement can be done in one shift. This results in K^2 single precision shifts, and 1 right shift of size K .

There are other modifications of the Montgomery Reduction [11, 12] for both software and hardware. And due to the number of variations, in this paper we will only compare our proposed algorithm against the two Montgomery Reductions algorithms, as shown in Algorithms 1 and 2.

2.3 The Diminished Radix Algorithm

The Diminished Radix Algorithm [13] has been designed so that certain moduli offer performance gains [10]. Because of this, it is not a universally applicable algorithm, so it will not be compared against in this paper.

2.4 Fast Modular Reduction Method

A proposed algorithm [7] by Cao et al. for modular reduction, uses a fixed size lookup table similar to precomputed tables described in [14]. Even though [7] is not one of the main algorithms for modular reduction, it will be described

in detail in this paper. This is because it shares some of the core theorems with our proposed algorithm in Sect. 3.1. Also [7] represents the common way of implementing a lookup table for modular reduction, and will help show that our proposed algorithm makes better use of a lookup table.

Cao et al. describe two algorithms, the second being an improvement over the first. Therefore we will discuss the first algorithm as shown in Algorithm 3, and just make note that the second algorithm is better for the worst-case situation.

Algorithm 3. Fast Modular Reduction Method

<p>Compute: $X \bmod Y$ $k = \text{bitlength}(Y)$ $r(\alpha) = 2^\alpha \bmod Y$ $\mathbb{T} = \{r(2k-1), r(2k-2), \dots, r(k)\}$</p> <p>1: if $X < Y$ then 2: return X 3: if $\text{bitlength}(X) = k$ then 4: return $X - Y$ 5: $s = \text{binary}(X)$</p>	<p>6: $m = 0$ 7: for $i = \text{bitlength}(X) - 1$ downto k do 8: if $s[i]$ <i>is high</i> then 9: $m = m + \mathbb{T}[i - k]$ 10: $m = m + \sum_{j=0}^{k-1} s[j]2^j$ 11: while $m \geq Y$ do 12: $m = m - Y$ 13: while $m < 0$ do 14: $m = m + Y$ return m</p>
--	---

Algorithm 3 uses a precomputed lookup table \mathbb{T} , which contains the modulus answer for each bit from index k to $2k - 1$. Then it loops through each bit index of X at or above k and adds the value from the lookup table. Once the loop is complete, it then adds the bits from index 0 to $k - 1$ of X onto the result m . This is the main reduction, but further reductions can be required, hence the last two while loops.

3 Proposed Algorithm

3.1 Overview

The algorithm we propose for calculating the modulus is shown in Algorithm 4, which computes $X \bmod Y$. The functions are shown before the pseudo code, where α for example denotes the parameter into the function. Lower case variables with a subscript represent bits at an index in the upper case variable, as shown by the definition of T . Finally $\hat{}$ denotes the variable is an array or set. These notations will be the same for the theorems and proofs in Sect. 3.3.

The bit shifting operation is the key feature of our algorithm, as it allows the use of a single precomputed value to find the modulus of all bits above the bit width of Y , unlike other solutions. We can also use multiple precomputed values in the form of a lookup table and use a larger bit shift to improve performance, which is described in Sect. 4.1. Where the algorithms in [7] require a precomputed value for each bit above the bit width of Y . This can be very costly as the precomputed values have a fixed size, where our algorithm can vary the number

Algorithm 4. Proposed Algorithm

<p>Compute: $X \bmod Y$</p> <p>$Width(\alpha) = \text{width of } \alpha \text{ in terms of bits}$</p> <p>$Split(\alpha, w) = \left\{ \sum_{i=0}^{w-1} \alpha_i 2^i, \dots, \sum_{i=0}^{w-1} \alpha_{Width(\alpha)-w+i} 2^i \right\}$</p> <p>$Num(\hat{\alpha}) = \text{number of elements in } \hat{\alpha}$</p> <p>$T = \sum_{i=0}^{Width(T)-1} t_i 2^i, t_i \in \{0, 1\}$</p> <p>1: $\hat{G} = Split(X, Width(Y))$</p> <p>2: $N = Num(\hat{G}) - 1$</p> <p>3: while $N > 0$ do</p> <p>4: $T = \hat{G}[N]$</p> <p>5: for $i = Width(Y) - 1$ downto 0</p> <p style="padding-left: 2em;">do</p>	<p>6: $T = T \ll 1$</p> <p>7: while $t_{Width(Y)} = 1$ do</p> <p>8: $t_{Width(Y)} = 0$</p> <p>9: $T = T + (2^{Width(Y)} \bmod Y)$</p> <p>10: $\hat{G}[N-1] = \hat{G}[N-1] + T$</p> <p>11: while $\hat{G}[N-1]_{Width(Y)} = 1$ do</p> <p>12: $\hat{G}[N-1]_{Width(Y)} = 0$</p> <p>13: $\hat{G}[N-1] = \hat{G}[N-1] + (2^{Width(Y)} \bmod Y)$</p> <p>14: $N = N - 1$</p> <p>15: while $\hat{G}[0] > Y$ do</p> <p>16: $\hat{G}[0] = \hat{G}[0] - Y$</p> <p style="padding-left: 2em;">return $\hat{G}[0]$</p>
---	--

of precomputed values. The other key property of our algorithm is that it only reads data of X once (reads an element in \hat{G} once), starting from the uppermost bits and working down to the 0^{th} bit. This gives our algorithm excellent spatial and temporal locality. In terms of a hardware implementation, it also means the data can be streamed from memory. Then because only a fixed amount of data is read and computed at a time, custom hardware can be faster, have better area usage, and be more power efficient. This guaranteed data width is another property that other solutions cannot offer easily.

3.2 Description

Before the algorithm is described in further detail, first the functions must be explained.

- **Width:** Given the parameter α , this function will return the minimum number of bits in α . Put simply, it results the index of the uppermost high bit, plus one. For example if 13 were inputted, the result would be 4.
- **Split:** The first parameter α is the value to be split, and the second w is a bit width value. This function splits α into a vector, so that each element is bit width w . If α cannot be split up evenly (i.e. the bit width of α is not a multiple of w), then α can be padded with zero bits. For example splitting 35 into a vector with an element bit width of 4, results in $\{3, 2\}$. In terms of binary values, $35 = 100011_2$, so the result is $\{0011_2, 0010_2\}$ (note the padded zeros at index 1).
- **Num:** Number of elements in the vector $\hat{\alpha}$ after the split function.

The first line of Algorithm 4 splits X into a vector \hat{G} so that the elements have the same width as Y . Then we set N to the uppermost index of \hat{G} , because we will process the elements in reverse. Once we enter the loop, we will set T to element N for ease of understanding the algorithm.

The For loop will count from 0, to the number of bits in Y . Each iteration we shift T left by one (i.e. double T). Then if an overflow occurs, meaning that the width of T is no longer equal to that of Y (i.e. the bit at index $\text{Width}(Y)$ is high), we clear this overflow bit, and add $2^{\text{Width}(Y)} \bmod Y$ to T . This value we add should be precomputed so that it is already in modulo Y , therefore it costs just one addition operation. The reason we do this is because if an overflow occurs, we are guaranteed that T is greater than Y . So we add the modulo value of this overflow bit back to the answer, which keeps T the same bit width as Y . Therefore this means that T is, or is close to the correct value. When adding $2^{\text{Width}(Y)} \bmod Y$ to T , another overflow may occur, which requires us to repeat line 7 until no overflow occurs.

Note that clearing the overflow bit could also be achieved by subtracting Y from T . But depending on implementation, it is possible that multiple subtractions of Y are required to clear the overflow bit. Also subtraction would not allow the use of a lookup table which is described in Sect. 4.1.

Once the we have finished the For loop, we have therefore performed $\text{Width}(Y)$ number of shifts on T . Then we add T to the next element in \hat{G} . Whenever we perform an add, it is possible for an overflow to occur, so we have to include the loop on Line 11 to deal with this when adding T .

When we reach element 0, which are the lowermost bits of X , so we know that we are already close to the correct result, meaning we can stop the loop. Then some subtractions could be required before the correct result is required (depending on implementation). However implementing the algorithm as is, at most only one subtraction would be required.

We are now left with the correct answer in element 0, which can be returned. In order to prove that this algorithm will produce the correct modulus answer, we must now discuss the theorems and proofs that the algorithm is based upon.

3.3 Theorems and Proofs

The algorithms main operation is to double the value of T , and by doing this, we are also doubling the modulus. This is shown in Theorem 1. Note that it is important to find the modulus of $2M$ because by doubling M , the result could become greater than Y .

Theorem 1. *If $X \bmod Y = M$, then $2X \bmod Y = 2M \bmod Y$*

Proof. Given $X \bmod Y = M$ where $X, Y \in \mathbb{Z}$ and $M \in \mathbb{Z}_0^Y$

$$\frac{X}{Y} = Q + M \quad (Q = \text{Quotient})$$

$$\therefore X \bmod Y \equiv M \bmod Y$$

$$\frac{2X}{Y} = 2(Q + M)$$

$$\frac{2X}{Y} = 2Q + 2M$$

$$\therefore 2X \bmod Y \equiv 2M \bmod Y$$

Because an integer represented in binary is made up of powers of 2s, it is said that the number is the sum of power of 2s. Therefore if we take the modulus of each power of 2 and sum them, it is equivalent to summing the power of 2s then taking the modulus, as shown in Theorem 2.

Theorem 2. *Given an integer X , which is the sum of power of 2s, then the sum of the modulus's of the power of 2s in Y , is equivalent to the modulus of X in Y .*

Proof. Given $X \bmod Y$ where $X, Y \in \mathbb{Z}$

$$\begin{aligned}
 X &= \sum_{i=0}^{n-1} x_i 2^i \quad (\text{where } x_i \in \{0, 1\}) \\
 \frac{x_i 2^i}{Y} &= Q_i + M_i \quad (\text{where } Q_i, M_i \in \mathbb{Z}) \\
 x_i 2^i \bmod Y &\equiv M_i \bmod Y \\
 \therefore X \bmod Y &\equiv \sum_{i=0}^{n-1} M_i \bmod Y
 \end{aligned}$$

Theorem 3 in the general context of the algorithm is the same as Theorem 1. However we require Theorem 3 because our proposed algorithm can use a lookup table (described in Sect. 4.1), which Theorem 1 does not make clear.

Theorem 3. *Given an integer X , which is to be shifted bit left by w , the modulus of X bit shifted by w is equivalent to finding the modulus of X after being bit shifted.*

Proof. Given $(X \ll w) \bmod Y = M$ where $X, Y, M, w \in \mathbb{Z}$

Because left bit shifting X by w is equivalent to doubling X w times, we can use Theorem 1 to prove that:

$$((X \bmod Y) \ll w) \bmod Y \equiv (X \ll w) \bmod Y \equiv M$$

The first 3 theorems are the basic underlying operations used in our algorithm. Now we need to prove the fundamental idea behind the algorithm. Theorem 4 proves the initial step of the algorithm, splitting up X into elements of an array/vector so that they can be concatenated together to give X . Then when finding the modulus of X , we can sum the modulus of each element (with bit shifting).

Theorem 4. *When finding $X \bmod Y$, if X is larger than Y in terms of number of bits, the modulus is equivalent to dividing X into elements of the same bit size as Y , follow by the bit shifting and summing the modulus values of each element.*

Proof. Given $X \bmod Y = M$ where $X, Y, M \in \mathbb{Z}$

$$\begin{aligned}
 Y &= \sum_{i=0}^{k-1} y_i 2^i \quad (\text{where } y_i \in \{0, 1\} \text{ and } k > 0) \\
 X &= \sum_{i=0}^{n-1} x_i 2^i \quad (\text{where } x_i \in \{0, 1\} \text{ and } n \bmod k = 0) \\
 \hat{X} &= \left\{ \left(\sum_{i=n-k}^{n-1} x_i 2^i \gg (n-k) \right), \dots, \left(\sum_{i=0}^{k-1} x_i 2^i \right) \right\} \\
 Y &= \sum_{i=0}^{k-1} y_i 2^i \quad (\text{where } y_i \in \{0, 1\} \text{ and } k > 0)
 \end{aligned}$$

$$\begin{aligned}
 X &\equiv \prod_{i=(n/k)-1}^0 \hat{X}_i \\
 &\equiv \sum_{i=0}^{(n/k)-1} (\hat{X}_i \ll ki)
 \end{aligned}$$

Because the concatenation of \hat{X} is equivalent to X . Then by shifting each item in \hat{X} to the correct bit position and summing, is equivalent to X . Therefore we can use Theorems 2 and 3 so that

$$\sum_{i=0}^{(n/k)-1} (\hat{X}_i \ll ki) \bmod Y \equiv X \bmod Y$$

Note: if $n \bmod k \neq 0$ then X can be padded with 0s until $n \bmod k = 0$. \square

Theorem 5 proves that a single precomputed value (or a lookup table), can be used to help find the modulus in each element greater than 0. This is an important property of the algorithm, as it allows the use of more precomputed values in the form of a lookup table.

Theorem 5. *When calculating the modulus of \hat{X}_i in Y where $i > 0$, the modulus is equivalent to multiplying \hat{X}_i by $(2^k \bmod Y \ll k(i-1))$.*

Proof.

$$\begin{aligned}
 &\hat{X}_i \ll ki \bmod Y \\
 &\equiv \hat{X}_i 2^{ki} \bmod Y \\
 &\equiv (\hat{X}_i \bmod Y) \times (2^{ki} \bmod Y) \\
 &\equiv (\hat{X}_i \bmod Y) \times (2^k \ll k(i-1) \bmod Y)
 \end{aligned}$$

Instead of multiplying (as in Theorem 5), we instead shift each individual bit in an element and add if an overflow occurs. These are proven to be equivalent in Theorem 6.

Theorem 6. *Left shifting \hat{X}_i by ki in modulo Y where $i > 0$, is equivalent to shifting \hat{X}_i one shift at a time. Then if the k th bit becomes high after any shift (or add operation), drop it, and add $2^k \bmod Y$ to \hat{X}_i . Therefore keeping the bit width of \hat{X}_i constant, while remaining in modulo Y .*

Proof.

$$\begin{aligned} & \hat{X}_i \times 2^{ki} \bmod Y \\ & \equiv \hat{X}_i \ll k \bmod Y \\ & \equiv (((\hat{X}_i \ll 1 \bmod Y) \dots) \ll 1 \bmod Y) \end{aligned}$$

After each shift we put \hat{X} back into modulo Y , and because \hat{X} has k bits, if the k th bit becomes high (i.e. overflow), we know that \hat{X} is definitely bigger than Y . Then by using Theorems 4 and 5, we can say that \hat{X} is equivalent in modulo Y to adding $2^k \bmod Y$ to \hat{X} (after dropping the k th bit).

Theorem 6 is the main theorem for this algorithms operation (i.e. producing the correct answer). However for implementation, it would suffer from performance issues. This is due to the amount of shifting and potential adding which we would need to perform. So instead of shifting an element to its correct position directly, we can just shift it by the number of bits in Y , then start shifting the next element as well. This is shown in Theorem 7, and in terms of performance of the algorithm, is the most important theorem.

Theorem 7. *If we start calculating the modulus of X in Y at $\hat{X}_{(n/k)-1}$, then instead of performing $k(((n/k) - 1) - 1)$ number of shifts, we can perform k shifts, then add $\hat{X}_{(n/k)-1}$ to $\hat{X}_{(n/k)-2}$. Repeating until we reach \hat{X}_0 which will contain the result.*

Proof.

$$\begin{aligned} & (\hat{X}_{(n/k)-1} \ll k((n/k) - 2)) + (\hat{X}_{(n/k)-2} \ll k((n/k) - 3)) \\ & \equiv (\hat{X}_{(n/k)-1} \ll k \ll k((n/k) - 3)) + (\hat{X}_{(n/k)-2} \ll k((n/k) - 3)) \\ & \equiv ((\hat{X}_{(n/k)-1} \ll k) + (\hat{X}_{(n/k)-2})) \ll k((n/k) - 3) \end{aligned}$$

3.4 Example

Below is an example on how to solve $1620 \bmod 11$ on a 4-bit processor using our proposed algorithm.

$$\begin{aligned} 1620 &= 011001010100_2 \\ 11 &= 1011_2 \\ \hat{G} &= \{0110_2, 0101_2, 0100_2\} \end{aligned}$$

$$\begin{array}{llll}
 T = \hat{G}[2] & T = 1010_2 + 0101_2 & T = T \ll 1 & T = 0000_2 + 0101_2 \\
 = 0110_2 & = 1111_2 & = 1\ 1010_2 & = 0101_2 \\
 T = T \ll 1 & T = T \ll 1 & T = 1010_2 + 0101_2 & T = T \ll 1 \\
 = 1100_2 & = 1\ 1110_2 & = 1111_2 & = 1010_2 \\
 T = T \ll 1 & T = 1110_2 + 0101_2 & T = T \ll 1 & T = T + \hat{G}[0] \\
 = 1\ 1000_2 & = 1\ 0011_2 & = 1\ 1110_2 & = 1010_2 + 0100_2 \\
 T = 1000_2 + & = 0011_2 + 0101_2 & T = 1110_2 + 0101_2 & = 1110_2 \\
 (10000_2 \bmod 1011_2) & = 1000_2 & = 1\ 0011_2 & T = T - 11 \\
 = 1000_2 + 0101_2 & T = T + \hat{G}[1] & T = 0011_2 + 0101_2 & = 1110_2 - 1011_2 \\
 = 1101_2 & = 1000_2 + 0101_2 & = 1000_2 & = 0011_2 \\
 T = T \ll 1 & = 1101_2 & T = T \ll 1 & \\
 = 1\ 1010_2 & & = 1\ 0000_2 & \therefore \mathbf{1620 \bmod 11 = 3}
 \end{array}$$

4 Theoretical Implementation

4.1 Lookup Table

A useful property of our algorithm is that it allows for the use of a variable sized lookup table. Lookup tables may not be suitable for all applications or functions, such as key generation, because the overhead in creating the table could be too expensive. However for applications where large amounts of data must be computed with the same modulo value, such as smart-cards or secure tunnels, there is a performance gain. The algorithm in its most basic form already uses a lookup table. For each shift, if an overflow occurs, we add a precomputed value, else we add nothing, giving a simple lookup table as shown in Table 1.

This can be extended to look at more bits at a time. For example, if we were to look at 2-bits, the lookup table would be that of Table 2. Allowing us to shift by two bits at a time (instead of a single shift), but we still only require a single add. However this makes a software implementation slightly more difficult, because we cannot use the overflow bit anymore. Therefore we must look at the upper 2-bits of T before shifting. The current software implementation uses an and operation then a shift to get these upper bits. On a 64-bit processor, a lookup table with a key size of up to 64-bits, only requires the and shift operations to be executed on a single 64-bit register, regardless of the size of Y . To possibly improve performance even more, one approach that could be explored is reversing

Table 1. Precomputed Lookup Table for 1-bit

Key	Value
0	0
1	$2^k \bmod Y$

Table 2. Precomputed Lookup Table for 2-bits

Key	Value
00	0
01	$2^k \bmod Y$
10	$2^{k+1} \bmod Y$
11	$(2^k + 2^{k+1}) \bmod Y$

the bits in each element (and keys), meaning only an and operation would be required.

Given that the lookup table is of variable size, it is important to make the whole table fit into the processors cache. This makes the lookup time require significantly less clock cycles than fetching from main memory, and thus improving performance. The size of Y has the biggest impact on the size of the lookup table. For example, if Y is a 2048-bit value, then each item in the table requires 2048-bits, plus the number of bits for each key.

4.2 Memory Access

This algorithm has been designed in such a way that memory access has been kept to a minimum. The lookup table can be configured so that it fits in cache. Once loaded, memory access is only required for X . Since the algorithm only works on segments of X at a time, it can load an element of X , process it, then load the next element. Therefore reducing the number of fetches (also cache misses), and improving performance. Each element loaded is the same number of bits as Y , meaning if Y is 2048-bits, then the processor only needs to load 2048-bits at a time. This allows the data to be streamed into the processor, which is also important for a custom hardware implementation.

5 Comparisons

We theoretically compared our algorithm to the algorithms presented in Sect. 2, and have not compared it to previously implemented designs or techniques of those algorithms.

5.1 Comparisons with Barrett Reduction

Comparing between our proposed algorithm and the Barrett Reduction is very difficult in theory. This is because the Barrett Reduction only seems to use a few operations. However when looking at implementation, it is the nature of these operations which cause performance issues. This is because when working with large numbers, for example 2048-bits, a 64-bit processor cannot simply execution a single instruction. Instead it must execution many instructions to compute the result.

Addition operations on large numbers are straight forward assuming the instruction set supports a carry bit. The words of the two input values can just added together. For example on Intel x86 processors, the *adc* instruction can be used to add two words together, plus add a carry bit if an overflow occurred on the previous *add* or *adc*. Therefore if we are computing $r = a + b$, our pseudo assembly code would be

```

add r[0], a[0], b[0]
adc r[1], a[1], b[1]
...
adc r[n], a[n], b[n]

```

where each word of a and b are added together. Therefore the number of instructions required is the number of words in the value. Subtraction can be achieved in a similar manner, by using the *sub* and *subb* instructions.

Multiplication is not as simple as addition or subtraction, because each word in a needs to be multiplied by all words in b . One technique to implement this is using basic long multiplication. For example if we are computing 33×52 on a 4-bit processor, we get

```

      0010 0001
    × 0011 0100
    -----
      1000 0100
+ 0110 0011 0000
-----
    0110 1011 0100

```

where we are only multiplying or adding 4-bits per cycle. Therefore in this example, we require 4 multiples, and 3 adds. This can get more complex if overflows occur when multiplying, often requiring more additions in modern instruction sets [4]. Scaling this up to multiplying two 2048-bit values on a 64-bit processor, 1024 multiplications and even more additions depending on the amount of overflows that occur. Ignoring other instructions such as fetch, move and store, just this simple multiplication requires over 2048 instructions. The number of registers within the processor will also impact the performance because large numbers (i.e. 2048-bits) can be difficult to fit into registers all at once.

This is the disadvantage of using the Barrett reduction, because even though the operations used seem simple, in reality they can be complicated to implement. The algorithm also makes use of the division operation which is more computationally intense than multiplication. However it depends on the value of b .

In contrast, our algorithm only uses simple operations like bit shifting and addition. It also uses memory efficiently as elements of the value are only accessed once. However the number of instructions required depends on the size of the key and lookup table. For example given $X \bmod Y$, if Y is only 1024-bits, and X is 2048-bits, therefore the main reduction is computed on the upper 1024-bits of X . Then the lookup table key can be 8-bits, resulting in a size of approximately 258Kb (1024-bit values + keys). This means that 1024/8 lookups are required. The bulk of the instructions are adding the lookup values each time. In this case, on a 64-bit processor, 2048 add instructions are needed. This is therefore comparable to a single multiplication in the Barrett Reduction (because the multiplication will be computed on 2048-bits). The shifts required will also

require approximately 2048 instructions, which makes equals two multiplication operations. There are other instructions that will be required of course, like some additions for joining the elements together, and some final subtractions to get to the correct result. But these heavy depend on the inputted values.

At this point it is not possible to definitely claim our proposed algorithm is better than the Barrett Reduction. However by analysing the instructions required, we can show that it will use less instructions when compared to the Barrett Reduction, because of the Barrett Reductions heavy use of multiplication and division instructions. Comparing implementations of both algorithms in software is a difficult approach. Because both algorithms would be needed to be implemented fairly, with neither having any better code than the other (i.e. optimisations). Therefore only once a fully operational hardware implementation of our algorithm is complete, will we be able to provide a definite answer.

5.2 Comparisons with Montgomery Reduction

Unlike the comparison for the Barrett Reduction, comparing our proposed algorithm against the Montgomery Reduction is more straight forward. This is because the main operation used in Algorithm 2 is addition, which is the same as our proposed algorithm. Given that an addition in Algorithm 2 only occurs if a bit is high, we will say that on average, half of the bits are high. Meaning if the input is 2048-bits, only 1024-bits are high. Which therefore requires 1024 addition operations. As discussed in the Barrett Reduction comparison, one addition uses many instructions on a standard processor. Therefore on a 64-bit processor, 16384 add instructions are required if the input is 2048-bit values. This is far more than our proposed algorithm would require, which was approximately 2048 add instructions, and approximately 2048 shift instructions, as shown in the Barrett comparison. The Montgomery Reduction also makes no guarantees on the number of bits required for each operation. For example, given $X \bmod Y$, where X is 4096-bits and Y is 1024-bits, then each add operation will need to compute over 4096-bits. However our proposed algorithm will only compute over 1024-bits at a time. This is a very useful property for hardware implementations, and for making efficient use of registers and lower level cache.

The Montgomery Reduction does not always give the correct answer first time, often requiring additional steps to compute the desired result. Where as our proposed algorithm will give the correct result after one run. Another point is that the Montgomery Reduction also requires a lot of shifting operations for the $2^k Y$ computation. Therefore using this simple theatrical comparison, our proposed algorithm should allow for better implementations in both software and hardware over the Montgomery Reduction algorithms shown.

5.3 Comparisons with Fast Modular Reduction Method

This newly proposed algorithm in [7] is actually similar to the Montgomery Reduction in the sense that it looks at high bits. The difference is that it only processes the bits above bits of the modulus (Y), like our proposed algorithm. For

example, given $X \bmod Y$, where X is 2048-bits and Y is 1024-bits, then it will only process the upper 1024-bits of X . Then for each high bit i , it uses a lookup table to get the result of $2^i \bmod Y$ and adds it to the result. Therefore if we again say that half of the bits are high, 512 add operations are required. The bit width of the result is not clearly stated, however given that only values are added, it has to be 2048-bits (the same as X). Meaning at least 8192 add instructions would be needed, but will probably be closer to 16384. Our proposed algorithm would use far less instructions, and it guarantees the width of the result. Also because it keeps the result at a fixed width, the amount of subtractions required should be less on average, but this depends on the input.

Comparing the lookup tables, this algorithm [7] has a fixed size lookup table. So for example, if Y is 2048-bits, then there must be 2048 entries in the table, each of a size of 2048-bits. Resulting in a size of approximately 4Mb. Also because of this, the bit width of the input X , can be no more than double the bit width of Y . This is because the lookup table only contains entries for the bits up to double that of Y , which for this example is 2^{2048} to 2^{4096} . This is a major limitation of this algorithm. However when looking at our lookup table, it can vary in size, and can support an arbitrary bit length of X . This is important to allow the table to fit in cache, and for devices which have limited storage. Using the same example where Y is 2048-bits, if the lookup table key is 8-bits, then the total size is approximately 0.5Mb. Our algorithm even allows for a lookup size of 1, if space is really limited.

Therefore our proposed algorithm is superior to that proposed in [7], both in terms of required instructions and the effectiveness of the lookup table. It is also a better option for hardware implementations because it can guarantee bit widths, and support varying sized lookup tables.

6 Future Work and Conclusion

We propose an algorithm for computing the modulus operation which is unique to other algorithms. Future work includes creating an implementation on a Field-Programmable Gate Array (FPGA). Then to compare not only the performance, but also the area and power required, as well as looking into the effectiveness of side channel attacks. We would also like to develop a software implementation, and compare its performance for computing the modulo operation over integers, as well as within already secure data. For example encrypting data which is already encrypted using a fully homomorphic encryption scheme.

The algorithm has been designed to keep memory access at a minimum, decreasing the time the processor has to wait for data to be fetched. It is more advanced than other algorithms using lookup tables, such as in [7] and [14], because our algorithm allows the table to be used on an arbitrary input, while supporting a variable size.

Acknowledgements. This research is supported by STRATUS (Security Technologies Returning Accountability, Trust and User-Centric Services in the Cloud) (<https://stratus.org.nz>), a science investment project funded by the New Zealand Ministry of Business, Innovation and Employment (MBIE). The authors would also like to thank Sabu M. Thampi for his kind invitation to submit this invited paper for the SSCC 2016 proceedings.

References

1. Daemen, J., Rijmen, V.: The Design of Rijndael: AES-The Advanced Encryption Standard. Springer, Heidelberg (2002)
2. Gentry, C.: Fully homomorphic encryption using ideal lattices. STOC **9**, 169–178 (2009)
3. Rivest, R., Shamir, A., Adleman, L.: A method for obtaining digital signatures and public-key cryptosystems. Commun. ACM **21**, 120–126 (1978)
4. Intel 64 and IA-32 architectures software developer’s manual. <http://www.intel.com/content/dam/www/public/us/en/documents/manuals/64-ia-32-architectures-software-developer-manual-325462.pdf>. Accessed 27 Aug 2014
5. Kaliski, B.: Twirl and RSA key size. RSA Laboratories Technical Note (2003)
6. Barrett, P.: Implementing the Rivest Shamir and Adleman public key encryption algorithm on a standard digital signal processor. In: Odlyzko, A.M. (ed.) CRYPTO 1986. LNCS, vol. 263, pp. 311–323. Springer, Heidelberg (1987)
7. Cao, Z., Wei, R., Lin, X.: A fast modular reduction method. IACR Cryptol. ePrint Arch. **2014**, 40 (2014)
8. Dupaquis, V., Venelli, A.: Redundant modular reduction algorithms. In: Prouff, E. (ed.) CARDIS 2011. LNCS, vol. 7079, pp. 102–114. Springer, Heidelberg (2011)
9. Montgomery, P.L.: Modular multiplication without trial division. Math. Comput. **44**(170), 519–521 (1985)
10. Denis, S.T., Rose, G.: BigNum Math: Implementing Cryptographic Multiple Precision Arithmetic. Syngress Publishing, Boston (2006)
11. Kwon, T.-W., You, C.-S., Heo, W.-S., Kang, Y.-K., Choi, J.-R.: Two implementation methods of a 1024-bit RSA cryptoprocessor based on modified montgomery algorithm. In: The 2001 IEEE International Symposium on Circuits and Systems, ISCAS 2001, vol. 4, pp. 650–653. IEEE (2001)
12. Batina, L., Muurling, G.: Montgomery in practice: how to do it more efficiently in hardware. In: Preneel, B. (ed.) CT-RSA 2002. LNCS, vol. 2271, p. 40. Springer, Heidelberg (2002)
13. Lim, C.H., Lee, P.J.: Generating efficient primes for discretelog cryptosystems, POSTECH Information Research Laboratories
14. Lim, C.H., Hwang, H.S., Lee, P.J.: Fast modular reduction with precomputation. In: Proceedings of Korea-Japan Joint Workshop on Information Security and Cryptology (JWISC 1997), pp. 65–79. Citeseer (1997)

Mutual Authentication Based on HECC for RFID Implant Systems

Asha Liza John^(✉) and Sabu M. Thampi

Indian Institute of Information Technology and Management - Kerala (IIITM-K),
Technopark Campus, Trivandrum, India
{asha.mphilcs3,sabu.thampi}@iiitmk.ac.in

Abstract. The Internet of Things (IoT) is an environment in which “things” (objects, animals or people) are provided with unique identifiers (IPv6 addresses) and the ability to communicate over a network without requiring human-to-human or human-to-computer interaction. Radio-Frequency Identification Technology (RFID) is the key enabler of the IoT. The RFID Implant System considered in the proposed work consists of an implantable, passive RFID tag which is a data carrying device that is attached to the object to be identified, RFID reader which communicates with the tag in order to read or write data to its memory and, the back-end database which stores information related to the identified object. There are several security issues associated with the use of RFID tags in IoT like eavesdropping, impersonation, cloning, replay attack, tag destruction, unauthorized tag reading, tag modification etc. To defend such attacks effectively, efficient security mechanisms are essential. So, the proposed system aims to provide a secure mutual authentication mechanism based on Hyper Elliptic Curve Cryptography (HECC) to authenticate the communication between the RFID tag and reader. The security of Hyper-elliptic Curve Cryptosystem depends on the hardness of solving hyper-elliptic curve discrete logarithm problem (HCDLP). This problem helps to avoid the eavesdropper from breaking into the security of the HECC cryptosystem. The proposed work also uses D-Quark hash algorithm.

Keywords: RFID Implant System · IoT · Security · Healthcare · Mutual authentication · Hyper-elliptic curve cryptography

1 Introduction

Internet of Things (IoT) involves the concept of connecting everything around, to the internet. These things may include wearable devices, metering devices, environmental sensors etc. As a result of this, in the near future, there may be trillions of connected devices which may communicate with each other by exchanging data. This communication between devices is made possible via wireless networks. Wireless networks use radio communication frequencies and follows radio regulations. Hence, Radio Frequency Identification (RFID) is the key enabler of IoT. IoT along with RFID technology has many applications in smart home, healthcare system, inventory management etc.

Providing health care was far simpler than it is today. But, advent of legislation, technology and reimbursement charges has forced the entire healthcare system to shift the way care is provided. However, there are opportunities to improve patient experience. Historically, physicians did not have access to a holistic view of the patients' health so they were forced to make treatment decisions, with limited or partial data. Soon the trend shifted from this to Electronic Medical Records (EMR) by which it is possible to collect complete medical records of a patient. When data from such EMR systems and consumer wearables are merged, it is possible to organize and process data beyond typical clinical scenarios. At the same time, advances in technology provided new and low cost ways to detect diseases. When all these combine, the patients and physicians benefit from more comprehensive views of patient health and treatment progress enabling physicians to more accurately adjust treatments. Hospitals may not have the resources to monitor everyone and people with such resources cannot monitor themselves. Meanwhile, funding constraints depend on optimization solutions to effectively and efficiently distribute and manage equipment. So, facilities need to rely on pervasive technologies like passive RFID tags to supplement monitoring and management efforts.

The RFID Implant System mentioned in the proposed work is a resource constrained system which has three main components – Implantable RFID tag which is a passive tag (almost the size of a rice grain) implanted into the patient's body, RFID Reader which communicates with both the tag and the back-end server, and a Back-end server which stores the information about the patient. The communication channel between the reader and the back end server is secure. But, the wireless communication channel between the reader and the tag is found to be insecure and hence, may be vulnerable to attacks like unauthorized location tracking, eavesdropping attack, impersonation attack, replay attack etc. Hence, both the tag and the reader must be assured that the other end is legitimate. In the healthcare scenario, providing robust and secure data communication is crucial so, the authentication of tag by reader is just not enough because the information about a particular patient, which the tag shares with the reader, is highly sensitive. So before sharing such sensitive data, the tag must make sure that the reader is legitimate. For this, a two way authentication or mutual authentication mechanism between the tag and the reader is essential. Hence in this work, we propose a mutual authentication mechanism to authenticate the communication between, RFID Reader and the RFID Tag.

As mentioned before, RFID Implant System is a resource constrained system since, the implanted RFID tag has only very less processing power. Hence, it requires efficient and optimized security solutions. The mutual authentication based on elliptic curve cryptography (ECC) or non-ECC mechanisms so far implemented for RFID systems in general are not adequately optimized to operate in resource constrained environments. So, in this work, we combine the concepts of Hyper-elliptic curves (HECC) and D-Quark hash algorithm to formulate an optimized and efficient mechanism for mutual authentication of RFID Reader and Tag.

The remainder of this paper is organized as follows: Sect. 2 provides an overview of related work and literature. Section 3 presents the proposed HECC-based mutual authentication scheme for the RFID implant systems. Section 4 provides a comprehensive security and computational performance analysis of our scheme. In this

section, the comparison of this work with similar existing approaches is also presented. Finally, Sect. 5 concludes and summarizes the work.

2 Related Works

Currently there are several security and privacy concerns which restrict the use of implantable tags. Mitrokotsa, Rieback and Tanenbaum classifies these RFID attacks based on its layer of operation [6]. Among these, the most popular attacks are the ones affecting network layer. The attacks on Network-Transport layer are classified into tag attacks and reader attacks. Attacks on tags are cloning, spoofing and many more and that on reader are impersonation, eavesdropping etc. There are several existing security mechanisms to defend these attacks. But this section, reviews only those literature dealing with authentication since the proposed system attempts to develop a mutual authentication mechanism.

Authentication can be ensured by generation and verification of digital signatures by both the communicating parties. In their paper Radu-Ioan Paise and Serge Vaudenay emphasizes the importance of mutual authentication [8]. A malicious reader can obtain unauthorized information from a tag, raising security or privacy issues. In order to fix this problem, besides tag's authentication, a protocol must ensure reader's authentication. To ensure this, a mutual authentication protocol is used. So far, several mutual authentication techniques have been proposed based on cryptographic algorithms like IDEA [2], AES [9], ECC [3]. Among these, algorithms based on elliptic curve cryptography are considered more suitable for application in constrained devices, because ECC uses shorter keys which results in faster execution and less memory utilization.

In 2006, Tuyls et al. proposed an ECC-based RFID identification scheme based on Schnorr identification protocol [10]. But, in 2008 Lee et al. found out that this identification scheme is vulnerable to location tracking attack and that it does not ensure forward security [5]. In 2007, Batina et al. proposed an ECC-based RFID identification scheme based on Okamoto's authentication algorithm [1]. But in 2008, Lee et al. proved that like Tuyls et al. scheme, this scheme is also prone to tracking and forward secrecy problem [5]. Hence, Lee et al. in 2010, proposed an ECC based RFID authentication scheme so as to solve the existing tracking problems [4]. Again, in 2011, Zhang et al. proposed an ECC-based randomized key scheme in order to improve Tuyls et al.'s and Lee et al.'s schemes [12]. This scheme defended almost all relevant attacks concerning the RFID systems. But, in all these schemes, the authors merely considered one-way authentication of tag by reader, excluding the possibility of authentication of reader by tag. This causes tags to reply to any malicious query being sent by an adversary. In 2013, Liao et al. proposed a secure ECC-based authentication scheme integrated with ID-verifier transfer protocol. But, the tag identification scheme suffered from lack of performance efficiency in terms of the tag's computation time and memory requirement [13]. Moosavi, Nigussie and Isoaho implemented a mutual authentication scheme based on the concept of Elliptic Curve Cryptography (ECC) on RFID Implant Systems [7]. But in 2014 Barsgade et al. compared elliptic curve curves and hyper elliptic curves in DLP and found that HECC is as good as ECC with less computational complexity [11] (Table 1).

Table 1. Comparison of existing schemes

	Batina et al. [1]	Zhang et al. [12]	Liao and Hsiao [13]	Lee et al. [5]	Moosavi et al. [7]
Eavesdropping	Yes	Yes	Yes	Yes	Yes
Impersonation	No	Yes	Yes	Yes	Yes
Replay attack	Yes	Yes	Yes	Yes	Yes
Forward security	No	Yes	Yes	Yes	Yes
Mutual authentication	No	No	Yes	No	Yes
Performance	Less	Less	Less	Less	Better

From the research conducted on existing security mechanisms it is found that conventional security and protection mechanisms are not adequate for RFID Implant Systems since it is resource constrained. So, RFID implant system still requires a robust, optimized, and lightweight security framework. So in the proposed system we combined concepts and algorithms with less power and memory requirements like HECC, D-Quark lightweight hash algorithm and Harley's algorithm for divisor computation to develop an optimized and efficient mutual authentication mechanism to ensure authentication between the implanted RFID tag and the RFID reader in a resource constrained RFID Implant System.

3 Proposed Authentication Mechanism

Mutual Authentication can be ensured via generation and verification of digital signatures by both the communicating parties. Hence the proposed algorithm for mutual authentication is implemented with reference to Digital Signature Algorithm in Digital Signature Standard published in Federal Information Processing Standards Publications (FIPS PUBS 186) which are issued by the National Institute of Standards and Technology (NIST). This standard specifies that a Digital Signature Algorithm (DSA) is appropriate for all applications requiring a digital signature. It is assumed that the reader side has a list of valid tag IDs and the tag side has a list of valid reader IDs. The proposed method is a two stage process. On entering the radio frequency field of the RFID reader, the passive implanted RFID tag, gets activated, and sends its ID to the reader. The reader checks with the database to see if this ID is already present in the list. If so, then the reader sends its ID to the tag. Further, a similar checking happens at the tag side. If both IDs are found to be valid, then mutually authenticated communication begins. The proposed algorithm has three modules (or phases):

1. Generation of Global Public Parameters and Key Generation
2. Signature Generation
3. Signature Verification

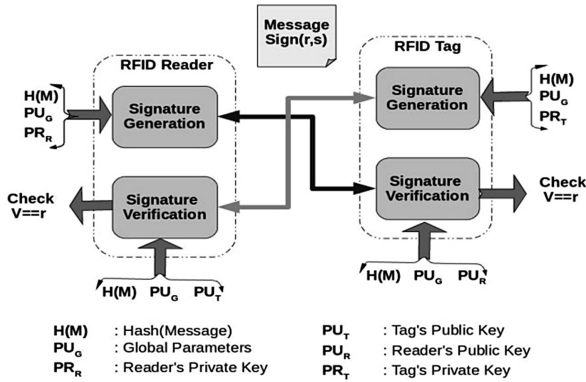


Fig. 1. Block diagram of proposed system

Figure 1 illustrates the proposed method of mutual authentication developed with reference to existing standard DSA. Each of the modules are explained in detail in the following subsections.

3.1 Generation of Global Public Parameters

Signature generation requires the use of some global parameters which are publicly available. In DSA, there are three global parameters. But in the proposed work, the global parameters introduced are all based on the concept of Hyper-elliptic curve cryptography. This is because; solving the Discrete Logarithm Problem of an 80-bit Hyper-elliptic Curve is as hard as that of a 160-bit Elliptic-curve. So, Hyper-elliptic curves are more suitable for resource constrained systems like RFID Implant System. The chosen global parameters are finite prime field F_p , Hyper elliptic curve C of genus 2 over prime field F_p , Unique Reduced Divisor D over Hyper elliptic curve C , a large random number p , large prime divisor q of $p-1$. The unique reduced divisor D over hyper elliptic curve C can be computed using either Harley's Algorithm or Cantor's Algorithm. Divisor D will be represented in Mumford form as $\langle u, v \rangle$.

After generation of global parameters, the next step is to generate the public and private keys required for the signing and verification process at both ends. Signature generation requires the use of private key for signing the message and signature verification uses public key of the corresponding party for verification of the signed message. Private Key PR is a random number such that $PR < q$ and public key PU is computed using the private key as $PU = PR * D'$ where parameter $D' = u + v$.

3.2 Signature Generation

In the proposed algorithm, the signature generation takes as input a message M and generates a signature pair (r, s) as output. Further the generated signature pair (r, s) is appended to the message to be transmitted to the receiver as (M, r, s) and is then send to

the receiver side. Although the overall process is similar to that in DSA, the computation of signature pair (r, s) is different in the proposed signature generation algorithm. Figure 2 shown below, illustrates the signing process. Algorithm 1 shown below explains the computation.

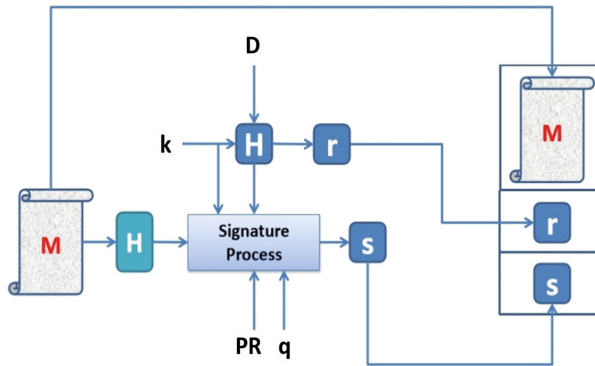


Fig. 2. Signature generation

Algorithm 1 : Signature Generation

Input: Message M

Output: Signature Pair (r,s)

Choose a per message secret random value $k \leq q$

Compute Hash value of the message $H(M)$ using D-Quark Lightweight Hash Algorithm

Compute signature pair r and s as follows

$$r = H(k * D')$$

$$s = [k^{-1}(H(M) - PR * r)] \bmod q$$

Send signature pair (r,s) to the RFID Tag

Here, the per-message secret random value k is newly generated each time a message is send and is destroyed and never reused. The message M as such is not used for calculating the signature pairs. Instead, the hash value of the message is calculated using a lightweight hash algorithm in the Quark series called the D-Quark Hash algorithm. Although, DSS Standard specifies the use of Secure Hash Algorithm (SHA-1), we use D-Quark because, hash value calculation using SHA algorithm is computationally intensive. D-Quark consumes less power and memory compared to SHA-1. Further, the signature pair (r, s) is calculated as shown in the algorithm.

3.3 Signature Verification

Signature verification takes as input the received message and signature pair (M', r', s') and computes four new parameters w, u1, u2 and V out of which, the value of V must be equal to r for the signature to be valid. As in the case of signature generation, the

signature verification process is also similar to that of DSA but, the formulae for computing V value is different. The computation of parameters w, u1, u2 and V are explained in Algorithm 3 described below. Figure 3 illustrates the whole process of verification.

Algorithm 2 : Signature Verification
 Input: Message and Signature Pair (M', r', s')
 Output: Verified Signature ($V = = r$)
 Compute hash value of the message M using D-Quark lightweight hash algorithm $H(M)$
 Compute
 $w = s^{-1} \text{ mod } q$
 $u1 = H(M')w \text{ mod } q$
 $u2 = r'w \text{ mod } q$
 $V = H[(u1+u2)D'+PU]$
 If ($V = = r'$) then, signature is correct and authentication is done.

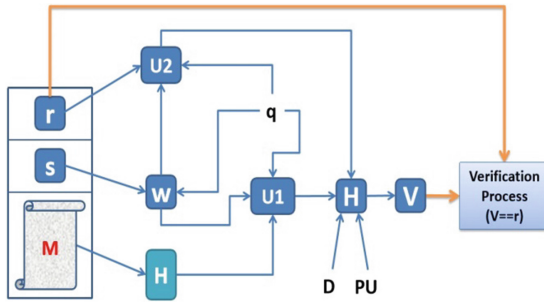


Fig. 3. Signature verification

The verification process mentioned in Algorithm 2 uses public key PU for verification. Both signature generation and signature verification algorithms should be implemented in both the RFID Reader as well as RFID tag side since the aim of the proposed work is to ensure mutual authentication. This is illustrated in the block diagram shown in Fig. 1.

3.4 Proof of $V = = r$

As mentioned before, for the proposed mutual authentication mechanism to be valid, the parameter V calculated during the signature verification phase must be equal to the signature value r. The statements below prove the credibility of this argument mathematically.

$$\begin{aligned}
V &= H[(u_1 + u_2)D' + PU] \text{ as per proposed algorithm} \\
&= H[(u_1 * D') + (u_2 * D') + PR * D'] \text{ (Distributive property and } PU = PR * D') \\
&= H[(u_1 + u_2 + PR)D'] \text{ (Distributive property)} \\
&= H[[H(M) * w \bmod q + r * w \bmod q + PR] D'] \text{ (as per equations of } u_1 \text{ and } u_2) \\
&= H[[H(M) * w \bmod q + r * w \bmod q + PR \bmod q] D'] \text{ since } PR < q \text{ } PR = PR \bmod q \\
&= H[[H(M) * w \bmod q + r * w * PR \bmod q] D'] \\
&= H[[H(M) + PR * r] w \bmod q] D'] \\
&= H[(k * s * w) \bmod q] D'] \text{ as per calculation of } s \\
&= H[(k * s * s - 1) \bmod q] D'] \text{ as per calculation of } w \\
&= H[k \bmod q] D'] \\
&= H[k * D'] \text{ since } k < q \\
&= r
\end{aligned}$$

4 Security and Performance Analysis

This section, analyses the security and performance of the proposed scheme in order to verify whether the essential requirements have been satisfied.

4.1 Security Analysis

Security of the proposed mutual authentication mechanism depends on the difficulty of solving the Hyper-elliptic Curve Discrete Logarithm Problem (HCDLP). Analyses of the proposed scheme against some of the relevant attacks are as follows:

- *Mutual Authentication* is achieved in this scheme by following DSA standard.
- *Availability* of the system is affected by DoS attacks. But this type of attack is possible only if adversary knows the per message secret value k . But this is impossible since the r value is hashed using D-Quark, before being transmitted to the receiving end. So, availability is ensured.
- *Forward Security* is ensured by destroying the per-message secret key k after sending each message.
- *Unauthorized Tracking* is not possible, since each communication between the reader and the tag are mutually authenticated.
- *Replay attack* is not possible since the per message secret key k is involved in each communication. Also, the key changes after each message

4.2 Performance Analysis

The performance of the proposed algorithm is influenced by three major factors - Hyper elliptic Curves, D-Quark Hash Algorithm and Harley's algorithm for computing unique

reduced divisor. From theoretical analysis it can be proved that the hardness of solving an 80 bit HCDLP is equal to the hardness of solving a 160 bit ECDLP. Hence the use of hyper elliptic curves instead of elliptic curves in the proposed work improves the performance. Also, the use of D-Quark lightweight hashing mechanism also improved the computational efficiency of the proposed mutual authentication algorithm. The third factor which influenced the computation is Harley’s algorithm. Usually, the algorithm used for divisor computation is Cantor’s algorithm which is a generic algorithm that involves polynomial arithmetic computation. But Harley’s algorithm converts polynomial arithmetic to field arithmetic and thereby decreasing the time and cost of computation. But, there are still a negligible number of exceptional cases in which Harley’s algorithm cannot find a divisor. For such cases alone, Cantor’s algorithm is used.

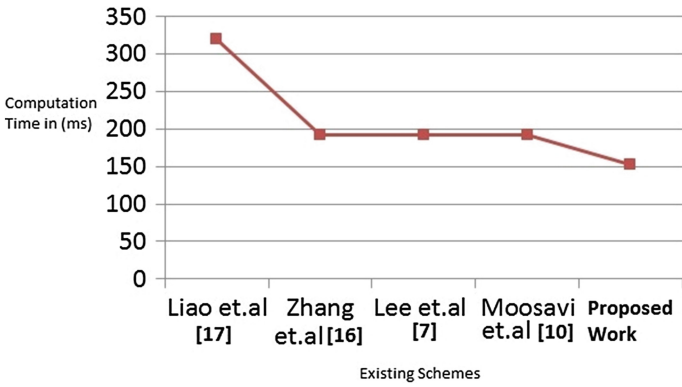


Fig. 4. Comparison with existing schemes

Figure 4 shows the performance graph of the proposed scheme in comparison with the existing schemes. From the graph, it is found that the computation time of the proposed hyper elliptic curve based mutual authentication algorithm is less than that of all the existing schemes based on elliptic curve cryptography. This is because of reduced key-size, use of lightweight D-Quark hash algorithm and use of Harley’s algorithm for calculating the divisor of the hyper elliptic curve.

Figure 5 shows the results of the simulation of the proposed mutual authentication algorithm in Python using a package called Sage-Math. The simulation was done for genus values 2, 3 and 4 and over a range of field order values. From the graph it is evident that execution time for genus 2 hyper-elliptic curves is much less than genus 3 and genus 4 curves and its execution time approaches the genus 3 and genus 4 curves for higher prime order values. Also, it is proven that genus 2 curves are more secure compared to the other two curves of genus 3 and 4. [14] Hence, in the proposed work we chose hyper elliptic curves of genus 2.

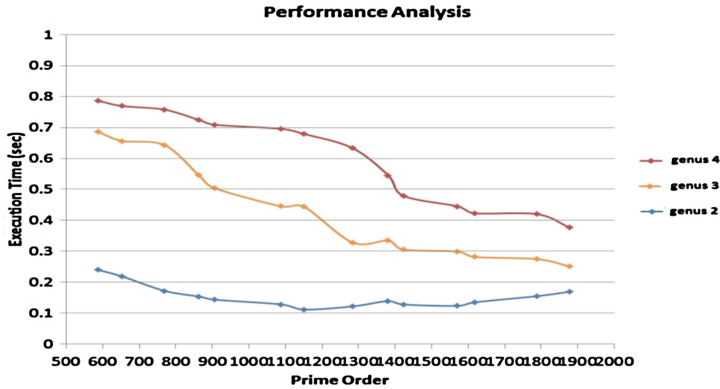


Fig. 5. Results of simulation

4.3 Comparison with ECC Based Signature and Verification Mechanism

The effort required by the best algorithms to solve the Discrete Logarithm Problem, in worst case, is $O(\sqrt{|G|})$ group operations. For curves of genus g over a finite field F_q , $|G| \approx q^g$ as $q \rightarrow \infty$. The minimum level of security recommended is 80 bits. i.e. $\sqrt{|G|} \approx 2^{80}$. Elliptic curves are hyper elliptic curves of genus $g = 1$. Therefore, the number of group operations required to solve ECDLP is $|G| = q^1 \approx 2^{160}$. From this we find that value of $q = 2^{160}$. For the proposed scheme, we use hyper-elliptic curves of genus $g = 2$. Therefore, number of group operations required to solve HCDLP is same as that required for ECDLP which is $|G| = q^2 \approx 2^{160}$. But for hyper-elliptic curves the value of q reduces to $q = 2^{80}$. Hence, from the above analysis it is evident that the hardness of solving an 80 bit HCDLP is equal to the hardness of solving a 160 bit ECDLP. Thus, shifting the focus from elliptic curves to hyper elliptic curves reduces key size which in turn leads to easier data management, minimum hardware and bandwidth requirement, increased battery life etc.

ECC based signature and verification mechanism uses SHA-1 for hashing. But SHA-1 is proved to be vulnerable and is not a lightweight hash algorithm. So in the proposed scheme we replaced SHA-1 with a light weight hashing mechanism called D-Quark. This consumes less power for execution and also provides more security since it has more number of rounds.

In ECC based signature and verification mechanism the coordinates of the points on the curve are directly employed for calculation. But, in the case of hyper-elliptic curves the divisor calculation is done using Harley's and Cantor's algorithm. Hence, the proposed scheme based on HECC performs better than the existing ECC based signature and verification schemes.

5 Conclusion and Future Work

A mutual authentication algorithm which operates in the RFID Implant System environment is developed with reference to Digital Signature Standard (FIPS 186). The algorithm uses the concepts of Hyper elliptic Curve Cryptography and D-Quark Hash. Proposed algorithm was theoretically proved and analysed for security and performance. A rough implementation of the algorithm was done using Python and Sage-Math Package. The implementation was simulated for a range of genus values and field orders. Also, a comparison of this work was evaluated against some of the existing mutual authentication schemes and was found to perform better.

The scope of this work may be extended to other mobile environments with similar requirement for mutual authentication. The proposed algorithm can also be combined with HECC based Diffie Hellman key exchange mechanism so that, symmetric key can be exchanged between both parties in communication and messages can be encrypted using this key for better security.

References

1. Batina, L., Guajardo, J., Kerins, T., Mentens, N., Tuyls, P., Verbauwhede, I.: Public-key cryptography for RFID-tags. In: Fifth Annual IEEE International Conference on Pervasive Computing and Communications Workshops, PerCom Workshops 2007, pp. 217–222. IEEE (2007)
2. Liu, D., Yang, Y., Wang, J., Min, H.: A mutual authentication protocol for RFID using IDEA. Auto-ID Labs White Paper WP-HARDWARE-048, March 2009
3. Chou, J.-S.: An efficient mutual authentication RFID scheme based on elliptic curve cryptography. *J. Supercomput.* **70**(1), 75–94 (2014). Springer
4. Lee, Y.K., Batina, L., Singelée, D., Verbauwhede, I.: Wide-weak privacy-preserving RFID authentication protocols. In: Chatzimisios, P., Verikoukis, C., Santamaria, I., Laddomada, M., Hoffmann, O. (eds.) *MOBILIGHT 2010*. LNICST, vol. 45, pp. 254–267. Springer, Heidelberg (2010)
5. Lee, Y.K., Batina, L., Singelee, D., Preneel, B., Verbauwhede, I.: An-counterfeiting, untraceability and other security challenges for RFID systems: public-key-based protocols and hardware. In: Sadeghi, A.-R., Naccache, D. (eds.) *Towards Hardware-Intrinsic Security*, pp. 237–257. Springer, Berlin (2010)
6. Mitrokotsa, A., Rieback, M.R., Tanenbaum, A.S.: Classification of RFID attacks. *G. E. N.* **15693**, 14443 (2010)
7. Moosavi, S.R., Nigussie, E., Virtanen, S., Isoaho, J.: An elliptic curve-based mutual authentication scheme for RFID implant systems. *Procedia Comput. Sci.* **32**, 198–206 (2014)
8. Paise, R.-I., Vaudenay, S.: Mutual authentication in RFID: security and privacy. In: *Proceedings of the 2008 ACM symposium on Information, Computer and Communications Security*, pp. 292–299. ACM (2008)
9. Pham, T.A., Hasan, M.S., Yu, H.: An RFID mutual authentication protocol based on AES algorithm. In: *2012 UKACC International Conference on Control*, pp. 997–1002. IEEE, September 2012
10. Tuyls, P., Batina, L.: RFID-tags for anti-counterfeiting. In: Pointcheval, D. (ed.) *CT-RSA 2006*. LNCS, vol. 3860, pp. 115–131. Springer, Heidelberg (2006)

11. Wankhede Barsgade, M.T., Meshram, S.A.: Comparative study of elliptic and hyper-elliptic curve cryptography in discrete logarithmic problem. *IOSR J. Math.* **10**(2), 61–63 (2014)
12. Zhang, X., Li, J., Wu, Y., Zhang, Q.: An ECDLP-based randomized key RFID authentication protocol. In: 2011 International Conference on Network Computing and Information Security (NCIS), vol. 2, pp. 146–149 (2011)
13. Liao, Y.-P., Hsiao, C.-M.: A secure ECC-based RFID authentication scheme integrated with ID-verifier transfer protocol. *Ad Hoc Netw.* **18**, 133–146 (2014)
14. Pelzl, J., Wollinger, T., Guajardo, J., Paar, C.: Hyperelliptic curve cryptosystems: closing the performance gap to elliptic curves. In: Walter, C.D., Koç, Ç.K., Paar, C. (eds.) CHES 2003. LNCS, vol. 2779, pp. 351–365. Springer, Heidelberg (2003)

On the Use of Asynchronous Cellular Automata in Symmetric-Key Cryptography

Biswanath Sethi^{1(✉)} and Sukanta Das²

¹ Department of Computer Science Engineering and Applications,
Indira Gandhi Institute of Technology, Sarang, Dhenkanal 759146, Odisha, India
sethi.biswanath@gmail.com

² Department of Information Technology, Indian Institute of Engineering Science
and Technology, Shibpur, Howrah 711103, West Bengal, India
sukanta@it.iiests.ac.in

Abstract. This paper addresses a symmetric key cryptosystem using rule 57 asynchronous cellular automata. It is experimentally shown that the proposed cryptosystem achieves the avalanche effect after 32000 iterations. The vulnerability of the proposed scheme is discussed and note that, brute-force attack is practically infeasible. The effectiveness of the scheme is compared with other cryptosystems and finally, it is also report that the proposed cryptosystem can easily be implemented in hardware.

Keywords: Asynchronous cellular automata (ACAs) · Reversibility · Block cipher · Symmetric key cryptosystem

1 Introduction

Cellular automata (CAs) were introduced by von Neumann for his study on biological self replication [1]. “CAs have been used in cryptography because of their complex behaviour, simple model and capable of generating complex and random patterns” [2]. The use of reversible CAs (RCAs) in cryptography was introduced by Kari [3]. Kari claimed that due to “the inherent parallelism, RCAs can be used as efficient encryption and decryption devices” [3]. However, all these studies on cryptography are mostly converge on synchronous CAs (deterministic), where, all the CAs cells are forced to update. After all, no work has been initiated in cryptography using *asynchronous cellular automata* (ACAs). This has motivated us to undertake this research. “We adopt one-dimensional two-state three-neighborhood CAs that are updated *fully asynchronously*” [4], i.e. an arbitrary cell is selected randomly and uniformly to update in each discrete time step.

In this paper, we address an ACAs based symmetric key cryptosystem for block cipher encryption. Our proposed scheme encrypts a plaintext using a class of CAs under fully asynchronous update and produce a ciphertext. We also decrypt the ciphertext using the same ACAs and obtain the plaintext. Since, encryption and decryption are opposite of each other, we utilize ACAs with

property of backward transitions for design of the cryptosystem. In view of the proposed target, we first diagnose ACAs with capable of backward transitions. There are total 256 local rules in one-dimensional two-state three-neighborhood system [5]. Here, we evolve entire 256 rules to find the ACAs with capability of backward transitions (Sect. 4). We design a theorem (Theorem 1) for the identification of such ACAs. However, ACAs with fixed points are not suitable for cryptography. Further, we find ACAs with no fixed points from the list of ACAs (Table 3) with backward transitions. Hence, in order to find such ACAs we use the idea of *fixed-point graph* (FPG) [6]. The security of the proposed scheme mainly relies on the key (update pattern of ACAs) used for the encryption. We also compare the proposed scheme with existing cryptosystems such as data encryption standard (DES) and advanced encryption standard (AES) [7]. We discuss the vulnerability of the proposed scheme to possible cryptanalytic attack. Finally, we also report it is possible to realize the cryptosystem in hardware.

2 Definitions

A cellular automaton (CA) contains a lattice of identical cells with a boolean value for each cell, referred as current state of cell. The state of each cell is updated at discrete time step according to a local update rule. The decimal equivalent of the 8 outputs is called ‘rule’ [5]. Three such rules (rule 57, 99 and 147) are shown in Table 1.

Definition 1. “The association of the neighborhood x, y, z to the value $f(x, y, z)$, which represents the result of the updating function, is called *Rule Min Term* (RMT). Each RMT is associated to a number $r(x, y, z) = 4x + 2y + z$ ” [6].

Any state of a CA can be expressed as a sequence of RMTs. For instance, let us consider the state 0110 in periodic boundary condition. Now this state can be viewed as $\langle 1364 \rangle$, where 1, 3, 6 and 4 are corresponding RMTs of first, second, third and fourth cell. In order to find the RMT sequence for a state, we imagine a 3-bit window which slides over the state. To find the i^{th} RMT, the window contains $(i - 1)^{th}$, i^{th} and $(i + 1)^{th}$ bits of the state. Then the window slides one bit right to find the $(i + 1)^{th}$ RMT. Now the window is loaded with i^{th} , $(i + 1)^{th}$ and $(i + 2)^{th}$ bits of the state. However, there is a relation between

Table 1. Look-up table for rule 57, 99 and 147

Current state (CSs):	111	110	101	100	011	010	001	000	Rule
(RMTs)	(7)	(6)	(5)	(4)	(3)	(2)	(1)	(0)	
(i) NSs:	0	0	1	1	1	0	0	1	57
(ii) NSs:	0	1	1	0	0	0	1	1	99
(iii) NSs:	1	0	0	1	0	0	1	1	147

Table 2. Relationship between i^{th} and $(i + 1)^{th}$ RMT

i^{th} RMT	0	1	2	3	4	5	6	7
$(i + 1)^{th}$ RMT	0, 1	2, 3	4, 5	6, 7	0, 1	2, 3	4, 5	6, 7

two consecutive RMTs present in any sequence of RMTs. If 3 (011) is the i^{th} RMT in a sequence, then $(i + 1)^{th}$ RMT is either 6 (110) or 7 (111). Similarly, if 4 (100) or 0 (000) is the i^{th} RMT, then 0 (000) or 1 (001) is the $(i + 1)^{th}$ RMT. These relations between consecutive RMTs are shown in Table 2.

During the evolution of CA under *fully asynchronous* mode, a sequence $(u_t)_{t \in \mathbb{N}}$ of cells can be marked where u_t denotes the cell updated at time t . This sequence of update is called *update pattern* [8]. For an initial state y and an *update pattern* U , the evolution of the CA is represented by the sequence of states (y^t) obtained by successive applications of the updates of U . Formally, we have: $y^{t+1} = F(y^t, u_t)$ and $y^0 = y$, with:

$$y_i^{t+1} = \begin{cases} f(y_{i-1}^t, y_i^t, y_{i+1}^t) & \text{if } i = u_t \\ y_i^t & \text{otherwise.} \end{cases}$$

This evolution can be illustrated in the form of a state transition diagram. For instance, $y = 0000$ and $U = (1, 4, 3, 1, \dots)$, the partial state transitions of 4-cell rule 57 ACA, is shown in Fig. 1. The cells evolved during the transitions are shown above the arrows.

Definition 2. “An RMT $r(x, y, z)$ for a rule R is active if $f(x, y, z) \neq y$; otherwise passive” [6].

For instance, consider the rule 57 ACA (see Table 1). The RMT 0 (000) is active for rule 57 ACA. Because, when a cell is acting on the RMT 0 of rule 57, then the cell’s current state is 0 and next state for the cell is 1 (see Table 1). On the other hand RMT 4 (100) of rule 99 is passive.

Definition 3. “A fixed point of an ACA is the ACA state, next state of which is the state itself for any cell update. That is, if an ACA reaches to a fixed point, the ACA remains in that particular state forever” [6].

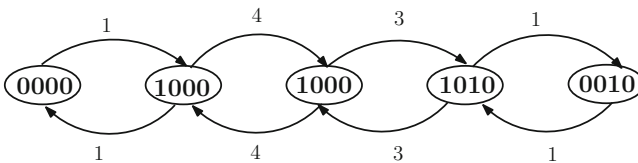


Fig. 1. Partial state transition diagram of rule 57 ACA

Let us consider, rule 147 ACA with 6 cells. The state 111111 is a fixed point for this particular rule. The next state of 111111 is always the same state (i.e. 111111) for any cell update as RMT 7 (111) for this rule is passive. Hence, the state 111111, is a fixed point. In general, all the RMTs of a fixed point are passive.

3 ACAs Based Cryptosystem

CAs have been used in cryptography because of their interesting behaviour of being reversible i.e. not only forward but also backward iterations are possible. CAs are said to be reversible, if the previous configurations can be uniquely reached from a given current configuration [9]. Kari, has reported in his article [3], that reversible CAs can be used for the design of cryptosystem. However, ref. [10], had claimed that it is hard to synthesize reversible one-dimensional ACAs. As cells of ACAs are independent, reversibility in ACAs are practically infeasible. Hence, we never find an ACA and its inverse ACA. The authors of [8] have studied the notion of reversibility in the context of fully asynchronous CAs. A set of ACAs (46 ACAs) are identified that are recurrent, that is, these ACAs are almost surely return to their initial configurations. In that study of reversibility, the evolution of ACA is represented by an initial configuration and an *update pattern* U . In this work, we intent to search a class of recurrent ACAs, which have interesting property of backward transitions considering reverse update pattern. These ACAs can be used in cryptography.

For an initial configuration S and an update pattern U , the evolution of an ACA is represented by the sequence of configurations (S^t) obtained from successive applications of the updates of U . Formally, we have $S^{t+1} = F(S^t, u_t)$, where u_t is the cell updated at time t . This evolution can be represented in the form of a state transition diagram. For example, Fig. 1 shows a partial state transition diagram of rule 57 ACA. The state $S = 0000$ is updated with update pattern, $U = (1, 4, 3, 1)$ and reached at 0010. Interestingly, the state $S = 0000$ can return back from 0010 if we use the same rule but an update pattern, $U^{-1} = (1, 3, 4, 1)$ (see Fig. 1 for verification). We name it as backward transition property of ACA. The U^{-1} is the inverse of U . So, ACAs with the property of backward transitions using reverse update pattern (U^{-1}) are eligible for the design of cryptosystem. This backward transitions of such ACAs are useful for the decryption of the ciphertext to plaintext. It is obvious that the number of forward and backward transitions are same for both the encryption and decryption. The update pattern that were generated in forward transitions are employed as the secret key for the cryptosystem.

Proposed Cryptosystem

The ACAs based cryptosystem is designed for 64-bit block cipher. A plaintext is a 64-bit block which is encoded as initial state of an ACA with capable of backward transitions. The process of encryption is performed by evolving an ACA on the 64-bit plaintext. After some specific number of iterations, the ACA

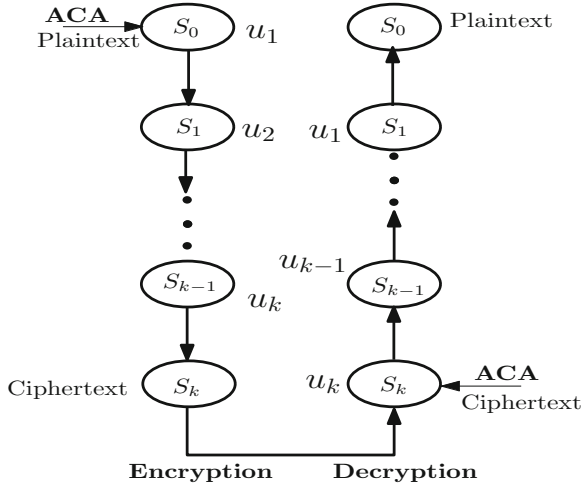


Fig. 2. Symmetric key cryptosystem using ACA

state is the ciphertext. Decryption is just reverse of the encryption. It is achieved by backward transitions of the same ACA on the ciphertext using the update pattern in reverse. The arbitrary sequence of updated cells (*update pattern*) generated during the forward transitions is the secret key for the cryptosystem. This update pattern is then securely shared to the receiver for decryption. The security of the scheme mainly relies on the update pattern of the ACA.

Figure 2 illustrates the detail procedure of encryption and decryption for the proposed ACA based cryptosystem. The state S_0 (see Fig. 2) is the plaintext, which is initial state to an ACA with capable of backward transition. The state S_0 (plaintext) is iterated for specific number of steps (k steps shown in this case) and S_k is the ACA state after the iterations. The update pattern, generated randomly during iterations, are shown in the figure as u_1, u_2, \dots, u_k . The state S_k is considered as the ciphertext for the cryptosystem (Fig. 2). Again the same ACA is evolved using the *update pattern* in reverse order (in the order u_k, u_{k-1}, \dots, u_1). Now, the state S_k is the initial state and u_k is the cell selected to update. After same number of iterations state S_0 is the ACA state which is the plaintext for the cryptosystem.

The above discussion proves that ACAs which have the property of backward transitions considering reverse update pattern are suitable for this cryptosystem. However, an arbitrary ACA can not be used in cryptography. Therefore, the following tasks are pointed out to find the relevant ACAs, that are useful for the design of cryptosystem.

1. All the 256 ACAs do not have the property of backward transitions. So, the initial task is to determine such ACAs that have the property of backward transitions starting from any initial seed.

2. An ACA with the property of backward transition may have fixed points. These ACAs are not suitable for the cryptosystem. Hence, the further task is to point out ACAs that have no fixed point from the above set.

The subsequent sections handle these issues to identify candidate ACAs for the cryptosystem.

4 Identification of ACAs with Backward Transition

This section identifies ACAs that have the property of backward transitions using reverse update pattern. It has been already discussed in the above section that these ACAs can be efficiently used in cryptography. We propose a theorem for identification of such ACAs. The following theorem states the necessary conditions, which identifies ACAs, where backward transitions are possible considering reverse update pattern.

Theorem 1. *An ACA does not follow backward transition using reverse update pattern if an RMT r from the given sets is active (passive) and the other one is passive (active)– $\{0, 2\}$, $\{1, 3\}$, $\{4, 6\}$ and $\{5, 7\}$.*

Proof. Case (i): Let us consider RMT 2 of an ACA is active but RMT 0 is passive. Now consider a state $X= 1010$. For the update pattern (3), we get state $Y= 1000$. However, it is not possible to return back to state X from state Y for the update pattern (3), as RMT 0 is passive. Similarly, consider RMT 0 is active and the other RMT 2 is passive. From a state $X=0000$, the state $Y=1000$ is reached by updating the first cell. However, using the same update pattern (1), we can not go back to the state X .

Case (ii): Similarly, let us consider RMT 1 is active and RMT 3 is passive. Now, assuming a state $X= 0100$ and updating the first cell (update pattern (1)), we get the state $Y=1100$. The state X can not return back from state Y for the same update pattern as RMT 3 is passive. Similarly, consider RMT 1 is passive and RMT 3 is active. Assuming a state $X=1100$ and update pattern (1), we get a state $Y=0100$. State X can not return back considering the same update pattern as RMT 1 is passive.

Case (iii) & Case (iv): We omit the detail proof of case (iii) for RMT set $\{4, 6\}$ and case (iv) for RMT set $\{5, 7\}$ as the rationale of the proof for these cases are same with the above cases.

Table 3. List of 16 ACAs where backward transitions are possible

51	54	57	60	99	102	105	108	147	150	153	156	195	198	201	204
----	----	----	----	----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

There are 16 rules in which both the RMTs of the above sets are either active or passive. All these rules are noted in Table 3. We conjecture that these 16 rules have the property of backward transition using reverse update pattern.

Example 1. Let us consider rule 57 ACA, in which RMTs 1 and 3 are passive and rest are active. Assume, initial seed is 1111 for the 4 cell ACA. Considering the state 1111 is updated five times utilizing the update pattern (4, 3, 1, 2, 4). The state transitions are $1111 (4) \rightarrow 1110 (3) \rightarrow 1100 (1) \rightarrow 1100 (2) \rightarrow 1000 (4) \rightarrow 1000$. Now, again applying the backward transitions on the state 1000 considering the reverse update pattern i.e. (4, 2, 1, 3, 4), we can reach to the initial state 1111. The detail transitions are $1000 (4) \rightarrow 1000 (2) \rightarrow 1100 (1) \rightarrow 1100 (3) \rightarrow 1110 (4) \rightarrow 1111$. The cells updated during transitions are shown in bracket.

5 Identification of ACAs with No Fixed Points

This section addresses a scheme to find out a list of ACAs with no fixed points. These ACAs are allowed only to participate for the design of cryptosystem. We now use the idea of a graph, named *fixed point graph* (FPG), that expedite to identify fixed points of an ACA.

Fixed point graph (FPG)

The FPG for an ACA is a directed graph, in which the vertices of the graph represents the passive RMTs of the ACA. In order to construct the FPG for an ACA, a forest is formed considering the passive RMTs as individual vertices. Now, we draw a directed edge from vertex u to vertex v , if u and v are related following Table 2. For example, if RMTs 1, 3 and 6 are passive, then we can draw directed edges from vertex 1 to vertex 3 and vertex 3 to vertex 6. However, it is not allowed to draw a directed edge from vertex 6 to vertex 1 as these two RMTs are not related (see Table 2).

Example 2. This example explains the procedure of constructing FPG for rule 105 ACA. RMTs 1, 3, 4 and 6 are passive for this rule. Now, the vertices of the FPG are 1, 3, 4 and 6 (Fig. 3). Now assuming, the first vertex as RMT 1, we find the next RMTs for vertex 1 as RMTs 2 and 3 (from Table 2). Since, RMT 3 is the only vertex, we draw directed edge from vertex 1 to vertex 3 (see Fig. 3). Similarly, RMTs 6 and 7 are the next possible RMTs for vertex 3. So, we only draw the directed edge from vertex 3 to 6 as RMT 7 is absent (Fig. 3). Subsequently, we draw the directed edges for all the vertices and the graph is the desired FPG for rule 105 ACA (Fig. 3).

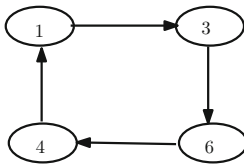


Fig. 3. FPG for rule 105 ACA

Considering the FPG of an ACA, the fixed points for the ACA can be easily identified. In order to find a fixed point of an n -cell ACA, we begin from a vertex and test whether the vertex can be reached after visiting n vertices (a vertex may repeated) including the starting vertex. If it is possible, then the sequence of vertices, visited, is the RMT sequence which is a fixed point.

Example 3. *Here, we exhibit the procedures for finding the fixed points of 4-cell rule 105 ACA. The FPG for rule 105 is shown in Fig. 3. Beginning from vertex 1, we can visit to vertex 3 then vertex 3 to vertex 6, vertex 6 to vertex 4 and finally reach at vertex 1 (see Fig. 3). So, the RMT sequence (1, 3, 6, 4) is the fixed point for this 4-cell ACA, as total number of visited vertices is 4 (see Fig. 3). Similarly, again from vertex 3 we can visit to vertex 6 and finally reach at vertex 3 visiting vertices, 4 and 1. The RMT sequence (3, 6, 4, 1) is the another fixed point. We can also find two more fixed points starting from vertex 6 (RMT sequence (6, 4, 1, 3)) and vertex 4 (RMT sequence (4, 1, 3, 6)). Hence, we get total 4 fixed points for 4-cell rule 105. It is also note that, an n -cell rule 105 ACA contains a fixed point if n is divisible by 4.*

Applying the concept of FPG on all the 16 ACAs of Table 3, we find three such ACAs (rule 51, 57 and 99), which do not have any fixed points for any cell length. These three ACAs are suitable for the design of cryptosystem. However, out of these three ACAs we choose rule 57 for the design of the proposed cryptosystem.

6 Performance Analysis

A desirable property for any cryptosystem is that a small change in plaintext should result a significant change in ciphertext. Changing the value of an arbitrary bit in the plaintext or in the key should produce a change of nearly half of the values of the bits in the ciphertext, which is called avalanche property [11]. In order to study the avalanche property, we have experimented the proposed cryptosystem. For the experiment, we evolve 64 cell rule 57 ACA under fully asynchronous update scheme. A randomly selected ACA state is the plaintext to the scheme. It is observed from the experiment that changing an arbitrary bit value in the plaintext, changes nearly half of the bit values in the ciphertext after around 32000 iterations. We plot the avalanche property of the cryptosystem considering number of iterations and the number of cell changed in Fig. 4. It is shown that after 32000 iterations, the proposed cryptosystem achieves the avalanche property (Fig. 4). Hence, we consider 32000 is the key length for the proposed 64-bit ACAs based cryptosystem. It is also note that, the proposed cryptosystem can also be implemented for 128-bit block cipher. However, it increases the number of iterations of the ACA to achieve the avalanche effect. Hence, it is note that the number of iterations required to achieve avalanche property depends on the size of ACAs.

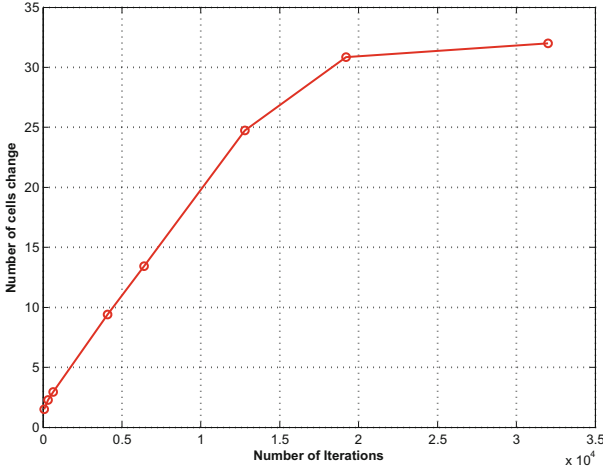


Fig. 4. Avalanche property for 64-cell rule 57 ACA

6.1 Illustration of the Cryptosystem Using Rule 57 ACA

Let us consider a statement “CAN MACHINE BEHAVE LIKE HUMAN?”. We now encrypt this statement using rule 57 ACA. We select 8 characters for the first block of encryption including the space between the characters (i.e. CAN MACH). First we convert each character into 8 bit binary from their ASCII values. Since, the proposed cryptosystem is designed for 64 bit block cipher, we select only 8 characters. We now apply rule 57 and evolve it for 32000 times on this 64 bits. Each time the arbitrary updated bit is stored for the future use. The sequence of randomly updated bit (*update pattern*) is the key for the cryptosystem. After 32000 iterations, we again convert all the scrambled 64 bits to hexadecimal characters. The desired ciphertext is “B5D6E6B2F2B637A5”.

The decryption of the ciphertext is just the opposite of the encryption using reverse update pattern. Similarly, we select another 8 characters for the next block of encryption. However, now the key is the update pattern that was generated during the first block of encryption.

6.2 Security Analysis

The security of any cryptosystem is treated as the process of analysing the robustness of the cryptosystem against various known attacks. However, the design rationale is the basic requirement to achieve the security of the cryptosystem. Diffusion and confusion are the most desirable criteria for a block cipher [12]. Diffusion is the process of rearranging or spreading out the bits in the message so that, any redundancy in the plaintext is spread out over the ciphertext. The resulting cryptosystem achieves this due to the use of ACA. Since, ACA is a non-deterministic system, so cells in ACA are independent and

it is hard to predict which cell is going to update for the next evolution. Hence, after some evolutions of an ACA, the plaintext characters affect many ciphertext characters. Similarly, confusion makes the relationship between the key and ciphertext as complex as possible. In the designed cryptosystem, the confusion can be accomplished by keeping the key size sufficiently large and the use of ACA. In order to achieve the avalanche property, we evolve the designed cryptosystem 32000 times (Fig. 4). So, the length of key for the cryptosystem is 32000. Hence, with this sufficiently large key space and the randomness provided by ACA, the proposed cryptosystem achieves confusion.

Since, the cryptosystem is designed for 64-bit block of plaintext, there are 64 possibilities of keys for each iteration. There are total 32000 iterations, so, total possible keys for the cryptosystem is $2^{6 \times 32000}$. Hence, brute-force attack is practically infeasible with such a huge key space. Further, the use of ACA makes the key complex, which puts the attacker in difficult to deduce the key. Changing the block size 64 to 128 bits, makes the key space large. This means greater security, however, it compromises the speed of encryption and decryption.

6.3 The Comparison

The performance of the proposed ACAs based cryptosystem is compared with other well known cryptosystems namely Data Encryption Standard (DES) and Advanced Encryption Standard (AES). Table 4 shows the comparison result between the cryptosystems considering various factors. The factors considered for the comparison are imported from [7]. Column 1 of Table 4 reports the factors used for the comparison while column 2 and column 3 shows the existing cryptosystem DES and AES respectively. Whereas, last column of Table 4 notes the proposed cryptosystem. It is shown that the proposed ACAs based cryptosystem is competitive with the existing well known cryptosystems.

The limitation of this cryptosystem is the large key length. It may difficult to store such a long key to implement. The reduction of the key length, however, is possible.

6.4 Reduction of Key Length

The main limitation of the cryptosystem during implementation is the key length. However, we can manage this key length using a pseudo random number generator (PRNG). At the encryption end we can generate the update pattern from a PRNG using an arbitrary seed of length 64 bit. The seed to the PRNG can be used as the key for the cryptosystem instead of the whole update pattern. Now we can securely send this key (seed to the PRNG) to the receiver. The receiver can generate the same update pattern from the particular PRNG using the seed. The receiver use the update pattern in reverse order for the decryption process. However, the key space for the cryptosystem reduces to 2^{64} . It may also increase the vulnerability of the scheme to possible attacks.

Table 4. Comparison between DES, AES and the Proposed Cryptosystem

Factors	DES	AES	Proposed cryptosystem
Block size	64 bits	128, 192 and 256 bits	64 bits
Cipher type	Symmetric block cipher	Symmetric block cipher	Symmetric block cipher
Cryptanalysis resistance	Vulnerable to differential and linear cryptanalysis	Strong against differential, truncated differential linear and square attack	Strong against Brute force attack
Key length	56 bits	128, 192 and 256 bits	28 KB
Key space	2^{56}	2^{128} , 2^{192} and 2^{256}	$2^{6*32000}$
Security	Proven inadequate	Considered secure	Confusion, diffusion and avalanche property achieved (secured)
Hardware realization	Yes	Yes	Yes

The main advantage of this cryptosystem is, it can easily be implemented in hardware, as ACAs can be realized in hardware [13]. Further, CAs can also be used as a good PRNG [14].

7 Summary

This work has proposed an ACAs based symmetric key cryptosystem for block cipher encryption. ACAs with the property of backward transitions using reverse update pattern are used for this design. 16 ACAs (out of 256), with property of backward transition are identified using a proposed theorem. 3 ACAs (out of 16) with no fixed are identified utilizing the concept of FPG. Rule 57 ACAs is used for the proposed cryptosystem. It has also experimentally shown, the proposed cryptosystem achieves the avalanche property after 32000 iterations. Randomly selected cells of the ACA (*update pattern*) are used as the secret key for the cryptosystem. The robustness of the scheme against some cryptanalytic attack is discussed and shown the brute-force attack is practically infeasible for this cryptosystem. The effectiveness of the cryptosystem is compared with other cryptosystems and report that the designed cryptosystem is competitive with other cryptosystems. The limitation of the designed cryptosystem is the large key length. However, it is managed by using PRNG. Finally, we have also reported, it is possible to realize the cryptosystem in hardware.

References

1. von Neumann, J.: The Theory of Self-reproducing Automata. University of Illinois Press, Urbana and London (1966). Edited by Burks, A.W
2. Bao, F.: Cryptanalysis of a new cellular automata cryptosystem. In: Safavi-Naini, R., Seberry, J. (eds.) ACISP 2003. LNCS, vol. 2727, pp. 416–427. Springer, Heidelberg (2003)

3. Kari, J.: Cryptosystems based on reversible cellular automata (1992, preprint)
4. Fatès, N., Thierry, E., Morvan, M., Schabanel, N.: Fully asynchronous behavior of double-quiescent elementary cellular automata. *Theor. Comput. Sci.* **362**, 1–16 (2006)
5. Wolfram, S.: *Theory and Applications of Cellular Automata*. World Scientific, Singapore (1986)
6. Sethi, B., Roy, S., Das, S.: Asynchronous cellular automata and pattern classification. *Complexity* (2016). doi:[10.1002/cplx.21749](https://doi.org/10.1002/cplx.21749)
7. Mahajan, P., Sachdeva, A.: A study of encryption algorithms AES, DES and RSA for security. *Comput. Sci. & Tech. Netw. Web Secur.* **13**, 15–22 (2013)
8. Sethi, B., Fatès, N., Das, S.: Reversibility of elementary cellular automata under fully asynchronous update. In: Gopal, T.V., Agrawal, M., Li, A., Cooper, S.B. (eds.) TAMC 2014. LNCS, vol. 8402, pp. 39–49. Springer, Heidelberg (2014)
9. Kari, J.: Reversible cellular automata. In: De Felice, C., Restivo, A. (eds.) DLT 2005. LNCS, vol. 3572, pp. 57–68. Springer, Heidelberg (2005)
10. Sarkar, A., Mukherjee, A., Das, S.: Reversibility in asynchronous cellular automata. *Complex Syst.* **21**, 71–84 (2012)
11. Feistel, H.: Cryptography and computer privacy. *Sci. Am.* **228**, 15–23 (1973)
12. Shannon, C.E.: Communication theory of secrecy systems. *Bell Syst. Tech. J.* **28**, 656–715 (1949)
13. Das, S., Sarkar, A., Sikdar, B.K.: Synthesis of reversible asynchronous cellular automata for pattern generation with specific hamming distance. In: Sirakoulis, G.C., Bandini, S. (eds.) ACRI 2012. LNCS, vol. 7495, pp. 643–652. Springer, Heidelberg (2012)
14. Karimi, H., Hosseini, S.M., Jahan, M.V.: On the combination of self-organized systems to generate pseudo-random numbers. *Inf. Sci.* **221**, 371–388 (2013)

A Random Key Generation Scheme Using Primitive Polynomials over GF(2)

Inderjeet Singh^(✉) and Alwyn R. Pais

National Institute of Technology Surathkal, Mangalore, Karnataka, India
inderjeet231990@gmail.com, alwyn@nitk.ac.in

Abstract. A new key generation algorithm is proposed using primitive polynomials over Galois Field GF(2). In this approach, we have used MD5 algorithm to digest the system time and IP address of the system. The combination of these digest values acts as random seed for the key generation process. The randomness test for the generated key is performed by using Blum Blum Shub (BBS), Micali-Schnorr and Mersenne Twister (MT19937) PRNG algorithms. The generated key has been compared on the basis of the combination of 2 bit, 3 bit, 4 bit and 8 bit count values of 0's and 1's. In this paper, we have used chi squared test, R squared test and standard deviation to check the randomness of the generated key. We have analyzed our result based on the above three criteria and observed that the proposed algorithm achieves lower dispersion in 72.5% of the test cases, lower error rate in 61.6% of the test cases and higher fitness value in 68.3% of the test cases.

Keywords: Primitive polynomials · Key generation · BBS · GF(2) · MT19937 · MD5 · IP

1 Introduction

In cryptography, key generation is the initial step for performing encryption and decryption of the message. The generated key can be used in symmetric-key algorithms like DES or AES where only single key is used for encryption decryption process or it can be used in asymmetric key algorithms like RSA where a pair of keys are required for preserving the privacy of the message. In asymmetric key algorithms, if the sender wants to send the secret message, he must encrypt it using his own private key and by using receiver's public key, he will encrypt again for maintaining the confidentiality of the message. After receiving the encrypted message, decryption will be performed by receiver using his private key and from the sender's public key. Though the symmetric key algorithms are extremely secure, it serves with a major disadvantage. Sharing of key with both sender and receiver may compromise the security if someone gets their hands on symmetric key.

In this paper, the key generation process is carried out using primitive polynomials over Galois field GF(2). As the system IP and system time are always unique, we are considering them as random seed. We are using MD5 algorithm

for digesting the combination of these two values. The MD5 is the cryptographic message-digest hash function which takes arbitrary length input and produces a 128-bit hash value as output.

Primitive polynomials are irreducible polynomials. Irreducible polynomials are those polynomials which cannot be factored into two non constant polynomials over the similar field. $x^2 + x + 1$ is considered to be irreducible over GF(2), but $x^2 + 1$ is not, since $(x + 1)(x + 1) = x^2 + 2x + 1 \equiv x^2 + 1 \pmod{2}$. Let q be any prime power and n be any positive integer. So the number of primitive polynomials of degree n over GF(q) can be determined by:

$$a_q(n) = \frac{\phi(q^n - 1)}{n} \quad (1)$$

where $\phi(n)$ is the euler totient function. GF(p) is called the prime field of order p where the p elements are the values from 0 to $p - 1$. Also, $a = b$ in GF(p) is equivalent to $a \equiv b \pmod{p}$.

The rest of the paper is organized as follows. An assessment of the existing task is given in Sect. 2. The proposed method is explained in Sect. 3. Experimentation and results are given in Sect. 4. Section 5 concludes the work with some future research directions.

2 Literature Review

The key which is used in [1] uses a random odd integer specified in a certain range which improves the security of fully homomorphic encryption over integers. The generation of primitive polynomials and performance analysis over the result is described in [2]. A chaos based pseudo random number generator is described in [3] which uses timing based reseeding method for seed generation. It analyses the random properties of binary number sequences and considered various algorithms for the generation of primitive polynomials. Random sequence has been generated using genetic algorithm [4] and the task has been done to accomplish the linearity of Linear Feedback Shift Register (LFSR) by adding Genetic Algorithm (GA) and try to make a quite non linear generator. Construction of Pseudorandom Number Generator (PRNG) with seed length $O(n \log n)$ has done in [5] and using the Fourier spectrum of the lower degree polynomials over finite fields. Labelled trees, logical formulas, XML files are described in [6] for the generation of seed. Using system date and time as seeds, secured random text sequence has generated for CAPTCHA in [7]. The formation of hardware pseudorandom number is discussed in [8] which passes the DIEHARD test. The analysis of the pseudorandom sequence generation of Feedback with Carry Shift Registers (FCSR) and its architecture is explained in [9] which is similar to the Galois architecture of LFSR. The analysis of binary sequence generators using LFSR and the total time needed to predict the randomness is described in [10]. The pseudo random number generators used in our proposed approach of key generation are Blum Blum Shub($x^2 \pmod{N}$ generator) [11], MT19937 [12] and Micali Shnorr [13].

3 Proposed Work

In this paper, we are introducing a random seed as an input to the key generation procedure. The random seed comprises of system time(milliseconds) and IP address of the system which are unique for different systems. These two values are hashed using MD5 [14] and then added so that computation can be done easily with enhanced security. The proposed work is shown in Fig. 1.

In this approach, instead of using a single big integer as a secret [1], we are using primitive polynomials for key generation. Primitive polynomial reduces the probability of breaking the secret key by the adversary. As shown in Fig. 1, the input provided for the key generation process comprises of 128 bit digest values for the system time and IP address. We are using MD5 hash function which takes arbitrary length input and gives 128 bit digest value as output. By adding these two digest values, we will get 128 bit random seed value in the worst case. The probability of finding this value will be $\frac{1}{2^{128}} \approx 0$ i.e. negligible. Now using equation(1), we will determine the number of primitive polynomials over $GF(q)$ for randomly selected degree n . The value of q for Galois field will be considered as 2. For checking the randomness of the seed which is obtained by adding hashed value of IP address and system time, we have checked whether the collision occurred or not after running the program for a specific amount of time. The outcome obtained with zero collision is shown in Fig. 2. The proposed algorithm for key generation is explained below:

After the key generation procedure, we provide the generated keys as input to existing PRNG methods. In this paper, we have used three pseudo random number generators to test our result. These are Blum Blum Shub(BBS), MT19937, Micali-shnorr. All these three algorithms are explained in Sect. 4. Similarly we take output of PRNG methods by providing standard inputs and then we compare the two outputs for randomness using three criteria.

Theorem 1. *The generated key using our proposed algorithm is pseudorandom:*

Proof. Formally, let S and T be finite sets and let $F = \{f : S \rightarrow T\}$ be a class of functions. A distribution D over S is ϵ -pseudorandom against F if for every f

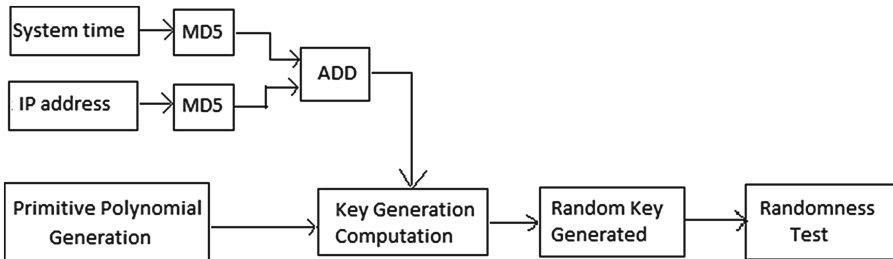


Fig. 1. Proposed system

```

ip hash code :174329070
sys time(in milliseconds) is 1465033209351
-----collision check-----
623691176
623692155
623693187
623694315
623695282
623696154
623697186
623698186
623699282
623700282
623701314
623702698
623703666
623704797
623705829
623706829
-----No collision detected-----

```

Fig. 2. Combination of the hashed value of system time and system IP with zero collision

Algorithm 1. Key Generation Algorithm Using Primitive Polynomials

Require: MD5(.) hash function, x (system time) and y (IP address)

Ensure: r as Randomized key

- 1: Let $x_1 = \text{MD5}(x)$ and $y_1 = \text{MD5}(y)$
 - 2: Let $x' = x_1 + y_1$
 - 3: Let N denote the number of primitive polynomials over GF(2). Calculate N using equation(1)
 - 4: Let $F_{2^q} = \{p_1, p_2, p_3, \dots, p_q\}$ be the set of primitive polynomials over finite fields of degree q .
 - 5: Consider $p(x) \in_R F_{2^q}$. Substitute x with x' in $p(x)$.
 - 6: $r = p(x')$
 - 7: Perform the randomness test over r
-

in F , the statistical distance between the distributions $f(X)$, where X is sampled from D , and $f(Y)$, where Y is sampled from the uniform distribution on S , is at most ϵ .

Our proposed algorithm A consist of the following functions:

$f_1 : \{0, 1\}^* \rightarrow \{0, 1\}^k$ represents MD5 hash function which maps the bit string of arbitrary length and gives k bit output, where $k=128$.

$f_2 : \{0, 1\}^k * \{0, 1\}^k \rightarrow \{0, 1\}^{k+1}$ be the mapping which shows the addition of the k bit hash values to obtain k or $k+1$ bit output (in worst case). Let $F_{2^q} = \cup_{i=1}^{i=q} p_i$ be the set of primitive polynomials p_i over finite field GF(2) from degree 1 to q .

$f_3 : F_{2^q} * \{0, 1\}^k \rightarrow \{0, 1\}^{k+q}$ be a function which takes two input values namely, $p \in F_{2^q}$ and the value obtained from the function f_1 and f_2 . Now, let us assume that $A(x_1, x_2)$ be our proposed algorithm which takes x_1 (system time) and x_2

(IP address) as seed parameters. Consider the given equation (2).

$$P[A(x_1, x_2) = r | x_1 \in Z_n, x_2 \in Z_n] - P[r \in_R \{0, 1\}^*] \leq \varepsilon \quad (2)$$

Now, we have to show that the probability of determining the generated key value r by the adversary is negligible when x_1 and x_2 are used as input to our proposed algorithm A . The equation (2) is equivalent to equation (3):

$$P[f_3(p_i, f_2(f_1(x_1), f_1(x_2))) = r | p_i \in_R F_{2^q}, x_1 \in_R Z_n, x_2 \in_R Z_n] - P[r \in_R \{0, 1\}^{q+k}] \leq \varepsilon \quad (3)$$

The equation (3) shows the probability of determining key value r of length $q + k$ is negligible when x_1 and x_2 are hashed using function f_1 and performed addition using function f_2 and the obtained value is provided as input to the polynomial p_i using function f_3 . Since p_i is randomly chosen from finite polynomial field F_{2^q} , equation (3) can be effectively reduced to the correctness of equation (4).

$$P[f_1(x) = r | x \in_R Z_n] - P[r \in_R \{0, 1\}^k] \leq \varepsilon \quad (4)$$

From the definition of MD5, the equation (4) shows that for any large string x which is an input to the function f_1 , over the range of Z_n , the probability of determining $f_1(x) = r$ is negligible. Since f_1 is pseudorandom (from equation (4)), thus we can say that f_2 is also pseudorandom which leads to the pseudorandomness of f_3 (shown in equation (3)). As equation (3) is equivalent to equation (2), so we can say that our proposed algorithm A is pseudorandom.

From the above function f_1 we can easily see that the probability of determining k bit hashed value as output is negligible. But in our case, $k = 128$ bits which is fairly less in number. Using primitive polynomial of degree q makes the key length size of $q + k$ bits which is far greater than k bits. Increasing the key length size from k to $q + k$ increases the randomness of the generated key. i.e. $\frac{1}{2^{q+k}} \approx 0$ which is more stronger case as compared to $\frac{1}{2^k} \approx 0$. \square

4 Experiments and Results

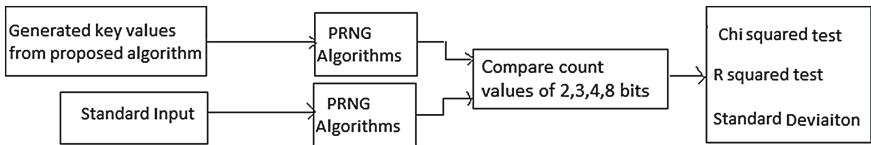
We have generated the keys randomly by executing our experiments for a specific amount of time so that the randomness test can be easily performed over it. Table 1 shows the key length of the generated keys from the primitive polynomials over GF(2). We can easily see that as the degree is increasing, the key length is also increasing. When the degree is selected randomly as 20, the key length of the generated key is 809 bits. The adversary will try 2^{809} possibilities for finding the key with the probability of $\frac{1}{2^{809}}$ which is negligible.

After the generation of the key values, randomness test needs to be performed over the generated key using these three criteria: standard deviation, chi squared test, R squared test. The experimental setup for performing the randomness test is shown in Fig. 3.

The randomly generated key values from the proposed approach are given as input to the standard PRNG algorithms such as Blum Blum Shub, MT19937,

Table 1. Length of keys generated from primitive polynomials over GF(2)

Degree(n)	No. of primitive polynomials	Keylength(bits)
2	1	150
3	2	172
4	2	194
5	6	203
6	6	243
7	18	283
8	16	324
9	48	374
10	60	405
11	176	445
12	144	487
13	630	526
14	756	570
15	1800	607
16	2048	647
17	7710	680
18	7776	728
19	27594	755
20	24000	809

**Fig. 3.** Randomness test

Micali-Schnorr. The result constitutes the chi squared, R squared and standard deviation values for n bit count values of 0's and 1's. Secondly, we have provided the standard input to their respective algorithms as input. The obtained result is compared on the basis of the above mentioned three criteria for n bit count values. In this paper, we have considered $n=2,3,4,8$. Here the 2 bit count values indicates the frequency distribution of this set of combinations 00,01,10,11 in the generated key. Similarly we can have different set of combinations when using $n=3,4,8$. We have performed randomness test using these pseudo random number generators: Blum Blum Shub(BBS), MT19937, Micali-Schnorr. These PRNG's are explained below:

Blum Blum Shub(BBS): BBS is a secured PRNG and can be represented in equation(5):

$$x_{n+1} \equiv x_n^2 \pmod{M} \quad (5)$$

where the x_{n+1} represents the bit parity. M is the product of two large primes p and q . The seed x_0 should be taken in such a way that the GCD of x_0 and M should be 1 i.e. coprime to M . The security of BBS is discussed in [15] by extracting $O(n \log M)$ bits on each iteration where M is the modulus (a Blum integer).

MT19937 [12]: It is also known as Mersenne Twister PRNG. The period length of this algorithm is mersenne prime which is $2^{19937} - 1$. It has higher order equidistribution than any other PRNG. In spite of depending on the system, it has proved to be the faster than `rand()`. This algorithm uses memory efficiently as it consumes only 624 words of working area.

Micali-Schnorr [13]: It is a modification to the RSA pseudo random bit generator and proves to be more efficient and cryptographically secure. Considering the distribution of $x^e \pmod{n}$ for any random r bit sequence. This proved to be indistinguishable by all polynomial-statistical tests from the uniform distribution of integers in the interval $[0, n - 1]$.

The randomness test for the generated keys is conducted based on the following three criteria as explained below:

Standard Deviation [16]: It is a measure which is used to determine the variation or dispersion of a set of data values. Standard deviation is said to be low when the data points are closer to the mean of the set and high standard deviation indicates that the data points are much more dispersed over a large range of values. Low dispersion value of generated keys implies high randomness.

Chi-squared test [17]: It is a statistical hypothesis test which are often constructed from a sum of squared errors. It is used to determine whether there is a significant difference between the expected frequencies and the observed frequencies. The lower the value of test, the more will be the randomness of the key.

R^2 test [18]: It is a measure which will give information about the goodness of fit of a model. It can be calculated by using the following equation:

$$R^2 = 1 - \frac{SS_{res}}{SS_{tot}} \quad (6)$$

where SS_{tot} is total sum of squares and SS_{res} is residual sum of squares. The value of R^2 lies between 0 and 1. The closer the value of R^2 to 1 implies more randomness to the generated key.

The comparison of our results with BBS randomization algorithm is shown in Tables 1 and 2. Table 1 shows the dispersion, fitness and error rate of our data compared to random value generated from BBS algorithm using 2 bit, 3 bit, 4 bit and 8 bit frequency distribution testing framework. Similar measure is obtained in Table 2 for random value as output from BBS algorithm alone.

Table 2. BBS with generated keys of proposed algorithm as input

	2 bit	3 bit	4 bit	8 bit
Std Dev(%)	0.2949	0.3761	0.9217	6.2631
R Squared Val	0.9998	0.9174	0.9894	0.9989
Chi Squared Val	2.2791	2.4725	11.1360	257.0781

Table 3. BBS with standard input

	2 bit	3 bit	4 bit	8 bit
Std Dev(%)	0.3680	0.5942	1.1028	6.0113
R Squared Val	0.9787	0.9021	0.9751	0.9961
Chi Squared Val	3.5495	6.1698	15.9404	236.8203

Table 4. Micali-Schnorr with generated keys of proposed algorithm as input

	2 bit	3 bit	4 bit	8 bit
Std dev(%)	0.1127	0.5677	0.9824	6.2727
R squared val	0.9242	0.9668	0.9684	0.9985
Chi squared val	6.3329	5.6324	12.6504	257.8594

The dispersion(standard deviation) value for the 2 bit count, 3 bit count and 4 bit count are less when the generated key is used as input(shown in Table 1) as compared to the dispersion value of Table 2. The R squared value for all the three counts are more in Table 1 as compared to Table 2 and the chi squared value is less for all the values in Table 1 except for 8 bit count value. Now considering the Micali-Schnorr algorithm for randomness test. Table 3 shows the result obtained when the generated keys of our proposed algorithm are used as input to Micali-Schnorr algorithm. The standard deviation values of Table 3 are less than the standard deviation of the Table 4 which represents the measures when standard input is given to Micali-Schnorr algorithm.

The more the R squared values is closer to 1, the more will be the goodness of fit. All the R squared values proves the generated keys are random expect for the 4 bit count. All the chi squared values support the randomness of the generated key as the chi squared value of Table 3 is less as compared to the chi squared value of Table 4.

Table 5 shows the dispersion, chi square value and R square value when the generated keys of our proposed algorithm are given as input. Similar measures are shown in Table 6 when standard input is used in Mersenne twister algorithm. Except chi squared value for 8 bit count, all the test values in Table 5 supports the randomness test for the generated key.

We execute the test cases, as explained in framework, in several samples (sample size is atleast 10) of random output obtained from proposed algorithm,

Table 5. Micali-Schnorr with standard input

	2 bit	3 bit	4 bit	8 bit
Std dev(%)	0.5031	0.5969	1.0109	6.8131
R squared val	0.7392	0.9289	0.9836	0.9970
Chi squared val	6.6356	6.2271	13.3938	304.2031

Table 6. Mersenne twister with generated keys of proposed algorithm as input

	2 bit	3 bit	4 bit	8 bit
Std dev(%)	0.2367	0.5802	1.0906	6.0909
R squared val	0.8889	0.9476	0.9282	0.9992
Chi squared val	1.4689	5.8836	15.5908	244.9453

Table 7. Mersenne twister with standard input

	2 bit	3 bit	4 bit	8 bit
Std dev(%)	0.2797	0.5975	1.1393	6.1136
R squared val	0.7175	0.9448	0.9280	0.9992
Chi squared val	2.0503	6.2381	17.0137	243.1328

BBS, Micali-Schnorr and Mersene twister. After analysis of the result on 2 bit, 3 bit, 4 bit and 8 bit test cases, it is observed that proposed algorithm achieves lower dispersion in 72.5 % of the test scenario. Similarly higher goodness of fit is achieved in 68.3 % of the test result and reduced error rate is obtained in 61.6 % of the test results.

5 Summary and Conclusion

In this paper, we have proposed an algorithm for random key generation using primitive polynomials over GF(2). We have used the combination of hashed value(using MD5) of system IP and system time(in milliseconds) for the generation of random seed and it enhanced the security. The generated key has passed through randomness test in all cases. We have compared our results with the three existing PRNG algorithms such as R squared, chi squared and standard deviation as testing measures and on the basis of bit counts of the generated key values. It can be concluded that the key generated using the proposed algorithm is random.

References

1. van Dijk, M., Gentry, C., Halevi, S., Vaikuntanathan, V.: Fully homomorphic encryption over the integers. In: Gilbert, H. (ed.) EUROCRYPT 2010. LNCS, vol. 6110, pp. 24–43. Springer, Heidelberg (2010)

2. Saxena, N., McClusky, E.J.: Primitive polynomial generation algorithms-implementation and performance analysis. Technical report, vol. 31, Center for Reliable Computing (2004)
3. Li, C.-Y., Chen, J.-S., Chang, T.-Y.: A chaos-based pseudo random number generator using timing-based reseeding method. In: Proceedings of 2006 IEEE International Symposium on Circuits and Systems, ISCAS 2006, p. 4. IEEE (2006)
4. Chegini, M.G., Mehrabi, A.: Intelligent random sequence generating. In: Fifth International Conference on Natural Computation, ICNC 2009, vol. 4, pp. 307–310. IEEE (2009)
5. Lovett, S., Mukhopadhyay, P., Shpilka, A.: Pseudorandom generators for $CC^0[p]$ and the fourier spectrum of low-degree polynomials over finite fields. *Comput. Complex.* **22**(4), 679–725 (2013)
6. Héam, P.-C., Nicaud, C.: Seed: an easy-to-use random generator of recursive data structures for testing. In: 2011 IEEE Fourth International Conference on Software Testing, Verification and Validation (ICST), pp. 60–69. IEEE (2011)
7. Yadav, V.K., Agarwal, S., Uprety, J., Batham, S.: SRTS: a novel technique to generate random text. In: 2014 International Conference on Computational Intelligence and Communication Networks (CICN), pp. 268–272. IEEE (2014)
8. Tkacik, T.E.: A hardware random number generator. In: Kaliski Jr., B.S., Koç, Ç.K., Paar, C. (eds.) CHES 2002. LNCS, vol. 2523. Springer, Heidelberg (2003)
9. Goresky, M., Klapper, A.M.: Fibonacci and Galois representations of feedback-with-carry shift registers. *IEEE Trans. Inf. Theor.* **48**(11), 2826–2836 (2002)
10. Key, E.L.: An analysis of the structure and complexity of nonlinear binary sequence generators. *IEEE Trans. Inf. Theor.* **22**(6), 732–736 (1976)
11. Ding, C.: Blum-Blum-Shub generator. *Electron. Lett.* **33**(8), 677–677 (1997)
12. Konuma, S., Ichikawa, S.: Design and evaluation of hardware pseudo-random number generator MT19937. *IEICE Trans. Inf. Syst.* **88**(12), 2876–2879 (2005)
13. Menezes, A.J., Van Oorschot, P.C., Vanstone, S.A.: Handbook of Applied Cryptography. CRC Press, Boca Raton (1996)
14. Rivest, R.: The MD5 message-digest algorithm (1992)
15. Sidorenko, A., Schoenmakers, B.: Concrete security of the Blum-Blum-Shub pseudorandom generator. In: Smart, N.P. (ed.) Cryptography and Coding 2005. LNCS, vol. 3796, pp. 355–375. Springer, Heidelberg (2005)
16. Bland, J.M., Altman, D.G.: Statistics notes: measurement error. *BMJ* **313**(7059), 744 (1996)
17. Lewis, P.A.W., Goodman, A.S., Miller, J.M.: A pseudo-random number generator for the system/360. *IBM Syst. J.* **8**(2), 136–146 (1969)
18. Wikipedia: Coefficient of determination – Wikipedia, the free encyclopedia (2016). https://en.wikipedia.org/w/index.php?title=Coefficient_of_determination&oldid=723297210. Accessed 4 June 2016

Multi-factor Authentication Using Recursive XOR-Based Visual Cryptography in Online Voting System

P. Sanyasi Naidu¹ and Reena Kharat^{1,2(✉)}

¹ Department of Computer Science and Engineering, GITAM University, Vishakhapatnam, India
snpasala@yahoo.com

² Department of Computer Engineering, Pimpri Chinchwad College of Engineering, Pune, India
reenakharat@gmail.com

Abstract. The democratic government is a system run by the people for the people. All eligible citizens of the country are involved in choosing their candidate and government through voting. People will be motivated to vote if they believe in the correctness of the underlying system. Authentication and confidentiality are the main issues in the online voting system. Proposed authentication system allows only authentic voter to vote. Non-transferable biometric credentials are used so that fake voter can't vote in lieu of legitimate voter. We have used two biometric factors fingerprint and photograph. XOR-based recursive visual cryptography is used for maintaining the confidentiality of voter's database.

Keywords: Security · Online voting · Authentication · Recursive XOR-based visual cryptography · Biometrics

1 Introduction

In democratic country, each individual has right to choose their own government through right to vote act. In current scenario, a voter has to carry a paper based voter ID card to authenticate him/her at the time of voting. This authentication process is manual and paper based. Fake voter may create fake voter identity card with his/her own photograph but with the name of eligible voter. On the voting day, authentication is done by officer. Officer matches photograph in voter identity card with present face of voter. If it is same then voter is allowed to vote. As voter identity card is paper based, fake voter can pass authentication as authentic voter. Fake voter gets access to voting system in lieu of eligible voter. Another problem with this system is fake voter can vote more than one time by creating fake voter identity cards of different eligible voter. Therefore, there is a need to replace manual paper based authentication with digital authentication.

There is a need to provide access to voting system to overseas people, military people and people not present in state on the day so that they can exercise their vote. Trustworthy secure remote voting system will address this need. Secure authentication is one of the important security considerations in online voting system.

NIST has given different security considerations for remote voting system.

- (a) Identification and Authentication - Identification is a process by which a voter is uniquely identified from other voters. In identification phase, voter has to provide unique identity. Authentication is a process by which trust is established in user identity. By this process, only eligible voter is allowed to vote. No voter can cast multiple votes. Therefore, this phase is very essential. In remote voting system, legitimate voter will be provided with credentials at the time of registration. These credentials will be verified in authentication phase. In remote voting system, the system will authenticate itself as legitimate voting system. This is important to establish trust of voter to voting system.
- (b) Confidentiality - Voting system must provide confidentiality of voter registration database which includes data required for authentication. The system must provide secrecy of ballot as well.
- (c) Integrity - Voter's trust in the data and functions provided by the system is referred as integrity.

In this paper we have proposed solution for authentication in online voting system using XOR based recursive visual cryptography. Multiple biometric features are used because it is non-transferable.

Section 2 discusses about different methods for authentication in online voting system proposed by different authors. Section 3 gives background of visual cryptography and recursive XOR-based visual cryptography. Section 4 introduces our proposed registration and authentication model. Section 5 gives detail security analysis of algorithm. Section 6 concludes the paper.

2 Related Work

Online voting system has been divided into four modules as registration, authentication, vote casting and vote counting. Our proposed system addresses the need of registration and authentication.

In [1], author has used PIN for authentication. Voter enters PIN during registration. At the time of voting, voter has to access voting system and enter login Id and PIN for authentication. Kerberos mechanism used in [3] directs user to enter his ID and password for authentication. In [1] and [3], the credentials used are login Id and password or PIN. These credentials are transferable from one user to another. Authentic voter may transfer these credentials to others with or without his will. Anyone having these credentials can login and vote. So, authentication through non-transferable credentials is the need of online voting system.

Biometrics is biological characteristics which includes fingerprints, retinal and iris scanning, hand geometry, voice patterns, facial recognition, DNA and other techniques. Every individual has unique biometric features. Voter can't transfer his biometric feature to anyone. If biometric features are used during authentication phase of voting, then we can identify legitimate voter and fake voter easily. Biometric is non-transferable feature which is used for authentication in online voting system [11].

In [13], Biometric feature and secret voting password provided at the time of registration is used during authentication. In this system, each voter gets smart card containing

his own fingerprint image. During authentication, fingerprint from smart card is matched with fingerprint from database. Drawback of this system is that match is not performed with live fingerprint of voter. So anyone having smart card of eligible voter can vote.

If cryptography and steganography is combined then security of embedded data is enhanced [2]. In [15], LSB technique is used for hiding data in cover image. If color image is used, each pixel contains 24 bits of RGB value. So, total 3 bits can be embedded in each pixel of color image [4]. In LSB technique of steganography, pixels are selected sequentially for embedding information. Malicious user can easily extract secret information from stego-image by reading LSB of each pixel. In [12], biometric security and password security is provided at the same time using cryptography and steganography. Fingerprint is used as secret key. Algorithm uses cover image. Password which is 4-digit PIN is stored in cover image using LSB technique of steganography. During steganography, Pixel selection is done randomly using password as seed. Therefore, it is difficult for malicious user to extract PIN information from stego-image even he gets voting card. During authentication same fingerprint image is used. System does not take live fingerprint of voter. So if eligible voter tells his PIN and gives his card to another person then another person will be able to pass the authentication step as eligible voter.

In [7], image of password is created. Using visual cryptography two shares of password image is created. One share of password is printed on transparency and sends to voter by mail. Another share of password is stored in authentication database server. At the time of voting, share from database is stored on monitor. Voter has to hold received transparency in front of monitor to superimpose it with database share so that original password is revealed. Voter has to take this password and authenticate himself for voting. It is low cost mutual authentication for voters and election servers. The drawback of system is it does not check the person entering password is the legitimate voter or not. So anyone intercepting mail can cast vote in lieu of legitimate voter.

Authors in [6] use identity number stored in card for identification and live fingerprint for authentication. This live fingerprint is matched with fingerprint stored in database. Drawback of this system is original fingerprint is stored as it is in single database at one place. Someone who gains access to this database can change the original fingerprint by fake fingerprint in database. Here, confidentiality of database is not maintained.

Biometric device like fingerprint scanner is costly and not available with every voter. As microphone is cheaper and commonly available device, voice is used for authentication [10]. But fingerprint has less false acceptance rate and false rejection rate compared to that for voice. Therefore, for authentication in voting system, fingerprint is more preferred. Also, multiple votes casting by single voter can be prevented due to biometric authentication.

3 Background

3.1 Visual Cryptography

Naor and Shamir proposed visual cryptography first time in 1994 [5]. The scheme divides each pixel of an image into two pixels forming two shares. When these two shares are superimposed with each other, original secret image is revealed. The original

secret image is revealed only if both shares are genuine. Drawback [9] of this scheme is reduced resolution and contrast.

3.2 XOR-Based Visual Cryptography

Figure 1 shows how black and white pixel of an image is divided into two shares in XOR-based visual cryptography. It shows two options for each pixel. Any one of the option is chosen randomly. From Fig. 1, it is visible that original pixel is reconstructed without any lose in resolution and contrast.

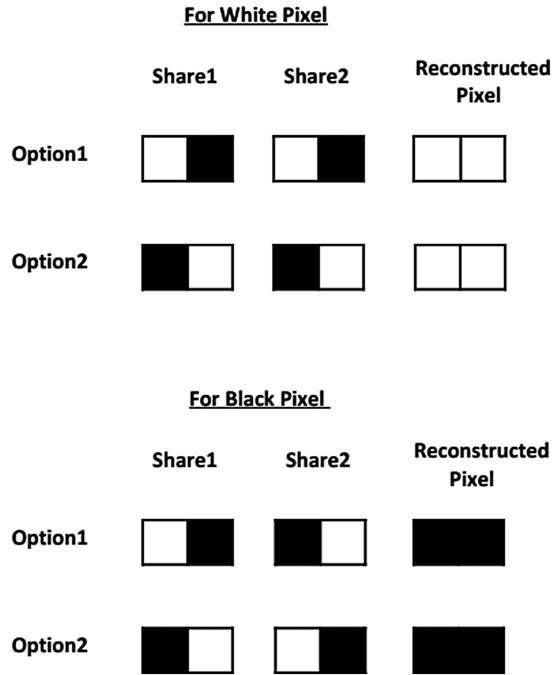


Fig. 1. XOR-based visual cryptography

XOR-based visual cryptography gives meaningful shares [8]. In (2, 2) XOR-based visual cryptography scheme, 2 X 2 matrix is generated. White pixel is represented by one of the column from matrix M1. Black pixel is represented by one of the column from matrix M2. Our proposed scheme also uses XOR-based VC.

$$M_1 = \begin{pmatrix} 1 & 0 \\ 1 & 0 \end{pmatrix} \quad M_2 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

3.3 Recursive XOR-Based Visual Cryptography

In first step shares of original image is generated using XOR-based visual cryptography. These shares are again divided into sub-shares recursively [14].

4 Proposed System

Authentication module in voting system aims at not to allow any other person to vote instead of eligible voter. It also aims at allowing one vote per voter avoiding multiple votes casting by single voter. As authentication is very crucial part in online voting system, we need to use credentials which are non-transferable like biometric features. Our system uses fingerprint and face as biometric measure. If we store fingerprint and photo image in database then registration database will not be confidential. Also if one gets access to database then he can replace images of original voter by fake person's images. He will create fake voter's identification card (VIC) for fake person and fake person will gain access to voting system as eligible voter.

In order to avoid storing original fingerprint and photo images as it is in database, we will store shares of fingerprint and photo images. XOR based (2, 2) secret sharing scheme is used to create two shares from original fingerprint and photo images. Another share of fingerprint and photo images will be stored in VIC which will be issued to voter. We use recursive XOR visual cryptography to split image share further into two different shares. In order to increase security we will use four different database servers each contain one share of one image.

Online voting system has four modules: registration, authentication, vote casting and vote counting. We have proposed system for registration phase and authentication phase.

4.1 Registration Phase

Every individual is required to come for registration with his/her current voting card, address proof and identity card issued by government. A government officer check if individual is eligible to vote by verifying all necessary documents. After verification test is passed, individual is allowed to register. During registration, voter's photograph and fingerprint is taken. Shares of fingerprint image and photo image are created using our proposed scheme shown in Fig. 2.

Steps in registration phase are as follows:

- (1) For each voter, voter identification number (VIN) is created which is unique. It is stored in VIC along with other details for voter.
- (2) Take live fingerprint of voter.
- (3) Use XOR-based visual cryptography to create two shares share_{T1} and share_{T2} of fingerprint-image.
- (4) Take live photograph of voter.
- (5) Use XOR-based visual cryptography to create two shares share_{P1} and share_{P2} of photo-image.
- (6) Share share_{T1} and share_{P1} are stored in Voter's Identification Card (VIC).

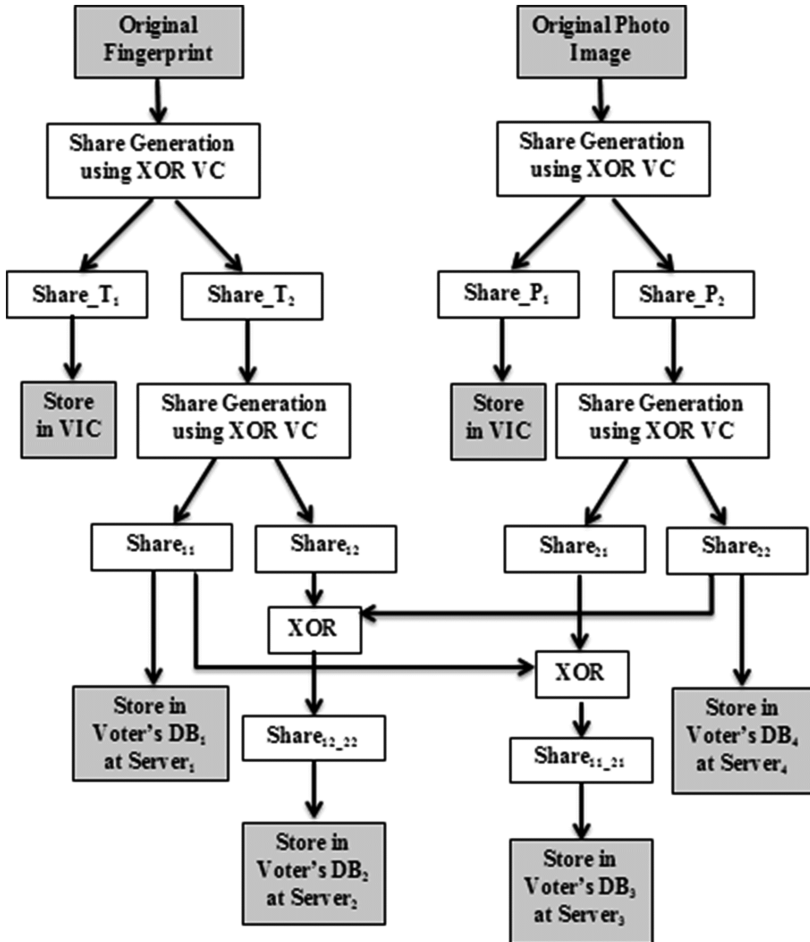


Fig. 2. Share generation in registration phase

- (7) Using recursive visual cryptography, share share_{T,2} is further divided into share_{1,1} and share_{1,2}.
- (8) Share share_{1,1} is stored on server₁.
- (9) Using recursive visual cryptography, share share_{P,2} is further divided into share_{2,1} and share_{2,2}.
- (10) Share share_{2,2} is stored on server₄.
- (11) Share share_{1,2,2,2} is created by XORing share_{1,2} with share_{2,2}. And it is stored on server₂.
- (12) Share share_{1,1,2,1} is created by XORing share_{1,1} with share_{2,1}. And it is stored on server₃.

Example of share generation in registration scheme is shown in Fig. 3. In example, the number 10110010 forms a fingerprint image and the number 11010100 forms a photo image. Figure 3 shows output of each step followed in registration phase.

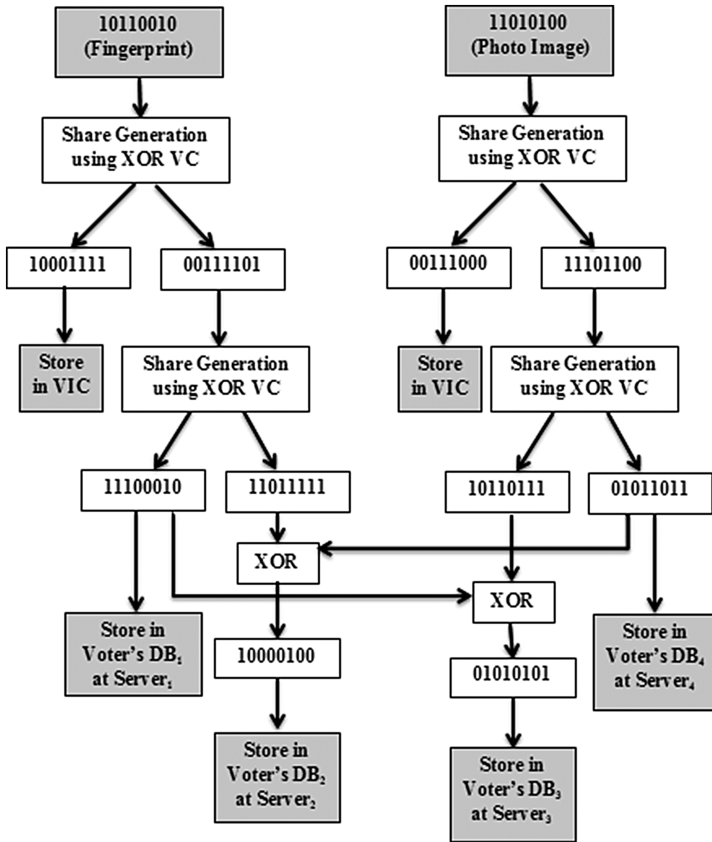


Fig. 3. Example of share generation in registration phase

4.2 Identification and Authentication Phase

At the time of election, a voter has to carry Voters Identification Card (VIC). It contains Voter Identification number (VIN), share of fingerprint image share_{T1} and share of photo image share_{P1}. Identification phase uses VIN and authentication phase uses shares. Figure 4 shows image reconstruction during authentication phase. Following steps are followed in this phase.

- (1) Scan Voters Identification Card (VIC) through smart card reader to read VIN.
- (2) Scan voter's database using VIN and get voters personal information. Validate this personal information with identity card. If it is matched then perform next step.

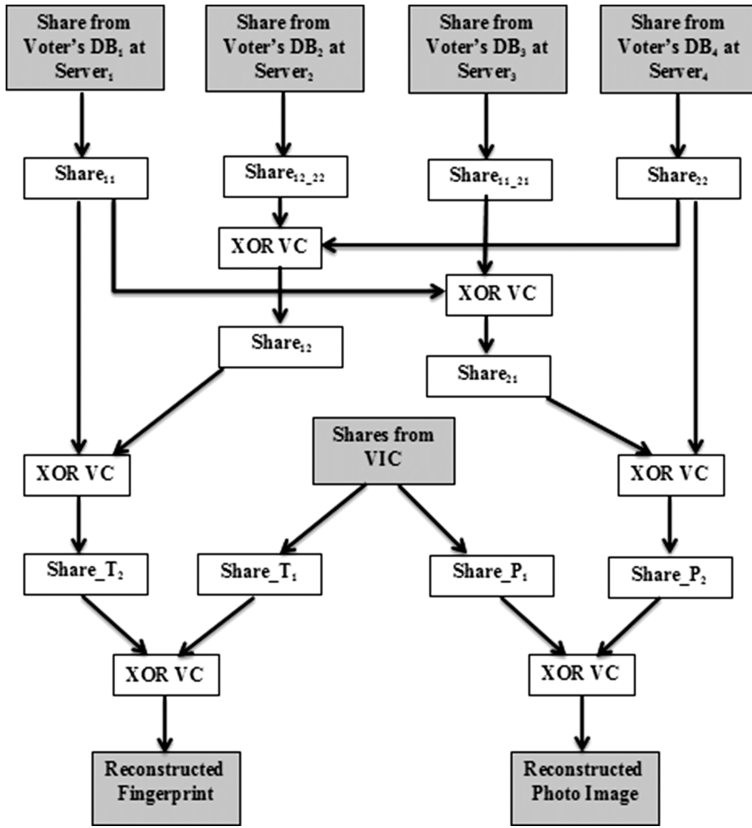


Fig. 4. Image reconstruction in authentication phase

- (3) Read voter's STATUS flag from database. If it is FALSE then reject the voter as vote has been already casted. If STATUS flag is TRUE then go to next step.
- (4) Read share₁₁ from server₁, share_{12_22} from server₂, share_{11_21} from server₃ and share₂₂ from server₄.
- (5) Share share_{12_22} is XORed with share₂₂ to get share₁₂.
- (6) Share share₁₁ is XORed with share₁₂ to get share_{T1}.
- (7) Original fingerprint image is reconstructed by XORing share_{T1} with share_{T2}.
- (8) Share share_{11_21} is XORed with share₁₁ to get share₂₁.
- (9) Share share₂₁ is XORed with share₂₂ to get share_{P2}.
- (10) Original photo image is reconstructed by XORing share_{P1} with share_{P2}.
- (11) Take live fingerprint of voter. If it matches with reconstructed fingerprint then go to next step. Otherwise give three chances to voter to enter fingerprint again. After three chances if fingerprint is not matched then reject him/her as unauthentic voter.
- (12) Take live photograph of voter. If it matches with reconstructed photo_image then the voter is allowed to vote. Otherwise give three chances to voter to take photo again. After three chances, if photo is not matched then reject him/her as unauthentic voter.

Example of image reconstruction during authentication phase is shown in Fig. 5.

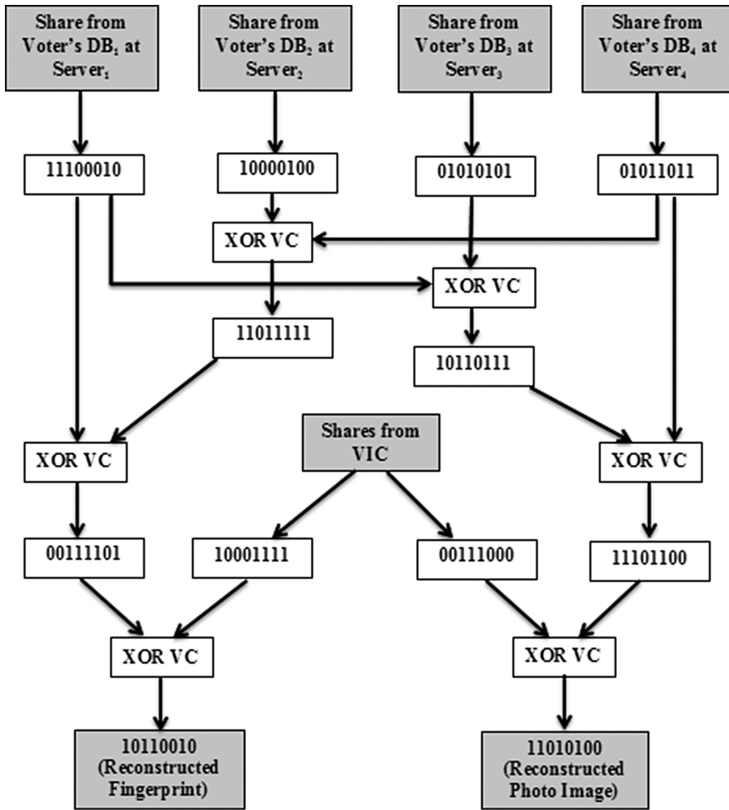


Fig. 5. Example of image reconstruction in authentication phase

5 Results and Security Discussion

During authentication phase, image is reconstructed by using shares from four different database located at different servers and shares from VIC. If fake VIC is generated, then it will contain fake shares. Fake shares are XORed with shares from database. As a result, correct fingerprint image and photo image will not be reconstructed. So, fake voter need to change database as well. There are four shares which are stored at different location. Thus, confidentiality of original fingerprint and photo-image is maintained.

Shares $share_{12,22}$ and $share_{11,21}$ are generated to increase complexity. We need three shares $share_{11}$, $share_{22}$, and $share_{12,22}$ to generate fingerprint. We need three shares $share_{11}$, $share_{22}$, and $share_{11,21}$ to generate photo-image. So if fake voter has to pass authentication with fake VIC, he needs to change shares at four different locations. To change only fingerprint, he needs to change shares at four different locations. It is difficult to gain access of three databases.

In authentication phase, live fingerprint and face is captured and compared with system generated images. If both matches then only voter is allowed for voting. If voter's VIC is lost, another person who found VIC can't cast the vote in lieu of legitimate voter. If VIC is lost, no information is leaked as it contains only shares of images. Original images of fingerprint and photo are not stored in the VIC. System maintains STATUS flag for each voter. Initially STATUS flag is set to TRUE for all voters. After authentication and successful vote casting, STATUS flag is set to FALSE. In authentication phase, the STATUS flag is checked first. If it is TRUE then system allows the voter to perform next authentication steps. If STATUS flag is FALSE then system rejects the voter. Therefore, system guarantees one vote per voter.

6 Conclusion

Biometric features are used for authentication. As biometric features are non-transferable, it gives more secure authentication compare to password authentication. Therefore only legitimate voter can pass authentication and can cast the vote. One vote per authentic voter at a time is guaranteed using status flag. Original fingerprint and photograph are not stored in database or in VIC. Shares created using recursive XOR-based visual cryptography are stored in databases and VIC. Therefore, voter's biometric information remains confidential.

References

1. Al-Anie, H.K., Alia, M.A., Hnaif, A.A.: E-voting protocol based on public key cryptography. *Int. J. Netw. Secur. Appl. (IJNSA)* **3**(4), 87–98 (2011)
2. Abdulzahra, H., Ahmad, R., Noor, N.M.: Combining cryptography and steganography for data hiding in images. *Appl. Comput. Sci.* 128–135 (2014). ISBN: 978-960-474- 368-1
3. Abd-alrazzq, H.K., Ibrahim, M.S., Dawood, O.A.: Secure internet voting system based on public key Kerberos. *IJCSI Int. J. Comput. Sci. Issues* **9**(2), 428–435 (2012). No 3
4. Johnson, N.F., Jajodia, S.: Exploring steganography: seeing the unseen. *IEEE Comput. Mag.* **31**(2), 26–34 (1998)
5. Naor, M., Shamir, A.: Visual cryptography. In: De Santis, A. (ed.) *EUROCRYPT 1994*. LNCS, vol. 950, pp. 1–12. Springer, Heidelberg (1995)
6. Khasawneh, M., Malkawi, M., Al-Jarrah, O., Hayajneh, T.S., Ebaid, M.S.: A biometric-secure e-voting system for election processes. In: *Proceeding of the 5th International Symposium on Mechatronics and its Applications (ISMA08)*, Amman, Jordan, 27–29 May 2008
7. Paul, N., Evans, D., Rubin, A., Wallach, D.: Authentication for Remote Voting. Workshop on Human-Computer Interaction and Security Systems, Ft. Lauderdale, FL, 6 April 2003
8. Tuyls, P., Hollmann, H.D.L., Lint, J.H., Tolhuizen, L.: A polarisation based visual crypto system and its secret sharing schemes. Available at the IACR Cryptology ePrint Archive. <http://eprint.iacr.org/2002/194/>
9. Tuyls, P., Kevenaar, T.A.M., Schrijen, G.-J., Staring, T., van Dijk, M.: Visual crypto displays enabling secure communications. In: Hutter, D., Müller, G., Stephan, W., Ullmann, M. (eds.) *Security in Pervasive Computing*. LNCS, vol. 2802, pp. 271–284. Springer, Heidelberg (2004)

10. Saini, S., Dhar, J.: An eavesdropping proof secure online voting model. In: 2008 International Conference on Computer Science and Software Engineering. IEEE (2008)
11. Vermani, S., Sardana, N.: Innovative way of internet voting: secure on-line vote (SOLV). IJCSI Int. J. Comput. Sci. Issues **9**(6), 73–78 (2012). No 3
12. Katiyar, S., Meka, K.R., Barbhuiya, F.A., Nandi, S.: Online voting system powered by biometric security using steganography. In: 2011 Second International Conference on Emerging Applications of Information Technology. IEEE (2011)
13. Sridharan, S.: Implementation of authenticated and secure online voting system. In: 4th ICCCNT 2013. IEEE, 4–6 July 2013
14. Monoth, T., Babu, A.P.: Recursive visual cryptography using random basis column pixel expansion. In: ICIT 2007, pp. 41–43. IEEE (2007)
15. Wayner, P.: Disappearing Cryptography. AP Professional Books, Boston (1996)

Enhanced Image Based Authentication with Secure Key Exchange Mechanism Using ECC in Cloud

Anurag Singh Tomar¹, Shashi Kant Shankar²(✉),
Manmohan Sharma³, and Aditya Bakshi³

¹ UPES, Dehradun, India

anuragtomar3105@gmail.com

² Shiv Nadar University, Greater Noida, India

shashi2y22@gmail.com

³ Lovely Professional University, Phagwara, India

manmohan_er@yahoo.co.in, addybakshi@gmail.com

Abstract. Cloud computing is the most emerging trend in computing. It provides numerous services like IaaS, PaaS and SaaS. It is a form of pay-per-use based computing. Although it provides tremendous services but there are numerous security issues which need to be resolved. User authentication in cloud computing is the most important step intended towards data security. Image-based authentication is one of the best techniques for user authentication based on the order of selected images. However, key exchange and data encryption in such a complex environment is very difficult to implement. Proposed scheme resolves existing issues of Image based Authentication with Secure key Exchange Mechanism and implements Captcha to detect machine user and Elliptic Curve Cryptography (ECC) for secure key exchange. ECC is the best asymmetric cryptographic algorithm which involves very less key size and computing steps. Hence, it provides a secure layer to cloud computing which deals with user authentication, key exchange and data encryption.

Keywords: Image based authentication · ECC · Cloud · CSP · Captcha · Session key exchange

1 Introduction

Cloud Computing has changed the complete scenario of using computer now-a-days. It provides almost all computing facilities and services that are provided by traditional computing and some new services [2]. Most prioritized and useful among those are data storage, processing capability, pay-per-use basis, availability, handling, data management etc. [3]. All such services are handled and managed at remote side known as Cloud and users are completely unaware about location of those resources as well as involved computing and processing. Cloud computing has witnessed rapid increase in strength and popularity [4].

Cloud computing is a technology which manages data and applications on remote servers. It is accessed by the internet. Cloud users do not need to have any specific

infrastructure to access services [5]. Computer having least resources which is able to access the internet can access any of cloud services. These services are basically categorized in three types. First one is Information-as-a-service (IaaS). It provides data services and management through cloud. Second one is Platform-as-a-service (PaaS). It provides platform facility to users. Platforms may be related to different sections like development platform, OS platform, etc. Last one is Software-as-a-service (SaaS). SaaS provides various software that are hosted on remote servers which can be accessed and used by their clients [6, 7].

Cloud computing is one of the most complex techniques and is very difficult to manage because it involves network, servers, virtual machines, data centers, gigantic machines, etc. It is very useful for users and has attracted many research scholars from different areas [14]. Although it is one of the key technologies in today world but it has numerous security flaws. Secure data access from Cloud Service Provider (CSP) is one of the major concerns for users [8]. Apart from that it is prone to various other threats ranging from database to network attacks. Major of the research in this field are focusing on the security aspect of cloud environment, still there are various issues [9]. However, cloud is expanding its reach to almost every users and providing numerous services but it has to resolve all such security flaws which would make it more secure, reliable, available and trustworthy [10, 11].

In this paper, our focus is concerned towards secure data access from CSP. Proposed scheme enhances existing technique of image-based authentication, key generation, secure session key exchange by using Captcha and Elliptic Curve Cryptography (ECC) [13]. Cloud may contain very sensitive data of users which need intensive care and powerful security. Paper is organized further in four sections. Section 2 reviews existing work whereas security issues in existing technique is dealt in Sect. 3. Section 4 covers proposed enhanced scheme and security analysis is covered in Sect. 5. Finally, the paper is concluded in Sect. 6.

2 Review of Image Based Authentication with Secure Key Exchange Mechanism in Cloud

This section reviews Image based authentication with secure key exchange mechanism in cloud [1]. It involves 3 phases as Registration, Image-based Authentication and Key Exchange phase. Notations that are used in this section is listed in Table 1.

2.1 Registration Phase

Every cloud user has to register itself to CSP. This registration process involves some personal details and password selection. CSP will show set of images and position of those images are completely random which get changed for each session. Every user has to select some images according to his own preference and position of those images based on the order of selected images is the password (pwd) of the user.

2.2 Image-Based Authentication

Proceeding further with the registration phase, every user has to be authenticated by CSP before accessing data. User will be provided with set of images. User has to select those images in same order that has been used at time of registration. Further login will be successful if both patterns match otherwise unsuccessful.

2.3 Key Exchange Phase

It consists some steps which can be used for exchange key in a secure manner. User has to select a large p and find g of group Z_p^* . Calculate key $K = g^r \text{ mod } p$. Then after, user has to calculate three values $P1$, $P2$ and $P3$ where $P1 = g^{r+H(\text{ID}||\text{PWD})} \text{ mod } P$, $P2 = H(\text{ID}||\text{PWD})$ and $P3 = \text{ID}$. CSP will fetch the password based on $P3$ and calculate the $P2$. After successful match of $P2$, CSP will calculate key $K = P1/g^{P2}$.

Table 1. Notations list

Notation	Description
CSP	Cloud service provider
Pwd	Password of user (order of selected images)
P	Prime number
Z_p^*	Group based on P
K	Session key
G	Generator of group Z_p^*
R	Random number
H(.)	Hash function

3 Security Issues in Image Based Authentication with Secure Key Exchange Mechanism in Cloud

This section deals with security issues of Image based authentication with secure key exchange mechanism in cloud [1] and that are as follows:

3.1 User Identification Issue

Proposed Image based authentication scheme [1] does not include any Captcha mechanism so it will not be possible for CSP to segregate between human and machine users. Any self-executable program may find out the correct order of images by applying all permutation and can successfully login into the system on behalf of any specific user and that is the main concern which may lead to security issues.

3.2 Brute Force Attack

Existing technique [1] uses a set of images where user has to select some specific images for password. Anyone can create such kind of program which will find all possible combinations based on m images where n images have to be selected. A single value will be exactly same to password of user. Emergence of high speed and distributed computing can compute all combinations within fraction of seconds. Hence, brute force attack is possible by very small human effort.

3.3 Compromised Session Key

Once the attacker would get the password by applying brute force attack or by some other mechanism [14] then he can capture the parameters $P1$, $P2$, $P3$ during transmission, with the help of that attacker can compute the key $K = P1/g^{H(ID||P^{WD})}$ and once key is compromised then the complete session and data transmission can be compromised very easily.

3.4 Denial of Service Issue

A program with multi-threading running on multiple processors can be designed for automated login to the system. It can login thousands of thousand times in a second and cause denial of service to users. As this scheme does not have any concept of Captcha so such kind of problem is possible and very easy to implement.

4 Proposed Security Improvements

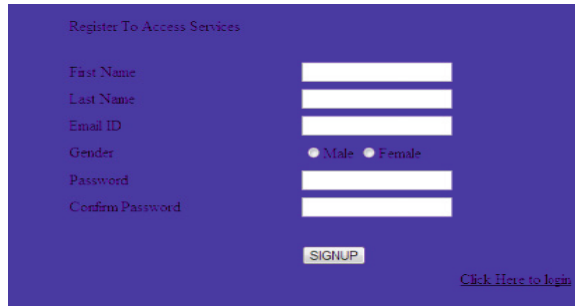
We have proposed image based authentication with Captcha validation for secure login and Elliptic Curve Cryptography (ECC) for secure session key exchange. This section covers both the phases whereas image based registration and login is same as in [1].

4.1 Registration Phase

User will select the set of Images displayed by server as in Fig. 1 and register these set of images in fixed order as in Fig. 2.

4.2 Image Based Login with Captcha Implementation

Proposed scheme involves an advanced security layer towards login step by using Captcha. Captcha is one of the most popular techniques which is automated public Turing test to differentiate between human and machine users [12]. It can be based on various objects like image, text, expression etc. This step really helps to prevent the



Register To Access Services

First Name

Last Name

Email ID

Gender Male Female

Password

Confirm Password

[Click Here to login](#)

Fig. 1. Registration phase

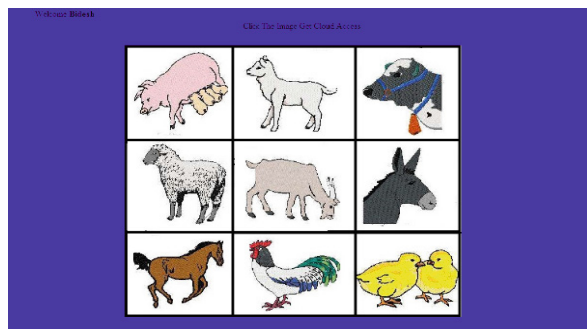
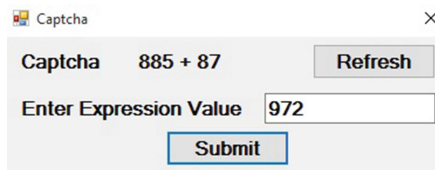


Fig. 2. Image-based authentication



Captcha

Captcha $885 + 87$

Enter Expression Value

Fig. 3. Captcha implementation

attacks on cloud services by machine users. As we know, Captcha is analyzed by human users alone and not by machine users. Only human users (accessing the cloud services) can read the Captcha and can take needed action as with some input form. So, using Captcha technique, no machine based user can crack the password or other confidential information of the users. It also helps to resolve the issue of DoS because user takes some time to enter value of Captcha. Proposed work includes expression based Captcha in Fig. 3 in which any mathematical expression is shown and user has to calculate and enter the result.

4.3 ECC for Secure Key Exchange

Existing technique [1] has issues which can easily compromise the session key. Hence, Elliptic Curve Cryptography (ECC) has been implemented to resolve those issues. It is an asymmetric cryptography which can be used for key exchange as well as encryption and decryption. It can also be used with Elgamal scheme for user authentication. It provides more security than RSA [15] with less steps involved in computation. Moreover, it uses less memory in contrast to RSA because of small key length. However, it provides same level of security with key size of 160 bits in comparison to RSA which uses key size of 1024 bits. Finally, it encodes message on Elliptic Curve and addition or multiplication on that curve is completely different than arithmetic mathematics. Therefore, it is another factor to prove ECC as more secure algorithm than any other cryptographic algorithms.

User will generate the key and that key will be exchanged by using Elliptic Curve Cryptography. ECC is one of the best asymmetric algorithms for encryption and decryption. It encrypts any message on elliptic curve [15] in form of coordinates. Table 2 lists all the steps involved in this phase.

First step in ECC involves agreement of common parameters which is shown in Fig. 4. After that both users choose their private keys and public keys are calculated as in Fig. 5. Message is encoded by receiver’s public key which is encoded on the curve and the same is displayed in the Fig. 6. Finally, message is encrypted and decrypted which is in Fig. 7. Finally session key is calculated as in Fig. 8.

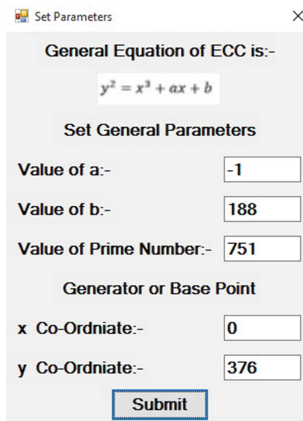


Fig. 4. Common parameters agreement

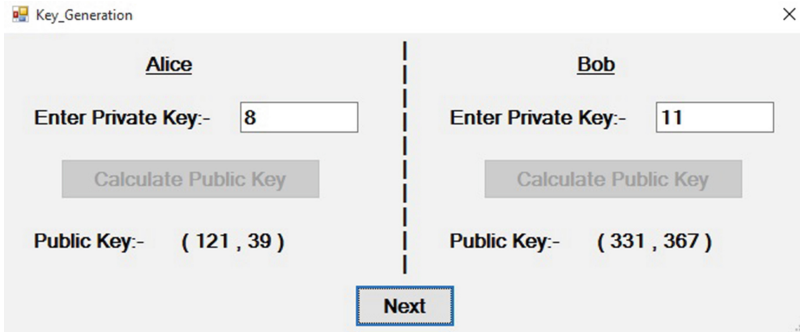


Fig. 5. Key generation

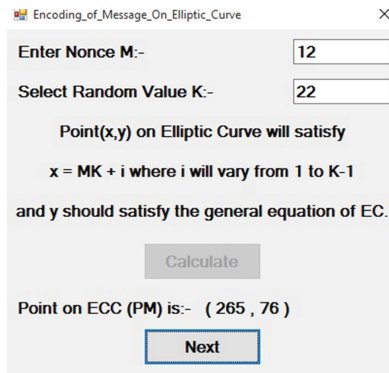


Fig. 6. Nonce encoding

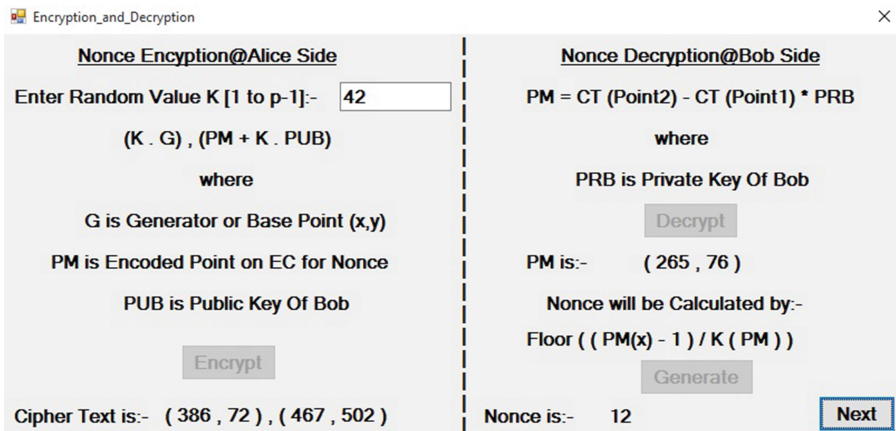


Fig. 7. Nonce encryption and decryption

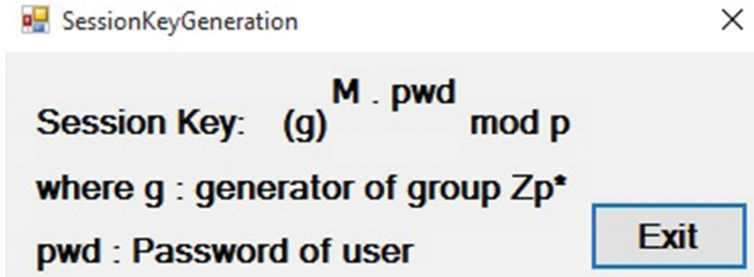


Fig. 8. Session key generation

5 Security Analysis

In this section security of proposed scheme is analyzed and it resists the following attacks.

5.1 Session Key Secrecy

Proposed scheme calculates session key as $(g)^{M \cdot \text{pwd}} \text{ mod } p$ where M is encrypted by using ECC and password (pwd) is only known either to user or to CSP. Session key is linked with user password which maintains its secrecy. Apart from that session key cannot be compromised because ECC encodes message on the curve and involved computation is very different from normal one.

5.2 Insider Attack

Any personnel from CSP may know password of the user but s/he cannot calculate session key because it involves M . Moreover M is encoded and encrypted by ECC. Hence proposed scheme resists against such kind of any inside attacks.

5.3 Man-in-Middle Attack

In the proposed scheme, if any one intercepts the session key exchange data then s/he would not be able to derive the actual session key because of involvement of password and ECC. Apart from that real session key is calculated after the decryption of exchanged data. Hence it resists against man-in-middle attack.

5.4 Replay Attack

In every session password can be changed due to change in order of images at server side as well value of M will be changed so if attacker is able to get the old session key

Table 2. ECC steps

User A (Sender)	User B (Receiver)
<p>Common Parameters Agreement</p> <ol style="list-style-type: none"> Equation:- $y^2 = x^3 + ax + b$ Prime Number p Base Point $B_{(x,y)}$ Co-efficient value 'a' and 'b' Any random value k 	<p>Common Parameters Agreement</p> <ol style="list-style-type: none"> Equation:- $y^2 = x^3 + ax + b$ Prime Number p Base Point $B_{(x,y)}$ Co-efficient value 'a' and 'b' Any random value k
<p>Key Generation</p> <ol style="list-style-type: none"> Select Private Key PR_a Calculate Public Key PU_a $PU_a(x, y) = PR_a * B_{(x,y)}$ where multiplication in Elliptic Curve is repeated addition. 	<p>Key Generation</p> <ol style="list-style-type: none"> Select Private Key PR_b Calculate Public Key PU_b $PU_b(x, y) = PR_b * B_{(x,y)}$ where multiplication in Elliptic Curve is repeated addition.
<p>Encoding of Nonce 'M' (any integral value which will change in every session) on Elliptic Curve as $P_m(x, y)$</p> <ol style="list-style-type: none"> $x = (M * k) + i$ where i will vary from 1 to $k-1$ Put value of x in equation $y^2 = x^3 + ax + b$ and find integral value of y. If integral value is not found then again calculate x by incrementing the value of i. 	
<p>M has been encoded as $P_m(x, y)$ which is further encrypted as CT</p> <ol style="list-style-type: none"> Select any random value R where $R \in (1, p - 1)$ Cipher Text $CT((x_1, y_1), (x_2, y_2))$ is computed as $CT = [(R * B_{(x,y)}), P_m(x, y) + R * PU_b(x, y)]$ and sent \longrightarrow Session key is calculated as: $(g)M.pwd \text{ mod } p$ where g : generator of group Zp^* pwd : Password of user[Table 1] 	<p>Decryption of Cipher Text $CT((x_1, y_1), (x_2, y_2))$</p> <ol style="list-style-type: none"> Compute $PR_b * CT(x_1, y_1)$ i.e. $(PR_b * R * B_{(x,y)})$ Compute $CT(x_2, y_2) - (PR_b * R * B_{(x,y)})$ i.e. $(P_m(x, y) + R * PU_b(x, y)) - (PR_b * R * B_{(x,y)})$ Now, $P_m(x, y)$ has been received. Calculate M as $M = \text{Floor}((x - 1) / k)$ Session key is calculated as: $(g)M.pwd \text{ mod } p$

he can't replay the old session key to hijack the new session hence the Proposed Scheme Resist the Replay Attack.

6 Conclusion

Cloud computing provides numerous services which can be accessed based on users' need and incurred rent. It is also known as pay-per-use based computing. The major issue in cloud computing is of data security. Although various schemes have been developed for user authentication but still there are many concerns related to user authentication and data security.

Image-based authentication with Captcha implementation resolves various existing security issues. It stops automated attacks and helps to identify robots. Users are authenticated by either CSP or any trusted third party. Moreover session key is exchanged between user and CSP by using ECC which provides an additional layer of security. It resolves various other attacks like insider attack, man-in-middle attack, etc. Finally secure transmission of session key leads a better roadway towards data access and transmission between user and CSP. Future work may include recent MCDM technologies which can be clubbed with this technique in cloud environment.

References

1. Tomar, A.S., Tak, G.K., Chaudhary, R.: Image based authentication with secure key exchange mechanism in cloud. In: International Conference on Medical Imaging, m-Health and Emerging Communication Systems (MedCom), pp. 428–431 (2014)
2. Mohamed, A., Grundy, J., Ibrahim, A.S.: Adaptable, model-driven security engineering for SaaS cloud-based applications. *Autom. Softw. Eng.* **21**, 187–224 (2013). Springer
3. Du, Y., Zhang, R., Li, M.: Research on a security mechanism for cloud computing based on virtualization. *Telecommun. Syst.* **53**, 19–24 (2013). Springer
4. Edurado, F.B., Monge, R., Hashizume, K.: Building a security reference architecture for cloud systems. *Requirements Eng.* **21**, 1–25 (2015). Springer
5. Jin, H., Dong, M., Ota, K., Fan, M., Wang, G.: NetSecCC: a scalable and fault tolerant architecture for cloud computing security. *Peer-to-Peer Netw. Appl.* **9**, 1–15 (2014). Springer
6. Hu, P., Sung, C.W., Ho, S., Chan, T.H.: Optimal coding and allocation for perfect secrecy in multiple clouds. *Inf. Forensics Secur.* **11**, 388–399 (2014). IEEE
7. Junwon, L., Cho, J., Seo, J., Shon, T., Won, D.: A novel approach to analyzing for detecting malicious network activity using a cloud computing testbed. *Mob. Netw. Appl.* **18**, 122–128 (2012). Springer
8. Jin, L., Li, Y.K., Chen, X., Lee, P.P.C., Lou, W.: A hybrid cloud approach for secure authorized deduplication. *IEEE Trans. Parallel Distrib. Syst.* **26**, 1206–1216 (2014)
9. Rahat, M., Shibli, M.A., Niazi, M.A.: Cloud identity management security issues and solutions: a taxonomy. *Complex Adapt. Syst. Model.* **2**, 1–37 (2014). Springer
10. Seungmin, R., Chang, H., Kim, S., Lee, Y.S.: An efficient peer-to-peer distributed scheduling for cloud and grid computing. *Peer-to-peer Networking Appl.* **8**, 863–871 (2014). Springer

11. Li, Q., Han, Q., Sun, L.: Collaborative recognition of queuing behavior on mobile phones. *IEEE Mob. Comput.* **15**, 60–73 (2014)
12. Tak, G.K., Badge N., Manwatkar, P., Rangnathan, A., Tapaswi, S.: Asynchronous anti phishing image captcha approach towards phishing. In: *International Conference on Future Computer and Communication*, vol. 3, pp. 694–698. IEEE (2010)
13. Malhotra, K., Gardner, S., Patz, R.: Implementation of elliptic-curve cryptography on mobile healthcare devices. In: *International Conference on Networking, Sensing and Control*, pp. 239–244. IEEE (2007)
14. Tomar, A.S., Jaidhar, C.D., Tapaswi, S.: Secure session key generation technique for group communication. *Int. J. Inf. Electron. Eng.* **2**, 831–834 (2012)
15. Shankar, S.K., Tomar, A.S., Tak, G.K.: Secure medical data transmission by using ECC with mutual authentication in WSNs. In: *4th International Conference on Eco-Friendly Computing and Communication Systems (ICECCS)*, pp. 455–461. Elsevier (2015)

Differential Fault Analysis on Tiaoxin and AEGIS Family of Ciphers

Prakash Dey¹, Raghvendra Singh Rohit², Santanu Sarkar³,
and Avishek Adhikari¹(✉)

¹ Department of Pure Mathematics, University of Calcutta, Kolkata 700019, India
`pdprakashdey@gmail.com`, `avishek.adh@gmail.com`

² Department of Mathematics and Statistics,
Indian Institute of Science Education and Research, Kolkata, India
`iraghvendraro hit@gmail.com`

³ Indian Institute of Technology Madras, Sardar Patel Road, Chennai 600036, India
`sarkar.santanu.bir@gmail.com`

Abstract. Tiaoxin and AEGIS are two second round candidates of the ongoing CAESAR competition for authenticated encryption. In 2014, Brice Minaud proposed a distinguisher for AEGIS-256 that can be used to recover bits of a partially known message, encrypted 2^{188} times, regardless of the keys used. Also he reported a correlation between AEGIS-128 ciphertexts at rounds i and $i + 2$, although the biases would require 2^{140} data to be detected. Apart from that, to the best of our knowledge, there is no known cryptanalysis of AEGIS or Tiaoxin. In this paper we propose differential fault analyses of Tiaoxin and AEGIS family of ciphers in a nonce reuse setting. Analysis shows that the secret key of Tiaoxin can be recovered with 384 single bit faults and the states of AEGIS-128, AEGIS-256 and AEGIS-128L can be recovered respectively with 384, 512 and 512 single bit faults. Considering multi byte fault, the number of required faults and re-keying reduces 128 times.

Keywords: Stream cipher · AEAD · Differential fault analysis

1 Introduction

Authenticated encryption with associated data (AEAD) is a class of cryptographic primitive for privacy of the plaintext and integrity of both plaintext and associated data. CAESAR [1], a competition for authenticated encryption, is targeting to identify a portfolio of AEAD. Initially, fifty seven authenticated encryptions were submitted to CAESAR. However, in the second round of the competition, 29 submissions survived. Tiaoxin and AEGIS family of ciphers are among the 29 selected second round candidates.

Avishek Adhikari—Research supported in part by National Board for Higher Mathematics, Department of Atomic Energy, Government of India (Grant No. 2/48(10)/2013/NBHM(R.P.)/R&D II/695).

Side channel attacks, such as timing analysis, power analysis and fault analysis, target the implementations of ciphers and test the strength of ciphers in such settings. Power and fault analyses are among the most explored types of side channel attacks.

Biham and Shamir [6] first introduced the idea of Differential Fault Analysis (DFA). Subsequently various symmetric ciphers were analyzed using DFA model. Fault attacks study the robustness of a cryptosystem, in a setting which is in general, weaker than its original or expected mode of operation. In a DFA model, during cipher operations, faults are injected. Since the faults flip the corresponding bits, the attack results in a difference in the state. The resulting faulty output, together with the fault free one, are analyzed to obtain full or a part of the secret information. Although optimistic, this model of attack has been shown to be successful against both stream ciphers and as well as against block ciphers. Most of the proposed ciphers in the eStream portfolio are vulnerable to the fault attacks [3–5, 7–14, 16, 20–22]. AES is also highly vulnerable to fault attacks [2, 17, 19, 23].

Tiaoxin [18] and AEGIS [25] are authenticated cryptographic algorithms submitted to CAESAR by Ivica Nikolić and Hongjun Wu et al. respectively. AEGIS family ciphers were first proposed in SAC 2013 [24]. In SAC 2014, Minaud [15] showed linear biases in AEGIS keystream. However, attack complexity in work of Minaud is higher than the exhaustive key search. There are many similarities in the design principle of Tiaoxin and AEGIS family. Both the ciphers use the same technique of injecting message directly into the state to achieve authentication almost for free. Both ciphers take advantage of AES-NI instructions to achieve outstanding speed in software. The security of both the ciphers relies on the following two assumptions:

- A Each Key-IV pair is used to protect only one message.
- B If the verification fails, the decrypted plaintext and the wrong authentication tag should not be given as output.

The Tiaoxin and AEGIS designers recommended that Key-IV pair should not be reused. They expressed security concern if all the assumptions are not fulfilled. However, no specific attack was provided. Nevertheless, in the security claims section of the submission document of Tiaoxin, it is stated that

“If the nonce is reused. Obviously in this case high probability trails (that do not need to end in a zero difference) for the Encryption of Tiaoxin-346 can be used to recover state bytes and to compromise the confidentiality.”

Note that one can protect only one message by each Key-IV pair in stream ciphers like Grain. However, there are many papers such as [3, 4] on Grain under fault attack where re-key is used. In [1], it is mentioned about fault attack as follows:

“Sometimes attackers can flip bits in a computation (for example, by firing a laser at a target chip), and deduce secret data from the resulting cipher output.”

The aim of this paper is to strengthen the designers claim by describing a fault attack in a nonce reuse setting that allow the complete key recovery for Tiaoxin and complete state recovery for AEGIS family.

CONTRIBUTION OF THE PAPER: The current paper proposes a differential fault attack model on Tiaoxin and the AEGIS family of ciphers when an adversary has precise control on the fault location and fault timing. The attacker injects single bit faults by re-keying each time to obtain particular state blocks. Then after getting a suitable number of state blocks, the entire state is recovered at a known cycle of operation of the cipher. For Tiaoxin, after reversing the state, the secret key can also be recovered. For AEGIS, the recovered complete state could be used for suitable purposes.

ORGANIZATION OF THE PAPER: The rest of the paper is organized in the following way: In Sect. 2 we provide description of Tiaoxin and AEGIS family of ciphers. The attack model considered in this paper and the attacks are described in Sect. 3. Section 3.5 briefly discusses another attack model to reduce the number of faults and re-keying. Finally Sect. 4 concludes the paper.

2 Description of the Ciphers

In this section we briefly describe (only the relevant parts are described) the ciphers Tiaoxin, AEGIS-128, AEGIS-256 and AEGIS-128L. For a descriptive version of the ciphers, the reader may refer to [18, 25]. Tiaoxin and the AEGIS family of ciphers extensively use one keyed round of AES. So we describe the one keyed round of AES first.

2.1 AES Round Function

A sequence of 16-bytes will be called a *word*. Let A and B be two words. We denote by $AES(A, B)$, the one keyed round of AES applied to A with B as the subkey (word to AES matrix conversion is the standard one). Thus

$$AES(A, B) = \tau(A) \oplus B \text{ where } \tau(\cdot) = \text{MixColumns}(\text{ShiftRows}(\text{SubBytes}(\cdot))).$$

One should note that the AES operations $\text{MixColumns}(\cdot)$, $\text{ShiftRows}(\cdot)$ and $\text{SubBytes}(\cdot)$ are all invertible. Thus if $\tau(A)$ is known one can obtain A uniquely and efficiently. Also if $AES(A, B)$ i.e. $\tau(A) \oplus B$ and B are both known, one can easily recover A .

2.2 Description of Tiaoxin

Tiaoxin-346 has three states T_3, T_4 and T_6 composed of 3, 4 and 6 words respectively. The state update mechanism of Tiaoxin uses a round transformation operation $R(T_s, M)$ with state T_s and a word M as input. The output T_s^{new} of $R(T_s, M)$ is the new state and is given by:

$$\begin{aligned} T_s^{new}[0] &= AES(T_s[s-1], T_s[0]) \oplus M, \\ T_s^{new}[1] &= AES(T_s[0], Z_0), \end{aligned}$$

$$\begin{aligned}
 T_s^{new}[2] &= T_s[1], \\
 \dots, \\
 T_s^{new}[s-1] &= T_s[s-2],
 \end{aligned}$$

where Z_0 is a Tiaoxin constant [18].

The state update operation $update(T_3, T_4, T_6, M_0, M_1, M_2)$ takes three additional words M_0, M_1, M_2 , i.e.

$$update : T_3 \times T_4 \times T_6 \times M_0 \times M_1 \times M_2 \rightarrow T_3 \times T_4 \times T_6$$

The function $update(T_3, T_4, T_6, M_0, M_1, M_2)$ is defined as (See Fig. 1):

$$\begin{aligned}
 T_3^{new} &= R(T_3, M_0); & T_3 &= T_3^{new} \\
 T_4^{new} &= R(T_4, M_1); & T_4 &= T_4^{new} \\
 T_6^{new} &= R(T_6, M_2); & T_6 &= T_6^{new}
 \end{aligned}$$

Tiaoxin ciphertext and tag generation are done in 4 stages: (1) The Initialization (2) Processing the Authenticated Data (3) The Encryption and (4) The Finalization.

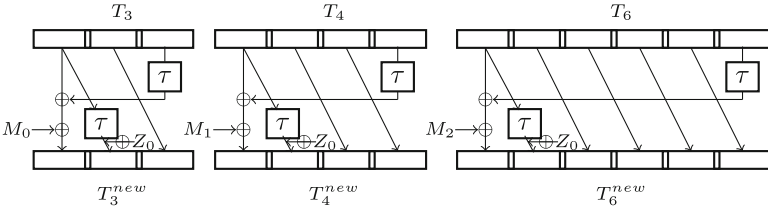


Fig. 1. The $update$ function in Tiaoxin-346

After initialization and processing of the authenticated data, in the encryption stage, at each round i , a plaintext $M_i = M_i^0 || M_i^1$, composed of two words M_i^0 and M_i^1 , is encrypted to the ciphertext $C_i = C_i^0 || C_i^1$, composed of two words C_i^0 and C_i^1 . The encryption at the round i is defined as:

$$\begin{aligned}
 &update(T_3, T_4, T_6, M_i^0, M_i^1, M_i^0 \oplus M_i^1) \\
 C_i^0 &= T_3[0] \oplus T_3[2] \oplus T_4[1] \oplus (T_6[3] \& T_4[3]), \\
 C_i^1 &= T_6[0] \oplus T_4[2] \oplus T_3[1] \oplus (T_6[5] \& T_3[2])
 \end{aligned}$$

2.3 Description of AEGIS-128

Five 128 bit substates S_0, \dots, S_4 constitutes the inner state of AEGIS-128. Let $S_{i,0}, \dots, S_{i,4}$ be the substates at the beginning of round i So we have $S_i = S_{i,0} || S_{i,1} || S_{i,2} || S_{i,3} || S_{i,4}$, where each $S_{i,j}$ is a word and $||$ is the concatenation operator.

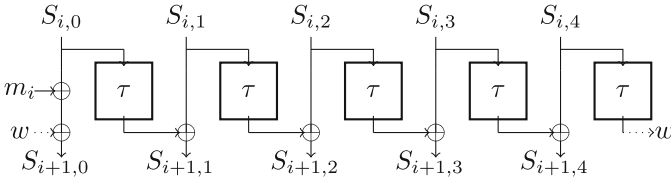


Fig. 2. The state update function of AEGIS-128

At each round i , a 16-byte data block m_i is used to update the state. The new state S_{i+1} is computed as follows:

$$\begin{aligned}
 S_{i+1,0} &= AES(S_{i,4}, S_{i,0} \oplus m_i), \\
 S_{i+1,1} &= AES(S_{i,0}, S_{i,1}), \\
 S_{i+1,2} &= AES(S_{i,1}, S_{i,2}), \\
 S_{i+1,3} &= AES(S_{i,2}, S_{i,3}), \\
 S_{i+1,4} &= AES(S_{i,3}, S_{i,4})
 \end{aligned}$$

Figure 2 represents the state update function of AGEIS-128.

AEGIS-128 ciphertext and tag generation are done in 4 stages: (1) The Initialization (2) Processing the Authenticated Data (3) The Encryption and (4) The Finalization.

AEGIS-128 takes a 128 bit key and 128 bit nonce. After initialization and processing of the authenticated data, in the encryption stage, at each round, a 16-byte plaintext block P is used to update the state, and P is encrypted to C as $C = P \oplus z_i$, where $z_i = S_{i,1} \oplus S_{i,4} \oplus (S_{i,2} \& S_{i,3})$ is the 16-byte block keystream.

2.4 Description of AEGIS-256

At the beginning of the i -th round, the (6-word) state of AEGIS-256 is given by $S_i = S_{i,0} || S_{i,1} || S_{i,2} || S_{i,3} || S_{i,4} || S_{i,5}$, where each $S_{i,j}$ is a word. At each round i , a 16-byte data block m_i is used to update the state. The new state S_{i+1} is computed as follows:

$$\begin{aligned}
 S_{i+1,0} &= AES(S_{i,5}, S_{i,0} \oplus m_i), \\
 S_{i+1,1} &= AES(S_{i,0}, S_{i,1}), \\
 S_{i+1,2} &= AES(S_{i,1}, S_{i,2}), \\
 S_{i+1,3} &= AES(S_{i,2}, S_{i,3}), \\
 S_{i+1,4} &= AES(S_{i,3}, S_{i,4}), \\
 S_{i+1,5} &= AES(S_{i,4}, S_{i,5}).
 \end{aligned}$$

Like AEGIS-128, AEGIS-256 ciphertext and tag generation is also done in 4 stages: (1) The Initialization (2) Processing the Authenticated Data (3) The Encryption and (4) The Finalization.

After initialization and processing of the authenticated data, in the encryption stage, at each round, a 16-byte plaintext block P is used to update the state. Also P is encrypted to C where $C = P \oplus S_{i,1} \oplus S_{i,4} \oplus S_{i,5} \oplus (S_{i,2} \& S_{i,3})$.

2.5 Description of AEGIS-128L

At the beginning of the i -th round, the (8-word) state of AEGIS-128L is given by $S_i = S_{i,0} || S_{i,1} || S_{i,2} || S_{i,3} || S_{i,4} || S_{i,5} || S_{i,6} || S_{i,7}$, where each $S_{i,j}$ is a word. At each round i , two 16-byte data block m_a and m_b are used to update the state. The new state S_{i+1} is computed as follows:

$$\begin{aligned} S_{i+1,0} &= AES(S_{i,7}, S_{i,0} \oplus m_a) \\ S_{i+1,1} &= AES(S_{i,0}, S_{i,1}), \\ S_{i+1,2} &= AES(S_{i,1}, S_{i,2}), \\ S_{i+1,3} &= AES(S_{i,2}, S_{i,3}), \\ S_{i+1,4} &= AES(S_{i,3}, S_{i,4} \oplus m_b), \\ S_{i+1,5} &= AES(S_{i,4}, S_{i,5}), \\ S_{i+1,6} &= AES(S_{i,5}, S_{i,6}), \\ S_{i+1,7} &= AES(S_{i,6}, S_{i,7}). \end{aligned}$$

AEGIS-128L ciphertext and tag generation are done in 4 stages: (1) The Initialization (2) Processing the Authenticated Data (3) The Encryption and (4) The Finalization.

After initialization and processing of the authenticated data, in the encryption stage, at each round, two 16-byte plaintext block P and P' are used to update the state. Also P and P' are encrypted to C and C' respectively as $C = P \oplus z_{2i}$, $C' = P' \oplus z_{2i+1}$, where $z_{2i} = S_{i,1} \oplus S_{i,6} \oplus (S_{i,2} \& S_{i,3})$, $z_{2i+1} = S_{i,2} \oplus S_{i,5} \oplus (S_{i,6} \& S_{i,7})$ are two 16-byte block keystream.

3 Attack Description

The current paper assumes the following attack model:

The attacker can run the cipher with the same secret key, public parameters and plaintext several times. The attacker is able to inject single bit faults. A single bit fault flips the value of the corresponding bit. The attacker has control on the fault timing i.e., the attacker is able to induce single bit fault at any chosen cycle of operation of the cipher. The attacker has control on the fault location i.e., the attacker is able to induce single bit fault at any chosen location. The plaintext and the corresponding normal/faulty ciphertext is available to the attacker.

3.1 Attack on Tiaoxin

Let us consider three consecutive ciphertext generation rounds $i, i + 1$ and $i + 2$. At round i , the plaintext $M_i = M_i^0 || M_i^1$, composed of two words M_i^0 and M_i^1 , is encrypted to the ciphertext $C_i = C_i^0 || C_i^1$ composed of two words C_i^0 and C_i^1 as:

$$\begin{aligned} & \text{update}(T_3, T_4, T_6, M_i^0, M_i^1, M_i^0 \oplus M_i^1) \\ C_i^0 &= T_3[0] \oplus T_3[2] \oplus T_4[1] \oplus (T_6[3] \& T_4[3]), \\ C_i^1 &= T_6[0] \oplus T_4[2] \oplus T_3[1] \oplus (T_6[5] \& T_3[2]). \end{aligned}$$

We first consider faults at round i . To be precise we inject faults at round i , just after the call to the state update function.

Let us now consider a single bit fault at the r -th bit of the j -th byte of the block $T_6[5]$ i.e., at the r -th bit of the byte $T_6[5][j]$, $0 \leq r \leq 7, 0 \leq j \leq 15$. Due to the fault, the faulty value of $T_6[5][j]$ becomes $T_6[5][j] \oplus f$, where the r -th bit of f is ‘1’, remaining bits being ‘0’s.

Now the fault free ciphertext is given by

$$C_i^1 = T_6[0] \oplus T_4[2] \oplus T_3[1] \oplus (T_6[5] \& T_3[2]),$$

whereas its faulty value becomes

$$C_{f_{\text{faulty},i}}^1 = T_6[0] \oplus T_4[2] \oplus T_3[1] \oplus ((T_6[5] \oplus F) \& T_3[2]),$$

where F is a word with its j -th byte as f , remaining 15 bytes being all 0’s. This shows that $C_i^1[j] \oplus C_{f_{\text{faulty},i}}^1[j] = f \& T_3[2][j]$. Since $C_i^1[j]$ and $C_{f_{\text{faulty},i}}^1[j]$ are both available to the attacker and r -th bit of f is known to being ‘1’, one can recover the r -th bit of the byte $T_3[2][j]$ directly and uniquely.

This shows that, by injecting single bit faults (at each re-keyed run) to the r -th bit of the j -th byte of the block $T_6[5]$ at round i , one can deterministically obtain the r -th bit of the j -th byte of the block $T_3[2]$ for any $0 \leq j \leq 15$ and $0 \leq r \leq 7$. Thus with 128 faults to $T_6[5]$, it is possible to recover the entire $T_3[2]$ block. Hence we arrive at the following proposition:

Proposition 1. *Given any ciphertext generation round i , by injecting 128 faults to the block $T_6[5]$ one can always recover the block $T_3[2]$.*

Key recovery procedure: We now present the key recovery procedure based on Proposition 1. For that we consider faults at rounds $i, i + 1$ and $i + 2$.

To avoid ambiguity, we use the superscript i , to denote the state values at round i . For example, with this new notation, T_s^i represents the state T_s at round i . At round i , by injecting faults to the block $T_6^i[5]$, just after the state update call, one recovers the block $T_3^i[2]$. At round $i + 1$, the state T_3 is transformed to

$$T_3^{i+1} = (AES(T_3^i[2], T_3^i[0]) \oplus M_0^i, AES(T_3^i[0], Z_0), T_3^i[1]).$$

Clearly by injecting faults to the blocks $T_6^{i+1}[5]$ and $T_6^{i+2}[5]$ respectively at rounds $i + 1$ and $i + 2$, just after the state update call, one can recover the

block $T_3^i[1]$ and $AES(T_3^i[0], Z_0)$. Since $AES(T_3^i[0], Z_0)$ and Z_0 are both known, $T_3^i[0]$ can now be recovered. Thus by injecting 3×128 faults at three consecutive ciphertext generation rounds $i, i + 1$ and $i + 2$ one can recover the entire T_3^i . One should note that,

$$T_3^i = (AES(T_3^{i-1}[2], T_3^{i-1}[0]) \oplus M_0^{i-1}, AES(T_3^{i-1}[0], Z_0), T_3^{i-1}[1]).$$

Thus

$$\begin{aligned} T_3^i[0] &= AES(T_3^{i-1}[2], T_3^{i-1}[0]) \oplus M_0^{i-1}, \\ T_3^i[1] &= AES(T_3^{i-1}[0], Z_0), \\ T_3^i[2] &= T_3^{i-1}[1]. \end{aligned}$$

Clearly from T_3^i we can recover T_3^{i-1} i.e., T_3 state update is invertible. Now during the initialization phase, the state T_3 was initialized by (key, key, IV) . Thus for Tiaoxin, the secret key can be recovered with 384 single bit faults. The attack strategy for Tiaoxin is illustrated in Fig. 3.

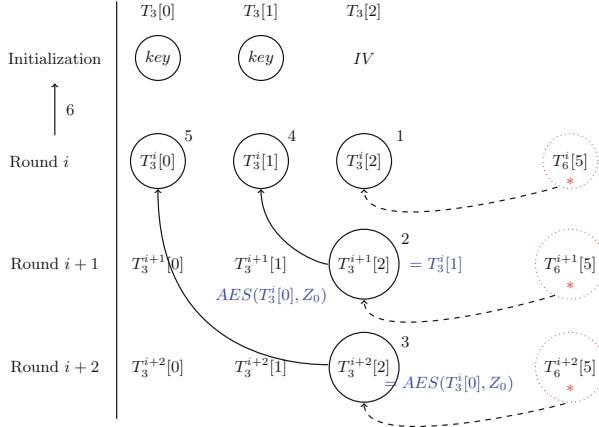


Fig. 3. Attack strategy on Tiaoxin: Here 1,2,... stand for the 1st, 2nd, ... steps of the attack procedure, "*" denotes the fault injection, the dotted arrow denotes the consequence of Proposition 1, the arrow from a state T_i to a state T_j denotes that the state T_j can be recovered from the state T_i .

3.2 Attack on AEGIS-128

Let us consider two consecutive ciphertext generation rounds i and $i + 1$. Under our attack model both the 16-byte block keystreams z_i and z_{i+1} will be available to the attacker. The state of the cipher at these rounds are given by

$$\begin{aligned} S_i &= S_{i,0} || S_{i,1} || S_{i,2} || S_{i,3} || S_{i,4}, \\ S_{i+1} &= S_{i+1,0} || S_{i+1,1} || S_{i+1,2} || S_{i+1,3} || S_{i+1,4} \end{aligned}$$

and the corresponding 16-byte keystreams are given by

$$z_i = S_{i,1} \oplus S_{i,4} \oplus (S_{i,2} \& S_{i,3}),$$

$$z_{i+1} = S_{i+1,1} \oplus S_{i+1,4} \oplus (S_{i+1,2} \& S_{i+1,3}).$$

As in the case of Tiaoxin, with 128 faults to $S_{i,2}$, it is possible to recover the entire $S_{i,3}$ block. Similarly by injecting 128 faults to $S_{i,3}$, it is possible to recover the entire $S_{i,2}$ block. Thus we arrive at the following proposition:

Proposition 2. *Given any ciphertext generation round i , by injecting 128 single bit faults to $S_{i,3}$ (or $S_{i,2}$) one can always recover the block $S_{i,2}$ (or $S_{i,3}$).*

State Recovery Procedure: We now present the state recovery procedure based on Proposition 2. For that we consider faults at rounds i and $i + 1$.

By injecting 3×128 single bit faults to $S_{i,3}, S_{i,2}$ and $S_{i+1,3}$ one respectively recovers the blocks $S_{i,2}, S_{i,3}$ and $S_{i+1,2}$. Now $S_{i+1,2} = \tau(S_{i,1}) \oplus S_{i,2}$. Since $S_{i+1,2}$ and $S_{i,2}$ are both known, $S_{i,1}$ can be recovered. Thus from $z_i = S_{i,1} \oplus S_{i,4} \oplus (S_{i,2} \& S_{i,3})$ one can recover $S_{i,4}$. At this moment $S_{i,1}, S_{i,2}, S_{i,3}$ and $S_{i,4}$ are known. Thus one can easily obtain $S_{i+1,3} = \tau(S_{i,2}) \oplus S_{i,3}$ and $S_{i+1,4} = \tau(S_{i,3}) \oplus S_{i,4}$. Now consider $z_{i+1} = S_{i+1,1} \oplus S_{i+1,4} \oplus (S_{i+1,2} \& S_{i+1,3})$ which gives $S_{i+1,1}$ as $S_{i+1,2}$ is also known. Finally $S_{i+1,1} = \tau(S_{i,0}) \oplus S_{i,1}$ gives $S_{i,0}$. Thus with 3×128 faults, we have the state $S_i = S_{i,0} || S_{i,1} || S_{i,2} || S_{i,3} || S_{i,4}$ at the i -th round. The attack strategy on AEGIS-128 is illustrated in Fig. 4.

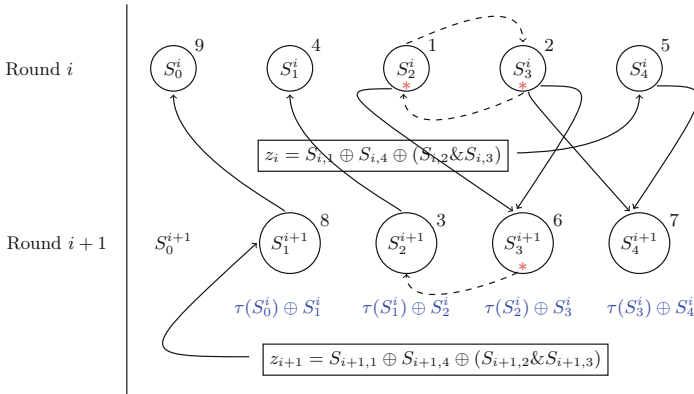


Fig. 4. Attack strategy for AEGIS-128: The notations are similar to that of Fig. 3.

3.3 Attack on AEGIS-256

In this case, we consider three consecutive ciphertext generation rounds $i, i + 1$ and $i + 2$. Under our attack model the 16-byte block keystreams z_i, z_{i+1} and

z_{i+2} are available to the attacker. The state of the cipher at these rounds are given by

$$\begin{aligned} S_i &= S_{i,0}||S_{i,1}||S_{i,2}||S_{i,3}||S_{i,4}||S_{i,5}, \\ S_{i+1} &= S_{i+1,0}||S_{i+1,1}||S_{i+1,2}||S_{i+1,3}||S_{i+1,4}||S_{i+1,5} \\ S_{i+2} &= S_{i+2,0}||S_{i+2,1}||S_{i+2,2}||S_{i+2,3}||S_{i+2,4}||S_{i+2,5} \end{aligned}$$

and the corresponding 16-byte keystreams are given by

$$\begin{aligned} z_i &= S_{i,1} \oplus S_{i,4} \oplus S_{i,5} \oplus (S_{i,2} \& S_{i,3}), \\ z_{i+1} &= S_{i+1,1} \oplus S_{i+1,4} \oplus S_{i+1,5} \oplus (S_{i+1,2} \& S_{i+1,3}), \\ z_{i+2} &= S_{i+2,1} \oplus S_{i+2,4} \oplus S_{i+2,5} \oplus (S_{i+2,2} \& S_{i+2,3}). \end{aligned}$$

As in AEGIS-128, with 128 faults to $S_{i,2}$, it is possible to recover the entire $S_{i,3}$ block. Similarly by injecting 128 faults to $S_{i,3}$, it is possible to recover the entire $S_{i,2}$ block. Thus we arrive at the following proposition:

Proposition 3. *Given any ciphertext generation round i , by injecting 128 faults to $S_{i,3}$ (or $S_{i,2}$) one can always recover the block $S_{i,2}$ (or $S_{i,3}$).*

We now present the state recovery procedure based on Proposition 3. For that we consider faults at rounds i , $i+1$ and $i+2$.

By Proposition 3, one obtains $S_{i,2}$ and $S_{i,3}$. $S_{i+1,3}$ is obtained from the relation $S_{i+1,3} = \tau(S_{i,2}) \oplus S_{i,3}$. At round $i+1$, $S_{i+1,2}$ can be recovered by injecting 128 faults to $S_{i+1,3}$. Now $S_{i+1,2} = \tau(S_{i,1}) \oplus S_{i,2}$. Since $S_{i+1,2}$ and $S_{i,2}$ are both known, $S_{i,1}$ can be recovered. At round $i+2$, one follows the same procedure to recover $S_{i+2,2}$, $S_{i+2,3}$, $S_{i+1,1}$ and $S_{i,0}$. At this moment four blocks $S_{i,0}$, $S_{i,1}$, $S_{i,2}$ and $S_{i,3}$ of i -th round are known. By $z_i = S_{i,1} \oplus S_{i,4} \oplus S_{i,5} \oplus (S_{i,2} \& S_{i,3})$ one knows the value of $S_{i,4} \oplus S_{i,5}$. Now consider

$$\begin{aligned} z_{i+1} &= S_{i+1,1} \oplus S_{i+1,4} \oplus S_{i+1,5} \oplus (S_{i+1,2} \& S_{i+1,3}) \\ &= S_{i+1,1} \oplus \tau(S_{i,3}) \oplus S_{i,4} \oplus \tau(S_{i,4}) \oplus S_{i,5} \oplus (S_{i+1,2} \& S_{i+1,3}). \end{aligned}$$

This gives $S_{i,4}$ as the rest are known. Finally $S_{i,4} \oplus S_{i,5}$ gives $S_{i,5}$. Thus with 4×128 faults, we have the state $S_i = S_{i,0}||S_{i,1}||S_{i,2}||S_{i,3}||S_{i,4}||S_{i,5}$ at the i -th round.

3.4 Attack on AEGIS-128L

We consider two consecutive ciphertext generation rounds i and $i+1$. The state of cipher at these rounds are given by

$$\begin{aligned} S_i &= S_{i,0}||S_{i,1}||S_{i,2}||S_{i,3}||S_{i,4}||S_{i,5}||S_{i,6}||S_{i,7}, \\ S_{i+1} &= S_{i+1,0}||S_{i+1,1}||S_{i+1,2}||S_{i+1,3}||S_{i+1,4}||S_{i+1,5}||S_{i+1,6}||S_{i+1,7}. \end{aligned}$$

The corresponding known 16-byte keystreams are given by

$$\begin{aligned} z_{2i} &= S_{i,1} \oplus S_{i,6} \oplus (S_{i,2} \& S_{i,3}), \\ z_{2i+1} &= S_{i,2} \oplus S_{i,5} \oplus (S_{i,6} \& S_{i,7}), \\ z_{2i+2} &= S_{i+1,1} \oplus S_{i+1,6} \oplus (S_{i+1,2} \& S_{i+1,3}), \\ z_{2i+3} &= S_{i+1,2} \oplus S_{i+1,5} \oplus (S_{i+1,6} \& S_{i+1,7}). \end{aligned}$$

For AEGIS-128L we have the next proposition similar to Proposition 3.

Proposition 4. *Given any ciphertext generation round i , by injecting 128 faults to each of $S_{i,3}, S_{i,2}, S_{i,7}$ and $S_{i,6}$ one can always recover the blocks $S_{i,2}, S_{i,3}, S_{i,6}$ and $S_{i,7}$ respectively.*

We now present the state recovery procedure based on Proposition 4.

By injecting 4×128 single bit faults, one obtains $S_{i,2}, S_{i,3}, S_{i,6}$ and $S_{i,7}$. z_{2i} and z_{2i+1} respectively give $S_{i,1}$ and $S_{i,5}$. Now $S_{i+1,1}$ and $S_{i+1,5}$ are recovered by considering the following relations

$$\begin{aligned} S_{i+1,2} &= \tau(S_{i,1}) \oplus S_{i,2}, \\ S_{i+1,3} &= \tau(S_{i,2}) \oplus S_{i,3}, \\ S_{i+1,6} &= \tau(S_{i,5}) \oplus S_{i,6}, \\ S_{i+1,7} &= \tau(S_{i,6}) \oplus S_{i,7}, \\ z_{2i+2} &= S_{i+1,1} \oplus S_{i+1,6} \oplus (S_{i+1,2} \& S_{i+1,3}), \\ z_{2i+3} &= S_{i+1,2} \oplus S_{i+1,5} \oplus (S_{i+1,6} \& S_{i+1,7}). \end{aligned}$$

Thus, $S_{i+1,1} = \tau(S_{i,0}) \oplus S_{i,1}$ and $S_{i+1,5} = \tau(S_{i,4}) \oplus S_{i,5}$ respectively give $S_{i,0}$ and $S_{i,4}$. This shows that with 4×128 faults, we have the state $S_i = S_{i,0} || S_{i,1} || S_{i,2} || S_{i,3} || S_{i,4} || S_{i,5} || S_{i,6} || S_{i,7}$ at the i -th round.

3.5 Reducing the Number of Re-Keying

Due to the nature of the ciphers, the attacker can reduce the number of re-keying of the ciphers by injecting faults parallelly. For this we consider another

Table 1. Summary of attacks

Cipher	Encryption round	Number of times Key-IV is repeated (Single bit fault model)	Number of times Key-IV is repeated (Multi byte fault model)
Tiaoxin	$i, i + 1, i + 2$	384	3
AEGIS-128	$i, i + 1$	384	3
AEGIS-256	$i, i + 1, i + 2$	512	4
AEGIS-128L	$i, i + 1$	512	4

fault model where the attacker can inject single bit faults to all the 128 bits of a 16-byte block at a time. In this case the number of re-keying will reduce by 128 times. With this fault model the attacker will now respectively require only 3, 3, 4 and 4 re-keying for Tiaoxin, AEGIS-128, AEGIS-256 and AEGIS-128L. The injected fault being visualized as a multi byte fault. We summarize the attacks in Table 1.

4 Conclusion

In this paper we presented a differential fault analysis on Tiaoxin and the AEGIS family of ciphers. We show one can find the key of Tiaoxin by injecting 384 single bit faults. Also we prove one needs 384 (respectively 512 and 512) single bit faults for AEGIS 128 (respectively AEGIS 256 and AEGIS-128L) to find the state. Reducing the number of single bit faults in an attack model where the adversary does not have the control over the fault injection timing as well as the fault injection location, could be a challenging future work.

References

1. CAESAR (Competition for Authenticated Encryption: Security, Applicability, and Robustness) (2013). <http://competitions.cr.yt.to/caesar.html>
2. Ali, S.S., Mukhopadhyay, D.: A differential fault analysis on AES key schedule using single fault. In: 2011 Workshop on Fault Diagnosis and Tolerance in Cryptography (FDTC), pp. 35–42. IEEE (2011)
3. Banik, S., Maitra, S., Sarkar, S.: A differential fault attack on the grain family of stream ciphers. In: Prouff, E., Schaumont, P. (eds.) CHES 2012. LNCS, vol. 7428, pp. 122–139. Springer, Heidelberg (2012)
4. Banik, S., Maitra, S., Sarkar, S.: A differential fault attack on the grain family under reasonable assumptions. In: Galbraith, S., Nandi, M. (eds.) INDOCRYPT 2012. LNCS, vol. 7668, pp. 191–208. Springer, Heidelberg (2012)
5. Berzati, A., Canovas, C., Castagnos, G., Debraize, B., Goubin, L., Gouget, A., Pailier, P., Salgado, S.: Fault analysis of GRAIN-128. In: IEEE International Workshop on Hardware-Oriented Security and Trust, HOST 2009, pp. 7–14. IEEE (2009)
6. Biham, E., Shamir, A.: Differential fault analysis of secret key cryptosystems. In: Kaliski Jr., B.S. (ed.) CRYPTO 1997. LNCS, vol. 1294, pp. 513–525. Springer, Heidelberg (1997)
7. Dey, P., Adhikari, A.: Improved multi-bit differential fault analysis of Trivium. In: Meier, W., Mukhopadhyay, D. (eds.) INDOCRYPT 2014. LNCS, vol. 8885, pp. 37–52. Springer International Publishing, Heidelberg (2014)
8. Dey, P., Chakraborty, A., Adhikari, A., Mukhopadhyay, D.: Improved practical differential fault analysis of grain-128. In: Proceedings of the 2015 Design, Automation & Test in Europe Conference & Exhibition, DATE 2015, pp. 459–464. EDA Consortium, San Jose (2015)
9. Dey, P., Rohit, R.S., Adhikari, A.: Full key recovery of acorn with a single fault. *J. Inf. Secur. Appl.* **29**, 57–64 (2016)
10. Hojsík, M., Rudolf, B.: Differential fault analysis of Trivium. In: Nyberg, K. (ed.) FSE 2008. LNCS, vol. 5086, pp. 158–172. Springer, Heidelberg (2008)

11. Hojsik, M., Rudolf, B.: Floating fault analysis of Trivium. In: Chowdhury, D.R., Rijmen, V., Das, A. (eds.) *INDOCRYPT 2008*. LNCS, vol. 5365, pp. 239–250. Springer, Heidelberg (2008)
12. Yupu, H., Gao, J., Liu, Q., Zhang, Y.: Fault analysis of Trivium. *Des. Codes Crypt.* **62**(3), 289–311 (2012)
13. Karmakar, S., Roy Chowdhury, D.: Fault analysis of grain-128 by targeting NFSR. In: Nitaj, A., Pointcheval, D. (eds.) *AFRICACRYPT 2011*. LNCS, vol. 6737, pp. 298–315. Springer, Heidelberg (2011)
14. Kircanski, A., Youssef, A.M.: Differential fault analysis of rabbit. In: Jacobson Jr., M.J., Rijmen, V., Safavi-Naini, R. (eds.) *SAC 2009*. LNCS, vol. 5867, pp. 197–214. Springer, Heidelberg (2009)
15. Minaud, B.: Linear biases in AEGIS keystream. In: Joux, A., Youssef, A. (eds.) *SAC 2014*. LNCS, vol. 8781, pp. 290–305. Springer, Heidelberg (2014)
16. Mohamed, M.S.E., Bulygin, S., Buchmann, J.: Using SAT solving to improve differential fault analysis of Trivium. In: Kim, T., Adeli, H., Robles, R.J., Balitanas, M. (eds.) *ISA 2011*. CCIS, vol. 200, pp. 62–71. Springer, Heidelberg (2011)
17. Mukhopadhyay, D.: An improved fault based attack of the advanced encryption standard. In: Preneel, B. (ed.) *AFRICACRYPT 2009*. LNCS, vol. 5580, pp. 421–434. Springer, Heidelberg (2009)
18. Nikolić, I.: Tiaoxin - 346 (2014). <http://competitions.cr.yip.to/round1/tiaoxinv1.pdf>
19. Saha, D., Mukhopadhyay, D., Chowdhury, D.R.: A diagonal fault attack on the advanced encryption standard. *IACR Cryptol. ePrint Arch.* **2009**, 581 (2009)
20. Esmaeili Salehani, Y., Kircanski, A., Youssef, A.: Differential fault analysis of SOSEMANUK. In: Nitaj, A., Pointcheval, D. (eds.) *AFRICACRYPT 2011*. LNCS, vol. 6737, pp. 316–331. Springer, Heidelberg (2011)
21. Sarkar, S., Banik, S., Maitra, S.: Differential fault attack against grain family with very few faults and minimal assumptions. *IACR Cryptol. ePrint Arch.* **2013**, 494 (2013)
22. Sarkar, S., Dey, P., Adhikari, A., Maitra, S.: Probabilistic signature based generalized framework for differential fault analysis of stream ciphers. *Cryptogr. Commun.* 1–21 (2016)
23. Tunstall, M., Mukhopadhyay, D., Ali, S.: Differential fault analysis of the advanced encryption standard using a single fault. In: Ardagna, C.A., Zhou, J. (eds.) *WISTP 2011*. LNCS, vol. 6633, pp. 224–233. Springer, Heidelberg (2011)
24. Hongjun, W., Bart Preneel, A.: A fast authenticated encryption algorithm. In: 20th International Conference on Selected Areas in Cryptography - SAC 2013, Burnaby, BC, Canada, 14–16 August 2013, Revised Selected Papers, pp. 185–201 (2013)
25. Hongjun, W., Bart Preneel, A.: A fast authenticated encryption algorithm (v1). CAESAR Submission, updated from Cryptology ePrint Archive Report 2013/695, updated from SAC 2013 version (2014). <http://competitions.cr.yip.to/round1/aegisv1.pdf>

A Comparison of Diffusion Properties of Salsa, ChaCha, and MCC Core

Rajeev Sobti¹(✉) and G. Geetha²

¹ School of Computer Science and Engineering,
Lovely Professional University, Phagwara, Punjab, India
rajeev.sobti@lpu.co.in

² Division of Research and Development, Lovely Professional University,
Phagwara, Punjab, India
gitaskumar@yahoo.com

Abstract. Salsa Core, ChaCha Core, and MCC (Modified ChaCha) Core are cryptographic primitives that take 64-byte input and mix this input to generate 64-byte output. Both Salsa and ChaCha cores, have been used to generate stream ciphers. Salsa is also listed as one of the eSTREAM profile cipher. This paper compares the diffusion properties of all these three cryptographic primitives and share the results. Comparison of Quarter rounds of all these competing cores had been done already but the Column and Row/Diagonal rounds of these cores differ considerably and comparison of full Double rounds of these cores is essential to determine their relative performance. Based on the diffusion characteristics and behavior of these cores, this study proposes alternative rotation distances for better diffusion. Comparative analysis reflects that MCC core performs better than both Salsa and ChaCha core.

Keywords: Salsa · ChaCha · MCC · Modified ChaCha · Diffusion · Diffusion matrix

1 Introduction

Salsa core [1], designed by D.J. Bernstein, has multiple variants depending upon the number of rounds. Salsa20, the primary proposal, consists of 20 rounds (10 Double rounds). Salsa20/12 and Salsa 20/8 are reduced round variants consisting of 12 and 8 rounds respectively. Stream cipher constructed using this core is also listed as one of eSTREAM [2] profile algorithm. ChaCha core [3], an improvement over Salsa by the same author, also has multiple variants like ChaCha8, ChaCha12, and ChaCha20 corresponding to Salsa20/8, Salsa20/12, and Salsa20. Modified ChaCha (MCC) Core is an improvisation of ChaCha by Sobti and Ganesan and is detailed in [4]. Like Salsa and ChaCha, different variants of MCC can also be defined based on number of rounds.

All these cryptographic primitives arrange input data in a matrix of 4×4 (where each element is a 32-bit word) and use ARX (Addition, Rotation and XoR) operations to mix the input data so that change in any one bit of input diffuses efficiently to maximum output bits. One full round (named as Double round) of these competing primitives consist of one set of Column and one set of Row (or Diagonal in case of

ChaCha) rounds. Each set of Column/Row/Diagonal round uses four Quarter rounds where each Quarter round operates on one Column or Row or Diagonal of the 4×4 matrix.

Section 2 briefly explains the Quarter round of these three competing primitives (Salsa, ChaCha and MCC). The comparative analysis of diffusion property of Quarter round of these candidate cores had been done already and presented in [4] by Sobti and Ganesan. However, not only the Quarter round, but the Double round of these three competing primitives also differ considerably. So with a motivation to have better and comprehensive comparison of these three primitives, this paper compares the diffusion property of full round (Double round) of each primitive instead of Quarter rounds as done in [4]. The authors of these three cores have prescribed specific rotation distances to be used in their respective Quarter rounds. The experiment presented in this paper analyze diffusion property of Double rounds of Salsa, ChaCha, and MCC cores, not only on prescribed rotation distances (constants) but on all possible rotation distances, to see how the diffusion changes with change in rotation constants and in turn proposes better and efficient rotation constants for more diffusion.

Organization of the Paper: Sect. 2 introduces Double round of Salsa, ChaCha, and MCC core. Section 3 details the experiment conducted to compare these primitives. Section 4 presents the results followed by conclusion and future work in Sect. 5.

2 Double Rounds of Salsa, ChaCha, and MCC Core

All these three primitives arrange data in a matrix of 4×4 , where each element is a 32-bit word. The input matrix ‘ x ’ may be represented as:

$$\begin{bmatrix} x_0 & x_1 & x_2 & x_3 \\ x_4 & x_5 & x_6 & x_7 \\ x_8 & x_9 & x_{10} & x_{11} \\ x_{12} & x_{13} & x_{14} & x_{15} \end{bmatrix}$$

The Double round of Salsa [1], ChaCha [3] and MCC [4] are briefly introduced below:

2.1 Salsa

Double round of Salsa takes matrix ‘ x ’ as input and returns matrix ‘ z ’ as output after mixing all words. Double Round of Salsa consists of four Column Quarter rounds followed by four Row Quarter rounds. The same may be represented as:

$$z = DoubleRD_{Salsa}(x) = RowRD_{Salsa}(ColumnRD_{Salsa}(x))$$

Column Quarter round of Salsa calls following four Quarter rounds, one for each column of matrix ‘ x ’.

$$\begin{aligned}
(y_0, y_4, y_8, y_{12}) &= \text{QuarterRD}_{Salsa}(x_0, x_4, x_8, x_{12}) \\
(y_5, y_9, y_{13}, y_1) &= \text{QuarterRD}_{Salsa}(x_5, x_9, x_{13}, x_1) \\
(y_{10}, y_{14}, y_2, y_6) &= \text{QuarterRD}_{Salsa}(x_{10}, x_{14}, x_2, x_6) \\
(y_{15}, y_3, y_7, y_{11}) &= \text{QuarterRD}_{Salsa}(x_{15}, x_3, x_7, x_{11})
\end{aligned}$$

Row Quarter round of Salsa calls following four Quarter rounds, one for each row of matrix ‘y’.

$$\begin{aligned}
(z_0, z_1, z_2, z_3) &= \text{QuarterRD}_{Salsa}(y_0, y_1, y_2, y_3) \\
(z_5, z_6, z_7, z_4) &= \text{QuarterRD}_{Salsa}(y_5, y_6, y_7, y_4) \\
(z_{10}, z_{11}, z_8, z_9) &= \text{QuarterRD}_{Salsa}(y_{10}, y_{11}, y_8, y_9) \\
(z_{15}, z_{12}, z_{13}, z_{14}) &= \text{QuarterRD}_{Salsa}(y_{15}, y_{12}, y_{13}, y_{14})
\end{aligned}$$

Quarter round of Salsa takes four words (a, b, c, d) as input and gives four words as output after performing following 32-bit ARX operations:

$$\begin{aligned}
b &= b \oplus ((a + d) \lll 7) \\
c &= c \oplus ((b + a) \lll 9) \\
d &= d \oplus ((c + b) \lll 13) \\
a &= a \oplus ((d + c) \lll 18)
\end{aligned}$$

Close look at Salsa’s Quarter round reflects that each input word is updated once with 32-bit addition, 32-bit rotation, and 32-bit XOR operation.

2.2 ChaCha

Double round of ChaCha also takes matrix ‘x’ as input and returns matrix ‘z’ as output. It consists of four Column Quarter rounds followed by four Diagonal Quarter rounds. The same may be represented as:

$$z = \text{DoubleRD}_{ChaCha}(x) = \text{DiagonalRD}_{ChaCha}(\text{ColumnRD}_{ChaCha}(x))$$

Column Quarter round of ChaCha calls following four Quarter rounds, one for each column of matrix ‘x’.

$$\begin{aligned}
(y_0, y_4, y_8, y_{12}) &= \text{QuarterRD}_{ChaCha}(x_0, x_4, x_8, x_{12}) \\
(y_1, y_5, y_9, y_{13}) &= \text{QuarterRD}_{ChaCha}(x_1, x_5, x_9, x_{13}) \\
(y_2, y_6, y_{10}, y_{14}) &= \text{QuarterRD}_{ChaCha}(x_2, x_6, x_{10}, x_{14}) \\
(y_3, y_7, y_{11}, y_{15}) &= \text{QuarterRD}_{ChaCha}(x_3, x_7, x_{11}, x_{15})
\end{aligned}$$

Diagonal Quarter round of ChaCha calls following four Quarter rounds, one for each diagonal of matrix ‘y’.

$$\begin{aligned}
(z_0, z_5, z_{10}, z_{15}) &= \text{QuarterRD}_{ChaCha}(y_0, y_5, y_{10}, y_{15}) \\
(z_1, z_6, z_{11}, z_{12}) &= \text{QuarterRD}_{ChaCha}(y_1, y_6, y_{11}, y_{12}) \\
(z_2, z_7, z_8, z_{13}) &= \text{QuarterRD}_{ChaCha}(y_2, y_7, y_8, y_{13}) \\
(z_3, z_4, z_9, z_{14}) &= \text{QuarterRD}_{ChaCha}(y_3, y_4, y_9, y_{14})
\end{aligned}$$

Quarter round of ChaCha takes four words (a, b, c, d) as input and gives four words as output after performing following 32-bit ARX operations:

$$\begin{aligned} a + &= b; & d \oplus &= a; & d \lll &= 16; \\ c + &= d; & b \oplus &= c; & b \lll &= 12; \\ a + &= b; & d \oplus &= a; & d \lll &= 8; \\ c + &= d; & b \oplus &= c; & b \lll &= 7; \end{aligned}$$

Close look at ChaCha's Quarter round reflects that each input word is updated twice. However, first and third words (word 'a' and 'c') always get updated with 32-bit addition operation, while second and fourth words (word 'b' and 'd') always get updated with an XOR operation followed by Rotation with a constant.

2.3 MCC

Double round of MCC is designed on the same lines as that of Salsa and ChaCha. It takes matrix 'x' as input and calls four Column Quarter rounds followed by four Row Quarter rounds to generate output matrix 'z'. The same may be represented as:

$$z = DoubleRD_{MCC}(x) = DiagonalRD_{MCC}(ColumnRD_{MCC}(x))$$

Column Quarter round of MCC calls following four Quarter rounds, one for each column of matrix 'x'.

$$\begin{aligned} (y_0, y_4, y_8, y_{12}) &= QuarterRD_{MCC}(x_0, x_4, x_8, x_{12}) \\ (y_5, y_9, y_{13}, y_1) &= QuarterRD_{MCC}(x_5, x_9, x_{13}, x_1) \\ (y_{10}, y_{14}, y_2, y_6) &= QuarterRD_{MCC}(x_{10}, x_{14}, x_2, x_6) \\ (y_{15}, y_3, y_7, y_{11}) &= QuarterRD_{MCC}(x_{15}, x_3, x_7, x_{11}) \end{aligned}$$

Row Quarter round of MCC calls following four Quarter rounds, one for each row of matrix 'y'.

$$\begin{aligned} (z_1, z_2, z_3, z_0) &= QuarterRD_{MCC}(y_1, y_2, y_3, y_0) \\ (z_6, z_7, z_4, z_5) &= QuarterRD_{MCC}(y_6, y_7, y_4, y_5) \\ (z_{11}, z_8, z_9, z_{10}) &= QuarterRD_{MCC}(y_{11}, y_8, y_9, y_{10}) \\ (z_{12}, z_{13}, z_{14}, z_{15}) &= QuarterRD_{MCC}(y_{12}, y_{13}, y_{14}, y_{15}) \end{aligned}$$

Quarter round of MCC takes four words (a, b, c, d) as input and gives four words as output after performing following 32-bit ARX operations:

$$\begin{aligned} b + &= a; & c \oplus &= b; & c \lll &= 4; \\ d + &= c; & a \oplus &= d; & a \lll &= 17; \\ c + &= a; & b \oplus &= c; & b \lll &= 8; \\ a + &= b; & d \oplus &= a; & d \lll &= 0; \end{aligned}$$

MCC's Quarter round also update each input word twice. However, contrary to ChCha's Quarter round that update two words with addition operation and other two words with XOR and Rotation, MCC's Quarter round exposes all four input words to

Addition, XOR, and Rotation operations. For example, word ‘*b*’ initially gets updated with a 32-bit addition operation, then next with a 32-bit XOR, and at last by a 32-bit rotation operation. Similarly, all other words are updated with all three operations.

3 Experiment Conducted to Compare Double Rounds

Double round of Salsa, ChaCha, and MCC core are compared by analyzing their diffusion property. Diffusion is an important and desired characteristic of a cryptographic primitive that represents how a small change in input spreads to multiple output bits. With regard to Double Rounds, diffusion may be defined as the number of output bits that change with change of one random input bit.

The Quarter round of all these competing designs have four prescribed rotation constants. These rotation distances are [7, 9, 13, 18], [16, 12, 8, 7], and [4, 17, 8, 0] for Salsa, ChaCha, and MCC respectively. The experiment conducted in this study compares diffusion property of Double rounds of these designs on prescribed rotation distances as well as on all possible rotation distances. Each rotation distance can have 32 possible value from 0 to 31. So for four rotation distances, we have $32 \times 32 \times 32 \times 32 = 1,048,576$ (more than a million) possible sets of rotation distances. For each possible set of rotation distances, three diffusion matrices - one each for Salsa, ChaCha, and MCC - were generated. The general diffusion matrix may be represented as:

$$\left[D_{a,b} \right] \text{ where, } a, b \in \{x_0, x_1, \dots, x_{15}\}$$

Value $D_{a,b}$ in the diffusion matrix represents change in output word ‘*b*’ with one random bit change in input word ‘*a*’.

The algorithm used to generate all possible diffusion matrices for all three competing designs is given below.

Step 1: Run the following steps (Step 2 to 8) for all possible values of *i*, *j*, *k*, and *l*; each varying from 0 to 31. So Steps 2 to 8 will be executed 1,048,576 times.

Step 2: Generate sixteen random values $(x_0, x_1, \dots, x_{15})$ of 32 bits each.

Step 3: Run the Double round of Salsa, ChaCha, and MCC core one by one and generate $(z_0, z_1, \dots, z_{15}) = DoubleRD(x_0, x_1, \dots, x_{15})$ for each design.

Step 4: Flip one bit of word x_0 randomly and keep all other words $(x_1, x_2, \dots, x_{15})$ unchanged. Call the Double round of Salsa, ChaCha, and MCC again and generate $(z'_0, z'_1, \dots, z'_{15}) = DoubleRD(x_0^{flipped}, x_1, \dots, x_{15})$ for each design.

Step 5: Compare $(z'_0, z'_1, \dots, z'_{15})$ with $(z_0, z_1, \dots, z_{15})$ and find how many bits are different in each word and store it in one dimensional array D_{x_0} having sixteen elements $[D_{x_0,x_0}, D_{x_0,x_1}, D_{x_0,x_2}, \dots, D_{x_0,x_{15}}]$. D_{x_0,x_1} represent change in word x_1 because of one bit change in word x_0 i.e. difference in z'_1 and z_1 . Example: If z_1 is 101111111100 11010111101111110 and z'_1 is 11100110111001101110011011100111 then it means change (no. of bits modified) in word x_1 because of one random bit change in word x_0 is 10. This is stored in D_{x_0,x_1} .

Step 6: Repeat steps 4 and 5 by flipping one bit of other words i.e. Step 4 and 5 will be called 15 times by flipping one bit of x_1 , then of x_2 , and so on. For example, by flipping one bit of x_1 and keeping all other words same, we will generate $(z'_0, z'_1, \dots, z'_{15}) = DoubleRD(x_0, x_1^{flipped}, x_2, \dots, x_{15})$ and will calculate D_{x_1} having sixteen elements $[D_{x_1, x_0}, D_{x_1, x_1}, D_{x_1, x_2}, \dots, D_{x_1, x_{15}}]$. Similarly, steps 4 and 5 will be called by randomly flipping one bit of other words.

Step 7: Repeat Step 2 to Step 6, 1000 times and find average of each element to get diffusion matrix for one set of rotation constants i, j, k , and l .

Using the above experiment, 1,048,576 diffusion matrices (DM) were generated for each design. The rounded values of diffusion matrices (DM) generated using prescribed rotation distances - [7, 9, 13, 18] for Salsa, [16, 12, 8, 7] for ChaCha, and [4, 17, 8, 0] for MCC - are given below.

$DM_{Salsa} =$	DM	x_0	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}	x_{12}	x_{13}	x_{14}	x_{15}
	x_0	16	12	17	15	7	10	3	3	4	5	10	0	6	9	11	15
	x_1	14	5	5	8	15	15	8	13	2	5	10	2	0	2	2	4
	x_2	4	0	1	1	6	8	2	2	10	13	16	4	0	0	0	0
	x_3	9	2	2	4	1	4	0	1	3	6	10	3	7	12	13	16
	x_4	17	9	13	15	5	8	2	2	2	2	6	0	5	6	8	14
	x_5	13	7	7	10	16	16	12	14	3	8	10	3	0	4	6	10
	x_6	7	0	2	3	10	15	6	6	15	15	14	9	2	2	6	11
	x_7	0	0	0	0	2	3	0	1	2	4	7	1	5	11	14	16
	x_8	15	5	10	12	0	0	0	0	1	1	4	0	2	3	4	7
	x_9	12	4	4	6	15	16	7	14	2	6	8	1	0	1	1	4
	x_{10}	10	0	3	5	7	12	6	6	14	14	17	13	2	2	6	10
	x_{11}	10	2	2	7	3	7	0	2	7	11	15	6	10	15	16	17
	x_{12}	16	7	13	15	5	9	2	2	2	2	5	0	4	6	9	15
	x_{13}	8	2	2	7	14	16	4	10	0	0	0	0	0	1	1	3
	x_{14}	4	0	1	2	8	11	3	5	11	14	17	7	2	2	4	9
x_{15}	8	1	2	5	4	9	0	4	7	10	14	6	13	15	14	15	
$DM_{ChaCha} =$	DM	x_0	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}	x_{12}	x_{13}	x_{14}	x_{15}
	x_0	6	7	9	14	14	15	14	14	11	16	14	12	9	9	15	9
	x_1	15	5	7	9	15	15	16	13	12	11	15	16	9	9	9	15
	x_2	10	14	7	7	12	14	17	14	13	11	11	16	14	11	9	10
	x_3	8	10	14	5	15	15	14	14	15	14	14	11	10	16	9	10
	x_4	10	9	11	16	16	14	14	15	12	15	15	13	11	11	16	12
	x_5	16	10	10	13	16	16	13	14	13	13	16	15	12	11	13	15
	x_6	11	15	10	10	15	16	17	14	16	15	13	16	15	12	13	11
	x_7	9	12	15	10	15	12	15	16	16	17	13	13	12	16	11	10
	x_8	3	2	4	13	15	9	8	6	6	15	8	5	4	4	13	4
	x_9	15	3	2	4	6	17	12	9	7	5	15	10	5	4	4	14
	x_{10}	4	14	4	3	9	7	15	10	8	8	7	16	15	6	5	4
	x_{11}	3	6	15	5	14	12	8	17	15	12	10	8	6	16	8	6
	x_{12}	5	5	7	16	14	12	10	10	8	14	10	8	6	7	15	7
	x_{13}	14	3	6	7	11	15	12	10	9	8	15	10	6	6	7	14
	x_{14}	9	15	5	5	10	14	16	13	11	8	10	17	15	7	5	9
x_{15}	5	9	15	3	13	11	13	16	15	12	10	10	9	12	6	6	

$$DM_{MCC} = \begin{bmatrix} DM & x_0 & x_1 & x_2 & x_3 & x_4 & x_5 & x_6 & x_7 & x_8 & x_9 & x_{10} & x_{11} & x_{12} & x_{13} & x_{14} & x_{15} \\ x_0 & 16 & 17 & 13 & 13 & 11 & 15 & 14 & 11 & 12 & 9 & 15 & 14 & 16 & 16 & 17 & 16 \\ x_1 & 16 & 15 & 15 & 15 & 4 & 10 & 7 & 4 & 6 & 6 & 12 & 8 & 11 & 7 & 5 & 12 \\ x_2 & 13 & 15 & 9 & 6 & 16 & 14 & 15 & 15 & 11 & 11 & 15 & 13 & 11 & 5 & 5 & 12 \\ x_3 & 16 & 15 & 11 & 11 & 5 & 14 & 13 & 11 & 17 & 16 & 17 & 16 & 14 & 11 & 11 & 15 \\ x_4 & 14 & 13 & 11 & 11 & 12 & 14 & 14 & 12 & 11 & 6 & 15 & 15 & 16 & 14 & 16 & 16 \\ x_5 & 15 & 17 & 16 & 15 & 16 & 15 & 16 & 16 & 11 & 11 & 17 & 16 & 15 & 12 & 10 & 17 \\ x_6 & 12 & 9 & 6 & 5 & 14 & 15 & 14 & 15 & 5 & 5 & 13 & 10 & 8 & 5 & 5 & 11 \\ x_7 & 12 & 12 & 7 & 7 & 7 & 16 & 16 & 11 & 15 & 14 & 17 & 14 & 10 & 7 & 7 & 12 \\ x_8 & 16 & 14 & 8 & 8 & 6 & 12 & 12 & 6 & 8 & 5 & 14 & 14 & 15 & 15 & 14 & 14 \\ x_9 & 14 & 16 & 14 & 15 & 12 & 15 & 13 & 12 & 14 & 14 & 14 & 14 & 14 & 10 & 6 & 13 \\ x_{10} & 16 & 17 & 13 & 12 & 14 & 16 & 15 & 16 & 13 & 13 & 16 & 16 & 15 & 10 & 10 & 16 \\ x_{11} & 8 & 7 & 4 & 4 & 6 & 14 & 12 & 8 & 15 & 15 & 15 & 17 & 8 & 5 & 5 & 11 \\ x_{12} & 14 & 10 & 5 & 5 & 5 & 11 & 9 & 5 & 7 & 5 & 13 & 12 & 16 & 17 & 15 & 15 \\ x_{13} & 15 & 14 & 14 & 15 & 8 & 13 & 13 & 8 & 6 & 6 & 14 & 14 & 14 & 11 & 7 & 14 \\ x_{14} & 14 & 13 & 11 & 7 & 15 & 16 & 15 & 17 & 11 & 11 & 16 & 13 & 14 & 11 & 11 & 14 \\ x_{15} & 16 & 16 & 11 & 11 & 10 & 16 & 14 & 12 & 16 & 17 & 15 & 16 & 15 & 12 & 12 & 15 \end{bmatrix}$$

Each cell of DM represents average number of bits modified in the word represented by a column, by randomly changing one bit of the word represented by a row. For example, value 17 in last column (represented by word x_{15}) of first row (represented by word x_0) signifies that with one bit change in x_0 , on an average 17 bits of x_{15} gets changed.

4 Results and Discussion

The first part of this section, presents the comparison of diffusion matrices of Double round of Salsa, ChaCha, and MCC core on prescribed rotation constants. In the second part of this section, the diffusion property of these designs are compared and analyzed for all the possible values of rotation distances.

4.1 Comparison of Diffusion Matrices Generated Using Prescribed Rotation Distances

Few important statistics that can be drawn from the DMs generated in previous section are listed below:

- Salsa's diffusion matrix has around **12.5 % words with zero value** i.e. on an average 12.5 % words are not changed with one-bit change in input words. However, in case of ChaCha and MCC, all words are affected with one-bit change in any input words.
- Ideally, it is desired to have diffusion value around 16 in each cell i.e. 50 % of bits in a 32-bit word gets updated with one-bit change in input word. For Salsa, we have just 43 such words (out of 256 words) where diffusion is close to ideal

(i.e. 14 or more). However, for ChaCha we have 92 such words which get close to ideal diffusion and for MCC, this value goes up to 124 i.e. on an average, about **35 % better performance than ChaCha and 188 % better performance than Salsa.**

- (c) The average diffusion generated in a word by Double round of Salsa, ChaCha, and MCC using prescribed rotation constants is **6.9645**, **11.4297**, and **12.6027** respectively and standard deviation in the same order is 5.32, 3.93, and 3.63.

4.2 Comparison and Analysis of Diffusion Matrices for All Possible Rotation Distances

Each diffusion matrix (*DM*) consists of 256 values. To compare all the diffusion matrices (1,048,576 *DMs* for each design), mean and standard deviation was calculated for each *DM* and plotted on graph with mean on *X* axis and standard deviation on *Y* axis. For every single diffusion matrix, one point is plotted on the graph. The graph in Fig. 1 represents 1,048,576 points for each design (corresponding to all possible permutations of rotation constants *i, j, k, and l*).

The major observations that can be drawn from the generated (*DMs*) and the graph plotted in Fig. 1 are listed below

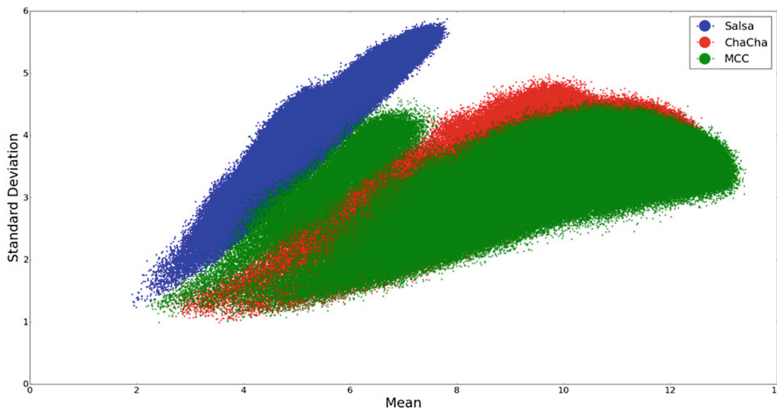


Fig. 1. Comparison of Double rounds of Salsa, ChaCha, and MCC core for all possible rotation constants

(a) The diffusion generated using rotation distances prescribed by the respective authors are not the best For Salsa, the *DM* corresponding to the prescribed rotation distances [7, 9, 13, 18], has mean of 6.9645 and standard deviation of 5.3213. However, experiment conducted in this study reflected that there are more than 290 thousand (approximately 28 %) sets of [*i, j, k, and l*] that generate *DMs* having mean more than the *DM* corresponding to the prescribed rotation distances of [7,19,13,18].

Similarly, for ChaCha there are more than 250 thousand (about 24 %) sets of rotation distances that performs better than the prescribed rotation distances of [16, 12, 8, 7]. However, for MCC the selection of rotation distances seems much better as there were only 1.7 % set of rotation distances that perform better than prescribed rotation distances of [4, 17, 8, 0].

(b) Proposal for Rotation Distances: The rotation distances that give highest diffusion for Salsa, ChaCha, and MCC core are [16, 7, 20, 26], [23, 14, 7, 30], and [5, 17, 8, 10] respectively. Table 1, 2, and 3 list 10 best set of rotation distances for these competing designs. Any one of these sets may be used looking at **DM's** mean value, standard deviation, and rotation efficiency. For example, for MCC the highest average diffusion of 13.4014 is obtained using [5, 7, 8, 10]. However, rotation distance of [16, 21,13, 20] is also a good choice as it gives average diffusion of 13.3838 (about 0.1 % less than best value of 13.4014) but improves standard deviation by more than 11 % (from 3.4565 to 3.0566). Similarly, byte aligned rotation distances may be preferred for better speed performance.

This study proposes to use rotation distances from Tables 1, 2, and 3 for better diffusion in place of rotation distances suggested by the authors of the respective design. With this change, diffusion can be improved considerably.

Table 1. Top ten rotation distances generating diffusion matrix (*DM*) with highest mean for Salsa

<i>i</i>	<i>j</i>	<i>k</i>	<i>l</i>	(<i>DM</i>)'s mean	(<i>DM</i>)'s SD	No. of Byte aligned rotations
16	7	20	26	7.8457	5.6541	1
16	7	20	22	7.832	5.6329	1
6	20	10	19	7.8115	5.5906	0
15	10	21	14	7.8096	5.8767	0
14	10	24	20	7.8086	5.6323	1
5	11	20	26	7.793	5.7193	0
17	8	26	3	7.7852	5.6607	1
22	19	15	4	7.7686	5.7292	0
6	21	13	10	7.7676	5.579	0
14	9	22	12	7.7617	5.7063	0

For Salsa core, the (*DM*) generated using the proposed rotation distances (Table 1) has mean varying from 7.8457 to 7.7617 against mean of 6.9645 with prescribed rotation distances i.e. improvement from **12.65 % to 11.45 %** depending on rotation distances opted from Table 1. **For ChaCha core** also, this change brings improvement up to **11.73 %**. Mean of (*DM*) corresponding to prescribed rotation distances is 11.4297 but with new proposed rotation distances it can increase up 12.7705. For MCC, this change brings increase up to **6.4 %** (i.e. from mean of 12.6027 using prescribed rotation distances to 13.4014 as listed in Table 3. Relatively less improvement in diffusion of MCC core (6.4 % compared to 12.65 % in Salsa and 11.73 % in ChaCha) with change in rotation distances reflects that the selection of

Table 2. Top ten rotation distances generating diffusion matrix (*DM*) with highest mean for ChaCha

<i>i</i>	<i>j</i>	<i>k</i>	<i>l</i>	(<i>DM</i>)'s mean	(<i>DM</i>)'s SD	No. of Byte aligned rotations
23	14	7	30	12.7705	3.7553	0
15	21	23	28	12.7656	3.9314	0
13	11	7	15	12.7344	3.842	0
8	17	20	7	12.7031	3.8919	1
26	21	19	10	12.7021	3.9047	0
8	17	21	18	12.6836	3.9084	1
5	9	15	6	12.6748	3.7754	0
8	5	17	7	12.6699	3.7712	1
14	10	9	1	12.667	3.6045	0
21	17	24	31	12.6572	3.8624	1

Table 3. Top ten rotation distances generating diffusion matrix (*DM*)'s with highest mean for MCC

<i>i</i>	<i>j</i>	<i>k</i>	<i>l</i>	(<i>DM</i>)'s mean	(<i>DM</i>)'s SD	No. of Byte aligned rotations
5	17	8	10	13.4014	3.4565	1
16	21	15	8	13.3906	3.3981	2
5	19	9	4	13.3857	3.1879	1
16	21	13	20	13.3838	3.0566	1
4	21	7	21	13.3535	3.4454	0
15	22	13	5	13.3535	3.317	0
6	11	25	20	13.3525	3.2889	0
26	10	24	3	13.3477	3.2605	1
15	21	12	25	13.3291	3.4676	0
14	6	11	5	13.3213	3.469	0

rotation distances for MCC by its authors was somewhat better than that of Salsa and ChaCha. Improvement in standard deviation of (*DM*) was also observed with these proposed rotation distances.

(c) MCC and ChaCha outperform Salsa. Figure 1 clearly reflects that Double round of MCC and ChaCha core perform better than Salsa’s Double round. For more than **950 thousand (about 91 %)** sets of rotation distances, ChaCha core generates more diffusion than the best possible diffusion generated by Salsa. For MCC, there are more than **970 thousand i.e. about 92.7 %** set of rotation distances that generate more diffusion than best possible diffusion by Salsa.

(d) Performance of MCC is found to be better than ChaCha. For MCC, we found more than 1900 sets of rotation distances generating (*DM*) with mean greater than 13. On the contrary, for ChaCha, this study could not find even a single set of rotation distances that generate (*DM*) with mean greater than 13. The graph in Fig. 2 reflects the similar scenario. This graph is zoomed version of Fig. 1 and presents only those

diffusion matrices where mean is greater than 12. For MCC, there are more than 330 thousand points (each representing one (DM) whereas for ChaCha, graph contains around 41000 points.

For at least **18 thousand** set of rotation distances, MCC generates diffusion matrices with mean greater than highest possible mean (12.7705) generated by ChaCha's diffusion matrix. However, highest possible mean of 12.7705 of ChaCha was not generated using prescribed rotation distances. Prescribed rotation distances generate (DM) with mean of 11.4297 and has more than 530 thousand (about 51 %) rotation distances that generate with higher mean than this.

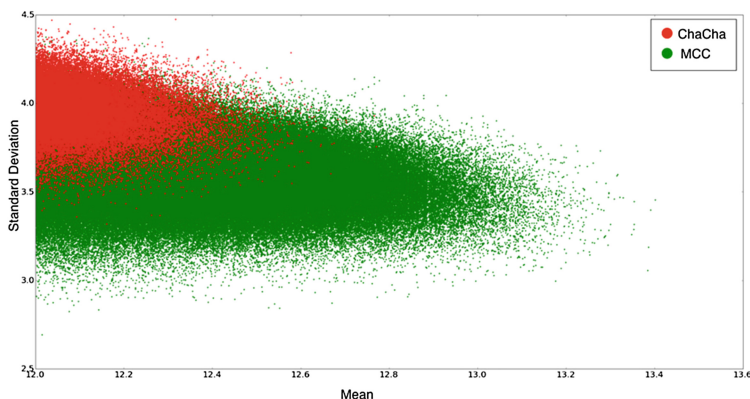


Fig. 2. Comparison of ChaCha, and MCC for diffusion matrices having mean greater than 12

5 Conclusion and Future Work

Double round of MCC core performs better than its counterparts by generating more diffusion. Comparison of performance on prescribed rotation distances reflects that MCC, on an average, has 35 % more words which get close to ideal diffusion. Double round of MCC produces average diffusion of 12.6027 compared to 6.9645 by Salsa and 11.4297 by ChaCha i.e. **improvement of about 10.3 % from ChaCha and 80.9 % from Salsa**. This study also proposes alternative rotation distances for all these competing cores and the proposed rotation distances increase diffusion up to 12.65 %, 11.73 %, and 6.4 % for Salsa, ChaCha, and MCC respectively.

This paper does not propose any specific number of rounds for MCC. In fact, it will depend on the cryptographic primitive constructed using MCC and the number of full diffusions (diffusion factor) required. The pseudo-run of MCC reflects that it achieves full diffusion in second round (even before 2nd round if completed). So to achieve diffusions factor of more than 5 (SHA-256 has diffusion factor of 4.6), ten rounds of MCC will be good enough.

As a future work, it will be interesting to evaluate whether the existing work on cryptanalysis of Salsa and ChaCha [5] is applicable on MCC also or not. Similar to Rumba [6] and Blake [7], MCC or its modification may be used to design compression function of a cryptographic hash also.

References

1. Bernstein, D.J.: The Salsa20 family of stream ciphers. In: Robshaw, M., Billet, O. (eds.) *New Stream Cipher Designs*. LNCS, vol. 4986, pp. 84–97. Springer, Heidelberg (2008)
2. ECRYPT: European Network of Excellence for Cryptology, The eSTREAM Project. <http://www.ecrypt.eu.org/stream/project.html>
3. Bernstein, D.J.: ChaCha, a variant of Salsa20. <http://cr.yp.to/chacha/chacha-20080128.pdf>
4. Sobti, R., Ganesan, G.: Analysis of quarter rounds of Salsa and ChaCha core and proposal of an alternative design to maximize diffusion. *Indian J. Sci. Technol.* **9**(3), 1–10 (2016)
5. Aumasson, J.-P., Fischer, S., Khazaei, S., Meier, W., Rechberger, C.: New features of Latin dances: analysis of Salsa, ChaCha, and Rumba. In: Nyberg, K. (ed.) *FSE 2008*. LNCS, vol. 5086, pp. 470–488. Springer, Heidelberg (2008)
6. Bernstein, D.J.: The Rumba20 compression function. <http://cr.yp.to/rumba20.html>
7. Aumasson, J.P., Henzen, L., Meier, W., Phan, R.C.-W.: SHA-3 Proposal BLAKE. http://csrc.nist.gov/groups/ST/hash/sha-3/Round3/submissions_rnd3.html

A Secure Keyword Ordered Multiuser Searchable Encryption Framework

Kulvaibhav Kaushik^(✉) and Vijayaraghavan Varadharajan

Infosys Ltd., Electronics City, Hosur Road, Bengaluru 560100, India
{kulvaibhav_kaushik, vijayaraghavan_v01}@infosys.com

Abstract. Recent trends in information technology have triggered the shift in various sectors from traditional methods of operation and data management to web based solutions. Cloud computing provides the best alternative, providing storage as a service and ensures efficient data operations. Data outsourcing reduces high cost and increases efficiency but it is vulnerable to leakage and manipulation hence rendering it unusable for most of the practical applications. Data encryption makes it safe but limits the scope for search and multiuser access. The model proposed supports efficient data encryption and search over the encrypted data by legitimate users and facilitates multiuser access to the data. Multiuser searchable encryption allows multiple users to access the data in both read and write mode with their distinct keys and facilitates addition and re-invocation of users with less overhead. The proposed scheme is based on encryption of data, keyword generation and search based on bilinear pairing. The system provides a proxy server which manipulates user queries making the system faster and more secure. The search is performed on keyword ordered list which makes the search faster as compared to the traditional methods of comparing trapdoor with keywords associated with the different files.

Keywords: Multiuser searchable encryption · Keyword ordering · Proxy server · Bilinear pairing · Discrete log problem

1 Introduction

The advancement in information technology has made data growth inexorable in all the sectors. A large percentage of organizations' expenditure is incurred on the data storage and maintenance. Infrastructure as a service provided by third party remote servers offer an alternative to the organizations thereby reducing the cost incurred on data storage and maintenance and provide efficient data management. Remote data storage and third party involvement triggers the need for strong data security and integrity mechanisms. There exists a tradeoff between data security and search efficiency of the encrypted data. Leading web service enterprises have come up with cloud computing solutions trying for a balance between the two. The quality of cloud provider is judged on the following parameters.

(1) Data Security: It is the most determining requirement. Data security involves an efficient data encryption, use of secure channels for data transmission and storage security. Even if compromised, the ciphertext should not leak any information.

The encryption method should be secure enough to prevent any learning on data pattern and queries by a curious server.

(2) Efficient search: The search must provide all valid results in minimum time. Searching over plaintext is easy and fast but provides the cloud with the access to the data. Data encryption highly limits the search capability over the data. In real life applications multiple users access same data and all the different users require search capability over the data.

(3) Dynamics: The system must be adaptive to change at the user end. The data stored at the cloud should be modifiable with provision of changing user access to the data. Addition of new user to system and re- invocation of existing users must be facilitated without affecting the data storage. Also addition of new files and removal of obsolete ones must be easy without involving any overheads.

1.1 Our Contribution

The framework provides solution to the multiuser searchable encryption. An architectural model has been proposed with data owner/user, proxy server, web server and key generator. The algorithm provides for multiuser search on encrypted data at the server without key sharing between users. The proxy maintains a keyword ordered table which aids in increasing efficiency and privacy in data search. Trapdoor based search using random elements makes the system more secure. The model when applied to healthcare scenario resulted in tremendous improvement in efficiency.

2 Related Work

Searchable encryption has emerged as an area of advanced research with the recent growth in data outsourcing and cloud computing [11]. Lots of work has been done on searchable encryption with Boneh and Franklin [1] trapdoor generation and corresponding search method being a pioneer in the field. The method is based on double hashing of the keyword for trapdoor generation. The cryptographic cloud storage scheme [7] has provided an architectural model for data encryption and search where three entities viz mega corporation, partner corporation and cloud storage are involved. This method has been widely implemented and has formed the base for all the architectures, with changes in the functionalities of each of the entities involved. The dynamic searchable symmetric encryption [10] provides a method for keyword based search but is limited to a single user scenario and the mechanism cannot be extended to multiuser system. These schemes are based on single user encryption and cannot be implemented in a multiuser scenario.

The current focus of searchable encryption is on multiuser search. The private query on encrypted data in multiuser setting [9], involves maintenance of a table at the server end, and for each of the keywords to be associated with the server, the corresponding ciphertext are generated as indices at the server and then provided to the user, who combines the indices with the original message which is stored at the provider. The scheme involves high amount of data transfer and also much of the control is at

providers end. A curious server may be able to extract large quantity of information, making the system vulnerable. The no shared key approach [8, 12] provides a scheme, involving an intermediary proxy which performs an additional encryption/decryption over it. The scheme provides multiuser access but the comparison queries performed is directly proportional to the number of files and thus may not be applicable where the number of files is quite high.

3 Preliminaries

Bilinear pairing [2, 3, 6] has been extensively used in cryptographic applications and forms an important part of our one time trapdoor scheme.

Consider two cyclic groups of prime order q , $(G_1, +)$ and (G_2, \cdot) . Then the map between the two is given by $\hat{e} : G_1 \times G_1 \rightarrow G_2$ with the following properties:

$$\text{Bilinear} : \forall P, Q \in G_1, \forall a, b \in \mathbb{Z}_q, \quad \hat{e}(aP, bQ) = \hat{e}(P, Q)^{ab} \tag{1}$$

$$\text{Non Degenerate} : \forall P \in \text{generator}(G_1), \quad \hat{e}(P, P) \in \text{generator}(G_2) \tag{2}$$

Computable: There exist algorithm to calculate $\hat{e}(P, Q)$ given any $P, Q \in G_1$. The Weil pairing and Tate pairing are amongst the method available for calculating elliptical pairing. A map with the above characteristics is said to be an admissible bilinear pairing.

We also elaborate over an existing complexity problem for bilinear pairing, Bilinear Diffie-Hellman problem. Consider a group G_1 with generator P and $\alpha, \beta, \gamma \in \mathbb{Z}_q^*$.

Given $P, \alpha P, \beta P, \gamma P \in G_1$ and the result $\hat{e}(P, P)^{\alpha\beta\gamma}$. If A is an algorithm to solve the given problem with probability ρ then

$$\text{Pr} \left[A(P, \alpha P, \beta P, \gamma P) = \hat{e}(P, P)^{\alpha\beta\gamma} \right] \geq \rho \tag{3}$$

Here the probability depends on random choice over, Generator $P \in G_1$

$$\alpha, \beta, \gamma \in \mathbb{Z}_q^* \tag{4}$$

From above it is deductible that for Bilinear Diffie-Hellman problem there exist no algorithm to solve the problem in polynomial time with non-negligible probability.

The discrete logarithm problem [4, 5] is defined over finite cyclic group. If g and h are elements of a finite cyclic group, then the solution x to the equation $g^x = h$, is called as the discrete logarithm of h to the base g . The discrete log problem has been extended to elliptical curves as well. Consider an elliptical curve E defined over finite field F_q of prime order q . For any point P on the curve i.e. $P \in E(F_q) \setminus \{O\}$ where ‘ O ’ is the point at infinity, we compute

$$Q = d \cdot P \tag{5}$$

Given P and Q there exists no polynomial time algorithm to find d with a non-negligible probability. This makes the problem quite relevant for usage in cryptography and has been extensively used in our approach for secret transmission of data.

4 Architectural Model

The framework proposes a proxy based search over a keyword ordered table. Figure 1 shows the different entities involved and the flow of data.

Data User: User saves the data and performs search over it. The user possess keys for data encryption, trapdoor generation and decryption. The keys are provided by the key generator. In the healthcare scenario user comprise of patients, provider (medical practitioner, organization, etc.) and payer (insurer).

Proxy Server: It is an intermediary server which receives the user query and performs a search over a set of tables existing there and fetches the corresponding files from the web server. It possesses a secondary key corresponding to each user and helps in data security and user revocation. It is maintained in-house by the provider or payer or any third party service provider.

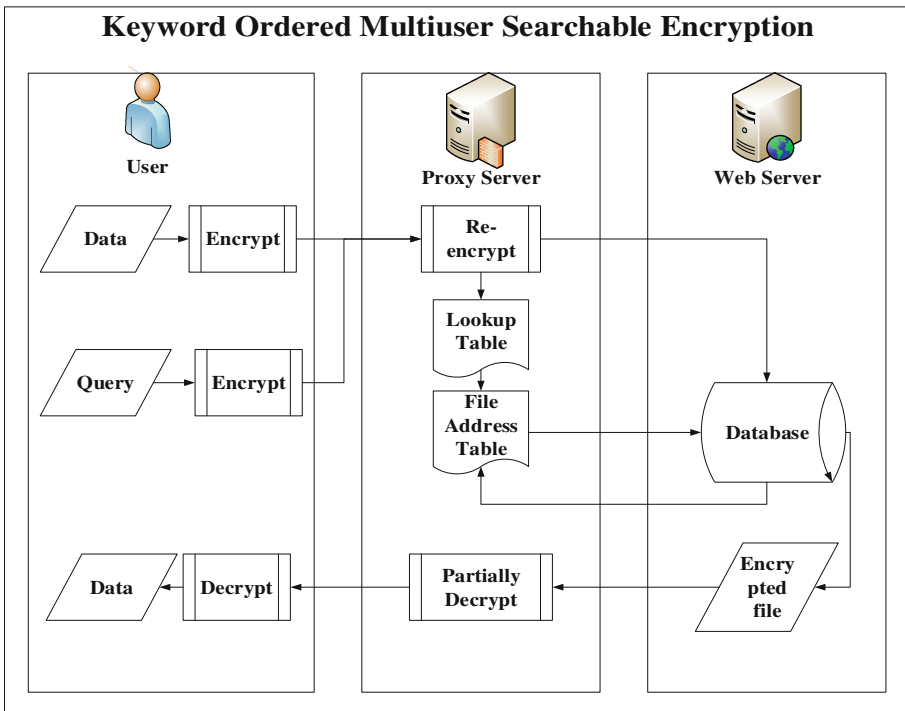


Fig. 1. Architectural model

Web Server: It is the server where the encrypted data is saved. The proxy server requests it for a particular file at a specific position and it returns the required file. No other operations are performed in the server. Its ownership is similar to proxy server.

In addition there exists a key generator which generates the keys for the user and the proxy server.

5 Algorithm

Multiuser keyword based searchable encryption requires data storage at the server via proxy. The data flow between the user and the server is given in the following steps:

5.1 Setup

The key generator generates system parameters based on the security parameter k . It returns cyclic group G_1 and G_2 of prime order q with admissible bilinear pairing $\hat{e} : G_1 \times G_1 \rightarrow G_2$. Here P is the generator for the group G_1 . Also it generates a random element $t \in \mathbb{Z}/q\mathbb{Z}$, secret to the key generator. We also define a hash function $H : \{0, 1\}^* \rightarrow \{0, 1\}^*$.

5.2 Key Generation

The key generator selects a random number s_i for every user. It also generates a random element $K_i \in G_1$. It generates the user key for user i as

$$UK_i = \left\{ \frac{t \cdot P}{s_i}, K_i^{1/s_i}, K_i \right\} \quad (6)$$

And sends it to the user i . It also generates a server key SK_i corresponding to user i and send it to the server.

$$SK_i = \{s_i\} \quad (7)$$

5.3 Encryption

Consider a case where the user wants to save the data item m . User associates the data with a set of keywords $KW = \{W_1, W_2, \dots, W_n\}$. The user generates a random number y and encrypts the data using key K_i^y using a symmetric encryption scheme. The user encrypts the key and keyword as

$$K'_i = K_i^{y/s_i} \cdot \hat{e} \left(P, \frac{t \cdot P}{s_i} \right) \quad (8)$$

$$W'_l = \hat{e}\left(H(W_l) \cdot P, \frac{t \cdot P}{s_i}\right) \tag{9}$$

where $l \in [1, n]$. It sends to proxy $\left\{Enc_{K'_i}^{SYM}(m), K'_i, W'_1, W'_2, \dots, W'_n\right\}$. Proxy server re-encrypts the key and keyword as

$$K_i^* = (K'_i)^{s_i} = \left(K_i^{y/s_i} \cdot \hat{e}\left(P, \frac{t \cdot P}{s_i}\right)\right)^{s_i} = K_i^y \cdot \hat{e}(P, t \cdot P) \tag{10}$$

$$W''_l = (W'_l)^{s_i} = \left(\hat{e}\left(H(W_l) \cdot P, \frac{t \cdot P}{s_i}\right)\right)^{s_i} = \hat{e}\left(H(W_l) \cdot P, \frac{t \cdot P}{s_i}\right)^{s_i} \tag{11}$$

$$= \hat{e}(H(W_l) \cdot P, t \cdot P)$$

$$W_l^* = H(W''_l) \tag{12}$$

The proxy generates a random number x , and sends to server $\left\{Enc_{K_i^*}^{SYM}(m), K_i^*, x \cdot P\right\}$

The server saves the encryption $\left\{Enc_{K_i^*}^{SYM}(m), K_i^*\right\}$ at a location f_l , and returns to proxy

$$\left\{f'_l = Enc_{xb \cdot P}^{SYM}(f_l), b \cdot P\right\} \tag{13}$$

Where b is a random number generated by server. Using the value of x and $b \cdot P$, proxy computes $xb \cdot P$ and retrieves $f_l = Dec_{xb \cdot P}^{SYM}(f'_l)$. Proxy maintains two lookup Tables 1 and 2. It checks if the keyword entry already exists in the Table 1.

Table 1. Keyword position table

Keyword Enc	Enc Position in Table 2
.....

Case I: If the entry doesn't exist in Table 1, Proxy saves in it: $\left\{W_l^*, Enc_{W_l^*}^{SYM}(b_l)\right\}$

Where b_l is an empty location in the Table 2 at position b_l in II, it saves $\left\{b_l, Enc_{W_l^*}^{SYM}(f_l), Enc_{W_l^*}^{SYM}(z)\right\}$ where b_z is null since only one data file exists for the keyword.

Case II: If the entry exists in Table 1. It retrieves b_l from 1: $\left\{W_l^*, Enc_{W_l^*}^{SYM}(b_l)\right\}$. It looks for b_l in 2 and retrieves the value $\{b_l, b_z\}$. It saves in Table 2 $\left\{b_l^*, Enc_{W_l^*}^{SYM}(f_l), Enc_{W_l^*}^{SYM}(b_z)\right\}$ where b_l^* is an empty location in Table 2 and modifies the Table 2 by saving for position b_l : $\left\{b_l, \dots, Enc_{W_l^*}^{SYM}(b_l^*)\right\}$.

Table 2. Corresponding address mapping table

Position	Encrypted file address	Encryption Position in Table 2
.....

5.4 Search

We consider a scenario where user j performs a search over the cloud for keyword W_1 . Proxy generates a random no v and sends to user $v \cdot P$. It serves as a session key, being useful for the session and will be deleted on session expiration. User sends $Qry_j = \hat{e}\left(H(W_1) \cdot v \cdot P, \frac{t \cdot P}{s_j}\right)$ to proxy. Proxy translates the query as:

$$Qry'_j = (Qry_j)^{s_j/v} = \hat{e}\left(H(W_1) \cdot v \cdot P, \frac{t \cdot P}{s_j}\right)^{s_j/v} = \hat{e}(H(W_1) \cdot P, t \cdot P) \quad (14)$$

Here, every query for same keyword by a particular user will be different and no learning may be made based on data sniffing from channel. Proxy computes $Qry^*_j = H(Qry'_j)$. It checks Table 1 for Qry^*_j . If no matching is found, the keyword being search doesn't exist. If a matching is obtained, the corresponding value b_1 is retrieved as

$$b_1 = Enc_{Qry^*_j}^{SYM}(Search\ result\ from\ table1) \quad (15)$$

The proxy looks for the position b_1 in Table 2 and retrieves the corresponding address for the file saved at server using Qry'_j . It also gets the next address in II where the next file is saved. Once retrieved the address for the different files it retrieves the corresponding entry from the server.

5.5 Decryption

The user j , sends to proxy along with the search query,

$$K'_j = K_j^{\alpha/s_j} \cdot \hat{e}\left(P, \frac{t \cdot P}{s_j}\right) \quad (16)$$

where α is a random number. The proxy re encrypts K'_j and returns K''_j .

$$K''_j = (K'_j)^{s_j} = \left(K_j^{\alpha/s_j} \cdot \hat{e}\left(P, \frac{t \cdot P}{s_j}\right)\right)^{s_j} = K_j^\alpha \cdot \hat{e}(P, t \cdot P) \quad (17)$$

$$K_j''' = \frac{K_j'}{K_j''} = \frac{K_i^y \cdot \hat{e}(P, t \cdot P)}{K_j^z \cdot \hat{e}(P, t \cdot P)} = \frac{K_i^y}{K_j^z} \tag{18}$$

User computes the decryption key $K_j^* = \frac{K_j^y}{K_j^z} \cdot K_j^z = K_i^y$. The user can decrypt the data as

$$m = Dec_{K_j^*}^{SYM}(encrypted\ data) \tag{19}$$

6 Analysis

The proposed scheme for keyword based multiuser searchable encryption provides as a faster and more secure way for data storage, search and retrieval. Here we describe the application of the proposed scheme to the healthcare domain. We consider an e-healthcare application where the medical data for various patients are stored online. The actual data is stored at a remote and outsourced data store i.e. at a cloud. The data stored at the cloud if leaked may lead to heavy losses. Also while transmission of data from the user to the cloud, it may be leaked. Hence the data requires an efficient encryption mechanism. The encryption of data renders it unsearchable.

In our scheme we associate every data item with a set of keywords and hence the search is performed using the keyword. Now let us assume that there are n files. We assume that each file is associated with k keywords. Let the total number of keywords be m. Now we have two options:

File based search: Every file is associated with the keywords [Table 3] and to retrieve the required files we match the keywords and the search trapdoors. For the given scenario it would require $n \times k$ search operations.

Keyword based search: The keywords are arranged and all the files that contain the particular keyword are mapped with it [Table 4].

Table 3. Storage mechanism in file based search

Filename	Keyword list
File 1	Keyword 1, ... Keyword k
...	...
File n	Keyword 1, ... Keyword k

Table 4. Storage mechanism in keyword based search

Keyword	File list
Keyword 1	Associated files
...	...
Keyword m	Associated files

On an average each keyword is mapped with $\frac{k \cdot n}{m}$ files. When the user search for a keyword the user performs a match with the m different keywords and once found the keyword the search stops and the no further search is performed.

Let the user search for a particular keyword. We assume that the probability of distribution of the keyword in the list be same. Hence the probability that the search with first keyword is successful is same as probability the keyword is stored at the last position in the search table. Hence on an average the number of search operations required to find the keyword are:

The summation [Table 5] is: $S = 1 + 2 + 3 + 4 + \dots + m = \frac{m \cdot (m + 1)}{2}$

Hence on an average number of search operations performed are:

$$\frac{S}{m} = \frac{\frac{m \cdot (m + 1)}{2}}{m} = \frac{m \cdot (m + 1)}{2 \cdot m} = \frac{m + 1}{2}$$

Now considering the healthcare scenario we will consider the type of search queries being made.

A patient may search for the data that belongs to the patient itself. Generally the patient performs search using the patient name or the patient ID as keyword which will return all the data that belongs to that particular patient. In this case it will be much faster for him to retrieve the data since all the required files are associated with the particular keyword the patient wants to search for.

A doctor may search for the data that belongs to a particular department, ex Cardiology or all the cases which the doctor has examined or for a patient’s medical data. Hence the examining doctor and the department may be additional keywords.

A hospital may search for data that is associated with the particular hospital and hence hospital name may act as other keyword.

We find that keywords form only a limited set and hence $m < n$. So the number of search queries in keyword based search will be less than queries in file based search. $\frac{m+1}{2} \ll k \cdot n$.

In the proposed scheme every time the user saves a data we associate it with keyword. The data storage and the keyword based search should be able to facilitate a multi user environment. In the proposed scheme, we support multiuser environment. For example, the data saved by the doctor is searchable and accessible by the concerned patient and hospital.

Also the problem of eavesdropping to learn search patterns has been eliminated by adding random parameters in data and keyword storage, search and retrieval. Ex: If a doctor searches a particular patient’s profile again and some person eavesdrop, he may be able to make a learning that the particular patient has fallen ill again. Our scheme removes such possibility.

Table 5. Computation for search operations

Keyword position	Number of search operations
1	1
...	...
m	m

7 Comparative Analysis

We provide a comparison between the proposed keyword ordered multiuser searchable encryption scheme (KOMSE) and the existing DLP based multiuser searchable encryption scheme (MSE) [8].

7.1 Methodology

MSE: File based search where each file is associated with a set of keywords.

KOMSE: Keyword based search where each keyword is associated with a set of files. Also the server doesn't learn which file was queried by which particular user. Providing keyword based search using current state of MSE involves challenges.

7.2 Computation Analysis

MSE:

Keyword storage: 2 exponentiation operations

Encryption key storage: 2 exponentiation operations

Query: 2 exponentiation operations

Encryption key decryption: 2 exponentiation operations

Comparison: for n files with k keywords each, no of comparison performed are equal to $n \times k$.

KOMSE:

Keyword storage: 1 multiplication, 2 pairing and 1 exponentiation

Key Encryption: 1 multiplication and 2 exponentiation

Query: 1 multiplication, 2 pairing and 1 exponentiation

Key decryption: 3 multiplication and 3 exponentiation

For a list of total w keywords, number of comparison performed are $\frac{w+1}{2}$

7.3 Security

MSE: It is based on DLP over finite field.

KOMSE: Based on the twin hard problems of ECDLP and pairing inversion and thus offers higher security.

7.4 Randomness

MSE:

In data storage: Every time a user save a particular keyword, it passes the same encryption which may lead to learning.

In search queries: Every time a makes a particular query, same encryption of query is generated. Hence learning may be performed.

KOMSE:

In data storage: Every time a user sends a keyword for storage, it is sent as different ciphertext. Hence no learning may be done.

In search queries: Every time a user search for a keyword, different ciphertext is sent over channel. Hence learning may not be performed.

Providing with randomness with the DSE involves challenges.

We analyze the healthcare search for the two mechanism using p, PS III machine. Let a machine makes l operations in one second. Hence time taken to perform z operations is

$$T_z = \frac{z}{p \times l} \tag{20}$$

For a PSIII machine $l = 27.67 \times 10^6$

For a provider owned healthcare organization, for every visit of a patient a separate file exists. There exists different files for pathological tests and personal details. Let number of patients be q with each patient visiting hospital r times an year. For each visit the number of files generated be s. Considering a life expectancy of ~ 68 (67.8) years, average life of a patient is $\frac{68+0}{2} = 34$. Hence total number of files in hospital is

$$\#(Files) = \text{personal data} + \text{visit data}$$

$$\#(Files) = q + 34 \times r \times s \times q$$

The keywords are patient ID, treating doctor, department, pathology lab, disease diagnosed.

Keywords associate with each file: k = 5

Since,

Number of patients \gg *no of doctors, departments, pathlabs, disease*

$$\text{number of keywords}(m) = \text{no of patients} = q$$

For a hospital catering to 5,000,000 patients with a patient visiting every quarter, on each visit files are made for prescription, pathology report and billing. Let the provider owns 10 PS III machines.

$$q = 5000000 \ \& \ r = 4 \ \& \ s = 3$$

Hence, Time taken for file based searchable encryption is:

$$T_z = \frac{z}{p \times l} = \frac{n \times k}{p \times l} = \frac{(q + 34 \times r \times s \times q) \times (5)}{p \times l}$$

$$T_z = \frac{(5 \times 10^6 + 34 \times 4 \times 3 \times 5 \times 10^6) \times (5)}{(27.67 \times 10^6) \times (10)}$$

$$T_z = 37 \text{ s}$$

Considering the proposed keyword based mechanism

$$T_z = \frac{(m+1)/2}{p \times l} = \frac{(q+1)/2}{p \times l}$$

$$T_z = \frac{(5 \times 10^6 + 1)/2}{(27.67 \times 10^6) \times (10)}$$

$$T_z = .009 s$$

Hence the time taken in the proposed approach is 4.1×10^3 times less.

8 Conclusions

The prime concern of the data owner lies in the security of the data outsourced. To avoid any lacuna in data security the proposed scheme eliminates the possibility of data sniffing by curious servers by usage of an intermediary proxy server which receives the user query, and routes it to the server. Also the search is faster as only matching operations are performed which are considerably lesser in number since the search is made over the keywords and not over the files. The keyword based multiuser searchable encryption scheme is based on non-polynomial hard problems of discrete logarithm and elliptical pairing inversion. It also facilitates addition of new records to the database and deletion of the existing records. The user revocation is also possible by deletion of the secondary key corresponding to the user at the proxy. The proposed framework provides with a secure, efficient and dynamic way of data storage at a remote server in a multiuser scenario.

References

1. Boneh, D., Franklin, M.: Identity-Based Encryption from the Weil Pairing. *SIAM J. of Computing* **32**(3), 586–615 (2003)
2. Certicom Research: Standards for Efficient Cryptography Group (SECG) – SEC 1: Elliptic Curve Cryptography. Version 1.0, 20 September 2000. http://www.secg.org/secg_docs.htm
3. Meffert, D.: Bilinear pairings in cryptography. Master's thesis, Radboud Universiteit Nijmegen (2009)
4. Studholme, C.: The discrete log problem. Research paper requirement (milestone) of the Ph. D. program at the University of Toronto, June 2002
5. Menezes, A.: Evaluation of security level of cryptography: the Elliptic Curve Discrete Logarithm Problem (ECDLP), University of Waterloo (2001)
6. Galbraith, S., Hess, F., Vercauteren, F.: Aspects of pairing inversion. *IEEE Transactions on Information Theory* **54**, 5719–5728 (2008)
7. Kamara, S., Lauter, K.: Cryptographic cloud storage. In: *Proceedings of Financial Cryptography: Workshop on Real-Life Cryptographic Protocols and Standardization*, January 2010

8. Dong, C., Russello, G., Dulay, N.: Shared and searchable encrypted data for untrusted servers. *J. Comput. Secur.* **19**(3), 367–397 (2011)
9. Yang, Y.J., Ding, X.H., Deng, R.H., Bao, F.: Multi-user private queries over encrypted databases. *Int J High Perform. Comput. Netw.* **1**(1/2/3), 64–074 (2008)
10. Kamara, S., Papamanthou, C., Roeder, T.: Dynamic searchable symmetric encryption. In: *The Proceedings of the 2012 ACM conference on Computer and communications security.* ACM (2012)
11. Raso Mattos, L.R., Varadharajan, V., Nallusamy, R.: Data protection and privacy preservation using searchable encryption on outsourced databases. In: Thampi, S.M., Zomaya, A.Y., Strufe, T., Alcaraz Calero, J.M., Thomas, T. (eds.) *SNDS 2012. CCIS*, vol. 335, pp. 178–184. Springer, Heidelberg (2012)
12. Varadharajan, V., Mani, R., Nallusamy, R.: Anonymous searchable encryption scheme for multi-user databases. In: *Proceedings of the 2013 IEEE International Conference on Cloud Engineering*, pp. 225–232

Cryptographic Assessment of SSL/TLS Servers Popular in India

Prakhar Jain^(✉) and K.K. Shukla

Indian Institute of Technology (BHU), Varanasi, India
{prakhar.jain.cse11,kkshukla.cse}@iitbhu.ac.in

Abstract. Major web sites use Secure Sockets Layer (SSL) or its updated version name called Transport Layer Security (TLS) to secure all communications between their servers and web browsers. It is very important to analyze the security of this protocol because the compromise of the banking accounts, health care directories, information of national importance, even vital information about business competitors is unacceptable.

SSL/TLS is not a simple encryption or hashing algorithm. It is a protocol which consists of bunch of cryptographic primitives which aim to provide secure communication. Moreover, this protocol has a long history of attacks and it needs to be revised since security field is changing. This paper presents the most commonly used configurations of this protocol among web servers, highlighting issues where it is insecure and areas where it can be improved. Specifically, parameters used in cryptographic primitives and certificates used by the web servers have been reported. The approach was to probe all web servers using a tool - TestSSLServer. There were sets of two experiments carried out. One in which top 500 most popular websites in India were probed and other in which 50 banking sites in India were probed. Some of the surprising results were that servers still possess SSLv2 and v3 despite of its insecurity. Also, banking sites were found not to support forward secrecy.

1 Introduction

With the widespread usage of Internet, there is a revolution taking place. This revolution started with the usage of web browsers and is continuing on it. Browsers were not designed with security in mind. SSL/TLS protocol provides secure communication within browser. The protocol is complicated because of its dependency on public key cryptography, digital signatures, symmetric encryption, MACs and hash functions. The security lies on what is least strongest within these dependencies. Thus, we have to ensure that all the dependencies work correctly.

The approach was to analyse the security of the SSL/TLS servers present in India. There are lot of Internet users in India and it is fishing ground of major software companies. But, India has not taken the issues of security seriously. This is evident from the lots of attacks being heard in news. The objective of this work

was to see what is the current scenario of security in India. So, we analysed the top 500 web servers visited in India [15]. Security is very important in banking region. So, we also analysed the Internet Banking sections of the 50 banking sites in India [16]. The work was to extract all host names of these sites and then compile a list for further analysis.

The tool used for probing all the host names is TestSSLServer by Pornin [17]. The tool is very simple, all you need is to provide the host name and some arguments as per requirements and a detailed report of the SSL/TLS connection of the web server is generated. The report is in the form of JSON. Now, to get a good scene of all the servers, a program to extract all important information from this list of JSONs was created. A simple C json parser developed by Dave Gamble [24] was used. A C++ program was required to get all the information. The project can be found on [25].

Several results were found which are explained later in the Results section. Some problems such as SSLv2, SSLv3 were supported by many servers despite its prohibition. Also, there were very less servers with Elliptic Curve support. Most of the top 500 websites visited in India gives an option to create account and thus transfers confidential information. But, the level of security was average. There were broken ciphers used. Even MD5 is still being popularly used despite so many attacks on it. The major banking sites were not supporting forward secrecy which is of utmost important especially in a country like India where there are many powerful adversaries.

The rest of the paper consists of Sect. 2 where the SSL/TLS protocol is described, Sect. 3 where some definitions related to public key infrastructure is described, and Sect. 4 where the result is described.

2 The Protocol

SSL/TLS is a protocol designed to provide secure communication over insecure infrastructure. It aims to provide confidentiality, integrity and authenticity between the two users. The Internet infrastructure which is built on top of TCP/IP is insecure. Basic applications level protocols such as HTTP are insecure. Anyone can tamper with it, read the contents and even masquerade some one else.

There are two broad ways in which the protocol can be attacked. Either by decrypting the encrypted data or by impersonating. When encryption (Here, encryption means encryption along with MAC) is deployed, the attacker might be able to gain access to the encrypted data, but he/she wouldn't be able to decrypt it or modify it. To prevent impersonation attacks, SSL/TLS rely on another important technology called PKI (public-key infrastructure), which ensures that traffic is sent to correct recipient.

TLS v1.2 which is the latest protocol version is discussed in the paper. At a high level, TLS is implemented via the record protocol which consists of sub-protocols. It has a short header which contains information about protocol version, sub protocol used and length. Message data follows the header. More about

protocol version will be discussed in Sect. 4.1. The record protocol is a useful abstraction which takes care of

- **Data Transmission:** The higher level protocol submits its data to record protocol which after handshake, applies encryption and integrity validation according to the negotiated connection parameters. Occasionally it breaks the data into multiple parts or joins them before applying encryption.
- **Compression:** It optionally provides compression. But, this feature has suffered from CRIME attack [3] and is no longer used.
- **Extensibility:** Since all the work is being done by sub protocols, it makes the protocol extensible. Record protocol does the job of data transmission and encryption.

There are four sub-protocols - handshake, alert, change cipher spec and application data.

2.1 Handshake

Handshake is a major sub protocol where the connection parameters are transferred between the server and client. The use of handshake is to:

1. Decide which connection parameters to use
2. Verify the certificate.
3. Exchange information to generate *mastersecret* which will be further used for key generation.
4. Verify whether the handshake is being modified by third party or not.

There are three major types of handshakes:

1. Full handshake with server authentication.
2. Full handshake with both client and server authentication.
3. Abbreviated handshake - used to resume an already established session.

I will explain full handshake with server authentication (the most common) with figure taken from TLS specification [13]:

1. **ClientHello:** Client submits a random value. This is used in generating master secret and ensures that new connection is negotiated. It submits in this case an empty session id since it wants to establish a new session. It also shares cipher suites, signature algorithms and Elliptic Curve parameters (if any) it supports.
2. **ServerHello:** Server submits its part of random value. It also submits the session id and chosen cipher suite.
3. **Certificate:** Server sends appropriate certificate for the public key to be used in connection and signature algorithms that client supports.
4. **ServerKeyExchange:** Server sends its part of parameters required to generate pre master secret which is used to generate master secret. This is optional since some key exchange mechanism dont require servers role in key exchange.
5. **ServerHelloDone:** Server sends a messages saying that it has completed its part of Hello.

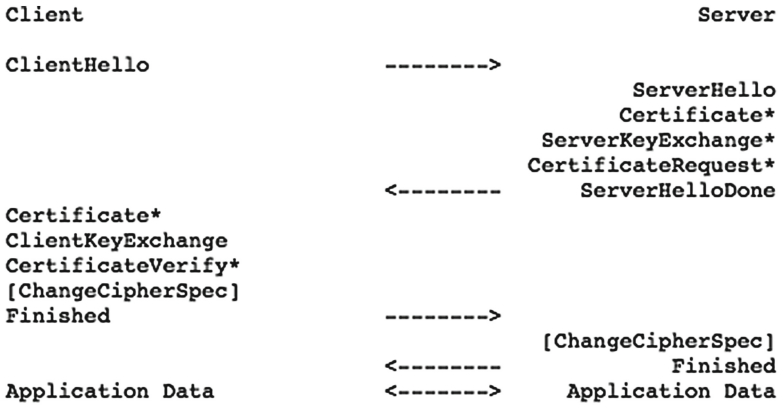


Figure 1. Message flow for a full handshake

* Indicates optional or situation-dependent messages that are not always sent.

Fig. 1. Message flow for a full handshake. * Indicates optional or situation-dependent messages that are not always sent

6. ClientKeyExchange: Client sends submits its part of parameters required for key exchange.
7. ChangeCipherSpec: It is a change_cipher_spec subprotocol message used to indicate that the party has computed the required parameters for secure connection and is shifting to encryption.
8. Finished: It is a hash of all handshake messages as each side saw them, mixed with the newly negotiated master secret.
9. Server similarly switches to encryption with ChangeCipherSpec and sends the Finished Message

2.2 Application Data

As far as TLS is concerned, this is just buffers of application data. Record protocol fragments and encrypts these messages.

2.3 Alert

Alert are simple notification mechanism for the other side. An example is close_notify alert which is send if one of the party wishes to close the connection.

2.4 Key Exchange and Authentication

The goal of the key exchange is to share *premastersecret* which is the value from which the *mastersecret* is constructed. TLS supports variety of key exchange mechanism. Authentication is tightly coupled with Key Exchange.

1. RSA: The client generates the *premastersecret* and transports it to the server, encrypted with server's public key. The server's public key is authenticated via certificate. For most web sites, security provided by 2048 bit RSA keys is sufficient. At 2048 bits, it provide 112 bits of security. But the problem with RSA is that it doesn't support forward secrecy. Forward secrecy is a protocol feature that enables secure conversations that are not dependent on the server's private key. That is, in RSA, if a passive attacker has access to the server's private key, can decrypt all the encrypted data. The attack doesn't have to happen in real time. A powerful adversary could record all the traffic and then use legal powers, bribery etc. to get the server's private key.
2. Diffie-Hellman Key Exchange: The DH Key exchange requires six parameters: two (dhp and dhg) are called domain parameters and are selected by the server. During the negotiation, the client and server each generate two additional parameters. Each sends one of its parameters (dhYs and dhYc) to the other end, and with some calculation, they arrive at the shared key. The parameters are signed from server's side to provide authentication. Ephemeral Diffie-Hellman (DHE) key exchange takes place when none of the parameters is reused. In simple DH, some of the parameters are embedded in server and client certificates. In this case there is no forward secrecy since every time same key is used.
3. Elliptic Curve Diffie-Hellman Key Exchange: An ECDH key exchange take place over a specific elliptic curve, which is for the server to define. The curve takes the role of domain parameters. In theory, static ECDH key exchange is supported, but in practice only the ephemeral variant (ECDHE) is used. Elliptic Curve Key Exchange is liked because its fast and provides forward secrecy. Here, ECDH parameter size should be 256 bits which provide 128 bits of security. Also, TLS takes care to support only authentic curves.

3 Public Key Infrastructure and X.509 Standard

A public key infrastructure (PKI) is a set of roles, policies, and procedures needed to create, manage, distribute, use, store, and revoke digital certificates [4]. The goal of PKI is to enable communication between two parties who haven't met each other. This is done with the help of trusted third parties called certification authorities (CA).

X.509 is an international standard for PKI. It manages digital certificates and all subtleties in the role of PKI. X.509 was adapted for use on the Internet by the Public Key Infrastructure (X.509) working group (PKIX). CA/Browser Forum (or CAB Forum) is a voluntary group of CAs, browser vendors, and other interested parties which enforces standards for certificate issuance. CAB issues *BaselineRequirements* which is applied to all CAs [2].

A digital certificate is a document which contains public key of an entity, some information about the entity and digital signature of the certificate issuer.

4 Results

Experiment one is of probing top 500 servers and experiment two is of probing 50 banking servers. URL wise details of individual servers are available at [25]. Results are generated in the form of tables in which no. of servers out of total are given which posses certain attribute. In Sects. 4.2, 4.3 and 4.4, it is given for each of the protocol version i.e. if the server supports a protocol version and a attribute, it is being marked.

4.1 Versions

There are five protocols in the SSL/TLS family: SSL v2, SSL v3, TLS v1.0, TLS v1.1, and TLS v1.2:

1. SSL v2 is insecure and should not be used. This protocol suffers from large number of flaws like weak MAC construction, cipher suite rollback attack, Drown attack [5] etc. More about prohibiting SSLv2 can be found on RFC 6176 [26].
2. SSL v3 is insecure when used with HTTP suffers from POODLE [6] attack and weak when used with other protocols. It's also obsolete and shouldn't be used. [27] More about SSL v3 can be found on paper. [7]
3. TLS v1.0 has replaced custom construction with PRF for master secret generation and other process where it is been used. More about TLS v1.0 can be found on [8]. TLS v1.0 suffers from BEAST [9] attack which has been mitigated in modern browsers but other problems remain.
4. TLS v1.1 and v1.2 are both without known security issues, but only v1.2 provides modern cryptographic algorithms. Its a must for a server to have TLS v1.2 support. In order to support older clients, a server can have support for TLS v1.0 and TLS v1.1 (Table 1).

Table 1. SSL/TLS version support

Exp1	Servers	Exp2	Servers
SSLv2	5	SSLv2	1
SSLv3	51	SSLv3	10
TLSv1.0	374	TLSv1.0	46
TLSv1.1	344	TLSv1.1	35
TLSv1.2	336	TLSv1.2	31

In experiment 1, 5 servers and in experiment 2, 1 server was found to support SSLv2. SSLv2 is totally insecure and should be removed at the earliest. Its strange that even 1 banking website support SSLv2 protocol. It can be subjected to an attack easily by an experienced attacker. SSLv2 and v3 are present only for old clients like Windows XP. But, being insecure it should be removed. Majority of servers were found to supports TLSv1.0 protocol.

Table 2. Key exchange

Exp1	DH	DHE	ECDH	ECDHE	RSA	RSAEXPORT
SSLv2	0	0	0	0	5	0
SSLv3	0	22	1	23	51	2
TLSv1.0	1	78	3	316	362	4
TLSv1.1	1	69	2	309	331	1
TLSv1.2	1	68	2	300	325	1
Exp2	DH	DHE	ECDH	ECDHE	RSA	RSAEXPORT
SSLv2	0	0	0	0	1	0
SSLv3	0	1	0	0	10	0
TLSv1.0	0	6	0	15	46	1
TLSv1.1	0	5	0	11	35	0
TLSv1.2	0	5	0	8	31	0

4.2 Key Exchange, Authentication and Forward Secrecy

There are effectively 3 key exchange mechanisms. The difference between DH and DHE and ECDH and ECDHE is that DHE and ECHDE are ephemeral versions and support forward secrecy. More about Key Exchange, Authentication and Forward Secrecy is discussed in Sect. 2.2.

Export algorithms are obsolete and insecure and should not be used.

The security of Diffie-Hellman key exchange protocol depends on the quality of domain parameters. A server can use weak or insecure parameters and the client even if willing to communicate with strong domain parameters cannot do anything if the server enforces weak parameters. This issue was highlighted in Triple Handshake Attack research paper [10].

DH parameters are largely ignored by servers and they use libraries with weak 1024 bit parameters and insecure 768 bit parameters Logjam Attack [11] and even 512-bit parameters. This problem is currently discussed in TLS Working Group draft [12].

In experiment 1, 48 servers were found to support DH parameter less than 2048 bit while in experiment 2, 4 servers.

Most of the servers were found to support RSA as key exchange followed by ECHDE, then DHE. (Table 2) Since RSA is supported, it means that these servers don't support forward secrecy. This is a serious drawback in case of experiment 2. Banking sites not supporting forward secrecy means that a powerful adversary can record all traffic to the server and if got hold of private key can get all user credentials by decrypting the traffic. Few servers support export algorithms which should be removed at the earliest.

Anon is authentication mechanism where there is no authentication to the server. Few servers in experiment 1 support anon authentication mechanism. Its good to see that no anon authentication mechanism is found in experiment 2.

Table 3. Authentication mechanism

Exp1	Anon	RSA	RSAEXPORT	ECDSA
SSLv2	0	5	0	0
SSLv3	1	51	2	0
TLSv1.0	4	366	4	13
TLSv1.1	3	336	1	13
TLSv1.2	3	328	1	13
Exp2	Anon	RSA	RSAEXPORT	ECDSA
SSLv2	0	1	0	0
SSLv3	0	10	0	0
TLSv1.0	0	46	1	0
TLSv1.1	0	35	0	0
TLSv1.2	0	31	0	0

In both experiment 1 and experiment 2, almost all servers are found using RSA as the authentication mechanism and very few supporting both RSA and ECDSA. (Table 3) This clearly shows that servers possess only 1 RSA certificate (the result of which is shown in Sect. 4.5).

4.3 Symmetric Encryption and Its Key Size

Symmetric Encryption is the heart of security. After the key has been exchanged, all the application data is secured with symmetric encryption algorithm.

Current security definition is to use 128 bits for the key. I divided all the cipher suites in 3 categories -

1. symmetric encryption algorithms with key size less than 128 bits: This include DES64CBC and DESCBC, both of which has key size less than 64 bits. This is insecure. Powerful dedicated machines can be made to break these algorithms.
2. symmetric encryption algorithms with key size greater than equal to 128 bits: This include AES128CBC, AES256CBC, DES192EDE3CBC, 3DESEDECBC, Idea, Camellia 128, Camellia 256, RC4128, RC2128CBC and SEEDCBC.
3. symmetric encryption algorithm with key size less than 40 bits: This include export algorithms such as RC440, RC2CBC40, DES40CBC. All of these have key size of 40 bits and can be easily broken. These algorithms are obsolete and should be removed (Table 4).

In both experiments, few server were found to support symmetric encryption algorithms (key size less than 128 bits) and even export algorithms (key size 40 bits). This is not good in case of experiment 2 where confidential banking information can be decrypted.

Table 4. Symmetric encryption

Exp1	Key size greater than 128	Key size less than 128	Export algorithms 40 key size
SSLv2	5	1	1
SSLv3	47	3	2
TLSv1.0	370	28	4
TLSv1.1	342	23	1
TLSv1.2	334	23	1
Exp2	Key size greater than 128	Key size less than 128	Export algorithms 40 key size
SSLv2	1	0	0
SSLv3	10	1	0
TLSv1.0	40	3	1
TLSv1.1	34	1	0
TLSv1.2	31	1	0

4.4 Hashing Algorithm for MAC or PRF

There is Pseudo Random Function (PRF) used in TLS for calculation of *mastersecret* and key. This PRF requires a hash function. In earlier protocol version this hash function is version dependent. From TLS1.2, the hash algorithm is specified in cipher suite.

Despite so many attacks on MD5, which has been broken, so many servers are found to use MD5. SHA1 has also been attacked but is comparatively more secure than MD5. All versions except TLSv1.2 support only MD5 and SHA hash algorithm for MAC or PRF. Because of the complexity of data being hashed here (contrary to what used in digital signature) its very difficult to come up with a good attack in real life scenario (Table 5).

Table 5. Hashing algorithm

Exp1	MD5	SHA	SHA256	SHA384	Exp2	MD5	SHA	SHA256	SHA384
SSLv2	5	0	0	0	SSLv2	1	0	0	0
SSLv3	30	50	0	0	SSLv3	9	10	0	0
TLSv1.0	76	374	0	0	TLSv1.0	26	46	0	0
TLSv1.1	63	344	0	0	TLSv1.1	18	35	0	0
TLSv1.2	60	336	321	310	TLSv1.2	11	31	18	15

4.5 Number of Certificates and Key Size

Only few servers were found to possess two certificates (RSA and EC). Possessing two certificates is a benefit because the server can switch to EC certificate for clients who support EC. In experiment 1, 49 servers were found without certificates, i.e. they don't support authentication or SSL/TLS connection.

Table 6. Number of certificates possessed by servers

	Exp1	Exp2
0	49	1
1	375	45
2	5	2
3	1	0

Table 7. Certificate type

	Exp1	Exp2
EC256	13	0
RSA1024	2	0
RSA2048	367	46
RSA4096	6	1

Almost all servers possessed RSA2048 certificate. This provides 112 bits of security. Few had EC256 support. EC support is supposed to increase in future due to faster performance offered by EC certificate (Tables 6 and 7).

4.6 Certificate Chain Length and Validity

The server must support valid certificate chain. Generally, there are at least two certificates in the certificate chain. Many certificates in the chain reduces security since there are high chances of security breach if any one of the CA is compromised. There is a browser warning if an invalid certificate chain is reported. The problem is even worsened because some browsers can reconstruct certificate chain and some can't. Its also common to see certificate delivered in incorrect order. Although, browsers tend to correct it.

Most of the servers were found to posses 2 or 3 certificate chain length. And, if there is a certificate it was found to be valid (Tables 8 and 9).

4.7 Signature Algorithm

Issuing CA issues a signature in the certificate for users to verify the public key of the server. The data is signed by CA after its being hashed. The problem comes with the hashing function used. MD5 hashing algorithm is totally insecure. SHA1 hashing algorithm despite of many attacks is secured when used with HMAC. Still, CAs are currently moving to SHA256 as their signature algorithm to be on more secured side. Results shows that most of the servers support SHA256 (Table 10).

Table 8. Certificate chain length

	Exp1	Exp2
1	8	0
2	165	20
3	179	15
4	36	11
5	0	1

Table 9. Certificate validity

	Exp1	Exp2
Valid	388	47
Invalid	0	0

Table 10. Signature Hash Algorithm

	Exp1	Exp2
SHA1	109	18
SHA256	314	45
SHA384	76	0

4.8 Secure Renegotiation

TLS [RFC5246] [13] allows either the client or the server to initiate renegotiation – a new handshake that establishes new cryptographic parameters. Unfortunately, although the new handshake is carried out using the cryptographic parameters established by the original handshake, there is no cryptographic binding between the two. This creates the opportunity for an attack in which the attacker who can intercept a client’s transport layer connection can inject traffic of his own as a prefix to the client’s interaction with the server. TLS [RFC5746] [14] gives ways to provide defense against this attack.

Secure Renegotiation was found by 309 servers in experiment 1 and 43 servers in experiment 2.

5 Conclusion

A good scenario of the security conditions in India was presented. This work can be extended in analysing any number of web servers. Given the importance of SSL/TLS in protecting sensitive information in the Internet, the need to understand security measures is very important today. Since, the security field is changing rapidly, there is insufficient information of the current scenario. With the help of this assessment the researchers will be able to know information about different cryptographic primitives popularly used in the servers. The researchers can target those primitives since there are lot of primitives in cryptography. The results generated here can be used by Security Administrators of web servers to harden their system. The security measurement of servers can be extended further to find particular attacks present in the web servers. The work can also be extended to include client side attacks present in the current scenario.

References

1. Wikipedia, Transport Layer Security. https://en.wikipedia.org/wiki/Transport_Layer_Security
2. Ristic, I.: Bulletproof SSL and TLS: Understanding and Deploying SSL/TLS and PKI to Secure Servers and Web Applications
3. Wikipedia, CRIME. <https://en.wikipedia.org/wiki/CRIME>
4. Wikipedia, Public Key Infrastructure. https://en.wikipedia.org/wiki/Public_key_infrastructure

5. DROWN. <https://drownattack.com/>
6. POODLE. <https://blog.qualys.com/ssllabs/2014/10/15/ssl-3-is-dead-killed-by-the-poodle-attack>
7. Wagner, D., Schneier, B.: Analysis of the SSL 3.0 Protocol
8. Rescorla, E.: SSL and TLS: Designing and Building Secure Systems. Addison-Wesley, Boston (2001)
9. BEAST. <https://blog.qualys.com/ssllabs/2013/09/10/is-beast-still-a-threat>
10. Bhargavan, K., Delignat-Lavaud, A., Fournet, C., Pironti, A., Strub, P.-Y.: Triple Handshakes and Cookie Cutters: Breaking and Fixing Authentication over TLS
11. LOGJAM. <https://weakdh.org/>
12. TLS Working Group Draft. <https://tools.ietf.org/html/draft-ietf-tls-negotiated-ff-dhe-10>
13. The Transport Layer Security (TLS) Protocol Version 1.2 - RFC 5246. <https://tools.ietf.org/html/rfc5246>
14. Transport Layer Security (TLS) Renegotiation Indication Extension - RFC 5746. <https://tools.ietf.org/html/rfc5746>
15. Alexa top 500. <http://www.alexa.com/topsites/countries/IN>
16. Directories of Banks in India. <http://www.banknetindia.com/banklinks.htm>
17. Pornin, T.: TestSSLServer. pornin@bolet.org, <http://www.bolet.org/TestSSL-Server/>
18. Davies, J.: Implementing SSL/TLS Using Cryptography and PKI
19. Lee, H.K., Malkin, T., Nahum, E.: Cryptographic Strength of SSL/TLS Servers: Current and Recent Practices
20. Boneh, D.: Coursera, Cryptography I
21. Katz, J.: Coursera, Cryptography
22. Schneier, B.: Applied Cryptography
23. Buchmann, J.A., Karatsiolis, E., Wiesmaier, A.: Introduction to Public Key Infrastructures
24. JSON Parser - cJSON. <https://github.com/DaveGamble/cJSON>
25. Source Code for Parsing JSON. <https://github.com/prakhs123/Parsing-TestSSLServer-Jsons>
26. Prohibiting Secure Sockets Layer (SSL) Version 2.0 - RFC 6176. <https://tools.ietf.org/html/rfc6176>
27. Deprecating Secure Sockets Layer Version 3.0 - RFC 7568. <https://tools.ietf.org/html/rfc7568>

Key Identifications Using Hebbian Learning

Bhavya Ishaan Murmu^(✉), Anu Kumari, Manu Malkani, and Sanjeet Kumar

Department of ECE, BIT Mesra, Ranchi, India

ishaanbhavya@gmail.com, kumarianu1729@gmail.com,
manumalkani@gmail.com, sanjeet@bitmesra.ac.in

Abstract. The increasing threat to data over public channels has brought about a need to secure sensitive data to avoid its misuse and tampering. This makes data security an important issue in the present which will continue to pose a problem in the future.

Over the successive years a lot of work has been done in this field, with the recent developments focusing on neural networks and its application in security. Neural network algorithms for the same like Multilayer Perceptron technique, Back Propagation Technique have been implemented. Multilayer Perceptron technique aforesaid model lacks in accuracy and is complex whereas the Back Propagation Technique is more accurate but its complexity is more than Multilayer Perceptron and also has result convergence faults.

This paper deals with data encryption using a random set of keys and the key identification using Hebbian learning. Here bits of data are taken at a time and encrypted using a key from given set of keys. The key is identified using Hebbian learning and hence data is decrypted. Main advantage of this method is its simplicity and that it is error free in lossless transmission.

Keywords: Neural network · Hebbian learning · Key identification · Hebbian learning for key identification

1 Introduction

The two topics, data security and neural networks in today's world offer huge scope of research and development. While the former is a well explored topic since start of communication and the latter one is a less worked upon area with tremendous possibilities in many directions. Neural Network algorithms focus on achieving a target based on trained neural network with desired target. Some of the works in this field include Back Propagation technique [1, 2] and Multilayer Propagation Technique [8] which are used in evaluation models for various threats to the system. Multilayer Perceptron technique is based on multiple layers of feed forward neurons desired to achieve the target. As the numbers of layers increase, the accuracy increases but so does the complexity. However, the foresaid model lacks in accuracy and is complex. The Back Propagation Technique is a Multilayer Perceptron with feedback links to achieve the target. It is more accurate but has higher complexity and result convergence faults. The above methods are used for network security analysis or detection of cyber-attacks. Other works of

neural network in network security include key generation for encryption or data encryption using genetic algorithm [4].

This paper brings together the application of neural network with data security using Hebbian Learning Technique for key identification. Hebbian learning [3] is the simplest and earliest discovered technique of neural network and can be used for many applications. The data to be encrypted is divided into blocks whose size depends on a randomly generated integer. For each block, keys are generated and a key is selected at random. Data is trained by Hebbian network to achieve a random target which gives us a value in approximation of target. This value is a unique number varies over a range of key and target combination. At receiver this number is used to identify the key and data is decrypted. The model is not complex due to single step iterations and hence the processing time is very less. It is error free in lossless transmission.

The advanced model works on variable bit encryption and hence the security is increased.

2 Network Security Model Using Hebbian Learning Technique

2.1 Proposed Model (Basic)

Hebbian learning is an algorithm used in neural networks to adjust the synaptic weights which helps in achieving the desired output. In the following model this algorithm has been used for identifying the correct key at the receiving side which was used to encrypt the data at the transmitting side.

In this model shown in Fig. 1 there is a fixed target matrix which consists of a set of randomly selected number, which is predefined at both the transmitting and receiving end. This prevents the need of transmission of the target value over the channel. The unencrypted data here is divided in blocks of six bits each. A number is randomly selected from the target matrix and this number along with the block of six bits is sent as an input to the Hebbian algorithm. The bias provided to the algorithm is the function of the keys and the target value. This gives a unique numeral key which is a value close to the target value but is never equal to the target itself. This unique number is different for every target value and key combination. This procedure is carried out so that we never directly have to transmit the desired target over the channel, hence improving the security of the network.

The value that are transmitted to the receiving side are the index of the target value which is randomly selected from the target matrix, the unique numeral k generated by the Hebbian algorithm, data encrypted with one of the keys randomly selected out of the three keys and the three keys encrypted amongst themselves using some predefined criteria.

At the receiving side the target index is utilized to retrieve back the target value used at the transmitting side. This target value is given as an input the Hebbian algorithm along with the encrypted data received, decrypted with each key one key at a time. This gives an output k' for decryption with each key. This value is hence compared with k received. If the values match, the key is identified and retrieved.

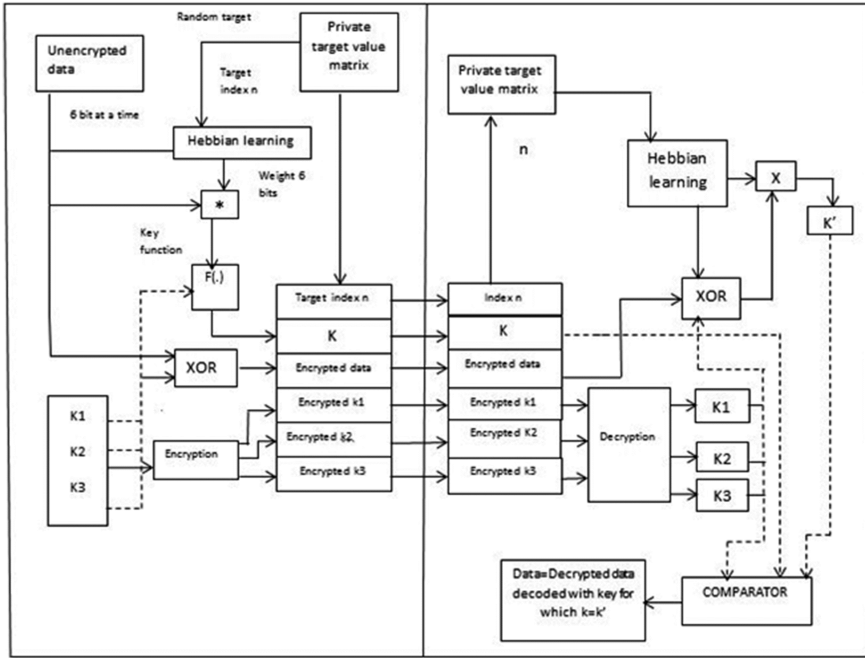


Fig. 1. Basic model using fixed bit encryption

Since the model implies the method of key identification using unique numeral obtained after Hebbian training, the integrity and traceability of the unique numeral plays an important role. But since this number is any real random integer depending on target and data both enclosed from the public, to the channel it is thus only a number which can be formed by any mathematical calculation.

Also since the model has simple and converging calculations, due to this, the accuracy is greatly reduced and hence unique number varies dynamically in a wide range around target making it more obscure.

In the Hebbian Learning the weight and bias are the most important parts of technique which are not sent to the user.

2.2 Proposed Model with Variable Bit Encryption (Advanced)

This model as shown in Fig. 2 is improved version of previous model by using randomly generated keys rather than a fix set of three keys.

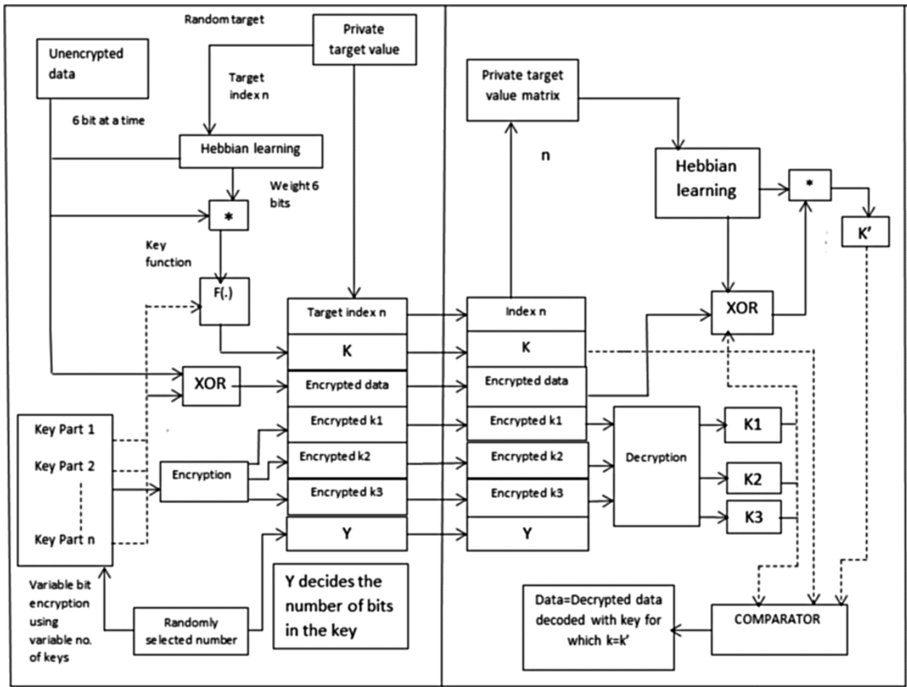


Fig. 2. Improved model using variable bit encryption in which keys are randomly generated for each block

In previous model there are two types of possible threats on the key. Firstly, a fixed set of three keys are used for encryption of every block of data so there is a chance that

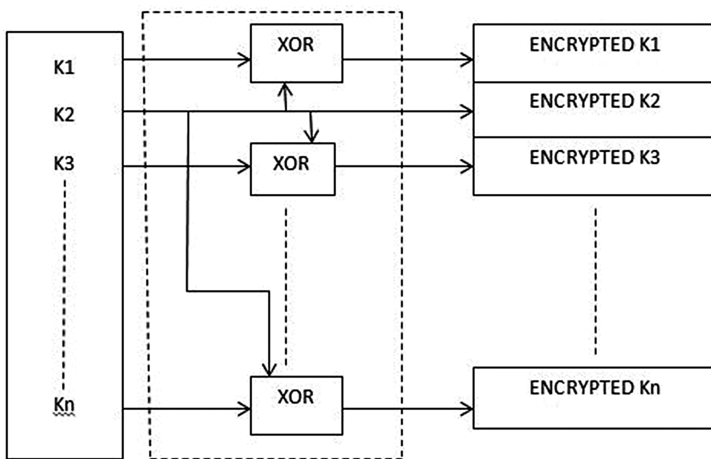


Fig. 3. Keys blocks encryption

the intruder might retrieve the keys by eavesdropping for a while. Also the keys and the block are of a fixed size so the hacker. To overcome this problem, we could use another method in which we randomly select a number between 6 to 10 and then we divide our data in a frame of size corresponding to that might also use brute force method to try and decode the right set of key number. Also we randomly create a set of keys whose amount is again same as the aforesaid number. Here as we use random bit encryption and the keys are also randomly generated each time, hence it will be impossible for the eavesdropper to guess each time the size of the current frame or the key as it will be continuously varying. The only drawback of this model is for each frame a large set of keys will be transmitted.

To overcome this drawback another model that has been implemented in this paper is having three keys of 128 bits, 256 bits and 384 bits. Though the key sizes are fixed but one of the key sizes will be randomly selected each time and the data will be encrypted with it. Also the data frames will be of different sizes each time depending on the key. This model gives a high security level to the data without actually increasing the data transmitted considerably.

2.3 Key Combination and Encryption in Advanced Model

In the advanced model, the keys are generated in blocks of 128 bits. Since it uses variable bit encryption, the numbers of encryptions possible are 256 bits, 384 bits and 512 bits. Hence for 256 bits, 2 blocks of 128 bits are generated randomly and then sent along the channel by xoring with the second block. These two blocks from two possible combinations and hence two keys can be formed. For 384 bits, if we consider blocks of 128 bits,

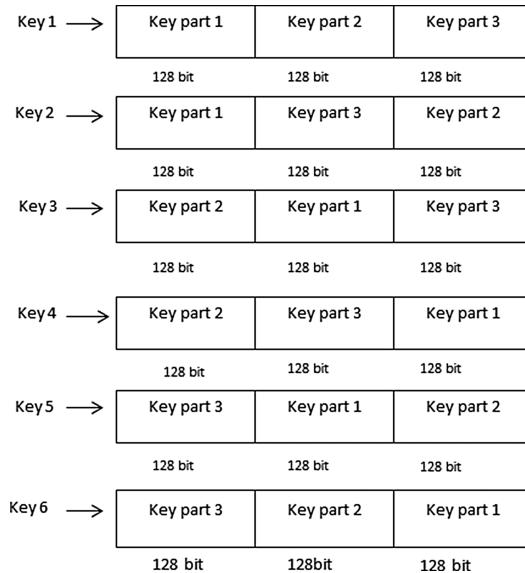


Fig. 4. Keys formed by combination

as shown in Fig. 4 combinations, hence 6 keys are formed. Hence, though only 384 bits are sent along the line, the numbers of keys that are available for use are 6 thus making the data size transmittable less but security even higher than the original model. All the key parts encrypted and number of keys used are denoted by another number y denoting the size of the key used or data encrypted at a time (Fig. 3).

3 Data Format and Calculation

3.1 Basic Model

The data after encryption (TX) consists of encrypted key 1, encrypted data, encrypted key 2 and encrypted key 3, encrypted data consists of each blocks of data separately encrypted where for each block of data, the unique number 'k', target index which decides the target 'n' and encrypted data is sent along the channel (Fig. 5).

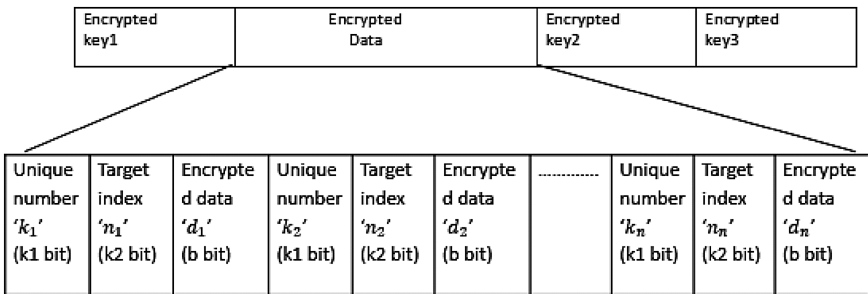


Fig. 5. Data format (advanced model)

Hence, the total transmission data (TX) size after block wise encryption = Encrypted key1 + encrypted data (unique number + target index + encrypted block of data) + encrypted key2 + encrypted key3.

The size of key and block of data is the same since here we implement a simple XOR for encryption. If the data to be encrypted is of size total_data, then

Number of blocks of data = blocks = Total_data/no. of bits to be encrypted at a time(b)

Hence, TX = enc_bit + (k1 + K2 + b) *blocks

Calculation:

If we divide and encrypt 128 bits at a time:

Blocks = total_data/128

TX = 128 + 128 + x+128 bits;

X = (k1 + k2 + 128) * blocks

If we divide and encrypt 512 bits at a time:

Blocks = total_data/512

TX = 512 + 512 + 512 bits;

X = (k1 + k2 + 512) * blocks

In advanced model, as the data format (Fig. 6) shows, for each block of data that is encrypted, keys are generated randomly and then sent along the line as explained before. Hence the total data size increases but so does the security too. For each variable block of data, a target index is sent along with number of keys or the size of key and hence data that is to be decrypted is sent. Then the key parts depending on number of data to be encrypted are sent in encrypted form with encrypted data and unique number at last. The number of keys is if it 00 then it means 256-bit encryption and if it 01 the 384 and 10 denotes 512-bit encryption.

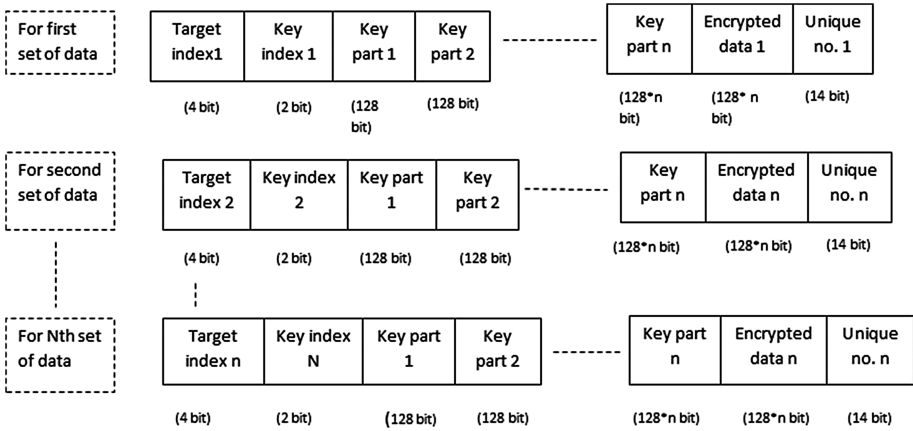


Fig. 6. Data format (advanced model)

Calculation:

Total size of transmitted data TX,

$$TX = \sum_{i=1}^N t_i + Y_i + 128 * n + encrypted\ data + 14$$

Where, t_i = target index for i^{th} data block

N = number of times data is encrypted

Y = denotes 'n' which denoted size of data to be Encrypted

Encrypted data size = $128*n$ (same as key)

4 Output

4.1 Basic Model (Fixed Bit Encryption)

The output table is calculated for 128 (Table 1) and 512 (Table 2) bits of data encryption at a time. For range of targets are taken from -5 to +5 and we consider 4 targets hence the index of the targets can be specified using only 2 bits and for range we take the unique number to be of 10 bits while working with 128 bits of data and 14 bits while working with.

Table 1. (For 128 bits):

Data to be encrypted (lengths in bits)	No. of keys used = No. of targets used	Encrypted data (total TX data length in bits)	Increased data size after encryption(roughly)	Error
5120	40	6064	1.184 times	0 bits
51200	400	57184	1.117 times	0 bits
512000	4000	568384	1.110 times	0 bits
5120000	40000	5680384	1.109 times	0 bits
51200000	400000	56800384	1.109 times	0 bits
512 Mb	4000000	568000384	1.109 times	0 bits

Table 2. (For 512 bits):

Data to be encrypted (lengths in bits)	No. of keys used = No. of targets used	Encrypted data (total TX data length in bits)	Increased data size after encryption(roughly)	Error
5120	10	6816	1.331 times	0 bits
51200	100	54336	1.061 times	0 bits
512000	1000	529536	1.034 times	0 bits
5120000	10000	5281536	1.031 times	0 bits
51200000	100000	52801536	1.031 times	0 bits
512 Mb	1000000	528001536	1.031 times	0 bits

It is observed that relative data size is less for 512 bits of data than 128 bits of data.

As the no. of data bits encrypted per algorithm process increases, relative encrypted data size decreases. However, it needs to be noted that the result is when there is no error in any of the bits. For large amount of data, the encrypted data size should be as less as possible.

4.2 Advanced Model (Variable Bit Encryption)

As we can see from the above output table (Table 3) that for this model the data sent over the channel after encryption is increased negligibly, although the security increases tremendously. Here the security is increased due to the fact that the keys are randomly generated in blocks of 128 bits each time the data is to be encrypted and also these blocks are shuffled amongst themselves to achieve different possible combinations of keys. The block and key size is also varied amidst 128,256 and 384 bits for every data frame, so it will be impossible for the intruder to guess each time the frame size and key correctly.

Table 3. :

Data to be encrypted (lengths in bits)	No. of keys used = No. of targets used	Encrypted data (total TX data length in bits)	Increased data size after encryption(roughly)	Error
5120	13	9998	1.952 times	0 bits
51200	181	105518	2.060 times	0 bits
512000	1699	1057478	2.065 times	0 bits
5120000	17167	10583606	2.067 times	0 bits
51200000	171428	105828058	2.067 times	0 bits
512 Mb	1713928	1028001536	2.007 times	0 bits

5 Conclusion

The method used in basic model is new and not yet fully explored. The disadvantage is that though key encoding using XOR provides a simple way of securing data over a network, but if not used wisely enough the data can still be decrypted by a hacker because of the key being sent over the insecure network to the output side. Hence in terms of security the model is at a loss but its uniqueness can be exploited with other integral techniques and can be used in more complex way to utilize this method to form a concrete secure model. This paper proposes this method and the future work could include using encryption without evident XOR technique or increasing the keys with increasing number of encryption bits to include more probabilities of combination making hacking difficult.

For advance model using variable bit encryption, clearly the security increases. In fixed bit encryption, a pattern may be formed but in variable bit due to randomness it is difficult to know from a series of binary digits that how many digits correspond to the target index or the unique number, the key, the arrangement of the key and decryption of key and size of data encrypted, all the factors remain unknown and hence the protection increases to a great extent. Hence the obscurity and integrity of pattern increases.

References

1. Luo, B., Liu, Y.: The risk evaluation model of network information security based on improved BP neural network. IEEE (2012)
2. Fu, J., Huang, L., Yao, Y.: Application of BP neural network in wireless network security evaluation. IEEE (2010)
3. Haykin, S.: Neural networks a comprehensive foundation. MacMillan Publishing Company, New York (1994)
4. El-Sharkawi, M.A., Huang, S.J.: Development of genetic algorithm embedded Kohonen neural network for dynamic security assessment. IEEE (1996)
5. Risk modelling (2010). http://www.owasp.org/index.php/Threat_Risk_Modeling. Accessed 24 Sep 2014
6. Dengguo, F., Yang, Z., Yuqing, Z.: Risk assessment of information security survey. J. China Inst. Commun. **25**(7), 10–18 (2004)

7. Liming, Z.: Artificial Neural Network Model and Its Application. Fudan University Press, Shanghai (1994)
8. Salek, Z., Madani, F.M., Azmi, R.: Intrusion detection using neural networks trained by differential evaluation algorithm

Security and Privacy in Networked Systems

An Automated Methodology for Secured User Allocation in Cloud

Srijita Basu^(✉), Anirban Sengupta, and Chandan Mazumdar

Jadavpur University, Kolkata, India

srijita.basu202@gmail.com, anirban.sg@gmail.com,
chandan.mazumdar@gmail.com

Abstract. Use of cloud infrastructure by enterprises for hosting their data and applications is growing at a rapid pace. This calls for proper security measures for protecting sensitive enterprise data within the Cloud. Absence of owners' control over data stored in Cloud introduces major security concerns that should be handled by the Cloud Service Provider. One of the most critical aspects is secure handling of data deletion; it should be ensured that sensitive data is completely deleted from the Cloud on an owner's request to avoid data leakages. This paper presents a methodology for effective and efficient allocation of Virtual Machines to users, while ensuring security of sensitive data that belongs to users within the same Conflict of Interest classes. The issue of secured data deletion has been addressed by the proposed methodology which, being a provider-end solution, does not incur any additional user overhead.

Keywords: Cloud security · Conflict of interest · Secured data deletion · Virtual machine reservation

1 Introduction

Cloud can be visualized as a distributed system comprising of a set of virtual machines that can be dynamically provisioned to meet the varying resource requirements of a consumer [1]. The organization/entity that maintains a public cloud is referred to as the Cloud Service Provider (CSP). Cloud relieves an enterprise of the overhead of physical installation and maintenance of its system, which automatically reduces the overall operational cost and enhances system efficiency. The modalities of CSP-Consumer relationship are governed by the terms and conditions defined within a Service Level Agreement (SLA). An underlying concern lies in the fact that the consumer has to rely completely on the CSP for the maintenance of privacy and security of sensitive data and services. The notion of mutual trust is achieved to some extent by negotiating the SLA, but still a good number of cloud-specific security issues become inevitable that need to be handled by either the CSP or the consumer [2].

Maintenance of data security in Cloud poses serious challenges owing to its distributed nature and multi-tenant architecture [3]. The data life cycle comprises of several phases, namely data generation, data storage, data usage, data distribution and data deletion. CSP should support all these phases with appropriate security mechanisms [4]. Most importantly, the process of data deletion is crucial in Cloud; this should

be handled carefully by the CSP to ensure permanent and complete deletion of data on a consumer's request. Moreover, the data backups (scope, saving intervals, saving times, storage duration, etc.) should be transparent and auditable for the consumers. Lack of secure deletion strategies may lead to leakage of sensitive data to unauthorized entities.

Traditional security models fail to address the requirements that are specific to Cloud systems [5]. Recently developed cloud-centric models are either too intricate, or fall short in properly representing the actual state and operations of Cloud systems [6]. Moreover, these models do not address the issue of secure data deletion, which continues to remain an open problem. This paper attempts to address this research gap by proposing a novel methodology that would enable a CSP to avoid data leakage that may occur due to incomplete removal of consumers' data. The proposed methodology also includes a technique for identifying the Virtual Machines (VMs) that are most suited as well as secured with respect to a consumer's requirements.

Rest of the paper is organized as follows. In Sect. 2, a survey of related work is given. Section 3 describes the System model and design goal. It presents the proposed methodology, including the service units involved and the algorithm for secure data removal. Section 4 illustrates the usefulness of the proposed methodology with the help of a case study. Finally, Sect. 5 concludes the paper.

2 Related Work

Several security models and mechanisms for cloud-based services have been proposed in recent years. Some of the significant contributions are surveyed here.

Wang et al. [7] proposed a Privacy-Preserving Public Auditing scheme meant for assuring data integrity/correctness within Cloud storage. Here, CSP is considered to be an untrusted party which may hide data losses or even free storage by deleting the blocks that are rarely accessed by the consumer. A Third Party Auditor and a public key based homomorphic authenticator have been used to prevent such breaches. However, it lacks dynamic file handling capabilities and involves large number of message transfers.

Yu et al. [8] introduced a fine-grained access control scheme for clouds. Each data file is stored in encrypted form, and is associated with a set of attributes. A logical expression is associated with each user which defines her access structure over the attributes, thus identifying the data files that she is allowed to access. However, complications arise when a user is to be removed from the server, which requires the data owner to re-encrypt all the data files accessible to that user.

Liu, Wang and Wu [9] proposed a time-based proxy re-encryption scheme that allows a user's access right to expire automatically after a pre-determined period of time. Each data is associated with an attribute-based access structure and an access time, and each user is identified by a set of attributes and a set of eligible time periods which denote the period of validity of the user's access right. The main drawback of this scheme is that it does not have provision for fine-grained time accuracy.

Wang, He and Tang [10] introduced a Cloud data integrity checking scheme based on identity-based proxy-oriented technique, which eliminates the tedious job of certificate maintenance required for verification. The verification is done based on the tag-block configuration of the stored files, a pseudo-random function and a pseudo-random permutation generated by the system, and the bilinear pairings.

Besides, some research has been carried out on cloud sensing/monitoring schemes which help in automating the provisioning of cloud services [11].

It is obvious that most of the existing cloud security models are either costly or too complex. Moreover, they are mostly consumer dependent, which is quite infeasible in a practical Cloud computing scenario. It is important to develop a comprehensive solution that could address cloud specific data security problems at the CSP end. This paper presents such a consumer-independent data security scheme.

3 Proposed Methodology

The Cloud model that has been presented here is composed of the CSP and the Cloud user or service owner (consumer) who deploys her service or stores her data in the Cloud system. The main aim is to formulate a detailed procedure that should be followed when a user requests for some VM instance to deploy her service. The goal here is to achieve user data confidentiality as well as proper utilization and balance between the VMs such that the most appropriate VM (based on user requirement, security issues as well as present condition of the Cloud system) is allocated to the user efficiently.

Basu et al. [12] addressed security based on the present content of each VM by applying suitable separation policies. This paper enhances that work by considering traces of deleted data before granting access to a particular VM. Thus, one of the most vital areas of Cloud data security, *data deletion and disposal issues*, have been addressed here. The problem with data deletion, when using a Cloud-based service, lies in the fact that when a request for data or service removal is made by a consumer, the CSP usually deletes only the pointers that point to the particular data that need to be deleted. However, problems might ensue when a user is assigned a VM which previously hosted data of another user with whom she has conflict of interest. A malicious user may try to obtain sensitive data of some other enterprise that may reside there, thus posing a threat to data confidentiality. Figure 1 illustrates this scenario with a sample case.

The present work is a small step towards handling such data deletion and access issues efficiently. An important assumption of the proposed methodology is the existence of mutual trust between the CSP and the Cloud user. The CSP is believed to be honest and the security issues that have been considered here are from the perspective of malicious users. The entire procedure from request initiation by a user to fulfillment of the request by allocating necessary resources (i.e. VM) by the CSP involves various functions and data structures. These are described in the following sub-section.

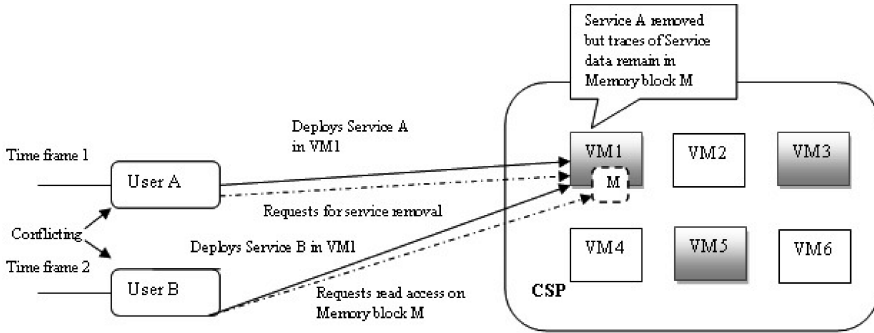


Fig. 1. Cloud with 6 VMs (Grey VMs contain applications/data; white VMs are empty), and 2 users with conflict of interest. Security breach occurs when *UserB* tries to access a particular memory location in VM1.

3.1 Service Units and Data Structures

The proposed methodology comprises of four units, namely **VM Allocation unit**, **Service Automation unit**, **Deletion Analyzer unit** and **Conflict Analysis unit**; they are used for managing VM allocation, VM reservation, complete removal of deleted data and analyses of Conflict of Interest classes [12], respectively.

3.1.1 VM Allocation Unit

This particular module inputs the user requirements (in terms of memory, hardware, storage, etc.) and searches for the best possible set of VMs that are able to meet those requirements. Once such a set is found, an LRU (Least Recently Used) algorithm is used to select a single VM out of this set. Use of LRU algorithm ensures proper and uniform utilization of CSP’s resources and helps avoid cloud sprawl or VM sprawl [13].

3.1.2 Conflict Analysis Unit

The VM selected by the **VM Allocation unit** is input to this module which checks for existing conflicts based on Conflict of Interest classes [12]. This module uses the Chinese wall security policy for conflict analyses as suggested in [12]. After a VM passes this unit, it enters the next module.

3.1.3 Deletion Analyzer Unit

This module uses a binary variable *flag*. When deleting enterprise data from a VM, the value of *flag* is set to “true”. A list *Org-index* is used to store the names of enterprises whose data have been deleted, along with the pointers which were actually removed from the memory. Both *flag* and *Org-index* are maintained on a per-VM basis. The field “status” in *User_details* table is set to “absent” and the timestamp value (explained in the next section) of the corresponding entry of the VM in *User_reservation* queue (if present) is set to zero. If the value of *flag* is “false”, the VM is assigned to the user. However, if it is “true”, the module checks *Org-index* list to identify names of

conflicting enterprises [12]. If no conflicts exist, the VM can be assigned to the user. On the other hand, if conflicts are detected, the deleted pointers, corresponding to the conflicting enterprises, are identified and the corresponding memory blocks are overwritten by some garbage value. After the completion of this process the VM can be safely allocated to the user. This helps to prevent illegal leakages of sensitive data.

3.1.4 Service Automation Unit

In addition to managing secure deletion of data, the proposed methodology also employs efficient means of handling user requests by using a reservation technique. The Service Automation (SA) unit reserves Virtual Machine Images (VMIs) for each user and manages the respective copies of reserved VMIs [14], as follows:

1. When a user request is made for the first time, a suitable VM is assigned to her using the procedure mentioned in the earlier modules. In addition to this, the SA unit constructs a reservation queue for that user containing the presently assigned VMI ID, along with those VMI IDs that have been found to be equally suitable for allocation by the earlier modules.
2. The same VMI may be reserved by the SA unit for more than one user. In such a scenario, any update (i.e. actual allocation of the VM to a particular user) must be reflected in all copies of reservation queues that contain the affected VMI IDs.
3. Periodic checks are conducted by the SA unit in order to detect security conflicts that might arise owing to new allocations. If security conflicts are detected, SA unit associates a status flag with the corresponding VMI ID in the reservation queue and sets its value to "FALSE". Later, if some VM is de-allocated, as a result of which previous conflicts are removed, the same change is reflected in the user reservation queue during periodic updates by setting the status flag to "TRUE".
4. When a user request is encountered for the second time, no further configuration checking or security checking is required, as a suitable VM could be assigned to the user readily from the reservation queue. Thus, though the implementation of the SA unit incurs some provider-overhead, it enables the CSP to serve its users efficiently.
5. It has been assumed here that a user follows the same trend of requests, which may not be the case always. It may so happen that the same user now has different requirements. In such a scenario, the user has the freedom to choose a new VM of suitable configuration from the list of available ones.
6. A problem may arise owing to the periodic update of reservation queues that is executed at specified time intervals. It may be the case that a certain allocation is done before the required update has taken place in the local VMI copy, thus resulting in conflicting enterprises sharing the same VM. Such situations can be handled by using a time-stamp. Whenever a VM is allocated to some user, the change is immediately reflected in the main VMI and a time-stamp is associated with the particular VMI (main copy) showing the time of allocation. Later on, when periodic checks are carried out, the change is propagated to all the local copies (residing in the user queues) corresponding to the particular VMI and a new time-stamp is assigned to the main as well as the local VMI copies. Now, when a user requests a VM, the SA unit can automatically allocate one, after checking the time-stamps associated with the local VMI copy in the reservation queue and that of

the main VMI copy. If both the time-stamps are same, the VM is allocated to the user. However, if the time-stamp of the main copy is found to be greater than that of the local copy, it implies that the local copy of the VMI is not updated. In such a scenario, SA unit first synchronizes all local copies of the VMI with the main copy and then performs the necessary allocation.

Figure 2 shows the detailed workflow needed to implement the proposed data deletion and VM allocation scheme, while the algorithms for implementing the methodology are described in the following sub-section.

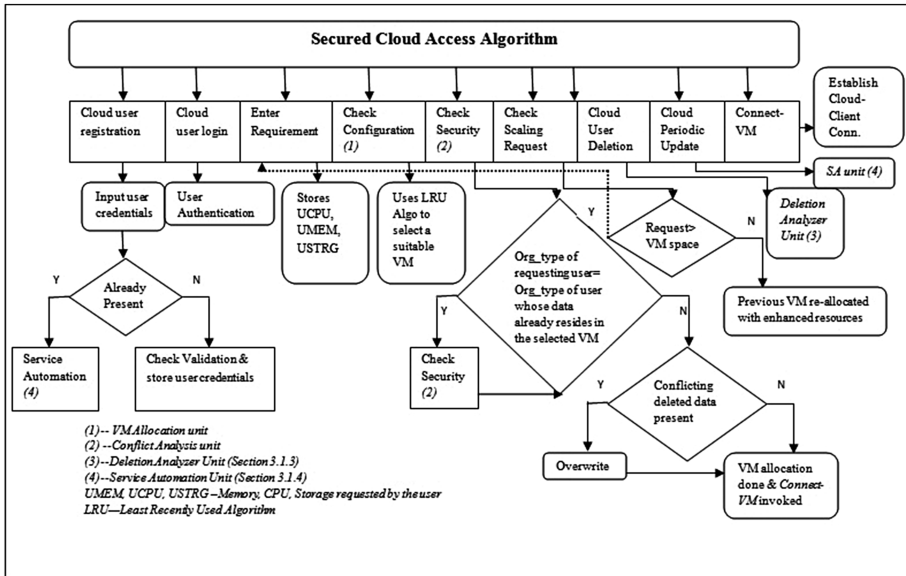


Fig. 2. Flowchart for secured cloud access.

3.2 Algorithm for Secured Cloud Access

Tables 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11 and 12 contain various algorithms that can be used to implement the proposed data deletion and VM allocation scheme.

The *Secured Cloud Access* algorithm (Table 1) has been designed to implement the service units described in the previous sub-section. It comprises of several subroutine calls which are detailed in Tables 2, 3, 4, 5, 6, 7, 8, 9, 10, 11 and 12. The entry point of the algorithm is at *Cloud User Registration* and the exit point occurs at line no. 51 or 54 when *Connect-VM(VM_ID)* either establishes a connection between the CSP and the Cloud user, or prints an Error message, as the case may be.

If the total number of VMs in the system is considered to be ‘n’, then a simple analysis shows that the Worst case Time Complexity of the proposed Algorithm is $O(n)$.

Table 1. Secured cloud access algorithm

<ol style="list-style-type: none"> 1. Name, Mail-id, Org_name, Org_type, password, Re-type password are fields containing the values of the Name, Email address, Organization Name, Organization Type, password (authenticating element), password repetition, respectively of the user who wants to use the Cloud service. Status is an additional field holding values "present" or "absent" signifying that the user is a present Cloud user or a deleted user. 2. UMEM,UCPU,USTRG are fields storing the user memory, CPU, and storage requirements that she wants to avail of the CSP 3. VCPU, VMEM, VSTRG are entities storing the memory, CPU, storage associated with each Virtual machine of the CSP. 4. SCPU, SMEM, SSTRG are the entities storing the memory, CPU and storage that the user wants to scale up. 5. VM_ID is the identifier/variable used to represent each virtual machine of the CSP. 6. VM_list[] is an array storing a list of VMs and integer variables i, j are initialized to 0 which acts as the counter for the lists used. 7. Name[] and Name''[] are string arrays for storing names of Cloud users. 8. User_details table is a component of the CSP database storing the values of Name, Email address, Organization Name, Organization Type, Status and password of a Cloud user. Primary key of the table is (Name, mail_id). 9. User_requirement table stores the values of Name, Email address, User memory, CPU, and storage requirement. Primary key of the table is (Name, mail_id). 10. VM_allocation table stores the values of Name, Email address, Time_stamp and VM identifier i.e. the VM_ID of the Virtual Machine that has been allocated to the particular user. Primary key of the table is (Name, mail_id) (It has been assumed that a user cannot be allocated more than one VM simultaneously). 11. VM_details table stores the values of VM_ID, Virtual Machine memory, CPU, storage. Primary key of the table is (VM_ID). 12. A variable flag_VMIDn stores either "true" or "false" for each VM. 13. Org-index is a structure storing the list of deleted user names and the list of particular pointers that have been deleted from the VM memory. 14. User_reservation queue stores list of reserved VMs (as VM_ID), associated Time stamps (TS) and a status flag for each VM_ID with an initial default value "True" for each user. "User_reservation queue.name" represents the name or identifier of each queue i.e. the user name to which the queue belongs. 15. If Cloud User Registration () = true 16. Var1 = Cloud User Login (); 17. Else 18. Terminate1=1; 19. End If 20. If Var1 = true 21. If(VM not allocated already) 22. Var3 = Enter Requirement(Name, Mail-id); 23. Else if(VM already allocated) 24. Terminate2=1; 25. Else if(User wants On-demand scaling of resources) 26. Var4 = Check Scaling(Name, Mail-id); 27. End If 28. Else 29. Go to Step 15; 30. End If 31. If Var3 = true 32. If Check Configuration(Name, Mail-id) = true 33. Var5= Check Security(Name, VM_ID, VM_ID); 34. End If 35. End If 36. If Var4 = 0 37. Var3 = Enter Requirement(Name, Mail-id); 38. Go to step 31; 39. Else if Var4 = 1 40. Terminate3=1; 41. End If 42. If Var5 = true 43. Terminate4=1; 44. End If 45. Call Cloud User Deletion(Name, Mail-id); 46. Call Cloud Periodic Update(User_reservation queue U); 47. If (Terminate1=1 or Terminate2=1 or Terminate3=1 or Terminate4=1) 48. Var2 = Connect-VM(VM_ID); 49. End if 50. If Var2 = true 51. End 52. Else 53. Print "Error in Connection" 54. End 55. End if

Table 2. Cloud user registration subroutine**Subroutine: Cloud User Registration ()**

1. *Input* Name, Mail-id, Organization Name, Organization Type, password, re-type password;
2. If *Check Validation()* = true and Name != "Name" and Mail-id != "Mail-id" in User_details table
3. *Store* Name, Mail-id, Organization Name, Organization Type, "present", password in User_details table;
4. Return True;
5. Else if (Status = "absent" && Name="Name") in User_details_table
6. Call *Service Automation*(Name, Mail-id)
7. Return False;
8. End If

Table 3. Cloud user login subroutine**Subroutine: Cloud User Login ()**

1. *Enter* Mail-id and password;
2. *Retrieve* stored values of Mail-id and password from User_details and *store* them as Mail-id' and password' ;
3. If (Mail-id' = Mail-id and password = password')
4. Return true
5. End if

Table 4. Enter requirement subroutine**Subroutine: Enter Requirement (Name, Mail-id)**

1. *Input* UCPU, UMEM, USTRG;
2. *Store* Mail-id, Name, UCPU, UMEM, USTRG in User_requirement table;
3. Return true;

Table 5. Check configuration subroutine**Subroutine: Check Configuration (Name, Mail-id)**

1. If (Name= "Name" and Mail-id= "Mail-id") in User_requirement table
2. *Retrieve* UCPU, UMEM, USTRG from User_requirement table
3. End If
4. If (VCPU >= UCPU and VMEM >= UMEM and VSTRG >= USTRG) in VM_details table
5. *Retrieve* VM_IDs from VM_details table and *store* in VMlist[]
6. End If
7. *Read* VM_ID1;
8. VM_ID1= *LRU* (VMlist[]);
9. *Delete* VM_ID1 from VMlist[]
10. Return true

Table 6. Check security subroutine

Subroutine: Check Security (Name, VM_ID1, VMlist[])

1. Read String [] Name', Name'';
2. If (Name= "Name" and Mail-id = "Mail-id") in User_details table
3. Retrieve Organization Type from User_details table and store in Org_type;
4. End If
5. If (Organization Type= "Org_type") in User_details table
6. Retrieve Name from User_details table and store in Name';
7. End If
8. If(VM_ID= VM_ID1)
9. Retrieve Name from VM_Allocation table and store in Name''
10. End If
11. If (Name' = Name'')
12. Message display "ALLOCATION FAILURE";
13. Secured= False;
14. VM_ID2= random (VMlist[]);
15. Check Security(Name',VM_ID2, VMlist[]);
16. Else if (flag_VM_ID1= true)
17. If (Name = "Org_index.name") in User_details table
18. Retrieve Organization Type from User_details table and store in Org_type ';
19. End If
20. If (Organization Type= "Org_type" && Name= "Name")
21. Retrieve Org_index.pointer
22. Overwrite memory locations pointed by Org_index.pointer
23. Secured = True;
24. Store Name, Mail-id, VM_ID2, TS in VM_allocation table
25. End If
26. Else if (Secured = True)
27. Store Name, Mail-id, VM_ID2, TS in VM_allocation table
28. End If
29. while (i < VMlist[].Length and Secured = True)
30. If(VM_ID = VMlist[i])
31. Retrieve Name from VM_allocation table and store in Name''
32. End If
33. If (Name'!= Name'')
34. Store VMlist[i] in User_reservation queue of this user
35. Store User_reservation.TS = current system time
36. End if
37. Increment i as i+1
38. End while
39. Return true;

Table 7. Connect-VM subroutine

Subroutine: Connect-VM (Name, VM_ID)

1. If (Name = "Name") in User_requirement table
2. Retrieve UCPU, UMEM, USTRG from User_requirement table ;
3. End If
4. If(VM_ID= "VM_ID") in VM_details table
5. Update VM_details table as VCPU= VCPU-UCPU, VMEM=VMEM-UMEM,VSTRG- USTRG ;
6. End If
7. Establish connection between the Client and the Cloud server and Return True;

Table 8. Check scaling request subroutine

Subroutine: *Check Scaling Request* (Name, Mail-id)

1. *Input* SMEM, SCPU, SSTRG;
2. *Read* VM_ID’;
3. If (Name = “Name”) in VM_allocation table
4. *Retrieve* VM_ID from VM_allocation table and *store* in VM_ID’;
5. End If
6. If (VM_ID = “VM_ID”) in VM_details table
7. *Retrieve* VCPU, VMEM, VSTRG from VM_details table;
8. End If
9. If (VCPU >= SCPU and VMEM >= SMEM and VSTRG >= SSTRG)
10. *Update* User_requirement table as UMEM=UMEM+SMEM and UCPU=UCPU+SCPU and USTRG=USTRG+SSTRG;
11. Return 1;
12. Else
13. If (Name= “Name”)
14. *Retrieve* UCPU, UMEM, USTRG from User_requirement table;
15. End If
16. If (VM_ID== “VM_ID”)
17. Call **Cloud User Deletion** (Name, Mail-id)
18. End if
19. Return 0;
20. End If

Table 9. Cloud user deletion subroutine

Subroutine: *Cloud User Deletion* (Name, Mail-id)

1. If (Name = “Name”) in User_requirement table
2. *Retrieve* UCPU, UMEM, USTRG from User_requirement table ;
3. End If
4. If (Name = “Name”)
5. *Retrieve* VM_ID from VM_allocation table
6. End If
7. If (VM_ID== “VM_ID”)
8. *Update* VM_details table as VCPU = VCPU+UCPU and VMEM=VMEM+UMEM and VSTRG+ USTRG
9. End if
10. If (Name== “Name”)
11. *Delete* entry from VM_allocation
12. End if
13. flag_VM_ID= true
14. Org_index.name= Name
15. Org_index.pointer= pointers to be deleted /*pointer addresses holding the data of the deleted user*/
16. *Delete* the necessary pointers
17. If (Name== “Name”)
18. *Update* User_details table as Status=”absent”
19. End if
20. If (VM_ID=User_reservation.VM_ID)
21. User_reservation.TS=0
22. End if

Table 10. Service automation subroutine**Subroutine: Service Automation (Name, Mail-id)**

1. Search User_reservation.VM_ID of user "Name"
2. Retrieve VM_ID from VM_Allocation table
3. If("User_reservation.VM_ID"!=VM_IDand"User_reservation.TS"!=0andUser_reservation.status==True)
4. Store Name, Mail-id, VM_ID, TS in VM_allocation table
5. Else if ("User_reservation.VM_ID"=VM_ID and (User_reservation.status = False or "User_reservation.TS"=0) and end of queue not reached)
6. Move to next entry in User_reservation queue
7. Go to Step 3
8. End if
9. If (end of queue reached and "User_reservation.VM_ID" = VM_ID) in VM_allocation table
10. Retrieve TS from VM_allocation table
11. If (User_reservation.TS < TS)
12. User_reservation.TS = TS
13. If "success" returned by Cloud Periodic Update ()
14. Store Name, Mail-id, VM_ID, TS in VM_allocation table
15. End if
16. Else if (User_reservation.TS>= TS)
17. Store Name, Mail-id, VM_ID, TS in VM_allocation table
18. End if
19. End if
20. If (end of queue reached and "User_reservation.TS"=0)
21. If (Name = "Org_index.name") in User_details table
22. Retrieve Organization Type from User_details table and store in Org_type ;
23. End if
24. If (Organization Type= "Org_type" && Name= "Name")
25. Retrieve Org_index.pointer
26. End if
27. Overwrite memory locations pointed by Org_index.pointer
28. Store Name, Mail-id, VM_ID2, TS in VM_allocation table
29. End If

Table 11. Cloud periodic update subroutine**Subroutine: Cloud Periodic Update ()**

1. Retrieve TS, VM_ID from VM_Allocation table and store in TS_list[] and VMlist[]
2. While i < TS_list[].length and j < VMlist[].length
3. If (User_reservation.VM_ID =VMlist[j])
4. Update User_reservation as User_reservation.TS= max(TS_list[i])
5. Retrieve Name from VM_Allocation table and store in Name'
6. If (Name= "Name'")
7. Retrieve Organization Type from User_details table and store in Org_type;
8. End if
9. If (Name = User_reservation.queue.name)
10. Retrieve Organization Type from User_details table and store in Org_type';
11. End if
12. If (Org_type = Org_type')
13. Set User_reservation.status=False;
14. Return "failure"
15. Else if (Org_type!=Org_type')
16. Set User_reservation.status=True;
17. Return "success"
18. End if
19. End if
20. Increment i as i+1
21. Increment j as j+1
22. End while

Table 12. Check validation subroutine

Subroutine: Check Validation ()

1. If (Name!=NULL and Mail-id!=NULL and Organization Name!=NULL and Organization Type !=NULL and password!=NULL and password = Re-type password)
2. Return true;
3. End If

4 Case Study

The proposed scheme is illustrated with the help of the following case study. The present state of the Cloud server is depicted in Tables 13, 14, 15, 16 and 17.

Table 13. User_details table

Name	Email address	Organization name	Organization type	Status
UserA	usra@abc.com	XYZ services	CRM (Customer Relationship Management)	Present
UserB	usrb@abc.com	PQR services	ERP (Enterprise Resource Planning)	Absent

Table 14. User_requirement table

Name	Email address	User memory	User CPU	User storage
UserA	usra@abc.com	1 GB	2vCPU	20 GB
UserB	usrb@abc.com	2 GB	1vCPU	15 GB

Table 15. VM_allocation table

Name	Email address	VM_ID	Time stamp
UserA	usra@abc.com	14	2016/04/14 14:22:13.656008

Table 16. VM_details Table (including *Flag* and *Org-index*)

VM_ID	VM memory	VM CPU	VM storage	Flag	Org-index
14	1 GB	1 vCPU	0 GB	False	–
23	4 GB	2 vCPU	30 GB	True	UserB [0xFFFF:000F, 0 × 9FFF:000F]
17	4 GB	2 vCPU	20 GB	False	–
9	4 GB	2 vCPU	15 GB	False	–
21	4 GB	2 vCPU	25 GB	False	–

Table 17. State of User_reservation queue

Name	Queue entry 1 (VM_ID, Time Stamp, status)	Queue entry 2 (VM_ID, Time Stamp, status)	Queue entry 3 (VM_ID, Time Stamp, status)
UserA	14, 2016/04/14 14:22:13.6,T	17, 2016/04/14 14:22:13.6,T	9, 2016/04/14 14:22:13.6,T
UserB	23,0,T	21, 2016/04/14 17:25:18.6,T	17, 2016/04/14 17:25:18.6,T

A new user, *UserC* wants to deploy her ERP application in Cloud. Tables 18 and 19 show the user details and requirements, respectively, after the subroutines *Cloud User Registration* and *Enter Requirement* (stated in Sect. 3.2) have been invoked.

Table 18. User_details table

Name	Email address	Organization name	Organization type	Status
UserA	usra@abc.com	XYZ services	CRM (Customer Relationship Management)	Present
UserB	usrb@abc.com	PQR services	ERP (Enterprise Resource Planning)	Absent
UserC	usrc@abc.com	PQR services	ERP (Enterprise Resource Planning)	Present

Table 19. User_requirement table

Name	Email address	User memory	User CPU	User storage
UserA	usra@abc.com	1 GB	2vCPU	20 GB
UserB	usrb@abc.com	2 GB	1vCPU	15 GB
User C	usrc@abc.com	2 GB	1 vCPU	28 GB

The algorithm checks for a suitable VM by invoking the subroutine *Check Configuration*. It is evident that VM-23 is the only one capable of meeting the user requirements (Table 16). VM-23 is now checked for security compliance by the subroutine *Check Security*. No conflicting enterprise data is found but the value of *flag* is “true”. As is evident from Table 16, *Org-index* contains *UserB* as an entry whose enterprise type is “ERP” which is the same as that of *UserC*. Therefore, the memory locations stored in *Org-index* are first overwritten and then VM-23 is allocated to *UserC* with the corresponding entry being done in VM_allocation table. It should be noted here that the User_reservation queue for *UserC* contains only VM-23, since this is the only one which matches the configuration requirements of the user. The post-allocation changes in the Cloud system are depicted in Tables 20, 21 and 22.

Thus, the case study illustrates the functions of the proposed scheme.

Table 20. VM_allocation table

Name	Email address	VM_ID	Time stamp
UserA	usra@abc.com	14	2016/04/14 14:22:13.656008
UserC	usrc@abc.com	23	2016/4/15 12:46:18:656001

Table 21. VM_details table (including *Flag* and *Org-index*)

VM_ID	VM memory	VM CPU	VM storage	Flag	Org-index
14	1 GB	1 vCPU	0 GB	False	–
23	2 GB	1vCPU	2 GB	True	UserB [0xFFFF:000F, 0 × 9FFF:000F]
17	4 GB	2 vCPU	20 GB	False	–
9	4 GB	2 vCPU	15 GB	False	–
21	4 GB	2 vCPU	25 GB	False	–

Table 22. State of User_reservation queue

Name	Queue entry 1 (VM_ID, Time Stamp,status)	Queue entry 2 (VM_ID, Time Stamp,status)	Queue entry 3 (VM_ID, Time Stamp,status)
UserA	14, 2016/04/14 14:22:13.6, T	17, 2016/04/14 14:22:13.6, T	9, 2016/04/14 14:22:13.6, T
UserB	23,0,T	21, 2016/04/14 17:25:18.6,T	17, 2016/04/14 17:25:18.6,T
UserC	23, 2016/4/15 12:46:18:6,T	–	–

5 Conclusion and Future Work

In this paper, a novel methodology for Cloud system security has been proposed. The different service units and their functions have been described. An algorithm for implementing the functionalities of the proposed scheme has been detailed. Cloud operations have been described considering aspects of security, data deletion and access control. The methodology can be used to manage the operations and security aspects of cloud services smoothly, and provide assurance to users about the safety of hosting such services. It ensures that a user is not able to access data that belongs to an enterprise within the same Conflict of Interest class [12] as that of the requesting user. This helps to protect the confidentiality of enterprise data. A detailed case study has been included to demonstrate the utility of the work.

The main overhead of this scheme lies in implementing the subroutine, *Cloud Periodic Update* which is to be executed at regular intervals of time for assuring updated and safe copies of VMI. An erroneous update would result in an incorrect and

unbalanced state of the Cloud system leading to improper allocation and de-allocation of the VMs and, possibly, security issues. Moreover, the assurance made by the scheme of overwriting memory locations of conflicting datasets would need hardware-level intervention which should be done carefully, avoiding other unrelated memory locations from getting affected. Future work is geared towards the development of an automated tool based on the proposed methodology. This would help in eliminating such errors while implementing the proposed operations.

References

1. Buyya, R., Yeo, C.S., Venugopal, S., Broberg, J., Brandic, I.: Cloud computing and emerging IT platforms: vision, hype, and reality for delivering computing as the 5th utility. *J. Future Gener. Comput. Syst.* **25**, 599–616 (2008)
2. Hashizume, K., Rosado, D.G., Fernández-Medina, E., Fernandez, E.B.: An analysis of security issues for cloud computing. *J. Int. Serv. Appl.* **4**(5), 1–13 (2013)
3. Rong, C., Nguyen, S.T., Jaatun, M.J.: Beyond lightning: a survey on security challenges in cloud computing. *J. Recent Adv. Technol. Theor. Grid Cloud Comput. Bio-Eng.* **39**(1), 4–54 (2013)
4. Sun, Y., Zhang, J., Zhiong, Y., Zhu, G.: Data security and privacy in cloud computing. *J. Dist. Sens. Netw.* **2014**, 1–9 (2014)
5. Subashini, S., Kavitha, V.: A survey on security issues in service delivery models of cloud computing. *J. Netw. Comput. Appl.* **34**(1), 1–11 (2010)
6. Majumder, A., Namasudra, S., Nath, S.: Taxonomy and classification of access control models for cloud environments. In: Zaigham, M. (ed.) *Continued Rise of the Cloud: Advances and Trends in Cloud Computing*. Computer Communications and Networks, Springer, London (2014)
7. Wang, C., Wang, Q., Ren, K., Lou, W.: Privacy-preserving public auditing for data storage security in cloud computing. In: *2010 Proceedings of INFOCOM*, pp. 1–9. IEEE Press, San Diego (2010)
8. Yu, S., Wang, C., Ren, K., Lou, W.: Achieving secure, scalable, and fine-grained data access control in cloud computing. In: *2010 Proceedings of INFOCOM*, pp. 1–9. IEEE Press, San Diego (2010)
9. Liu, Q., Wang, G., Wu, J.: Time-based proxy re-encryption scheme for secure data sharing in a cloud environment. *J. Inf. Sci.* **258**, 35–370 (2014)
10. Wang, H., He, D., Tang, S.: Identity-based proxy-oriented data uploading and remote data integrity checking in public cloud. *J. IEEE Trans. Inf. For. Secur.* **11**(6), 1165–1176 (2016)
11. Aceto, G., Botta, A., de Donato, W., Pescapè, A.: Cloud monitoring: definitions, issues and future directions. In: *2012 Proceedings of IEEE CloudNet*, pp. 63–67. IEEE Press, Paris (2012)
12. Basu, S., Sengupta, A., Mazumdar C.: Implementing Chinese wall security model for cloud-based services. In: *ICGCIoT 2015*, pp. 1083–1089. IEEE Press, Noida (2015)
13. Cloud Management and Monitoring. <http://searchcloudcomputing.techtarget.com/definition/cloud-sprawl>
14. Infrastructure as a Service Cloud Concepts. <http://www.ibmpressbooks.com/articles/article.asp?p=1927741&seqNum=7>

Provenance-Aware NoSQL Databases

Anu Mary Chacko^(✉), Munavar Fairouz, and S.D. Madhu Kumar

National Institute of Technology Calicut, Kozhikode, India
anu.chacko@nitc.ac.in

Abstract. NoSQL stores are very widely used for BigData Analytics. These stores are built with inherent scalability and fault tolerance. But there are not much mechanism to provide security guarantees like integrity and auditability. Provenance is a metadata which captures the details of how the data reached its current state. By way of capturing provenance it is possible to enhance the functionality of NoSQL stores to verify the integrity of results. This paper presents an approach to capture provenance of NoSQL databases using logs generated by the database. A proof of concept was implemented in MongoDB and examples are used to illustrate the use of ‘Why provenance’ and ‘How-provenance’ captured.

Keywords: Data provenance · NoSQL databases · MongoDB · MapReduce · How-provenance · Why-provenance

1 Introduction

With the growth of information technology, the volume of data has grown enormously in the last decade. This large volume of data available is usually unstructured in nature which when mined can give a lot of useful information. For meeting this requirement new paradigms of storage and analytics have evolved, and NoSQL databases is an example of the same. NoSQL stands for Not Only SQL and is an evolution of traditional Relational Databases to handle the large volume of unstructured data. NoSQL databases usually have a simple data model, support basic operations, have weak security but have high availability and scalability. The key attraction of these databases is the dynamic schema that allows the different types of unstructured data to reside in the same collection/table. Most of the NoSQL databases support only CRUD (Create, Read, Update, and Delete) operations. So when the analytic query demands more complex join or aggregation operations, analytic frameworks like MapReduce are used.

When a huge amount of data is processed, and decisions are derived based on it, users need some mechanism to ensure the credibility of decision. Data provenance can help here. Data provenance is the metadata that captures the creation and subsequent modification of the data as it is processed in and across systems.

In the case of NoSQL data analytics, data in the range of terabytes and petabytes are being processed. The credibility of the result produced by analyzing the big data is dependent on the goodness of data. Provenance is the metadata which captures the information about how a data reached its current state. Hence, provenance can be of great help in verifying and ensuring the “goodness” of data.

The relationship between provenance and security can be considered symbiotic [1]. To ensure security attributes like auditability or to make good security decisions users need well-captured provenance. To ensure good security practices provenance needs to be audited.

Security is a major concern for the adopters of Big Data and Cloud processing. Ensuring efficient capture and storage of provenance for data processed will provide the users a mechanism to verify and debug results obtained. Also, provenance can throw light on potential security breaches. In this paper, we evaluate an option to capture provenance of NoSQL databases and explore the uses of the collected provenance. There is no provision for capture of provenance in any of the existing NoSQL implementations. Rest of the paper is organized as follows: Sect. 2 discusses the related work in the area, Sect. 3 explores the detail of the proposed design, Sect. 4 gives the details about the implementation of proof of concept and Sect. 5 concludes the paper by exploring future work possible.

2 Related Works

Initial works in provenance was done in 1970s in the area of eScience where provenance was collected to ensure reproducibility and verification of scientific experiments. Examples of such systems are Chimera [2], MyGrid [3], PASOA [4], etc. The approach adopted for provenance capture was to redesign existing application/workflow to capture provenance.

PASS [5] (Provenance-Aware Storage System) attempts to capture provenance at the storage level. PASS is built by modifying Linux kernel to automatically deduce provenance information by observing operating system calls at read/write level. There have been significant contributions in the domain of relational databases as well.

Buneman et al. [6] categorizes database provenance as ‘Why-provenance’ and ‘Where-provenance’. ‘Why-provenance’ lists all source data items that contributed to the creation of result data item. ‘Where-provenance’ lists the originating sources of the result. This categorization has been extended to include ‘How-provenance’ that explains how the individual derivations have been carried out according to the query [7].

A practical example of making a relational database provenance-aware is seen in the project done by the University of Illinois called PERM (Provenance Extension of Relational Model) [8] built by extending PostgreSQL engine. Here provenance is captured by query rewriting and is displayed along with the query results as additional columns.

Kulkarni [9] suggests a generalized provenance model for key-value systems. The proposed system has the capability to capture tuple provenance and schema provenance. The author proposes application to explicitly select the data/collection for which provenance need to be captured. For updates, the value before modification is also captured. The scheme provides provision for a logical marker which will help in tracking set of columns as a single logical unit. Provenance queries are provided for finding the provenance information required. A proof of concept was implemented on by modifying Cassandra to make it provenance-aware.

A NoSQL metadata management tool called Wasef was proposed by Alkhalidi et al. [10]. Wasef captures provenance as one of the metadata. Provenance capture and use are quite primitive in the initial model. A proof of concept was implemented in Cassandra and evaluated.

Both KVPM and Wasef is implemented by changing the code of Cassandra to make it provenance-aware. It will be a better approach to have a generic provenance collection methodology which automatically capture provenance by observing transactions in the database. In this paper we propose a novel approach of automatic provenance collection in NoSQL database by monitoring the logs and demonstrate a proof of concept of our idea in MongoDB. The practical use of provenance collected is illustrated through an example.

3 Design

When we do analytics on data stored in NoSQL store, the primary use of provenance will be to explain unexpected results. For this purpose ‘why-provenance’ and ‘how-provenance’ will be useful. ‘Why-provenance’ will list all the source data tuples that contributed to the creation of result data tuple. ‘How-provenance’ will list the operations that caused the data tuple to reach the current value. So combining ‘how-provenance’ and ‘why-provenance’ we will have a holistic answer to how the result tuple was formed. NoSQL databases are designed to be scalable and can partition across many servers. The system scales transparently, and built-in fault tolerant mechanisms via replication, make it difficult to capture the ‘Where-provenance’.

There are two approaches for capturing provenance. One approach is to redesign individual application to make it provenance-aware and the second approach is to automatically deduce provenance by observing the transformations to data. In this paper, we propose a novel approach to capture provenance by using existing logs in NoSQL database. NoSQL databases are designed with logs to enable replication of changes to ensure transparent scalability. The information in the logs can be reused to capture provenance of data transactions. As a proof of concept, MongoDB is made provenance-aware using this approach.

4 Proof of Concept - Provenance Aware MongoDB

MongoDB is an example of document-oriented datastore that captures data in the form of key-value pair. To capture provenance we use the concepts in key value provenance model proposed by Kulkarni [9].

4.1 Requirements

MongoDB supports only basic CRUD operation. Complex analytic queries are supported by built-in MapReduce Framework. So in this work the ‘How Provenance’ of data stored in MongoDB and ‘Why Provenance’ of data queried using MapReduce

is captured. Combining the ‘How-provenance’ with the ‘Why-provenance’ the holistic picture of the contributing tuples of a result can be obtained.

The requirements of the Provenance Aware MongoDB are listed below.

1. Users should be able to track tuple level and schema level provenance.
2. Users should be able to know the contributing sources for a result tuple (why-provenance) and know the operations that caused the tuple to have current value (how-provenance).
3. The solution should be generic as far as possible and should have minimal instrumentation of existing application/data store.
4. As the volume of provenance and data is huge, the user should be provided the option to select the tuple or table for which provenance needs to be captured.
5. Overheads should be minimal as far as possible.

4.2 Capturing ‘How-Provenance’

To capture ‘How-Provenance’ system needs to capture the operations through which a tuple reached its current state. MongoDB stores humungous amount of data and provenance may not be relevant to all data. Hence, the user/programmer of the database is given flexibility to identify the document for which provenance needs to be collected. Resource for which provenance needs to be captured can be listed as ‘resource expression’.

Resource expression can be written in the following formats.

If the provenance is to be tracked.

- For a particular document inside a collection -<Database/Collection/Id>.
- For a collection - <Database/Collection>(provenance is tracked for all documents for the collection).

MongoDB maintains a capped collection called Olog for storing all operations that happened in the database so as to replicate in secondary servers. Capped Collections are fixed size collections in MongoDB where documents are retrieved in the order of insertion, and when the size gets exhausted, space for new documents are made by overwriting the oldest documents in the collection.

Olog entries in MongoDB include the timestamp, unique id, operation name, namespace with the details of the database, collection and document affected by the operation and the new state of the document after performing the operation. Primary olog captures all the operations that are applied on the primary node. The secondary nodes copy and apply these operations in an asynchronous process to achieve eventual consistency. Thus, olog entries give primary information to track ‘how-provenance’. In addition to the above information, details about the user executing the action need to be captured.

The ‘how-provenance’ is captured by setting up a tailable cursor to the olog. This script runs parallel to the MongoDB process to detect any entry being made in the olog. When a new entry comes to olog, the resource expression file is checked to identify whether the tuple is listed for provenance capture. If so, information from Olog like timestamp converted to ISO date time, operation type are retrieved and stored in a

separate append only provenance collection. This provenance information is augmented with user information to make provenance complete.

The following example illustrates the ‘how-provenance’ captured for a document.

Suppose in the MongoDB database called ‘hospital’ there exists a collection called ‘patients’. Assume that we want to track the provenance for a particular patient, say ‘P123’. In the beginning we specify resource expression as <hospital/patients/P123>

The current state of the patient record is given in Fig. 1

```
{
  "_id" : "P123",
  "Name" : "John",
  "Doctor" : "Dr. jacob",
  "Disease" : "Asthma",
  "Medication" : [ "Doxil4", "Laxin" ],
  "Allergy" : "Sneezing "
}
```

Fig. 1. Document in MongoDB for P123

‘How-provenance’ for the document is given in Fig. 2.

```
{
  "_id" : "hospital.patient.P123",
  "Provenance" : [
    {
      "Op_Type" : "i"
      "Operation" : "{ 'Name': 'John', 'Disease': 'Asthma',
        'Medication': ['Doxil4', 'Aadrone'],
        'Doctor': ' Dr. James '
      }",
      "Time" : ISODate("2015-04-29T12:56:49Z"),
      "user" : "Dr. James",
    },
    {
      "Op_Type" : "u"
      "Operation" : "{ '$set': {'Medication': ['Laxin'],
        {'Doctor': 'Dr. Jacob'}}",
      "Time" : ISODate("2015-04-29T1:57:08Z"),
      "user" : "Dr. Jacob",
    },
    {
      "Op_Type" : "u"
      "Operation" : " { '$set': {'Allergy': 'Sneezing'},
        {'Medication': [Doxil4, Laxin]}}",
      "Time" : ISODate("2015-04-29T32:57:16Z"),
      "user" : "Dr. Jacob",
    }
  ]
}
```

Fig. 2. How-provenance captured

The above example shows how both data and schema provenance is available for querying.

4.3 Capturing ‘Why-Provenance’

The work was extended to capture MapReduce provenance in MongoDB. The provenance collected characterizes as ‘Why-provenance’ as it gives reason/witness for why an output was obtained.

Ikeda et al. [10] proposed a model to capture provenance in MapReduce workflows in Hadoop called RAMP (Reduce and Map Provenance). They use a wrapper-based approach to capture provenance by building wrappers for the various components of MapReduce framework like Record Reader, Mapper, Reader, and Writer. They apply an eager approach and record provenance as the workflow proceeds. The eager approach of computing provenance along with MapReduce computation introduces a lot of computational overhead and delay in job execution time. Hence a lazy approach of computing provenance was used in this work. The mapper and document writer was modified to write intermediate data into two temporary files. Mapper emits key value pairs (ki, vi). For each of the mapper output, a pair (ki, pi) is recorded in the file1 where pi is the 12-byte document id of the input document being processed. Reducer processes all the values with the same key (ki, [v1,v2,...vn]) and produces an output of the form (ki, vi). The document writer writes the key-value pair in the result collection as well as in file2. After the execution of the job, provenance file is generated by a python script, which takes the two files as input and maps input provenance with output value using the key. So the output provenance value will be (V, {p1, p2 ... pn}) and will list all input documents which contributed towards output record. The algorithm to capture MapReduce Provenance is diagrammatically explained in Fig. 3.

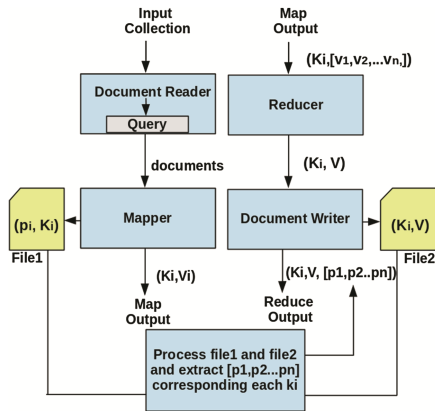


Fig. 3. Modified MapReduce to capture ‘Why-Provenance’

The example given below illustrates the practical application of the ‘Why-provenance.’

In a hospital application, the collection of patient’s medication bills at different times are captured in hospital database is illustrated in Fig. 4.

The total bill for a patient can be consolidated by running a MapReduce job. The output for the same is shown in Fig. 5.

The provenance captured via MapReduce (Why Provenance) and MongoDB(How Provenance) will provide a good explanation for the result as shown in Fig. 6.

Patient id	Bill Date	Prescribed Doctor	Items	Price(₹)
P127	2012-12-13 22:00:00	Dr.Jacob	{ "Medicine": "Aidol7", "qty": 10, "price": 2.5 } { "Test": "MRI", "qty": 1, "price": 1250 }	1275
P133	2012-09-04 00:00:00	Dr.Ajeeb	{ "Medicine": "Laxin", "qty": 5, "price": 10 } { "Medicine": "Mentol", "qty": 5, "price": 2.5 } { "Test": "Blood Test", "qty": 1, "price": 50 }	111.5
P123	2012-10-03 14:00:00	Dr.Ajeeb	{ "Medicine": "Ameco7", "qty": 5, "price": 20 } { "Medicine": "Mentol", "qty": 5, "price": 2.5 } { "Test": "ECG", "qty": 1, "price": 125 }	234.5
P127	2012-12-13 22:00:00	Dr.Jacob	{ "Medicine": "Aidol7", "qty": 10, "price": 2.5 }	25
P127	2012-12-13 22:00:00	Dr.Ajeeb	{ "Medicine": "Demol Tab", "qty": 20, "price": 2.5 } { "Test": "ECG", "qty": 1, "price": 250 }	300
P123	2012-12-13 22:00:00	Dr.Jacob	{ "Medicine": "Abeol", "qty": 10, "price": 25 }	250
P123	2012-10-04 00:00:00	Dr.Jacob	{ "Medicine": "Laxin", "qty": 5, "price": 2.5 } { "Test": "ECG", "qty": 1, "price": 125 }	137.5
P133	2012-12-04 04:00:00	Dr.Ashly	{ "Medicine": "Laxin", "qty": 5, "price": 2.5 } { "Medicine": "Alo xenol", "qty": 25, "price": 25 }	625
P333	2013-01-04 04:00:00	Dr.Hema	{ "Medicine": "Laxin", "qty": 5, "price": 2.5 } { "Medicine": "Alo xenol", "qty": 25, "price": 25 } { "Test": "ECG", "price": 125 }	150

Fig. 4. Hospital database

Key	Value
P127	1600
P333	150
P123	622
P133	736.5

Fig. 5. Output of the query for total bill

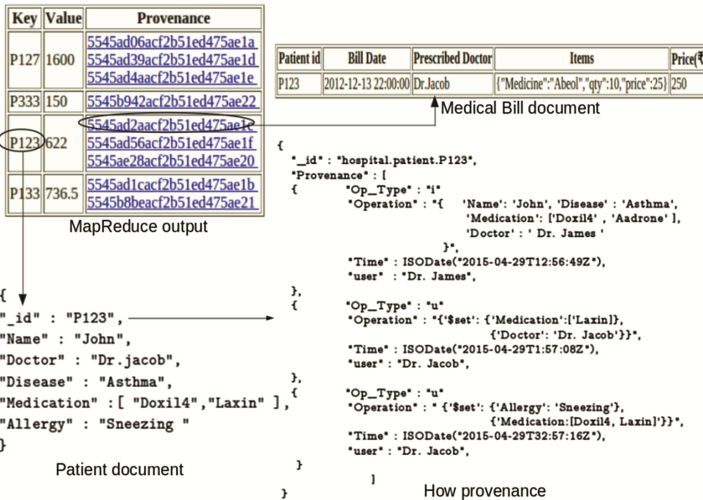


Fig. 6. Why and how-provenance combined

4.4 Performance Analysis

The experiment was conducted on a single node MongoDB 2.6.1 on an Intel i3CPU Linux Laptop with 4 GB RAM.

How-Provenance’ Overhead. As the strategy to capture ‘how-provenance’ is by running a tailable cursor parallel to MongoDB server process, there is no time overhead associated with the same. However, the storage overhead is proportional to the number of operations done on the document/collection.

Why-Provenance’/MapReduce Provenance Overhead. We ran MapReduce on MongoDB on various randomly generated datasets of different sizes. The time given is cumulative time for processing MapReduce workflow and running the python script to generate provenance. Time overhead is about 70–73 %. Space overhead depends on the documents that are processed, as for every input document 12 bytes are required to track the document id. In the 260 MB dataset used, space overhead is very large as we used more than 700000 documents in the dataset. Experiment results are graphically represented in Fig. 7.

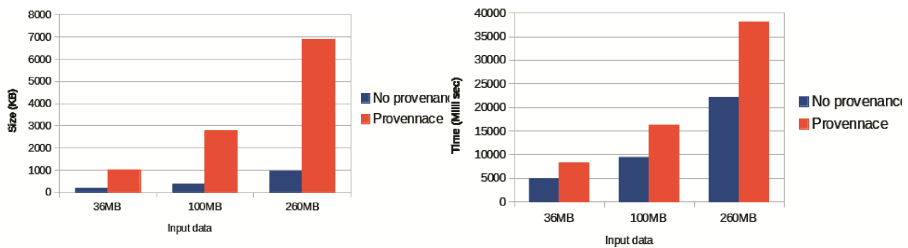


Fig. 7. Space overhead and time overhead for why-provenance against data size

5 Conclusion and Future Work

Provenance is a metadata that helps us audit the security of our systems and ensure that our system is trustworthy. In this era of data explosion, big data analytics is extensively used for decision making. In this context, capturing provenance is critical as it provides a mechanism to ensure trustworthiness of result. In this paper, the option of capturing of the provenance of NoSQL database by using system logs is explored. The proof of concept was demonstrated by building a basic prototype in MongoDB and capturing ‘how-provenance’ and ‘why-provenance’ of queries.

Using our approach, any application built on top of MongoDB can capture provenance of all database operations without adding any code for the same. The user is given the flexibility of selecting documents for which provenance need to be tackled via resource expression.

The work is based on the assumption of fault tolerance that is built into NoSQL databases via replication mechanism. In this basic prototype, we have explored the type of provenance and their uses. As future extension of this work, storage and performance optimizations to make the provenance model more usable can be explored. Currently an append-only table is used to capture provenance so that provenance remains immutable. Securing the provenance with integrity and confidentiality guarantees will be an interesting line of work.

References

1. McDaniel, P.: Data provenance and security. *J. IEEE Secur. Priv.* **9**(2), 83–85 (2011)
2. Foster, I., Vöckler, J., Wilde M., Zhao, Y.: Chimera: a virtual data system for representing, querying, and automating data derivation. In: *Proceedings of the 14th Conference on Scientific and Statistical Database Management* (2002)
3. Ikeda, R., Salihoglu, S., Widom, J.: Provenance- based refresh in data-oriented workflows. In: *Proceedings of the 20th ACM International Conference on Information and Knowledge Management* (2011)
4. Moreau, L., Groth, P., Miles, S., Vazquez, J., Ibbotson, J., Jiang, S., Munroe, S., Rana, O., Schreiber, A., Tan, V., Varga, L.: The provenance of electronic data. *Commun. ACM* **51**(4), 52–58 (2008)
5. Muniswamy-Reddy, K., Holland, D., Braun, U., Seltzer, M.: Provenance-aware storage systems. In: *Proceedings of the 2006 USENIX Annual Technical Conference*, Boston, June 2006
6. Glavic, B., Dittrich, K.R.: Data provenance: a categorization of existing approaches. In: *Proceedings of the 12th GI Conference on Datenbanksysteme in Buisness, Technologie and Web (BTW)* (2007)
7. Cheney, J., Chiticariu, L., Tan, W.-C.: Provenance in databases: why, where and how. *Found. Trends Databases* **1**(4), 379–474 (2009)
8. Galvic, B.: Perm: efficient provenance support for relational databases. Ph.D. thesis, University of Zurich (2010)
9. Kulkarni, D.: A provenance model for key-value systems. In: *TaPP 2013 Proceedings of the 5th USENIX Workshop on the Theory and Practice of Provenance* (2013)
10. Park, H., Ikeda, R., Widom, J.: RAMP: a system for capturing and tracing provenance in MapReduce workflows. In: *International Conference on Very Large Data Bases*, pp. 1351–1354 (2011)

Efficient Key Management in IoT Using Mobile Aggregator

Sumit Saurabh, Alwyn R. Pais, and Sumanta Chatterjee^(✉)

National Institute of Technology Karnataka, Surathkal, Mangalore, India
ssaurabh262@gmail.com, alwyn.pais@gmail.com,
Sumanta.Chatterjee.nitk@gmail.com

Abstract. Managing keys in Internet of Things (IoT) is challenging. With this proposed work we are trying to address an efficient key management protocol for specific application based scenario which enforces secure connectivity of devices and minimizes node capture attacks. There are a number of protocols that have been enforced and implemented for wireless sensor networks (WSN) and internet-enabled devices. We propose a protocol with mobility interface using combinatorial designs for key management in IoT devices. Mobile devices follow a dedicated path to collect data securely from installed devices in the network. We also compare our work with existing protocols and few mobility models.

Keywords: IoT · Key predistribution · WSN · Combinatorial design

1 Introduction

The connectivity of *smart* and *intuitive* devices, more popularly known as Internet of Things (IoT) opens exciting range of opportunities. Kevin Ashton (Co-founder of Auto-ID at MIT) proposed term Internet of Things (IoT) in the year 1982 and he termed this “thing” in a conference in 1999. International Telecommunication Unit (ITU) and European Research Cluster on the Internet of Things (IERC) [1] officially defined as “*Things*” are active participants in business, information and social processes where they are enabled to interact and communicate among themselves and with the environment by exchanging data and information sensed about the environment, while reacting autonomously to the real/physical world events and influencing it by running processes that trigger actions and create services with or without direct human intervention.

1.1 IoT and Security

IoT constitutes of items that forms a system, and it encompasses sensors within the system formed, are connected to the Internet via wireless and wired Internet connections. These connections can be of type RFID, Wireless Fidelity, Bluetooth, Near Field Communication (NFC) and Zigbee to connect in local area. For wide area connection system requires GSM/CDMA, GPRS and 3G/4G/LTE.

The sole purpose of IoT is to communicate & connect *machine-to-machine, machine-to-man, and man-to-man*. In short we can say that, *IoT as smart things are combination of Sensors & actuators, connectivity and people & processes*. IoT also comes with a vision that individual objects of everyday life can measure, track and analyze useful information about environment. It provides convenience for identification, management and control. There are wide range of applications that IoT provides across private and public areas that includes healthcare, production of goods and their transportation, oil/gas, military applications, etc.

As IoT aims to be present in a ubiquitous manner, there is increase in the level of communication and the exchange of data. IoT, fusion of technologies like WSN, Bluetooth, mobile communication network not only involves existing problems of security, but it also incurs more issues like privacy protection problem, authentication for network and access control, data storage and management. Security is a vital component for the IoT devices. However, there is a lack understanding by organizations and people for IoTs specific security requirements that somewhere affects its growth potential. So to design IoT environment, first we need to understand the specific requirements for an application and then we can play with trade offs between security and performance of IoT-enabled devices. As the IoT is flourishing, large numbers of heterogeneous devices will have to be managed. In this context, the problem of how to manage different sensors and devices on different locations at once is of paramount importance. Key management offers a secure way to manage these devices in the networks and minimizes risk of devices capture.

1.2 Key Management in IoT

Key management is one of vital part which can used for securing IoT devices. It can contribute to end-to-end security and mitigate various risks. An efficient and novel approach can securely authenticate the device. However, key management in IoT has to go through several challenges. IoT-enabled devices are mostly unattended. Key management provides secure way of communication among devices in the network. There are many key management schemes which are proposed and implemented for WSN, Ad-Hoc networks, WiFi etc. Based on existing protocols on these technologies, from these protocols few have been proposed for IoT. Induction of smart devices also adds a number of concerns in terms of security. Symmetric key cryptography provides faster way to manage keys in the devices. However devices are “constrained”, devices consist of sensors and mix of technologies. They have limited CPU power and storage. Security vs. performance has always been a trade-off. Therefore, an efficient key management protocol is required to securely use or monitor the devices.

2 Background

In cryptography, keys are being used, ranging from Caesar’s cipher to Diffie-Hellman, RSA in private or public way. Keys involved in these protocols often

requires to be generated, distributed and used properly, in short they need to be managed properly. Whenever there is need of secure communication in a suspicious ambiance, key management is used in most of the part.

2.1 Key Management

Key management is an omnipresent term that is associated with the entire process of cryptographic keys. An IoT-device can “talk” with another device over the internet if the second device lies in the range of the first device considering the range as in the circle, or if there lies a common key between them. To manage keys, key management protocols follow a lifecycle of keys. Lifecycle starts from *key generation* and then *key distribution & establishment, key storage, key use, key update* and ends with *key destruction*.

We are discussing the phase after key generation, i.e. key distribution and establishment. Key management schemes are broadly classified as: Key Predistribution, Session-Key Agreement & Key Agreement. In our protocol, we use Key predistribution as a key management scheme.

2.2 Key Predistribution

Key predistribution is defined as distributing keys in nodes prior to deployment, i.e. Key predistribution is the method of distribution of keys onto nodes before deployment. Therefore, the nodes build up the network using their secret keys after deployment, that is, when they reach their target position.

The first literature on key predistribution was in 1985, by R. Blom [2]. Thereafter many schemes were proposed. A key predistribution scheme comprises of three phases [3]. They are *key predistribution, shared-key discovery, and path-key establishment*. Key predistribution phase is about selecting a set of keys from a key pool to form a group. Next phase, i.e. shared-key discovery phase deals with the communications of two nodes and how they can obtain a common key if they are lying in each others communication range. And in path-key establishment phase two nodes A and B can communicate if they can find intermediate node say, C which shares a common key between them.

WSN often requires complete connectivity of the network. To design a WSN with full connectivity, keys can be predistributed in two major ways: First is having one single master key for entire system topology, known as master key predistribution. And the second is pairwise key predistribution in which keys are predistributed in a discrete manner. Master key predistribution is scalable and ensures fully connected network topology. But individual node capture exposes its whole designed system to the attacker. To counter this attack pairwise key distribution can be used, as keys are distributed in a way that the single node attack will not create an impact on a larger scale.

Key predistribution schemes for WSN can broadly be classified into three schemes: *Random, polynomial-based and Combinatorial* [4]. Polynomial based key predistribution is probabilistic while combinatorial design key predistributions are deterministic. In our proposed work, we are using combinatorial design.

2.3 Combinatorial Design

Combinatorial design theory is about a possible arrangement of elements of a given finite set into subsets to fulfill certain properties which are definite by nature. It describes the families of subsets with various pronounced regularity characteristics and patterns. Combinatorial designs comes with distinguishable “patterns”, and this unique, interesting property becomes to be useful for shared key discovery and path-key establishment. Using combinatorial design makes these two very efficient in terms of computation and communication complexity. Mitchell [5] were first to design a key distribution in networks using combinatorial design. Later, this contrasting property was mentioned by Ruj and Roy [6].

A set system or design is a pair of X and A represented as (X, A) , where A is a set of subsets of X , known as blocks. Universal set, S elements are called varieties in set design, and *blocks* are subsets selected from that. Some of regular designs are shown in [7,8] as:

Balanced Incomplete Block Design (BIBD) This design is represented as (v, b, r, k, λ) , where v represents distinguishable objects, b is used for blocks. And each block b is having distinct objects as k . Each k occurs in exactly r different blocks, and every pair of distinct k occurs together in exactly λ blocks. Where: $\lambda(v-1) = r(k-1)$ and $bk = vr$. **Symmetric BIBD (SBIBD)** with $v = b$. Pattanayak [3] gives combinatorial design for $q = 4$ (non prime).

Evaluation Metrics. The following discusses the important factors based on which a key management protocol can be weighed.

- *Key connectivity* is the probability that the two nodes share one/more keys with which they can securely obtain a link for communication.
- *Scalability* means the ability to add new nodes in the network and assign them key-chains from the existing key pool.
- *Communication cost* refers to number of transmit messages exchanged during the key management protocol.
- *Resiliency* refers to the number of nodes uncompromised in the network after an attack.
- *Computation cost* refers to the number of computations (number of CPU cycles) required to establish a node-to-node common key.
- *Storage cost* refers to the amount of memory for the storage of keys for a single node.
- *Mobility* refers to path (based on mobility model) the device follows or use in the network for some purpose like data aggregation.

2.4 Mobility

As the electronic and world of Internet has grown rapidly, leading to invention of smart mobile devices. Smartphone is one example, having attractive features on the screen. Mobility is an essential feature of these smart devices. It comes handy in various situations. In IoT ambience, Devices having mobility can add

great advantage to it. Mobile elements perambulating the nodes in the network can collect data from sensor nodes when they arrive to node. Adding mobility in devices can be helpful alternative for avoiding multi-hop communication and thus reducing the relay overhead and also they can be used for recharging devices/sensors in the defined path. Mobility in nodes can be broadly classified as *random, predictable or controlled*. Considering the data mule (Humans & animals), where a mobile element can be installed as “data mule” and it moves based on programmed coordinates. The use of a predictable mobile element is similar to the sensing element mounted on the bus. The static devices or nodes learn when the bus comes near them, and wake up as required to communicate or it can also have some features like charging that node by the bus or vice-versa. Controlled mobility is mobility which is based on given coordinates where the programmed node acting as the mobile device, moves on a predetermined path, but its speed can be varied depending on specific requirements while moving across in that path and also on availability of the communication and node population. Based on these various mobility models have been defined in [9].

Somasundara et al. [10] discusses about vehicular routing problem (VSP) and various problems in mobility of nodes.

Existing protocols on key management and mobility have been evaluated based on evaluation metrics. Blom and Blundo schemes are x -secure in resiliency, where x represents those nodes which are secure among total number of nodes [2,6]. Ruj scheme shows high resiliency against node capture attacks and uses very less storage as compared to other two schemes. These three schemes mention that they support fully mobile network. However, they do not mention about making selected nodes mobile who will also perform data aggregation in the network. The protocol in [10] describes how mobility can be useful in various applications. However, this protocol does not talk anything about security. In this project, we aim to provide a protocol which will ensure the task of providing secure communication in the network via these aggregator mobile nodes.

2.5 Example Scenario

Let us consider a military campus in a remote geography. Here we need to monitor various things, like the movement of convoys over a bridge, to inspect the condition of connecting bridge between two points in a campus. Then we can also monitor climatic conditions of that area. To accomplish these activities securely, an efficient protocol is required which can effectively do the things and report to the concerned authority.

3 Proposed Work

This section describes our proposed work and solution framework using combinatorial design and enabling mobility in aggregator agents.

3.1 Problem Definition

We contemplate a network scenario that consists of IoT devices. This group of devices forms a hierarchical topology, where end-to-end security needs to be enabled between any two legitimate devices in the network. We also consider secure exchange of data and mobile communication through an efficient protocol. In this scenario the following properties are considered:

- An efficient key management protocol that will enable end-to-end security for devices of the IoT.
- A protocol with optimal storage is required for devices and sensors because of their limited resources (Storage & Computation). A good design is required to judiciously and securely use the IoT devices.
- For some selected nodes in the network, mobility should be enabled. These mobile nodes will have a predefined path on which they will traverse.
- Mobile node is a data aggregator, that perambulates through devices/nodes in the network. It is expected to have more power than trivial sensor nodes in the network.
- Since aggregation require significant power consumption of the aggregator node, efficient protocols are required to be implemented to reduce the energy depletion rate of the aggregator.
- All aggregated data via mobile node should flow into the Sink node. Sink node is having more computational and storage resources than mobile and sensor nodes.

To meet the standards of an efficient protocol we consider following details.

- Hierarchical topology over a square grid. Sensor nodes and devices can deployed randomly in the remote and corner regions of the considered environment as shown in Fig. 1(b).
- A set of devices, with prime order, q with total $q^2 + q + 1$ devices.
- A SBIBD design over a square grid that predistributes the $q + 1$ keys each in the $q^2 + q + 1$ devices by assigning it to respective key identifiers.
- Out of $q^2 + q + 1$ devices, q devices are mobile and $q^2 + 1$ have fixed location coordinates.
- Mobility models: Manhattan, Random walk and Random Way point.

For this network model, we make the following assumptions:

1. The Sink Node (S) is honest.
2. Agent node (M) is considered mobile, while devices (D) will be fixed but can be deployed randomly.
3. Network prior to deployment is uncompromised.
4. Adversaries have limited capability. Both inside and outside adversaries can be present.

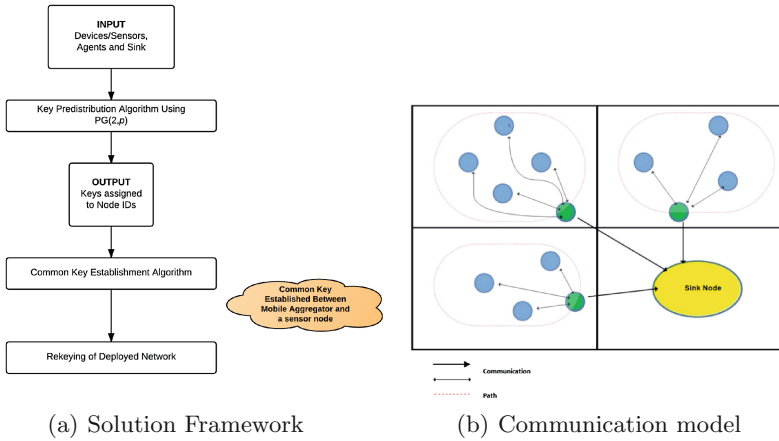


Fig. 1. Solution Framework with Communication model

3.2 Solution Framework

A hierarchical system design is considered. Let us consider a scenario of military region with remote geography, wherein sensors nodes are deployed. The following summarizes the proposed solution framework for the problem definition.

- Sensor nodes and devices (D) will be task specific, deployed in large numbers and with limited computational power for required purpose.
- Agent Node (M) visits every device in the dedicated path based on mobility models taken from [9]. Key predistribution using combinatorial design is used for deployment of keys in the given network.
- In Shared Key Discovery for Mobile agents, Agent device visits every node in the dedicated path. Shared Key Discovery Algorithm is implemented using combinatorial SBIBD Design. As per Key predistribution, after shared key discovery one has to apply path key establishment. Instead of that we have made q sensor nodes nodes mobile.
- Mobility of agents has been enabled on three mobility models: Manhattan Grid, Random WayPoint and Random Walk. As per positions generated by these three, a mobile node moves on predetermined path. Mobile node traverses through the path and communicates with nodes it came across on the path with the shared key obtained from Shared key discovery. Adding mobility eliminates the path key establishment phase of key predistribution and offers a better alternative at optimal cost
- Rekeying for designed network.

3.3 Design

Proposed scheme takes a step further to basic predistribution schemes. Before deployment of the IoT system, Topology of the network has to have a definite set

of keys out of a reasonably giant key pool. However, we now consider our design for small scale purposes, such as operating few smart devices in an office with controlled mobile device. Following describes the design of the scheme briefly.

Combinatorial Design. We use type of SBIBD transversal design for generating a key pool to be distributed in the network. A *finite projective plane* consists of a finite set of points P and a set of subset of P called lines. For an prime power q a finite projective plane consists of $q^2 + q + 1$ points, $q^2 + q + 1$ lines where each line contains $q + 1$ points and each point occurs in $q + 1$ lines. If we consider lines as blocks and points as elements, then a finite projective plane of order q is a Symmetric-BIBD known as SBIBD. Beauty of this design is that any two nodes will share a common key. Based on this design we have mapped our of set systems to IoT environment as shown in Table 1. Consider a set system (X, A) . $X = \{0, 1, 2, 3, 4, 5\}$ $A = \{\{0, 1, 2\}, \{0, 1, 3\}, \{0, 2, 4\}, \{0, 3, 5\}, \{0, 4, 5\}, \{1, 2, 5\}, \{1, 3, 4\}, \{1, 4, 5\}, \{2, 3, 4\}, \{2, 3, 5\}\}$. Here, this set system forms a BIBD design $(v, b, r, k; \lambda)$, where $v = |X| = 6$, $|A| = b = 10$, $r = 5$, $k = 3$ and $\lambda = 2$. BIBD design for this example can be represented as BIBD $(6, 10, 5, 3; 2)$. This is very basic BIBD design with parameters described as below:

- b denotes the size of key pool and v number of nodes/devices. Each subset in $|A|$ has exactly r elements, denoting size of key chain.
- Any pair of nodes have exactly λ in common. Two nodes in a topology will share atleast λ keys. This parameter is considered important while enabling mobility, as mobile node needs to establish a connection with sensing node during its visit. λ needs to be chosen carefully to observe better security between pair of nodes. Neither too less which can hamper communication nor too much which can have resiliency risks.

Table 1. Mapping set system to IoT environment

Block	Devices or sensors
Elements	Key identifiers
Elements present in each block	Keys present in each sensor

Key Predistribution. In key predistribution, key predistribution using PG $(2, p)$ is applied [6] for particular square grid to generate a key pool or preferably called as key table. In this construction design, each key identifier and its keys are in form (i, j, k) and (x, y, z) . Key identifiers for devices are based on prime power, p . Each key identifier node will be having $p + 1$ keys stored in it. We index Sensor/devices in the form of $(1, j, k)$, $(0, 1, k)$, & $(0, 0, 1)$ where $j, k \in GF(p)$. So there will be total of $p^2 + p + 1$ devices. Similarly the keys present in nodes are indexed by (x, y, z) where $x, y, z \in GF(p)$. The key identifiers are

in these forms: $(x, y, 1)$, $(x, 1, 0)$, $(1, 0, 0)$. Thus, a total of $p^2 + p + 1$ keys will be distributed. A key will be assigned to a node if it satisfies following property:

$$ix + jy + kz \equiv 0 \pmod{p} \quad (1)$$

After that p nodes will be chosen randomly among key identifier nodes. Then we organize devices based on applications. And nodes with fixed can be initialized with random or predefined fixed locations. After deployment of all devices in the defined topology, shared key discovery algorithm [6] is applied to establish a common key between mobile node and a sensor node.

Mobility of Agents. Adding mobile agents in network add up many possibilities in the existing topology. Proposed solution considers mobile agents nodes, which are similar to mobile sensor networks. But here only few agents are mobile unlike in MWSN, where every node is mobile. Mobility of agents considers following:

- Can move from node to neighboring node.
- communication is available over time and space, making broadcast, routing and computations feasible.

Mobility Models. We have considered three mobility models for carrying out experimental analysis: Manhattan grid, Random Walk & Random Way Point described in [9]. Manhattan grid uses Manhattan distance to compute distance between two nodes in the grid. Random Walk and Random way point follows a random path generated.

3.4 Rekeying

For updating the key pool for key redistribution, we need to update the value of prime factor q . This will lead to new key table for $q^2 + q + 1$ devices. This operation will be costly and can be used when scaling of the topology is required. So to minimize this cost, while designing the network we need to choose number of devices optimally. For example, if we require 50 devices, then $q=7$ will give keys for 57 devices. Hence, we are left with 7 unused rows in the key pool table. These 7 nodes can be used in updating the key-chains present in node identifier. We have not considered in-depth analysis of rekeying in this work.

4 Experimental Results

We performed our experiments of proposed work on Intel Core I7 CPU. We are using ConitkiOS along with Cooja Simulator for evaluation of model and different mobility scenarios of our proposed protocol.

4.1 Implementation

We have taken two values of prime q and the mobility models (Manhattan grid, Random Walk & Random Way Point) to perform simulations. For sensors we have considered cc2420 2.4 GHz IEEE 802.15.4 compliant RF transceiver. Topology comprises of $q^2 + q + 1$ devices with q mobile nodes & $q^2 + 1$ fixed nodes: Keys are distributed to 13 devices based on SBIBD design. Taking prime $q = 3$, gives a total of 13 devices that can be deployed. Out of $q^2 + q + 1$, q are having privilege with enabled mobility. Here, 3 devices are mobile and the rest 10 are deployed. When $q = 5$, gives a total of 31 devices can be deployed. Here, 5 devices are mobile and rest 26 are having fixed coordinates. Each device in the network has $q + 1$ i.e. 6 keys, distributed among 31 devices similar to 13 nodes from the key table.

4.2 Results

We discuss the results of performed simulations on proposed framework. In our simulation environment we have taken two primes $q = 3$ & 5 , resulting into 13 and 31 nodes. Similarly it can be done for next primes, $q = 7, 11, 13$ and so on. We have considered small and compact design to simulate the results of our proposed protocol.

1. Total nodes = $q^2 + q + 1$ & Keys per node = $q + 1$.
2. Energest values for Low Power Mode (LPM), CPU computation, Transmission and Listen mode.
3. Mobility: Manhattan Grid, Random Walk & Random WayPoint.

In this section we compare the result of introducing mobility in the aggregation process as proposed in the protocol. We compare the energy consumption of the sensor node and aggregator nodes. Energest value is used to measure the energy consumption in Low Power Mode (LPM), CPU computation, Transmission and Listen modes. Energest value does not give the power directly, it provides the required data in timer ticks. We record these values in an interval of 10 second for each nodes. Equation 2 is used to compute the total power consumption from energet reading.

$$Energy(in\ mW) = \frac{1.8 \times CPU + 0.545 \times LPM + 20 \times Rx + 17.7 \times Tx}{RTIMER_SECOND \times 0.33 \times 10} \quad (2)$$

RTIMER_SECOND for a sky mote is 37628. We compute the energy consumption in two different topologies. One consists of 3 mobile nodes and 10 sensor nodes ($q = 3$). And second the topology consists of 5 mobile node and 26 sensor satisfying the criteria of key distribution with $q = 5$. We compare the energy requirement of three different mobility model with static node placement in given topologies as given in Fig. 2.

We also compare energy saving across different mobility models in given topology. It can be observed that though in 13 node cluster, placement of the

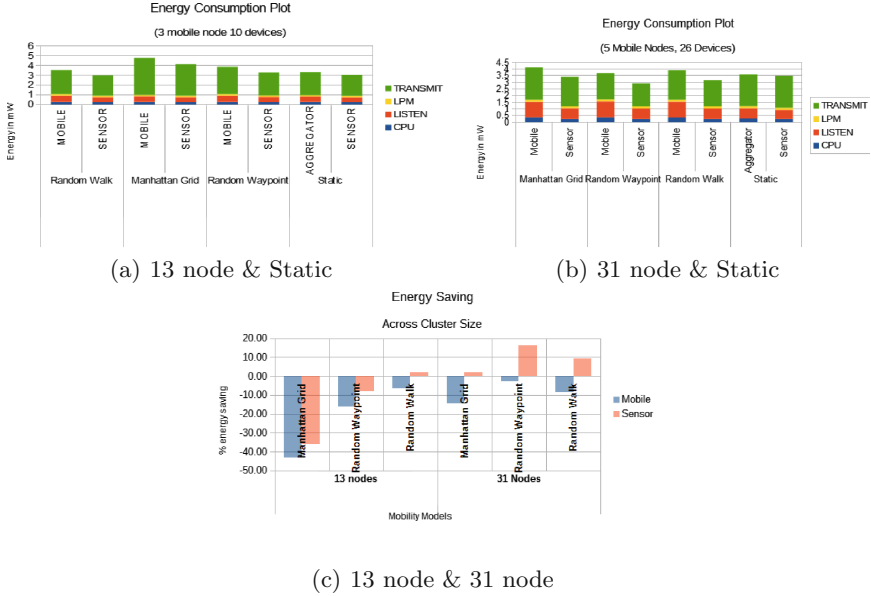


Fig. 2. Energy consumption plots

static aggregator node at random position performs better than mobility model, with high node significant energy saving can be obtained when the aggregator is mobile instead of static. Also the energy saving depends on the mobility model implemented. It can be observed that random walk model achieves maximum energy reduction compared to static aggregation. Comparison result is shown in Fig. 2(c).

We have compared our key management protocol with mobility with the schemes given by [2, 6, 10, 11]. From Fig. 2(a) we infer that, when the number of nodes increases, there is a fair amount of increase in the energy savings for network with mobile aggregator. Figure 2 summarizes the comparison of proposed protocol. We also observe that out of three different mobility models, Random WayPoint gave maximum energy savings for our proposed protocol on key management.

5 Conclusion

Our proposed solution has added a number of advantages over existing schemes. It requires less storage requirements because of combinatorial design and efficient key distribution. It also shows high Resilience against node capture attacks. Adding mobility for agents further adds efficiency and provides better services such as communication and better availability of information in the network model. And one major advantage of this protocol is that complexity of path-key establishment phase is eliminated. As a future work various key management

scheme with efficient rekeying techniques from WSN can be integrated to get valuable results. In a topology that can consist of many applications, we can have prime order q based applications. This will be hybrid design. Each application will have their own mobile nodes and key table designed as per the value of q . Routing algorithms can also be incorporated for better mobile features.

References

1. Sundmaeker, H., Guillemin, P., Friess, P., Woelfflé, S.: Vision and challenges for realising the Internet of Things. In: EUR-OP, vol. 20, no. 10 (2010)
2. Du, W., Deng, J., Han, Y.S., Varshney, P.K., Katz, J., Khalili, A.: A pairwise key predistribution scheme for wireless sensor networks. *ACM Trans. Inf. Syst. Secur. (TISSEC)* **8**(2), 228–258 (2005)
3. Pattanayak, A., Majhi, B.: Key predistribution schemes in distributed wireless sensor network using combinatorial designs revisited. *IACR Cryptol. ePrint Arch.* **2009**, 131 (2009)
4. Ruj, S., Nayak, A., Stojmenovic, I.: Pairwise and triple key distribution in wireless sensor networks with applications. *IEEE Trans. Comput.* **62**(11), 2224–2237 (2013)
5. Mitchell, C.J., Piper, F.C.: Key storage in secure networks. *Discrete Appl. Math.* **21**(3), 215–228 (1988)
6. Ruj, S., Roy, B.: Key predistribution using combinatorial designs for grid-group deployment scheme in wireless sensor networks. *ACM Trans. Sens. Netw. (TOSN)* **6**(1), 4 (2009)
7. Çamtepe, S.A., Yener, B.: Combinatorial design of key distribution mechanisms for wireless sensor networks. *IEEE/ACM Trans. Netw.* **15**(2), 346–358 (2007)
8. Street, A.P., Street, D.J.: *Combinatorics of Experimental Design*. Oxford University Press Inc., Oxford (1986)
9. Aschenbruck, N., Ernst, R., Gerhards-Padilla, E., Schwamborn, M.: BonnMotion: a mobility scenario generation and analysis tool. In: *Proceedings of the 3rd International Conference on Simulation Tools and Techniques (ICST)*, p. 51 (2010)
10. Somasundara, A.A., Ramamoorthy, A., Srivastava, M.B.: Mobile element scheduling for efficient data collection in wireless sensor networks with dynamic deadlines. In: *Proceedings of the 25th IEEE International Real-Time Systems Symposium*, pp. 296–305. *IEEE* (2004)
11. Blundo, C., De Santis, A., Herzberg, A., Kutten, S., Vaccaro, U., Yung, M.: Perfectly-secure key distribution for dynamic conferences. In: Brickell, E.F. (ed.) *CRYPTO 1992*. LNCS, vol. 740, pp. 471–486. Springer, Heidelberg (1993)

Cloud Resources Optimization for Air Pollution Monitoring Devices and Avoiding Post Pillar Problem

Parampreet Singh^(✉) and Pankaj Deep Kaur

Computer Science Department, Guru Nanak Dev University,
Regional Campus, Jalandhar, India
param.pasricha@gmail.com, pankajdeepkaur@gmail.com

Abstract. Cloud Computing is 21st century's precious gem that is revolutionizing the computing world. Cloud computing is progressively transforming the world through its wide applicability in diverse fields. One such field is environment monitoring. Today cloud computing is being utilized for monitoring the air pollution levels in association with different sensory devices and aid the ecologists around the globe to derive subtle ways to lower down its impact factor. But the major problem with such noble application is the elasticity factor of resource provision in cloud for handling the gargantuan amount of data that is generated by sensors. This elasticity cause troublesome to the service provider as the need of resources are very erratic and spontaneous. In this paper we present an algorithmic technique that attempts to quash this problem and provide a way to optimally allocate and utilize the resources. The evaluated simulation results reveals a very positive side and suggest an increase in utilization factor by 25 %–40 %.

Keywords: Cloud computing · Cloud computing resource optimization · Air pollution monitoring

1 Introduction

The modern era of 21st century encompasses a digital backbone. Fresh principles are entering our lives each and every day by use of technology. Things which were once dreams for our forefathers are now existing in reality and serving the basic human needs. Cloud computing is one such aspect that is propagating contemporary human civilization to a next level by its unmatched services. The fundamental definition of cloud computing is “the usage of computing resources over the internet”. Today cloud computing has acquired a serious level of popularity as compared to existing techniques. The chief contributing aspect to this substantial growth is the “Elasticity” behavior catered by the cloud computing to the user. This elasticity permits the user to increase or decrease the demand of computing resources at any instant of time as per their requirements. This allows the users to lower down the superfluous costs that are generally taxed on them if physical or personnel resources were deployed from their end. Now only usage fee is charged upon them and nothing else. Moreover the maintenance costs also lowers down by shifting to cloud. Cloud computing has

completely revolutionized the scenario of business computing. But the continuous evolution of this gem has now turned its application towards diverse fields. One such aspect where cloud computing has spread its wings is “Air Pollution Monitoring”. Currently, a lot of research work is being carried out by different researchers who are leaving no stone unturned to find an antidote to this perilous mess. Number of pollution monitoring devices are deployed by various organizations to carry out the surveillance work. The major problem associated with the assigned devices is that they generate huge amount of data which requires some serious computation. Now cloud computing can be efficient solution to this due to its elastic nature. But what turns out to be boon for one, sometimes turns out to be a bane for the other. The resources that are allocated by the service provider to the devices are sometimes overused and sometimes underused. The major influential factor that leads to this muddle is of improper management of resources. The provider has to bear huge losses for the underutilization part as unutilized resources are just eating up the power without providing any substantial benefits. The over utilization taxes the resources pretty brutally and leads to deterioration of the resources. Hence there is some serious need to address this issue and a dynamic approach is required which can find a blend to allocate appropriate amount of resources in such a manner that utilization factor the resources is always high and succor the provider with some good amount of benefits.

In this paper we have tried to adopt a completely different technique to increase the utility of the resources. This algorithmic technique named as “OCR-SAPM” (optimizing cloud resources to sate air pollution monitoring) endeavors to utilize the unused resources of peculiar host by adjusting the load on different hosts along with considering their capacities to handle the load. The future part is divided into 5 sections. Section 2 discusses the literature survey and analysis the work that has been carried out earlier by various authors to gain the best possible solution. Section 2.1 describes the problematic nature of the mentioned techniques. Section 2.2 discusses the major motivational force behind the work. Section 3 firstly describes the terminology required for the smooth understanding of the algorithm. This section further contains Sects. 3.1 and 3.2 which discusses the “Post and Pillar Problem” and OCR-SAPM algorithm respectively. The subsequent part the results and analysis part followed by conclusion respectively. At last all the references of different works are listed out without help of which this work wouldn’t have been possible.

2 Related Work

For a service provider, an effective management of resources is becoming a convoluted task as with the popularity of cloud computing. Thoughtful efforts are being made towards achieving the desired outcomes. The research under this is not fresh but had been there for quite a time. [2] Under his work provided an appropriate “MLE” method of estimating the resource requirement of the users under severe complexities of IAAS and E-Learning system. The practical implementations were carried out with the help of “Samsung School Solutions” to monitor the effectiveness of the algorithm. Few Additional Resources were always reserved in case of emergency requests. Nagpure in [3] employs a special load balancing server, whose sole purpose is to envisage the

upcoming demands. “Skewness” was used to determine the factors responsible for creation of chaos under resource optimization and how it can be dealt. [4] Provides a methodology to counter back and forth occurrence of resource demand using certain bound values. [1] Proposes a proficient technique named as “auto-scaling algorithm” to provide resource allocation for different workflow inputs. [5] Provides a technique to balance the load on the system by considering the response time associated with each task. The author first address the contemporary issues with other load balancing algorithms out there in the world. The algorithms predicts the response time for the incoming requests. In [6], author strives to publish a perfect difference between the static and dynamic techniques out there that helps to balance the load. It also caters a new elasticity policy to help the provider to cater needs in a better and appropriate manner. [7] Works for the multimedia cloud and helps the cloud industry by an algorithm for allocation of resources on “Near-Client-Datacenter” basis. The network analysis was taken into due consideration along with the allocation part in his work. The problem of catering sudden outburst in workload with efficient management and allocation of resources was handled in [8]. Here traffic burst was considered for different applications under different scenarios like using dataset of 1998 World Cup to find out the desired results. Saad and Babar in their taxonomy [9] tries to present the analysis of various categories under “Resource Management” which includes Energy-aware management, SLA-aware management and many more categories. A detailed comparative analysis was provided by the authors to process the fortes and flaws of different techniques. [10] Addresses a very critical, unmanageable problem of energy consumption of the data centers and tries to propose a holistic approach based on multi-criteria decision system. S. Manvi in his work [11] studies the various issues of one of the services of cloud computing namely IAAS and suggests various schemes out in the computing world that are being utilized to counterpoise the issue. An adaptive migration technique for various resources by using skewness and prediction methodology was provided by [12]. The next section explores a brief comparison between certain specified algorithms for specifying the problems that these algorithms are facing (Table 1).

Table 1. Brief comparative analysis between certain specified algorithms.

Algorithm name	Average based	Centrally monitored
Maximum Likelihood Estimation.	No	Yes
Skewness based Allocation System.	Yes	Yes
Auto-Scaling Algorithm.	No	Yes

2.1 Problem Description

Though the proposed works of all the above mentioned author is pretty impressive, but the major stage used for their deployment is either “Mean/Average” based or a

completely different “Central monitoring system” based whose purpose is to be a spectator and just monitor the other hosts for any adverse circumstances or for switching task if required. The problem with the latter is that deploying a complete system just for monitoring purpose is not valuable taking into effect that the contemporary systems are way more powerful and efficient. This “Patterned” or synchronous approach is not appropriate as the system can be utilized for some other useful computing and scientific work. Moreover, the problem with the former “Mean” method is that analysis are based on average of a parameter as compared to a bigger picture. As “Mean” can only lower the gap between the available and the optimal but cannot provide an optimal solution which eventually hampers the optimization of resources. Certain Algorithms also uses “Prediction” based methodology to assign resources which most of the times backfires. OCR-SAPM address these problems by switching from a “Patterned” approach to a “Non-Patterned” approach. Here there is no deployment of any central monitoring system, rather there is an integration of the existing systems.

2.2 Motivational Work

Whenever we talk about the advancements in the technology, one name is there that can't be left unnoticed. The name is “Google”. As we all are very well aware about the drastic effects that are being foreseen by the environmentalists regarding climate change due to snowballing pollution levels, Google has come upfront and took the initiative to put curtains to this mess. At the 11th Annual Clinton Global Annual Google declared his partnership with Aclima, a San-Francisco based sensor manufacturing organization, to utilize its street view cars to generate pollution maps [15] which would help the users using google maps to get instantaneous updates of the pollution levels of different areas. The Cars will be properly equipped with different sensors to monitor different pollutants like CO₂, NO₂, and Particulate Matter (PM) etc. The data collected by the sensory nodes is transferred to cloud systems, but before transmitting it, the values are amplified refined as per requirements to make a hassle free transfer. The cloud system analyzes the data and provide feedback as RED zones or GREEN zones according to severity of pollution level. Moreover, the data that has been collected is further made available to ENVIRONMENTALIST, public Government agencies etc. to use that data to cater certain useful measures that can help to put curtains to this ever growing hazardous problem. The following figure illustrates the appropriate stepwise process of the above mentioned scenario (Fig. 1).

Data generation by these sensory devices varies from moderate to gigantic level. This fitful generation makes data processing at the cloud side difficult in terms of resource allocation. Provider cannot simply predict the amount of resources that are required to be allocated. This unevenness of data depends upon the working area of the sensors. For instance, a crowded market place will generate an enormous amount of data after monitoring as compared to a residential area. Hence, our algorithm “OCR-SAPM” considers this chaos and delivers an efficient and effective algorithmic solution for the plight of provider. The next section will address the work that has been proposed under “OCR-SAPM”.

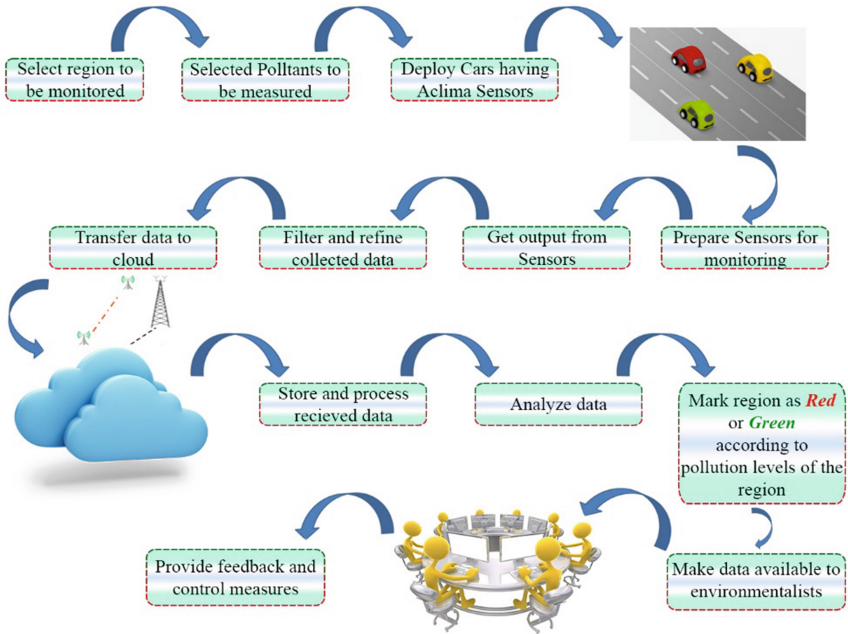


Fig. 1. An overview of general steps under specified Air pollution monitoring scenario.

3 Proposed Work

There is a definite set of terminology associated with OCR-SAPM that needs to be addressed before leaping into any subsequent part. The following table describes them all (Table 2).

Table 2. Terminology and Definition.

Terminology	Definition
Common_Tab_Map	This is a common mapping table that contains all the hosts that are available and number of resources they can serve.
CL_Data	Data transferred by monitoring device.
TOBO	The Top bound that is assigned to host
BOBO	The Bottom bound that is assigned to host.
HDL,HDLR_ID	Represents the host ID's that are available to aid the loaded host.
OL_Cond	Overloaded condition of host.
UL_Cond	Under loaded condition of host.
RAP	Resource availability proportion. i.e. amount in percentage of resources that are available at any instant.
MAX_RA	Maximum resource availability under a specific host.
CL_Data_Min	Data Item that entails minimum resources.
CL_Data_Max	Data Item that entails maximum resources.

3.1 Post and Pillar Effect

Before jumping into the pool of OCR-SAPM, there is a concept associated with BOBO and TOBO values that needs to be addressed. The concept is named as ‘‘Post and Pillar Effect’’. The concept is named after back and forth flow that occurs sometimes due to allocation and deallocation of load i.e. on increasing the load, the overload condition occurs, whereas on decreasing the load, the under load condition occurs. This is known as ‘‘Post and Pillar effect’’. To make sure that such situation never occurs, the values of BOBO and TOBO for each hosts are assigned using two methods which are discussed as under:

- Inert Allocation: Under this technique the value BOBO and TOBO are assigned in a static context. Here the TOBO is assigned higher to achieve a greater utilization factor. Now assigning a higher bound also increases the chances of underutilization. To counterpoise this problem we limit the value of BOBO as:

$$BOBO < \frac{TOBO}{2} \tag{1}$$

- Relative allocation: Under relative allocation the TOBO is settled first, but the value of BOBO is assigned by scrutinizing TOBO. Under this method, there is occurrence of frequent changes to BOBO value according to the load on the system. Under this BOBO is given in terms of TOBO as:

$$BOBO < \frac{(MAX_RA - RAP) - (MAX_RA - RAP')}{MAX_RA - RAP} . TOBO \tag{2}$$

- Declaration: Decreasing load does not simultaneously generates under loading situation if BOBO and TOBO are assigned as stated above.
- Evidence: Initial total load before decrease in load can be given as:

$$(MAX_RA - RAP) . BOBO \tag{3}$$

Now after the load is decreased, the new total load according to (3) can be given as,

$$\frac{(MAX_RA - RAP)}{(MAX_RA - RAP) - (MAX_RA - RAP')} . BOBO$$

Were RAP’ is the new proportion of available resources. To make sure that on increasing load, overload condition does not occurs, we must set

$$TOBO > \frac{(MAX_RA - RAP)}{(MAX_RA - RAP) - (MAX_RA - RAP')} . BOBO$$

Or

$$BOBO < \frac{(MAX_RA - RAP) - (MAX_RA - RAP')}{MAX_RA - RAP} \cdot TOBO \quad (4)$$

- *Declaration:* Increasing load does not simultaneously generates over loading situation if BOBO and TOBO are assigned as stated above.
- *Evidence:* Initial total load before increase in load can be given as:

$$(MAX_RA - RAP) \cdot TOBO \quad (5)$$

Now after the load is increased, the new total load according to (5) can be given as,

$$\frac{(MAX_RA - RAP)}{(MAX_RA - RAP) + (MAX_RA - RAP')} \cdot TOBO$$

To make sure that on removing, under load condition does not occurs, we must set,

$$BOBO < \frac{(MAX_RA - RAP)}{(MAX_RA - RAP) + (MAX_RA - RAP')} \cdot TOBO \quad (6)$$

Now from (4) and (6),

$$\frac{(MAX_RA - RAP)}{(MAX_RA - RAP) + (MAX_RA - RAP')} \cdot TOBO > \frac{(MAX_RA - RAP) - (MAX_RA - RAP')}{MAX_RA - RAP} \cdot TOBO > BOBO$$

Hence to make sure that post pillar problem never occurs, set

$$BOBO < \frac{(MAX_RA - RAP) - (MAX_RA - RAP')}{MAX_RA - RAP} \cdot TOBO$$

3.2 OCR-SAPM Algorithm

The modern industry computers have become so powerful that handling hundreds of users at a particular instance of time is just like a piece of cake. Appointing a scrutinizer over these powerful machines is just a wastage of computing power which can be labeled with some other scientific and beneficial work. In order to put an end to this approach, OCR-SAPM algorithm uses “entangled” approach as stated in diagram. Under this each computing node has a copy of Common_Tab_Map table which gets apprised by each and every other node once the resource are manipulated by incoming or outgoing of Data that requires resources for processing. As below in Fig. 3, as soon as the resources are consumed from node A, all other nodes are apprised by it from A and the Common_Tab_Map is refreshed by all (Fig. 2).

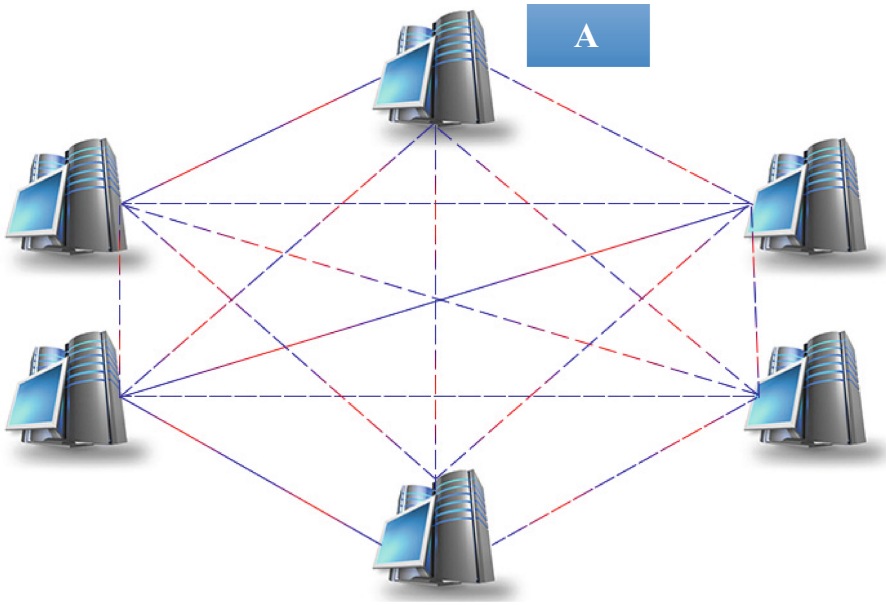


Fig. 2. Entangled Approach for Common_Tab_Map.

The Common_Tab_Map and RAP performs the core functionality of the algorithm. The successful working of algorithm depends upon these two parameters. The greater value of RAP signifies that there is substantial amount of underutilization is prevailing among hosts and vice-versa. Genesis of OCR-SAPM contains a module as Regular_Optimize() which first gets introduced to the incoming data. The incoming CL_data is simply assigned to Host [x] presuming it to be capable enough to provide service. Now, Common_Tab_Map is revised for the existing resources and is sorted out in an aspiring manner. Check_HDL() now joins the party and looks for the ideal host that satisfies two conditions namely, Fulfill Data resource requirement and Improve resource utilization. These conditions are surveyed from Common_Tab_Map in such a manner that the specific host has ample RAP value (minimum) available to provide resource supply without fracturing its TOBO. Now if there is an availability, then the ID of that host is stored in HDL and that CL_Data is assigned to it for processing. Common_Tab_Map is again restructured via sorting.

But, this simple thing can become problematic if there is not even a single host to provide service and the data arrived sticks to the existing host which also unfortunately becomes overloaded (Monitor_OL()) and demands immediate offloading. If this kind of situation arrives then OCR-SAPM transforms itself from “intra host” where exchange was from one host to another of a machine to “inter host” where now it will be carried out between hosts under different machines. OL_OCR_SAPM() is asked for a solution. The OL_OCR_SAPM() first derives the TOBO and BOBO values and along with this, a list named as “CL_Data_List” is prepared for the current data that is under processing on that host. Now in order to fully optimize the resource usage, it is not appropriate to assign any ith CL_Data. To achieve the best results, The CL_Data that requires

minimum amount of resources is selected and is stored under, `CL_Data_Min`. To assign this under a suitable Host, `get_HDLR_ID_Common_Tab_Map_OL()` is summoned which scrutinizes the `Common_Tab_Map` for the host (other than current) that can successfully satisfy optimization conditions as well fulfil resource catering service for the `CL_Data`. The RAP again plays an important role here as selection of host is based on the minimal level i.e. the host having minimum RAP is preferred, provided it does not tax its resources upon selection. If under extremely unfavorable circumstances, a situation comes when there is no host that can cater the service, then we have to apply new nodes to lower the burden on current host.

Till now our central theme for optimization was restricted to overloading situation. But optimization of resources doesn't always associated with overloading. Sometimes the situation of under loading can also severely obstruct the optimization. Most of the contemporary algorithm doesn't take this situation into their radar for optimization. OCR-SAPM challenges this problem and provides an efficient solution. `MonitorUL()` raises an under load condition whenever the RAP becomes greater than BOBO i.e. the maximum resources are unwaged. `UL_OCR_SAPM()` is called which works likewise as `OL_OCR_SAPM()` but instead of selecting the `CL_Data_Min`, it chooses `CL_Data_Max` i.e. the supreme resource yearning is chosen. On the similar guidelines `get_HDLR_ID_Common_Tab_Map_UL()` is invoked and on accomplishment, the unrequired freed hosts are Signed-Off to provide resource provider ample and high yielding amount of benefits. The following section discusses the complete pseudo-code OCR-SAPM algorithm (Tables 3, 4, 5, 6, 7, 8, 9 and 10).

3.3 OCR-SAPM Algorithm

Table 3. Algorithm: `Regular_Optimize(Common_Tab_Map)`.

```

1: Host[x] ← CL_Data[x]; //Host[x] ∈ machine[x].
2: Refresh and Sort Common_Tab_Map;
3: HDL ← Check_HDL(Common_Tab_Map, Resources[CL_Data]);
4: if (HDL) {
    Host [HDL] ← CL_Data[x] and repeat 2: only;
}
5: Inspect if Overloaded/Under loaded;
6: If Overloaded, assign OL_Cond=true;
7: If Under loaded, assign UL_Cond=true;
8: if (OL_Cond==true) {
    Call OL_OCR_SAPM();
}
9: else if (UL_Cond==true) {
    call UL_OCR_SAPM();
}
10: More CL_Data, Goto 1;

```

Table 4. Algorithm: Check_HDL (Common_Tab_Map, Resources).

```

1: for (count ← 0 to Size(Common_Tab_Map)) {
    If (TOBO[count] > RAP[count] ≥ Resources) {
        Return count;
    }
    Break; }

```

Table 5. Algorithm: OL_OCR_SAPM (Host [x], Common_Tab_Map).

```

//Host[x] signals an overloaded condition.
// CL_Data_List is initialized as an empty list.
1: TOBO[x] ← Host[x];
   BOBO[x] ← Host[x];
2: For Host[x] {
   CL_Data_List ← Host[x]; //Preparing list for data processing under a
   specific host.
}
//Step 3 & 4 fetches the CL_Data_Min
3: CL_Data_Min ← CL_Data_List[0];
4: For (count ← 0 to Size (CL_Data_List)) {
   If (CL_Data_List [count] < CL_Data_Min) {
       CL_Data_Min ← CL_Data_List [count];
   }
}
5: Sort (Common_Tab_Map);
6: for (j ← 0 to Size(Common_Tab_Map)) {
   HDLR_ID ← get_HDLR_ID_Common_Tab_Map_OL (Common_Tab_Map, Resources
   [CL_Data_Min]);
}
7: if (HDLR_ID) {
   Host[HDLR_ID] ← CL_Data_Min;
   Remove CL_Data_Min from Host[x];
   Refresh RAP[x] in Common_Tab_Map;
   Refresh RAP[HDLR_ID] in Common_Tab_Map;
} else {
   Assign_New_Host ← CL_Data_Min;
}
8: if (Host[x] == Overloaded) {
   Go to 2;
} else {
   Return;
}

```

Table 6. Algorithm: get_HDLR_ID_Common_Tab_Map_OL (Common_Tab_Map,Resources).

```

1: For (Every hosts[j] in Common_Tab_Map) {
  If (TOBO [j] >RAP [j] > =Resources){
    HOVL← j;//local variable for storage
    break;
  } else {
    continue;
  }
}
2: If (HOVL ==null) {
  HOVL← -1;
}
return HOVL;}

```

Table 7. Algorithm: MonitorUL(Common_Tab_Map).

```

1: If (RAP [x] > BOBO [x]) {
  Return "under loaded";
}

```

Table 8. Algorithm: MonitorOL(Common_Tab_Map).

```

1: If (RAP [x] <TOBO [x]) {
  Return "overloaded";
}

```

Table 9. Algorithm: UL_OCR_SAPM (Host [x], Common_Tab_Map).

```

//Host[x] signals an under load condition.
// CL_Data_List is initialized as empty list.
1: TOBO[x]←Host[x]; //Fetching TOBO for host[x].
BOBO[x]← Host[x]; //Fetching BOBO for host[x].
2: for Host[x] {
  CL_Data_List ← Host[x]; //Preparing list for data processing under a
  specific host.
}

3: CL_Data_Max← CL_Data_List [0];
4: For (count=0 to Size (CL_Data_List)) {
  If (CL_Data_List [count]> CL_Data_Max) {
    CL_Data_Max ← CL_Data_List [count];
  }
}
5: Sort (HashMap);
6: for (j←0 to Size(HashMap)) {
  HDLR_ID ←
  getHDLR_ID_Common_Tab_Map_UL(Common_Tab_Map,Resources[CL_Data_Max]);
}
7: if (HDLR_ID) {
  Host[HDLR_ID] ← CL_Data_Max;
  Remove CL_Data_Max from Host[x];
  Refresh RAP[x] in Common_Tab_Map;
  Refresh RAP[HDLR_ID] in Common_Tab_Map;
} else {
  Assign_New_Host ← CL_Data_Max;
}
8: if (Host[x]==Under loaded) {
  Go to 2:
} else if ((Size)CL_Data[Host[x]]==0){
  Host[x]← Sign-Off;
  Return;
}

```

Table 10. Algorithm: getHDLR_ID_Common_Tab_Map_UL (Common_Tab_Map, Resources).

```

1: For (Every hosts[j] in Common_Tab_Map) {
  If (TOBO [j] >RAP [j] >BOBO [j] && Resources<=RAP[j]) {
    HUDL←j; //local variable for storage.
    break;
  } else {
    HUDL←-1;
  }
}
return HUDL;
}

```


Table 11. Different values of different pollutants received from different areas. (In AQI).

PM 2.5	O ₃	NO ₂	SO ₂
184	8	12	4
168	16	4	2
185	10	2	1
164	4	3	3
205	7	12	1
107	4	17	1
159	2	20	2
177	4	19	4
145	5	17	2

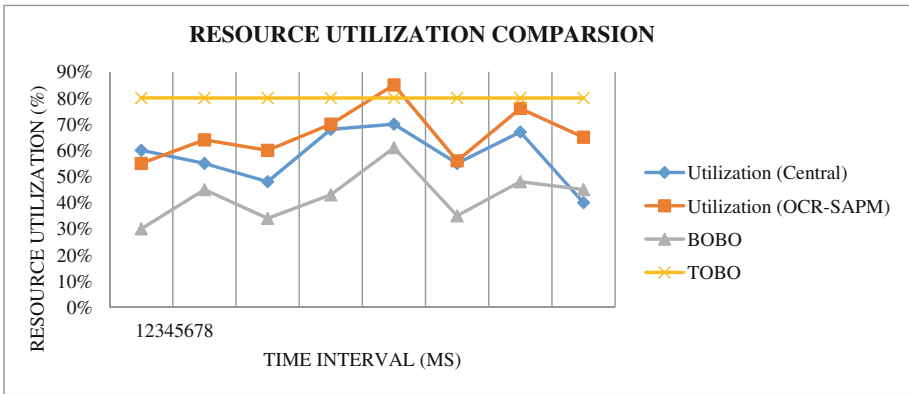


Fig. 3. Resource utilization comparison. (Color figure online)

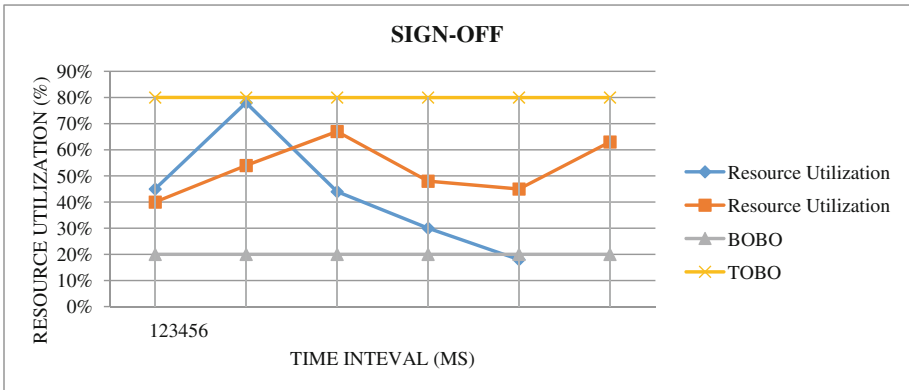


Fig. 4. Sign-Off. (Color figure online)

4 Results and Analysis

Under this section a detailed graphical analysis is presented containing the percentage increase in the optimization levels using OCR-SAPM algorithm. The Sample data from [13, 14] was used to calculating the results. Pollutants such as Particulate Matter (PM), Ozone (O₃), Nitrogen Dioxide (NO₂) and Sulfur Dioxide (SO₂) were taken into consideration as shown in Table 11.

Simulation experiments were carried out using CloudSim 3.0.3. The inception load was assigned to different hosts on different machines using the random data generator function and further the run was augmented by the dataset for evaluation. Comparative analysis were performed using different number of task under different time intervals as described in Figs. 3 and 4. Preliminary analysis clearly reveals that there is significant resource utilization improvement if we switch to the given algorithm.

The simulation results clearly indicates the increase in utilization factor of the resources. Yellow and Grey lines indicate the TOBO and BOBO respectively Here TOBO is static, but BOBO keeps on adjusting as per the load. Blue and Orange line signifies the improvement factor discovered with the use of OCR-SAPM as compared to centrally appointed systems.

Figure 4 address the site of immediate load shrinkage which leads to increase in the RAP and as soon as RAP goes greater than BOBO, The load of that host is transferred to other host that can service its needs and also remain under TOBO and the current host is logged off.

5 Conclusion

Air pollution control is one of the utmost important agendas that needs some serious addressing. This paper proposes a dynamic resource optimization technique that caters its services to air pollution monitoring devices for efficient working. The Experimental analysis does reveal that there is significant improvement achieved in utilization part if OCR-SAPM is deployed. The algorithm benefits the resource providing authority with some serious reductions in cost factors by signing off certain unused resources and effectively utilizing the deployed ones. The future work can include to further optimize the algorithm to adapt network fluctuations and unavoidable circumstances related to optimization.

References

1. Fakhfakh, F., Kacem H.H., Kacem A.H.: A Provisioning approach of cloud resources for dynamic workflows. In: 2015 IEEE 8th International Conference on Cloud Computing, New York City (2015)
2. Koch, F., Assuncao, D.M., Cardonha, C., Netto, M.A.S.: Optimizing resource costs of cloud computing for education. *Future Generation Computer Systems* **55**, 473–479 (2015). Elsevier

3. Nagpure M.B., Dahiwale P., Marbate P.: An efficient dynamic resource allocation strategy for VM environment in cloud. In: 2015 International Conference on Pervasive Computing (ICPC), Pune, pp. 1–5 (2015)
4. Khatua, S., Manna, M.M., Mukherjee, N.: Prediction-based instant resource provisioning for cloud applications. In: 2014 IEEE/ACM 7th International Conference on Utility and Cloud Computing (UCC), London, pp. 597–602 (2014)
5. Sharma, A., Peddoju, S.K.: Response time based load balancing in cloud computing. In: 2014 International Conference on Control, Instrumentation, Communication and Computational Technologies (ICCICCT), Kanyakumari, pp. 1287–1293 (2014)
6. Aslazandeh, S., Chaczko, Z., Chiu, C.: Cloud computing-the effect of generalized spring tensor algorithm on load balancing. In: 2014 Asia-Pacific Conference on Computer Aided System Engineering (APCASE), South Kuta, pp. 5–8 (2014)
7. Gong, W., Chen, Z., Yan, J., Qianjun, S.: An optimal VM resource allocation for near-client-datacenter for multimedia cloud. In: 2014 Sixth International Conference on Ubiquitous and Future Networks (ICUFN), Shanghai, pp. 249–254 (2014)
8. Zhang, Q., Chen, H., Shen, Y., Ma, S., Lu, H.: Optimization of virtual resource management for cloud applications to cope with traffic burst. *Future Gener. Comput. Syst.* **58**, 42–55 (2016)
9. Mustafa, S., Nazir, B., Hayat, A., Khan, A.R., Madani, S.A.: Resource management in cloud computing: taxonomy, prospects, and challenges. *Comput. Electr. Eng.* **47**, 186–203 (2015)
10. Arianyan, E., Taheri, H., Sharifian, S.: Novel energy and SLA efficient resource management heuristics for consolidation of virtual machines in cloud data centers. *Comput. Electr. Eng.* **47**, 222–240 (2015)
11. Manvi, S.S., Shyam, G.K.: Resource management for Infrastructure as a Service (IaaS) in cloud computing: a survey. *J. Netw. Comput. Appl.* **41**, 424–440 (2014)
12. Ashalatha R., Agarkhed J.: Dynamic load balancing methods for resource optimization in cloud computing environment. In: 2015 Annual IEEE India Conference (INDICON), New Delhi, pp. 1–6 (2015)
13. Air Quality Forecasting in Northern India (2014). <http://aqicn.org/faq/2016-02-28/air-quality-forecasting-in-northern-india/>
14. Air Pollution in India: Real-time Air Quality Index Visual Map. <http://aqicn.org/map/india/#@g/26.8439/77.5961/5z>
15. Schiffman R.: Air pollution – live online. In: *New Scientist*, vol. 228, pp. 20

Credibility Assessment of Public Pages over Facebook

Himanshi Agrawal and Rishabh Kaushal^(✉)

Department of Information Technology,
Indira Gandhi Delhi Technical University for Women, Delhi, India
hagarwal.281@gmail.com, rishabh.kaushal@gmail.com

Abstract. With the growing use of online social media and presence of users on many such platforms, their interaction with social networks is huge. They are free to spread wrong information without any accuracy, integrity and authenticity checkpoints masquerading as legitimate content. All these wrong, unrelated, unwanted, manipulated information are distributed for some hidden reasons. Even, their distribution network is not limited to one social media platform. Sometimes, they use the social network as a market place either for advertising, promotion of particular website, product and an application. But these advertisements do not provide any incentive to Facebook as this content is just spam, irrelevant for Facebook. Dissemination of unwanted, unrelated information has become a huge problem not only on blog, discussion forum but also on online social network like Facebook. Due to lack of marking over content posted, this information become online and reader has no barometer to check either the credibility of commenter or poster or the credibility of facts. To address these issues, we have derived an equation to weigh the credibility of public pages and applied machine learning algorithms (MLA) over collected data to validate our prediction.

Keywords: Credibility · Online social media · Unrelated content · Machine learning algorithms · Public pages · Facebook

1 Introduction

Among the various social media platforms, Facebook is very popular. While there are many advantages that are offered by online social media, there are few issues faced as well. In this virtually connected world of online social media platforms, many aspects of our social interactions are disguised by garb of anonymity. With whom we are on chat, what kind of people are sending us emails, to whom we are friends on Facebook, Twitter, Google+ and most importantly our decisions, opinions are influenced or diverted by what kind of misleading information is diffusing over social space. In spite of thousands of innovation and policy implementation, fraudulent users target the information that is available for public view on online social media platforms like Facebook, Twitter. These users indulge in fraudulent activities like spreading rumors, advertising content for promotion

of their business, etc. On public pages of Facebook, users that are not allowed to post typically contribute by commenting over the posts. It is often observed that users put unrelated and unwanted comments over multiple posts, multiple pages and in very limited duration. Such misplaced spam comments with this information can misguide the other users. In our work, we focus on this problem and assign credibility to these facebook pages taking into account the spam comment count out of the total comment count and fake user count out of total user count. As an extension, we performed prediction over collected dataset by using three conventional classifiers namely J48, Naive Bayes and Random Forest to take comment level, user level, userfeed level and page level features into consideration.

Key contributions of our work are as follows.

- We have used a comprehensive set of features taking into account all three dimensions namely message, source and media to assess credibility over Facebook public pages.
- We have taken care of *hinglish* text in comments.
- We proposed a metric to assess the credibility taking spam comment count over total comments count and fake user count over total users count into account and calculated score for public pages.

Further, paper is structured as follows. Related Work is elaborated in Sect. 2. Our proposed work and its results and observations have been described in Sect. 3 and Sect. 4, respectively. Conclusion and Future Work is discussed in Sect. 5.

2 Related Work

The problem of unsolicited content no longer restricted to blogs and discussion forums but has marked its presence over online social media with their offensive footprints. Along with this, other derivatives challenges have emerged like credibility of message content, publisher of content and media, where these activities are taking place, etc. Following are some previous works that address both base and derived issues.

2.1 Unrelated Spam Content

Wang et al. [1] have categorized diversionary comments into five types based on their observations and propose an effective framework to identify and flag them. They have also conducted a user study to verify the effect of identifying diversionary comments by asking the following question “Assume that you hold interest in the post discussion topic, is this comment of interest to you?” [1] In their proposed work, they compute the relatedness between a comment and the post content, and the relatedness between a comment and its reply-to comment, which involves co reference resolution, extraction from Wikipedia, and topic modeling. They have evaluated 4,179 comments from Digg and Reddit. Dewan and Kumaraguru [2] classified the unsolicited malicious content generated

during news making events. For classification, an extensive feature set was taken comprising of 42 features. They had provided REST API and browser plug-in as well. URLs were extracted from the post and then those URLs were visited using Python request or LongURL API to remove the invalid URLs to find the final destination URL of URL mentioned in the post. Each URL was passed to six blacklist lookups e.g. Google SafeBrowsing, PhishTank etc. Abu-Nimeh et al. [3] assessed the prevalence of malicious and spam posts in Facebook. They have analyzed more than half a million posts with the help of Defensio, a Facebook application that protects users from such content as well as filters profanity and blocks URL categories. They surveyed the temporal and network-level properties of those posts containing URLs that Defensio had determined to be malicious or spam. They have concluded by their research that much more research is needed to gain a better grasp of the true extent and nature of security threats in online social networks.

2.2 Credibility Assessment of Source, Content and Media

Metzger et al. [4] has defined *“credibility is defined as the quality of being trustworthy. In communication research; information credibility has three parts, message credibility, source credibility, and media credibility.”* Thai et al. [5] have worked to identify the origin point of misinformation on OSN platform. They had studied k-suspector problem which works for identification of the top k most suspected origin point of misinformation. For this, they had proposed two efficient approaches, ranking-based and optimization-based algorithms. They had applied their approaches on differently structured networks. Thus their experiments and observations show that their proposed approach is very helpful to discover the origins of misinformation up to 80% accuracy and hence increase the solidity of facts sharing on OSN. Lin et al. [6] have identified the need of a framework for all online social media to analyze consequences for the dissimilar practices of posting on content generated by user over different OSN workspace. This discovery of dissimilar practice had become prior knowledge for ensemble classifier to measure the quality of content. The proposed approach had shown good results over different OSN like Slashdot and Apple discussion forum. Ismail and Latif [7] have tried to address the issues like uncertainty sources of it and last but not the least is quality of content published over online social network. Basically, they had arranged a questionnaire survey to get an outline of social network trends in Association of Southeast Asian Nations (ASEAN) countries and in Malaysia particularly. Through this, they had identified features like credibility of publishers, quality of information they shared and lack of continuity in online content. For the analysis and proof of the framework, they had applied multiple regression method and from the results they had concluded that lack of continuity of online contents create more uncertainty among users in comparison to lack of quality of information and even the publisher credibility. AlMansour et al. [8] had examined the different organizations for credibility of information based on methodologies and parameters used and classify Twitter surveys based on parameters used for credibility assessment. They had brought out a model for

assessment of credibility in different context and will help Arab people. Abbasi and Liu [9] have proposed a CredRank algorithm to examine the user credibility in online network based on the users behavior. In their research work, they had detected arranged behavior and mark that users less credibility score who are involved in these actions. Their proposed work will help to identify those personnel who have their accounts on multiple social networks and distribute wrong information using those accounts and ranking algorithm will help to save from rumors trap, bogus product feedbacks. For better result, they had suggested that all three; messages, source and media credibility should be taken into consideration. Kang [10] have proposed 14-components evaluation for credibility assessment of blog. They had taken two dimensions source and content for credibility assessment. Based on the observation results, authority and reliability were highly effective factor for blogger credibility and accuracy and focus are best indicators of message credibility. The proposed methodology is helpful to examine behavior effects on consumers of blog information.

From the study of above researchers work over OSN workspace, it is evident that to measure credibility, it is important to address all three dimensions message, source and media. In our proposed work, we have tried to include all three addressing comment, user and Facebook public pages and along with this validate our results using Machine learning algorithms in Weka.

3 Proposed Approach

In this section, we propose an approach to quantify the scope of distribution of unrelated content over public pages posts and to label each public page with a credibility score. We apply machine learning algorithms using Weka. Figure 1 shows the workflow of our proposed approach. Every step of flow chart is described in detail in the following subsections.

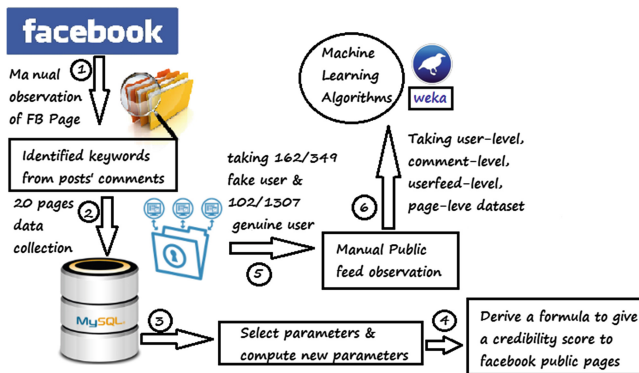


Fig. 1. Proposed approach

3.1 Manual Observation of Public Pages

We manually observed 56 public pages over Facebook. Out of them, advertisements activities were detected on 40 pages where it was found that users are placing advertisement comments which are unrelated to posts on those public pages. Even URLs are shared with text. Spammers are posting same comment over multiple pages. We have identified some patterns in comments for data collection. Here we have shown one such public page’s post in Fig. 2 as an example of our problem:



Fig. 2. Unrelated content over Wikipedia

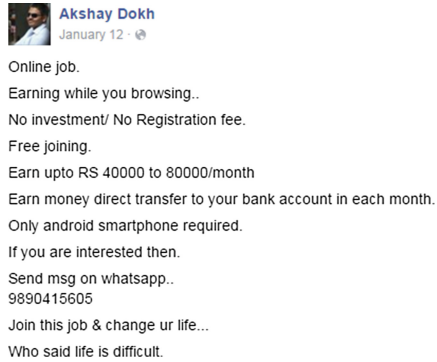


Fig. 3. Public feed example

3.2 Data Collection with and Without Identified Keywords

After keyword identification, we collected two types of comments. One type of comments contain those identified keywords and second type do not contain those keywords using Facebook Graph API v2.5 and stored all collected attributes for posts and comments in MySQL database. As some of comments are in *hinglish*, so for translation, we used Text Blob. Amount of data collected is shown in Table 1.

Identified Keyword Patterns in Comments: Here in Table 2, we have listed those keywords observed manually and taken into consideration during data collection:

Table 1. Data set with and without keywords

Data collection	#Pages	#Posts	#Posts per Page	#Comments	Duration
With keywords	20	1997	100	1523	Jan-Feb, 2016
Without keywords	20	1997	100	1626	Jan-March, 2016

Table 2. List of identified keywords

Identified keywords patterns
Good knowledge of the Internet, spiritual astrologer,
Unemployment in India, All World Famous Astrologer Tulsi Das,
Champcash, Many way to earn unlimited, play store,
Earntalktime, Android mobile, Minimum Recharge,
SPONSOR ID, Just download and install 8-9 Apps,
Level income, ONLINE JOB Digital India, Investment
Part time job, Online Home Based Job, Friends Refers Someone

3.3 Public Feed Observation of Fake and Genuine Users

To strengthen our results, we also analyzed the public feeds of both users who have posted the comments having identified keywords and those who have not. Observations of fake user's public feeds are shown in Table 3.

Table 3. Observations of public feed dataset of fake user

Observed points over fake user	Count
No of users with advertisement comments	349
No of users for whom public feeds are observed	162
Users commented present on more than one page	98 [28.08 %]

Observations of genuine users' public feed are shown in Table 4.

Table 4. Observations of public feed dataset of genuine user

Observed points over genuine user	Count
No of users with genuine comments	1307
No of users for whom public feeds are observed	102
Users commented present on more than one page	28 [2.14 %]

3.4 Selected Parameters

After data collection and manual observation of public feed of both fake and genuine users, we selected some parameters from our collected dataset and calculated some new parameters. All selected and calculated features on all levels are shown in following tables. Table 5 shows the features of comment.

Table 5. Comment level features

Features	Description
Is_Fake	Does comment contain spam keywords or not
Has_URL	Comment containing URL or not

All responses to users' comments like reply count, like count and other features like total comments per day by user, total urls in users' comment etc. are shown in Table 6.

Table 6. User level features

Features	Description
Total_Cmtcount	Total comment posted by user
Total_Likecount	Total Likes on their Comments
Total_Urlcount	URL posted In the comments
Totalcmt_Perday	Count of Comment posted by each user per day
Total_Replycount	Total reply to their comments
Avg_Likecount	Number of total like count over total comment count
Avg_Replaycount	Number of total reply count over total comment count
Avg_Urlcount	Number of total url count over total comment count

Amount of spam comments with fake users and genuine comments with genuine users on Facebook pages are shown in Table 7.

Table 7. Page level features

Features	Description
Totalcmt_Posted	Cumulative comment count on all collected posts
Spam_Cmt	Total number of Spam comment out of total comments
Genuine_Cmt	Cumulative genuine comment count on all collected posts
URL_Count	Cumulative url count on all collected posts
Fakeuser_Count	Cumulative fake user count on all collected posts
Genuineuser_Count	Cumulative genuine user count on all collected posts

Table 8 shows the details of users' feeds which are observed manually:

Table 8. User feed level features

Features	Formulas
Posted_Comment_Count	Total comment posted by users
Comment_Type	What identified keyword, comment contain or not contain
IS_PublicFeed	Does user contain such unrelated content in their feed or not
Active_On_Facebook	Whether user is active on Facebook or not
Feed_Count	Total count of such spam feed
CmtCount_OnFeed	Total comment posted on such spam feed
LikeCount_OnFeed	Total like count on such spam feed
ShareCount_OnFeed	Total share count on such spam feed
DiffPageCount	Total no of public pages on which user posted comment

3.5 Formulation of Credibility Score for Public Pages

We propose a formulation (metric) to give credibility score to Facebook public pages. The algorithm takes as input all the 20 pages in the list PL and keywords list $Keylist$ based on which comments of one type are separated from others. It outputs corresponding *Credibility* score.

Algorithm. Find-Credibility-Score ($PL, Keylist$)

```

1: ScalingFactor  $\leftarrow$  10
2: for all  $Page_i \in PL$  do
3:   spamcmtCount  $\leftarrow$  0
4:   genuinecmtCount  $\leftarrow$  0
5:   fakeuserCount  $\leftarrow$  0
6:   genuineuserCount  $\leftarrow$  0
7:    $Post_i \leftarrow getPosts(Page_i)$ 
8:   for all  $p_{ij} \in Post_i$  do
9:      $Comment_{ij} \leftarrow getComments(p_{ij})$ 
10:     $TotalComment \leftarrow getLength(Comment_{ij})$ 
11:    for all  $c_k \in Comment_{ij}$  do
12:      if any (word in  $c_k$  for word in  $Keylist$ ) then
13:        spamcmtCount  $\leftarrow$  spamcmtCount + 1
14:        fakeuserCount  $\leftarrow$  fakeuserCount + 1
15:      else
16:        genuinecmtCount  $\leftarrow$  genuinecmtCount + 1
17:        genuineuserCount  $\leftarrow$  genuineuserCount + 1
18:      end if
19:    end for
20:  end for
21:   $Score(Page_i) \leftarrow \left( \frac{spamcmtCount}{TotalComment} + \frac{fakeuserCount}{fakeuserCount + genuineuserCount} \right) * ScalingFactor$ 
22: end for

```

Score less than and equal to 0 shows page is very high credible and score value between 16 to 20 shows page is very low credible. Other labels like high, medium and low lie between these ranges.

4 Results and Observations

After applying credibility formula over pages data set, public pages are labeled as “Very High, “High, “Medium, “Low and “very Low credibility as shown in the following Table 9.

Table 9. Credibility label for public pages

Credibility score	Label	For pages
0 and <0	Very high	Star One, Fox, HBO India, Star World
1–5	High	Life OK, Indiaforum.com , Channel [v] India
6–10	Medium	And TV, Zee Cinema, sony entertainment television, Zee business, Etc Bollywood, Star Plus, National Geograhic, Colors TV
11–15	Low	Star Gold, Bindass, 9XM, Zoom TV
16–20	Very low	Sony Max

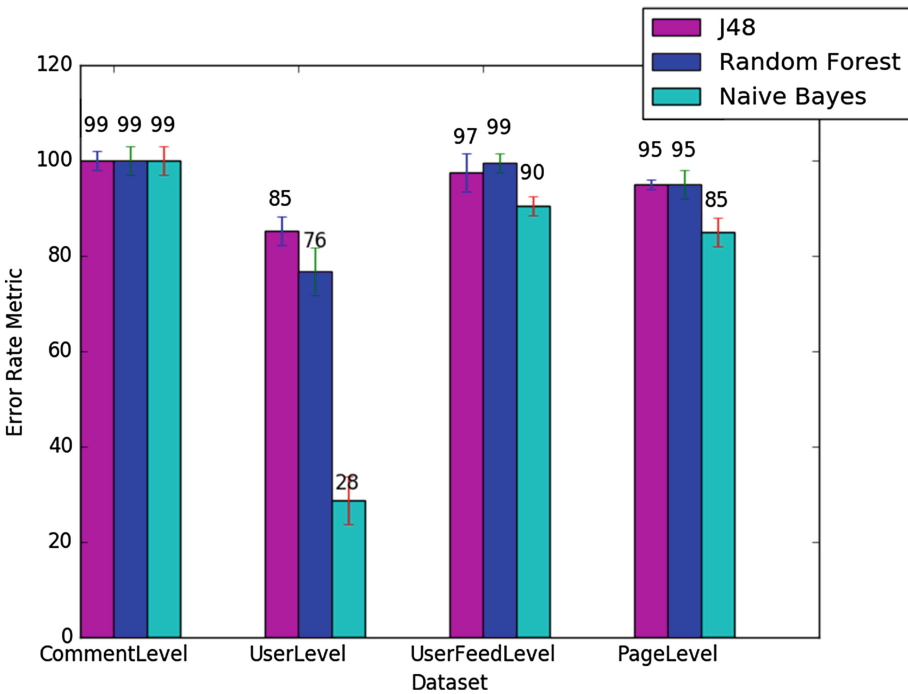


Fig. 4. Experimental result after execution of Machine Learning Algorithms

Using comment level, user level, userfeed level and page level features, we apply classifiers based on three machine learning algorithms namely J48, Naive Bayes and Random Forest using Weka to test our training dataset. Here we have shown the classification accuracy resulted from this experiment in the following graph.

4.1 Other Observations

Along with the above results, we have listed some of users (out of total 337 users) who had posted the same comments over different public pages in Table 10.

Table 10. Commenter on different public pages

User	On_diff_page
Abhishank Goyal	AndTV, ETC Bollywood, Bindass, Sony MAX, Zoom TV
Ravi Sharma Khajuria	AndTV, STAR Gold, Sony Entertainment Television, 9XM, National Geographic Channel
Vanita Gore	AndTV, STAR Gold, Sony Entertainment Television, ETC Bollywood, India-Forums.com , 9XM, Sony MAX, STAR Plus, Zoom TV
Anu Gaurav Gupta	ZEE Cinema, STAR Gold, Sony Entertainment Television, STAR Plus, Zoom TV
Atul Jain	Sony Entertainment Television, 9XM, Sony MAX, STAR Plus, Zoom TV

5 Conclusion and Future Work

Unsolicited and unrelated content over online social platforms have been analyzed in depth in the above research work. Using the collected data set of source (spammer), content (unrelated comment) and medium (public pages), user activities have been classified with maximum 85 % accuracy, user feed with 97.6 % accuracy, comments with 99 % accuracy as genuine or fraudulent and pages with 95 % accuracy as credible or not credible. But to use our analysis, we plan to build an application for Facebook that can show the credibility score for the public page.

References

1. Wang, J., Yu, C.T., Yu, P.S., Liu, B., Meng, W.: Diversionary comments under blog posts. *ACM Trans. Web (TWEB)* **9**(4), 18 (2015)
2. Dewan, P., Kumaraguru, P.: Towards automatic real time identification of malicious posts on Facebook. In: 2015 13th Annual Conference on Privacy, Security and Trust (PST), pp. 85–92. IEEE, 21 July 2015

3. Abu-Nimeh, S., Chen, T.M., Alzubi, O.: Malicious and spam posts in online social networks. *Computer* **12**(9), 23–28 (2011)
4. Metzger, M.J., Flanagin, A.J., Eyal, K., Lemus, D.R., McCann, R.M.: Credibility for the 21st century: integrating perspectives on source, message, and media credibility in the contemporary media environment. *Commun. Yearb.* **20**(27), 293–336 (2003)
5. Nguyen, D.T., Nguyen, N.P., Thai, M.T.: Sources of misinformation in Online Social Networks: who to suspect? In: 2012-MILCOM Military Communications Conference, pp. 1–6. IEEE, 29 October 2012
6. Lin, C., Huang, Z., Yang, F., Zou, Q.: Identify content quality in online social networks. *IET Commun.* **6**(12), 1618–1624 (2012)
7. Ismail, S., Latif, R.A.: Authenticity issues of social media: credibility, quality and reality. In: *Proceedings of World Academy of Science, Engineering and Technology*, vol. 74, p. 265. World Academy of Science, Engineering and Technology (WASET), 1 February 2013
8. AlMansour, A.A., Brankovic, L., Iliopoulos, C.S.: A model for recalibrating credibility in different contexts and languages—a twitter case study. *Int. J. Digit. Inf. Wirel. Commun. (IJDIWC)* **4**(1), 53–62 (2014)
9. Abbasi, M.-A., Liu, H.: Measuring user credibility in social media. In: Greenberg, A.M., Kennedy, W.G., Bos, N.D. (eds.) *SBP 2013. LNCS*, vol. 7812, pp. 441–448. Springer, Heidelberg (2013)
10. Kang, M.: Measuring social media credibility: a study on a measure of blog credibility. *Inst. Public Relat.* 59–68 (2010)
11. Stringhini, G., Kruegel, C., Vigna, G.: Detecting spammers on social networks. In: *Proceedings of the 26th Annual Computer Security Applications Conference*. ACM (2010)
12. Stein, T., Chen, E., Mangla, K.: Facebook Immune System in European Conference on Computer System (EuroSys) (2011)
13. Robertson, M., Pan, Y., Yuan, B.: A social approach to security: using social networks to help detect malicious web content. In: 2010 International Conference on Intelligent Systems and Knowledge Engineering (ISKE). IEEE (2010)
14. Cvijikj, I.P., Michahelles, F.: Monitoring trends on facebook. In: 2011 IEEE Ninth International Conference on Dependable, Autonomic and Secure Computing (DASC). IEEE (2011)
15. Abu-Nimeh, S., Chen, T.M.: Proliferation and detection of blog spam. *IEEE Secur. Priv.* **8**(5), 42–47 (2010)
16. Agrawal, H., Kaushal, R.: Analysis of text mining techniques over public pages of Facebook. In: 2016 IEEE International Advance Computing Conference(IACC). IEEE (2016)

Elliptic Curve Based Secure Outsourced Computation in Multi-party Cloud Environment

V. Thangam^(✉) and K. Chandrasekaran

Department of Computer Science and Engineering,
National Institute of Technology Karnataka, Surathkal, India
vedhathangam@gmail.com, kchnitk@ieee.org

Abstract. Secure Multi-Party Computation (SMPC) is a scheme where a set of trusted users will calculate a certain function on their inputs where the inputs will be always in an encrypted format for security purpose. In many cases, outsourcing of these calculations to an untrusted cloud server is desirable because of huge computational power of cloud server and storage space provided by them to process the data. However, the existing secure computation approaches are based on either a single key setup or old traditional encryption methods. In this paper, we suggested two secure multi-party computation techniques based on the latest elliptic curve cryptosystem. In which, we used two non-colluding cloud servers to co-operatively compute the outsourcing calculation with minimum number of interactions between them. However, it is ensured that the inputs, intermediate and final results all remain secret throughout the calculation.

Keywords: Secure multi-party computation · Outsourced computation · CTR1-SMPC · Improved CTR1-SMPC

1 Introduction

In current trends, cloud computing is becoming an emerging and efficient tactic for storing and processing those stored data. Many users have started to use the services of cloud server. However, the data stored in these servers could be sensitive at some cases. As we know, the confidential or sensitive data should always be encrypted before uploading them in cloud server. So the users might store their data in encrypted form and want to do outsourcing on it [1, 2]. Doing outsourced computation on these data is having many benefits like building accurate classifiers in machine learning, improving the disease diagnosis [1], price optimization in smart meters, face recognition system, etc. Obviously, the users might have used many different keys [3] for encrypting their data. This situation brings a big problem in outsourced calculation over the cloud data uploaded by many users. The outsourced computation would become easier if the uploaded entire data set is encrypted by using a single key.

Many of the current schemes for secure outsourced computation are constructed on a single key setup by using Fully Homomorphic Encryption (FHE) [4–7] and some of them are far away from practical implementation because of huge number of

interactions required with end users. Recently Peter et al. [8] (represented as PTK in this paper) suggested a method with two non-colluding servers by using Bresson-Catalano-Pointcheval (BCP) encryption [9]. In their approach, the ciphertexts (or confidential messages) encrypted by using multiple keys are converted into ciphertexts of a single key through heavy server to server communications ahead of starting the calculation. Then, the two non-colluding servers collaboratively calculate the arithmetic functions. But, the huge number of communications required between the two non-colluding servers suffers the effective calculation. Therefore, achieving an efficient secure outsourced computation technique over the cloud data which is encrypted by multiple different keys remains an unsolved problem.

In our work, we aimed to solve the above said problem with the help of 2 non-colluding servers and CTR1-Proxy Re-Encryption (CTR1-PRE) [24] technique. In our proposed schemes, every user is able to encrypt their information using their own public key and upload the encrypted information in the cloud. Then, at any time, they can reclaim and decrypt their uploaded information without negotiating the security. Moreover, the arithmetic function (addition) can be effectively computed on many users' encrypted information without disclosing the input, intermediate and final outcomes to any of the two non-colluding servers.

Furthermore, our proposed approaches are scalable and effective to end users. Especially, users are not required to share any private key among them. Most significantly, the calculation is non-interactive to users. After uploading their encrypted data, users do not need to do any activity for computation; they can go offline until retrieving the encrypted outputs.

We have chosen two non-colluding servers model for effectual calculation due to the fact that a single server technique for secure outsourced calculation cannot be done without huge interactions between users and server.

2 Related Work

Old secure multiparty calculation methods [6, 10, 11] concentrate on how to do any computation securely for many users without any trustworthy third party. Those approaches required exhaustive communications among end users which is not suitable in our situation we consider. As we need to compute the multiusers data in a bit-by-bit manner, we cannot use one time garbled circuit. Although the latest improvement suggested by Goldwasser et al. [7] is capable to recycle jumbled circuits, it is still constructed upon a single key setup, which is not appropriate for the context of data encrypted by many different keys. Fully homomorphic encryption (FHE) [4] allows arbitrary functions to be calculated over the ciphertexts without disclosing any confidential information. But, most current approaches [5, 12] are limited to single key setup, which is not suitable for our case. Gentry [4] suggested a method for secure multiparty calculation where a single secret key is first securely shared among users. Moreover, it needs huge number of communications among users at the end to decrypt the final results.

A variety of different approaches [13–16] target to enhance the performance of secure multiparty outsourced calculation by utilizing the benefits of cloud computing,

where the cloud server will calculate a major part of computation while the users only need to calculate a less portion of computation. For sample, Halevi et al. [14] suggested an approach by avoiding user communications throughout multiparty calculation. Nikolaenko et al. [13] and Chow et al. [17] have projected a method by using two non-colluding servers. Unfortunately, all of the above works are not suitable for outsourced calculation over encrypted cloud data of many different keys. Wang et al. [18, 19] also suggested an approach for selected real problems, such as linear programming. Still, they are also not fit for the scenario of many keys.

Going forward, López et al. [3] first considered the usage of fully homomorphic encryption in many keys environment, called Multi-key FHE, which can be used as a probable solution for the problem we want to deal with. Unluckily, the performance of their proposed approach is quiet at a same level matched to FHE because of heavy communications amongst users. Recently Boyang Wang et al. [21] proposed a scheme (denoted as Vitamin⁺ and Vitamin^{*} in this paper) with the help of 2 non-colluding server and BBS [20] proxy re-encryption. Even though, their scheme eliminates huge interactions between two servers, it leaks out the privacy (i.e. m_1/m_2) in its multiplicative homomorphic version. Later the same authors in paper [22] (denoted as Banana⁺ and Banana^{*} in this paper) tried to fix the privacy leakage by introducing one more server. Unluckily, it still discloses the similar kind of facts (i.e. $(m_1 + m_2)/m_1$). In our approach, we consider the elliptic curve based proxy re-encryption (i.e., CTR1-PRE [24]) for secure computation. As the elliptic curve encryption is only additively homomorphic in nature, here we proposed a scheme only for secure addition operation. Moreover, with slight change in procedure, our second approach can be extended to work for multiplication as well.

3 Problem Statement

3.1 System Model

Our proposed system model has n users $\{U_1, U_2, \dots, U_n\}$ and two cloud servers S and T as shown in Fig. 1. Both the cloud servers are able to afford data storing and computational services to users. Especially, in our model, data storing facility is only provided by cloud server S , while the computational services are offered by both the cloud servers. End users are expected to upload their secure information in cloud server S and they can easily access their own uploaded data at any time later. Moreover, the computation task will be executed over the secure cloud data to expose the outcome of a function $f(m_1, m_2, \dots, m_n) = m_1 + m_2 + \dots + m_n$, where m_i denotes the separate data of each user U_i .

3.2 Multi-key Model

In our work, we presume that both the cloud servers S and T are semi-honest (i.e. honest but curious) and non-colluding. It is also assumed that both servers will follow the procedure as designed and attempt to disclose the content of users input data, computed intermediate results or final outcomes of calculation without interacting with

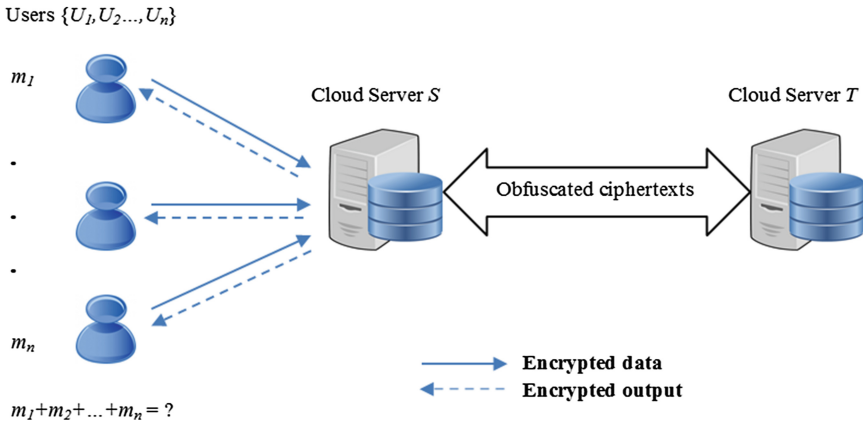


Fig. 1. Our proposed system model with n users and 2 non-colluding servers

each other. At the same time, a user U_i may also be curious and attempts to acquire the original information of other users' data uploaded in cloud server S .

Every user usually encrypts their content ahead of uploading them to cloud server S , because of the concern of privacy leakage in cloud server. In real scenario, every user will encrypt their content by using their public key. As an outcome, the outsourced calculation over these encrypted cloud data needs to be performed under many keys. This is the main reason why we call our model is of multi-key. As all the information uploaded in cloud server S is encrypted, all the data transformation between the two servers is also obfuscated to preserve security. This same kind of multi-key model can be found in papers [8, 21, 22].

3.3 Real and Ideal Model

In our paper, we analyzed the security of our proposed approaches by using Real and Ideal model. Especially, in the real world, the execution of a function f will be accomplished by an imaginary believed third party with input m_i received from every user U_i , where $1 \leq i \leq n$. This believed third party then calculates $y = f(m_1, m_2, \dots, m_n)$ and forwards y to all n users. Finally, every user gets only y without learning other secretive content. On the other hand, in the real world, this imaginary believed third party does not exist at all, and function f will be computed by executing protocol Ω (through the 2 servers S and T in this paper). A protocol Ω is said to be secure in the context of semi-honest paradigm if it satisfies the below two conditions,

1. The ideal world adversary F^{SH} should be able to replicate all the sight of real world adversary A^{SH} .
2. Both the sights of real world adversary A^{SH} and ideal world adversary F^{SH} should be computationally indistinguishable (represented by \equiv^C).

$$\text{IDEAL}_{f,F}^{SH}(i) \equiv^C \text{REAL}_{\Omega,A}^{SH}(i)$$

where i denotes the input of protocol Ω . Detailed description about the Real and Ideal model can be found in paper [23].

4 Technical Preliminary

Proxy re-encryption (PRE) is a scheme where a semi-trusted proxy changes a ciphertext of one party into a ciphertext of another party without knowing the original plaintext. In this paper, we used the PRE technique proposed in CTR1-PRE [24]. A detailed review of each step in CTR1-PRE is shown in Fig. 2.

CTR1-PRE structure is a group of procedures (Setup, KeyGenerate, ReKeyGenerate, Encrypt, Re-encrypt, Decrypt) based on elliptic curve cryptosystem.

- **Setup:** Let $E(F_q)$ be an elliptic curve above the fixed field F_q where q is large prime number (at least 160 bits) and G be a point on E of order n . Publish the system parameters as $SP = (E, q, G, f, n)$. Here f denotes the message embedding function which is used to convert arbitrary message into a point on elliptic curve.
- **KeyGenerate**(SP) $\rightarrow (sk_a, pk_a)$. Given SP , output secret key $sk_a = a \in Z_n^*$ and public key $pk_a = aG \in$ point on E .
- **ReKeyGenerate**(sk_a, sk_b) $\rightarrow rk_{a \rightarrow b}$. Given secret keys sk_a and sk_b calculate re-encryption key $rk_{a \rightarrow b} = b/a \in Z_n^*$.
- **Encrypt**(pk_a, m) $\rightarrow (C_a)$. Given public key pk_a and message $m \in Z_n^*$, generate a random $r \in Z_n^*$ output $C_a = (A, B) = (r \cdot pk_a, rG + P_m)$, where $P_m = f(m)$.
- **ReEncrypt**($rk_{a \rightarrow b}, C_a$) $\rightarrow (C_b)$. Given a rekey $rk_{a \rightarrow b}$ and ciphertext C_a , compute $C_b = (A', B') = (r \cdot pk_a \cdot rk_{a \rightarrow b}, rG + P_m)$.
- **Decrypt**(C_a, sk_a) $\rightarrow m$. Given ciphertext C_a and private key sk_a , output $P_m = B - sk_a^{-1}A$. To get back the original message m from P_m , we need to apply inverse of the function f .
- **Encrypt***(pk_a, m) $\rightarrow (C_a)$. Given public key pk_a and message $m \in Z_n^*$, generate a random $r \in Z_n^*$ output $C_a = (A, B) = (r \cdot pk_a, rG + mG)$.
- **Decrypt***(C_a, sk_a) $\rightarrow m$. Given ciphertext C_a and secret key sk_a , output $mG = B - sk_a^{-1}A$. To get back the original message m from mG , we need to solve ECDLP.

Fig. 2. Details of CTR1-PRE

The CTR1-PRE approach is additively homomorphic as it is built on the ElGamal based elliptic curve cryptosystem. Specifically, we can describe the additive homomorphic nature of ElGamal based elliptic curve cryptosystem as follows,

$$\text{Enc}_{pk}(P_{m1}) \diamond \text{Enc}_{pk}(P_{m2}) = \text{Enc}_{pk}(P_{m1} + P_{m2})$$

$$\text{Enc}_{pk}(P_{m1})^z = \text{Enc}_{pk}(P_{m1}^z),$$

where P_{mi} represents a point on elliptic curve after converting a message m_i , $\alpha \in \mathbb{Z}_q^*$ and \diamond denotes an additive operation over the points on elliptic curve.

More precisely, given ciphertexts $A = (\beta, \gamma)$ and $B = (\beta', \gamma')$, $A \diamond B$ is computed as follows, $A \diamond B = (\beta + \beta', \gamma + \gamma')$.

The decryption of this kind of additive homomorphic method requires computing elliptic curve discrete logarithm problem (ECDLP), which is assumed to be hard in practical. So it is assumed that the size of the message is small enough to solve the ECDLP efficiently.

5 Outsourced Computation in Encrypted Cloud Data Under Multiple Keys

5.1 Overview

In order to achieve outsourced calculation over the encrypted cloud data, we first transform the ciphertexts of multiple keys into ones encrypted by using the single key with the help of CTR1-PRE technique. At the same time, the original ciphertexts are also retained in cloud server, so that each user can recover and decipher their content later by using their private key.

5.2 CTR1-SMPC

Details of CTR1-SMPC. Our first scheme, CTR1-SMPC consists of six procedures such as **Setup**, **Upload**, **TransEncrypt**, **Compute**, **TransDecrypt** and **Retrieval**. At first, every user and two non-colluding servers produce their own public and secret keys in **Setup** phase. Moreover, the re-encryption keys for converting ciphertexts of different keys into ciphertexts of a single key are also generated in **Setup**. Here it needs interactions among server S , server T and users U_i via secure channel such as SSL. In **Upload** phase, each user encrypts their data by using their public key with CTR1-PRE (i.e. **CTR1-PRE.Encrypt**), and then uploads their encrypted information to cloud server S for computation.

After then, cloud server S first transforms the received encrypted data into ones encrypted under the public key of server T by using the pre-computed re-encryption keys (or rekeys). This transformation is done in **TransEncrypt** phase. After the conversion of ciphertexts, addition over them can be calculated by server S independently without the interaction of server T . Then the last outcome of calculation will be converted to user with the help of procedure **TransDecrypt**, so that each user can able to decrypt the last outcome received from server S by using their own secret key in **Retrieve**. The details of each procedure are explained in Fig. 3.

Discussion. The proposed approach done in paper PTK also supports the many keys concept with 2 non-colluding servers. But, in their approach, the number of server-to-server communications in the alteration of ciphertexts of many keys into a single key is getting increased linearly with respect to the number of data uploaded by

Let $E(F_q)$ be an elliptic curve over the fixed field F_q where q is big prime number (at least 160 bits) and G be a point on E of prime order n . Publish the system parameters as $SP = (E, q, G, f, n)$. Here f denotes the message embedding function which is used to convert arbitrary message into a point on elliptic curve.

Setup: Each User U_i selects their key pair as $(sk_i, pk_i) = (a_i, a_iG)$, S generates $(sk_s, pk_s) = (a_s, a_sG)$, T generates $(sk_t, pk_t) = (a_t, a_tG)$. S produce an arbitrary k_i , and forwards it to all users U_i ; then U_i calculates k_i / a_i and sends it to server T ; T computes $a_t * k_i / a_i$ and sends it to S ; finally, S recovers $rk_{i \rightarrow t} = a_t * k_i / a_i * k_i^{-1} = a_t / a_i$.

Upload: Each user U_i encrypts data m_i as

$$C_i(m_i) = \mathbf{CTR1-PRE.Encrypt}(pk_i, m_i) = (r_i * pk_i, r_iG + P_{m_i})$$
, and uploads C_i to S .

TransEncrypt: S calculates $C_i(m_i)$ with $rk_{i \rightarrow t}$ as

$$C_i(m_i) = \mathbf{CTR1-PRE.ReEncrypt}(rk_{i \rightarrow t}, C_i(m_i)) = (r_i * pk_i, r_iG + P_{m_i})$$
.

Compute: Given $C_i(m_1)$ and $C_i(m_2)$, cloud server S outputs,

$$C_i(y) = C_i(m_1 + m_2) = C_i(m_1) \diamond C_i(m_2) = (pk_i(r_1+r_2), G(r_1+r_2) + (P_{m_1} + P_{m_2}))$$
.

TransDecrypt: Given $C_i(y)$, where $y = f(m_1, m_2, \dots, m_n)$, S outputs,

$$C_i(y) = \mathbf{CTR1-PRE.ReEncrypt}(rk_{i \rightarrow t}^{-1}, C_i(y)), \quad 1 \leq i \leq n$$
;

Retrieve: User U_i retrieves $C_i(y)$ from S , and outputs, $y = \mathbf{CTR1-PRE.Decrypt}(sk_i, C_i(y))$, where $y = m_1 + m_2 + \dots + m_n$.

Fig. 3. Details of each procedure in CTR1-SMPC

user, which is not an efficient feature. On contrary, our approach CTR1-SMPC does not need any server to server communications during the alteration of confidential messages. This is because we have already generated the re-encryption keys earlier in the **Setup** phase of our protocol. Moreover, our approach persists secure in case of collusion between any end user and server S or T , whereas in PTK it is assumed to be collusion free between any two entities.

Security Analysis of CTR1-SMPC. We examine the security of our approach CTR1-SMPC under semi-honest adversary model. We have used the Real-and-Ideal context to evaluate our each procedure based on Composition Theorem (in the context of semi-honest paradigm) [23].

Theorem 1: *In CTR1-SMPC, as long as CTR1-PRE is semantically secure and the 2 cloud servers are non-colluding, it is computationally infeasible for cloud server S or T to differentiate the inputs, intermediate outcomes, and end results of outsourced calculation done over encrypted cloud data in multi-keys environment.*

Proof: Let's analyze the security of each procedure one by one in the context of semi-honest paradigm.

Setup: As per the security of CTR1-PRE, the calculation of rekeys by cloud server S will not disclose the private key of any user or the private key of cloud server T or the rekey of T to S .

Upload: In this procedure, every user uploads their secret content to cloud server S , where all content are encrypted under **CTR1-PRE.Encrypt** by using their own public

key. Hence, given two ciphertexts (e.g., $C_1(m_1)$ and $C_2(m_2)$) produced by user (e.g., U_1), it becomes computationally infeasible for server S to differentiate these 2 ciphertexts as long as the ciphertexts generated by **CTR1-PRE.Encrypt** is semantically secure and indistinguishable.

TransEncrypt: In this step, cloud server S transforms every user's confidential messages into the confidential messages of cloud server T by using the **CTR1-PRE.ReEncrypt**. Therefore, it becomes computationally infeasible for server S to disclose users' original content during the re-encryption process as long as the re-encryption method of CTR1-PRE is semantically secure.

Compute: We analyze the security of this procedure with Real-and-Ideal framework. In the proof, our aim is to replicate the sight of semi-honest adversary in real world by a simulator in ideal world. An algorithm is said to be secure, if both the sights of real and ideal world are computationally indistinguishable.

In this procedure, as server S is independently computing addition operations on ciphertexts, we need to show that this procedure is secure against a semi-honest adversary A_S^{SH} corrupting S in the real world. We construct a simulator F^{SH} in the ideal world to replicate the sight of semi-honest adversary A_S^{SH} in the real world. The sight of A_S^{SH} in this procedure contains its input $\{C_t(m_1), C_t(m_2)\}$ and output $C_t(m_1 + m_2)$.

Simulator F^{SH} computes $C_t(m_1') = \text{CTR1-PRE.Encrypt}(pk_p, 1)$, $C_t(m_2') = \text{CTR1-PRE.Encrypt}(pk_p, 2)$, where $m_1' = 1$ and $m_2' = 2$ without loss of generality. Then, it calculates $C_t(m_1' + m_2') = C_t(m_1') \diamond C_t(m_2')$, and returns $\{C_t(m_1'), C_t(m_1')$, $C_t(m_1' + m_2')\}$ to A_S^{SH} . Since the view of A_S^{SH} are ciphertexts created under CTR1-PRE and A_S^{SH} has no idea of the secret key of server T , if A_S^{SH} could differentiate the confidential messages of ideal world from the real world, then in designates A_S^{SH} could have an algorithm to differentiate the confidential messages produced by **CTR1-PRE.Encrypt**, which controverts to the presumption that CTR1-PRE is semantically secure. Therefore, it is proved that the confidential messages of both real and ideal world are computationally indistinguishable by a semi-honest adversary A_S^{SH} . Thus, we have

$$\text{IDEAL}_{f,F}^{SH}(C_t(m_i)) \equiv^C \text{REAL}_{\text{Compute},AS}^{SH}(C_t(m_i))$$

where $i \in \{1, 2\}$.

For the situation of conspiracy between server S and any user (e.g., user U_1) in **Compute**, the corresponding semi-honest adversary A_{S,U_1}^{SH} corrupting S and user U_1 at the same time will only have an insignificant amount of benefit (i.e., the benefit brought by a message/ciphertext pair $\{m_1, C_t(m_1)\}$ in the security game of CTR1-PRE) to differentiate ciphertexts $\{C_t(m_2), C_t(m_1 + m_2)\}$ between the real world and the ideal world under the input of $\{m_1, C_t(m_2)\}$ in the above simulation. So we can conclude,

$$\text{IDEAL}_{f,F}^{SH}(\text{Input}) \equiv^C \text{REAL}_{\text{Compute},A\{S,U_1\}}^{SH}(\text{Input})$$

where $\text{Input} = \{m_1, C_t(m_2)\}$.

TransDecrypt: In this step, server S transforms the ciphertext of result encrypted by the private key of server T into the ciphertext of every user with the help of rekey and **CTR1-PRE.ReEncrypt** method. We can analyze the security of this procedure similar to **TransEncrypt**.

Retrieval: In this step, the ciphertexts encrypted by each user's public key, will be retrieved by each user. Moreover, server S is said to be computationally infeasible for differentiating the outcome of outsourced calculation as long as CTR1-PRE is semantically secure.

5.3 Improved CTR1-SMPC

Details of Improved CTR1-SMPC. In our first approach, the cloud server T is just used as a place holder. It is not involved in any computation rather than the re-encryption keys generation. Moreover, after getting the final results from cloud server S , all the users need to solve the ECDLP individually in order to see the original result, which is not desirable to do. So, we modified the steps in CTR1-SMPC and utilized the cloud server T to solve the ECDLP as it is having huge computational power. Same like CTR1-SMPC, Improved CTR1-SMPC also consists of six procedures such as **Setup**, **Upload**, **TransEncrypt**, **Compute**, **TransDecrypt** and **Retrieval**. The Setup phase is exactly same as in CTR1-SMPC. In **Upload** phase, each user encrypts their data by using their public key with CTR1-PRE (i.e. **CTR1-PRE.Encrypt***), and then uploads their encrypted content to server S for computation.

After receiving these encrypted data, Cloud Server S first transforms these data into ones encrypted under the public key of server T by using the pre-computed rekeys. This transformation is done in **TransEncrypt** phase. After the transformation, the ciphertexts are further obfuscated and temporary addition is performed on them at cloud server S . Then, it is forwarded to cloud server T to solve ECDLP. Once the ECDLP is solved at T , the result is encrypted (i.e., **CTR1-PRE.Encrypt**) by using its own public key and sent back to cloud server S . Finally the end outcome of calculation will be sent to user with the help of procedure **TransDecrypt**, so that every user can able to decrypt the end result received from server S by using their own secret key in **Retrieve**. The details of each procedure are explained in Fig. 4.

Discussion. Similar to our first approach CTR1-SMPC, Improved CTR1-SMPC can also remove server to server communications during the alteration of confidential messages in **TransEncrypt**. The following Table 1 explains the number of server-to-server communications required at each step in our proposed schemes along with existing approaches.

Moreover, the second approach also requires solving elliptic curve discrete logarithm problem during the decryption of ciphertexts at cloud server T . So, it is assumed that the message size is small enough to efficiently solve ECDLP. Here, after receiving the outsourced encrypted (i.e., by using **CTR1-PRE.Encrypt***) data from users, the

Let $E(F_q)$ be an elliptic curve over the fixed field F_q where q is big prime number (at least 160 bits) and G be a point on E of prime order n . Publish the system parameters as $SP = (E, q, G, n)$.

Setup: Each User U_i selects their key pair as $(sk_i, pk_i) = (a_i, a_iG)$, S generates $(sk_s, pk_s) = (a_s, a_sG)$, T generates $(sk_t, pk_t) = (a_t, a_tG)$. S select an arbitrary k_i , and forwards it to all users U_i ; then U_i calculates k_i / a_i and sends it to server T ; T computes $a_i * k_i / a_i$ and sends it to S ; finally, S recovers $rk_{i \rightarrow t} = a_i * k_i / a_i * k_i^{-1} = a_i / a_i$.

Upload: U_i encrypts data m_i as
 $C_i(m_i) = \mathbf{CTR1-PRE.Encrypt}^*(pk_i, m_i) = (r_i * pk_i, r_iG + m_iG)$, and uploads C_i to S .

TransEncrypt: S calculates $C_i(m_i)$ with $rk_{i \rightarrow t}$ as
 $C_i(m_i) = \mathbf{CTR1-PRE.ReEncrypt}(rk_{i \rightarrow t}, C_i(m_i)) = (r_i * pk_i, r_iG + m_iG)$.

Compute: Given $C_i(m_1)$ and $C_i(m_2)$, cloud server S calculates
 $C_i(m_1 + x_1) = C_i(A, B + x_1G)$, $C_i(m_2 + x_2) = C_i(A', B' + x_2G)$,
 $C_i(m_1 + x_1 + m_2 + x_2) = C_i(m_1 + x_1) \diamond C_i(m_2 + x_2)$
 $= (pk_i(r_1+r_2), (r_1+r_2)G + (m_1 + x_1 + m_2 + x_2)G)$, and
 $z = x_1 + x_2$,

where x_1 and x_2 are two random numbers selected from Z_n^* , and sends $C_i(m_1 + x_1 + m_2 + x_2)$ to cloud server T . Then, T computes ECDLP as follows,
 $m' = m_1 + x_1 + m_2 + x_2 = \mathbf{CTR1-PRE.Decrypt}^*(sk_t, C_i(m_1 + x_1 + m_2 + x_2))$,
and returns $C_i(y') = C_i(m') = \mathbf{CTR1-PRE.Encrypt}(pk_i, m')$ to cloud server S .

TransDecrypt: Given $C_i(y')$, where $y' = f(m_1, m_2, \dots, m_n) = m_1 + x_1 + \dots + m_n + x_n$, S outputs,
 $C_i(y') = \mathbf{CTR1-PRE.ReEncrypt}(rk_{i \rightarrow t}^{-1}, C_i(y'))$, $1 \leq i \leq n$;

Retrieval: User U_i receives $C_i(y')$ and z from S , and outputs,
 $y = \mathbf{CTR1-PRE.Decrypt}(sk_i, C_i(y')) - z$, where $y = m_1 + m_2 + \dots + m_n$.

Fig. 4. Details of each procedure in improved CTR1-SMPC

Table 1. Comparison of server to server communications in each step with n users

Name of step	PTK	Vitamin ⁺	Vitamin [*]	Banana ⁺	Banana [*]	CTR1-SMPC	Improved CTR1-SMPC
TransEncrypt	O(n)	Null	Null	Null	Null	Null	Null
Compute (add)	Null	Null	O(1)	Null	O(1)	Null	O(1)
TransDecrypt	O(n)	Null	Null	O(n)	O(n)	Null	Null

server S transform them to ciphertexts of server T with the help of rekeys. Then server S adds blind factors x_1, x_2, \dots, x_n with each of ciphertexts. The different blind factors added to ciphertexts ensure the security of outsourced data against security attack in cloud server T . As it is assumed that the two servers are non-colluding, the cloud server T cannot differentiate the decrypted result $m_1 + x_1 + m_2 + x_2 + \dots + m_n + x_n$ from original result $m_1 + m_2 + \dots + m_n$.

As we said earlier in related work, our second approach Improved CTR1-AMPC can be extended to work for secure multiplication operation as well. The Fig. 5. shows how to do multiplication securely.

Compute (multiplication): Given $C_t(m_1)$ and $C_t(m_2)$, cloud server S calculates $C_t(x_1 \cdot m_1) = C_t(x_1 \cdot A, x_1 \cdot B)$, $C_t(x_2 \cdot m_2) = C_t(x_2 \cdot A', x_2 \cdot B')$, and send them to cloud server T .

$$z = x_1 \cdot x_2,$$

where x_1 and x_2 are two random numbers selected from Z_n^* . Then, server T computes

$$x_1 \cdot m_1 = \mathbf{CTR1-PRE.Decrypt}^*(sk_t, C_t(x_1 \cdot m_1)),$$

$$x_2 \cdot m_2 = \mathbf{CTR1-PRE.Decrypt}^*(sk_t, C_t(x_2 \cdot m_2)),$$

and $m' = x_1 \cdot m_1 \cdot x_2 \cdot m_2$, and returns $C_t(y') = C_t(m') = \mathbf{CTR1-PRE.Encrypt}(pk_t, m')$ to cloud server S .

Retrieval: User U_i receives $C_t(y')$ and z from S , and outputs $y = \mathbf{CTR1-PRE.Decrypt}(sk_i, C_t(y')) / z$, where $y = m_1 \cdot m_2 \cdot \dots \cdot m_n$.

Fig. 5. Detail of multiplication operation in improved CTR1-SMPC

Security Analysis of Improved CTR1-SMPC. Alike to the security study of our first approach, we can also explain the security of Improved CTR1-SMPC in the context of semi-honest paradigm.

Theorem 2: *In Improved CTR1-SMPC, as long as CTR1-PRE is secure, obfuscating elements are arbitrarily chosen and the 2 cloud servers are non-colluding, it is computationally infeasible for cloud server S or T to differentiate the inputs, intermediate outcomes and end results of outsourced calculation over encrypted cloud data in multi-keys environment.*

Proof: As the particulars of **Setup**, **Upload**, **TransEncrypt**, **TransDecrypt** and **Retrieval** in Improved CTR1-SMPC are as identical as to CTR1-SMPC except the format of confidential messages, we can show the security of those procedures with the similar manner in the respective portions of Theorem 1. Now, we show the security of **Compute** in Improved CTR1-SMPC with Real and Ideal model.

Compute: Since both the cloud servers are collaboratively contributes together in this procedure, we need to show that the procedure **Compute** is not only secure against semi-honest adversary A_S^{SH} corrupting server S in the real world but also secure against semi-honest adversary A_T^{SH} corrupting cloud sever T .

Security Against Cloud Server S : We create a simulator F^{SH} in the ideal world to replicate the sight of A_S^{SH} in the real world. The sight of A_S^{SH} in **Compute** comprises inputs $\{C_t(m_1), C_t(m_2)\}$, random numbers $\{x_1, x_2, x_1 + x_2\}$, ciphertexts $\{C_t(m_1 + x_1), C_t(m_2 + x_2), C_t(m_1 + x_1 + m_2 + x_2)\}$ and output $C_t(m_1 + m_2)$.

Simulator F^{SH} computes $C_t(m_1') = \mathbf{CTR1-PRE.Encrypt}^*(pk_t, 1)$, $C_t(m_2') = \mathbf{CTR1-PRE.Encrypt}^*(pk_t, 2)$, where $m_1' = 1$ and $m_2' = 2$ without loss of generality. Then, it will generate random numbers x_1', x_2' , and computes $x_1' + x_2'$. Then, it evaluates $C_t(m_1' + x_1') = C_t(A, B + x_1'G)$, $C_t(m_2' + x_2') = C_t(A, B + x_2'G)$, $C_t(m_1 + x_1 + m_2 + x_2) = \mathbf{CTR1-PRE.Encrypt}^*(pk_t, 1 + x_1 + 2 + x_2)$, $C_t(m_1 + m_2) = \mathbf{CTR1-PRE.Encrypt}^*(pk_t, 1 + 2)$. At last, F^{SH} returns all these calculated values $\{C_t(m_1'), C_t(m_2'), x_1', x_2', x_1' + x_2', C_t(m_1' + x_1'), C_t(m_2' + x_2'), C_t(m_1 + x_1 + m_2 + x_2), C_t(m_1 + m_2)\}$ to A_S^{SH} . It is known that the view of A_S^{SH} is ciphertexts generated by CTR1-PRE algorithm and A_S^{SH} has no clue about the private key of server T . Moreover, if A_S^{SH} is able to differentiate the ideal

world from the real world, then it says that A_S^{SH} could have an algorithm to differentiate the confidential messages produced by **CTRI-PRE.Encrypt**, which opposes to the pre-supposition that CTRI-PRE is semantically secure. Therefore, adversary A_S^{SH} is computationally infeasible to differentiate the ideal world from the real world. Thus, we have

$$\text{IDEAL}_{f,F}^{SH}(C_t(m_i)) \equiv^C \text{REAL}_{\text{Compute},AS}^{SH}(C_t(m_i))$$

where $i \in \{1, 2\}$. For the situation of conspiracy between cloud server S and any user U_i , we can show the security in the same way as proved above with inputs $\{m_1, C_t(m_2)\}$, and finally we have,

$$\text{IDEAL}_{f,F}^{SH}(\text{Input}) \equiv^C \text{REAL}_{\text{Compute},A\{S,U_i\}}^{SH}(\text{Input})$$

where $\text{Input} = \{m_1, C_t(m_2)\}$.

Security Against Cloud Server T: We create a simulator F^{SH} in the ideal world to simulate the sight of A_T^{SH} in the real world. The sight of A_T^{SH} in **Compute** comprises input $C_t(m_1 + x_1 + m_2 + x_2)$ [#] (# note that the result of summation $m_1 + x_1 + m_2 + x_2$ is converted to a point on elliptic curve and then encrypted by using public key of server T), blinded message $m_1 + x_1 + m_2 + x_2$ and output $C_t(m_1 + x_1 + m_2 + x_2)$.

Simulator F^{SH} computes $m_1' + x_1' = 1 + x_1'$ and $m_2' + x_2' = 2 + x_2'$, and $m_1' + x_1' + m_2' + x_2' = 1 + x_1' + 2 + x_2'$, where x_1 and x_2 are randomly selected two numbers, and send $m_1' = 1, m_2' = 2$ without loss of generality. Then, it computes $C_t(m_1' + x_1') = C_t(1 + x_1')$, $C_t(m_2' + x_2') = C_t(2 + x_2')$, and $C_t(m_1 + x_1 + m_2 + x_2) = C_t(1 + x_1 + 2 + x_2)$. Finally it sends $\{m_1 + x_1 + m_2 + x_2, C_t(m_1 + x_1 + m_2 + x_2)\}$ to A_T^{SH} .

Even though, the adversary A_T^{SH} is able to decipher the ciphertext in its sight with its own secret key, the deciphered message is obfuscated, which is arbitrarily distributed due to the use of arbitrary numbers x_1' and x_2' . Therefore, it is said to be that A_T^{SH} cannot differentiate ideal world from the real world, and we have

$$\text{IDEAL}_{f,F}^{SH}(C_t(m_i + x_i)) \equiv^C \text{REAL}_{\text{Compute},AT}^{SH}(C_t(m_i + x_i))$$

where $i \in \{1, 2\}$. For the situation of collusion between cloud server S or T and any user U_i , we can show the security in the same way as proved above with the input as $\{m_1, C_t(m_2 + x_2)\}$, and finally we have,

$$\text{IDEAL}_{f,F}^{SH}(\text{Input}) \equiv^C \text{REAL}_{\text{Compute},A\{T,U_i\}}^{SH}(\text{Input})$$

where $\text{Input} = \{m_1, C_t(m_2 + x_2)\}$.

6 Conclusions and Future Work

In this paper, we projected two effective approaches to support secure outsourced calculation on encrypted cloud data with the help of 2 non-colluding cloud servers. In our scheme, users need not to communicate with other user or cloud server throughout the calculation. We have also shown that our approaches require minimum number of

server to server communications, which considerably improves the performance of outsourced calculation. Our approaches have certain weakness as they support only small size messages and addition (in the case of first scheme). Our future work will focus to eliminate these two constraints.

References

1. Song, D., Shi, E., Fischer, I., Shankar, U.: Cloud data protection for the masses. *IEEE Comput.* **45**(1), 39–45 (2012)
2. Ren, K., Wang, C., Wang, Q.: Security challenges for the public cloud. *IEEE Internet Comput.* **16**(1), 69–73 (2012)
3. L'opez, A., Tromer, E., Vaikuntanathan, V.: On-the-fly multiparty computation on the cloud via multikey fully homomorphic encryption. In: *Proceedings of STOC 2012* (2012)
4. Gentry, C.: Fully homomorphic encryption using ideal lattices. In: *Proceedings of STOC 2009* (2009)
5. Brakerski Z., Vaikuntanathan, V.: Efficient fully homomorphic encryption from (standard) LWE. In: *Proceedings of FOCS 2011* (2011)
6. Yao, A.: Protocols for secure computations. In: *Proceedings of FOCS*, pp. 160–164 (1982)
7. Goldwasser, S., Kalai, Y., Popa, R.A., Vaikuntanathan, V., Zeldovich, N.: Reusable garbled circuits and succinct functional encryption. In: *Proceedings of STOC 2013* (2013)
8. Peter, A., Tews, E., Katzenbeisser, S.: Efficiently outsourcing multiparty computation under multiple keys. *IEEE Trans. Inf. Forensics Secur.* **8**(12), 2046–2058 (2013)
9. Bresson, Emmanuel, Catalano, Dario, Pointcheval, David: A simple public-key cryptosystem with a double trapdoor decryption mechanism and its applications. In: Laih, Chi-Sung (ed.) *ASIACRYPT 2003*. LNCS, vol. 2894, pp. 37–54. Springer, Heidelberg (2003)
10. Chaum, D., Crepeau, C., Damgard, I.: Multiparty unconditionally secure protocols. In: *Proceedings of STOC*, pp. 11–19 (1988)
11. Bendlin, Rikke, Damgård, Ivan, Orlandi, Claudio, Zakarias, Sarah: Semi-homomorphic encryption and multiparty computation. In: Paterson, Kenneth G. (ed.) *EUROCRYPT 2011*. LNCS, vol. 6632, pp. 169–188. Springer, Heidelberg (2011)
12. Smart, N.P., Vercauteren, F.: Fully homomorphic encryption with relatively small key and ciphertext sizes. In: *Proceedings of PKC*, pp. 420–443 (2010)
13. Nikolaenko, V., Weinsberg, U., Ioannidis, S., Joye, M., Boneh, D., Taft, N.: Privacy-preserving ridge regression on hundred of millions of records. In: *Proceedings of IEEE S&P 2013* (2013)
14. Halevi, Shai, Lindell, Yehuda, Pinkas, Benny: Secure computation on the web: computing without simultaneous interaction. In: Rogaway, Phillip (ed.) *CRYPTO 2011*. LNCS, vol. 6841, pp. 132–150. Springer, Heidelberg (2011)
15. Kamara, S., Mohassel, P., Raykova, M.: Outsourcing Multi-Party Computation. <http://eprint.iacr.org/2011/272>
16. Kamara, S., Mohassel, P., Riva, B.: Salus: A system for server-aided secure function evaluation. In: *Proceedings of ACM CCS 2012*, pp. 797–808 (2012)
17. Chow, S.S.M., Lee, J.H., Subramanian, L.: Two-party computation model for privacy-preserving queries over distributed databases. In: *Proceedings of NDSS 2009* (2009)
18. Wang, C., Ren, K., Wang, J., Secure and practical outsourcing of linear programming in cloud computing. In: *Proceedings of INFOCOM*, pp. 820–828 (2011)

19. Wang, C., Ren, K., Wang, J., Urs, K.M.R.: Harnessing the cloud for securely solving large-scale systems of linear equations. In: Proceedings of ICDCS (2011)
20. Blaze, Matt, Bleumer, Gerrit, Strauss, Martin J.: Divertible protocols and atomic proxy cryptography. In: Nyberg, Kaisa (ed.) EUROCRYPT 1998. LNCS, vol. 1403, pp. 127–144. Springer, Heidelberg (1998)
21. Wang, B., Li, M., Chow, S.S.M., Li, H.: Computing encrypted cloud data efficiently under multiple keys. In: Proceedings of CNS-SPCC (2013)
22. Wang, B., Li, M., Chow, S.S.M., Li, H.: A tale of two clouds: computing on data encrypted under multiple keys. In: Proceedings of CNS (2014)
23. Goldreich, O.: Foundations of Cryptography: Volume 2, Basic Applications. Cambridge University Press, Cambridge (2009)
24. Thangam V., Chandrasekarn, K.: Elliptic curve based proxy re-encryption. In: Proceedings of ICTCS (2016)

Secure and Privacy Preserving Mobile Healthcare Data Exchange Using Cloud Service

Doyal Pal^(✉), Gobinda Senchury, and Praveenkumar Khethavath

Mathematics, Engineering and Computer Science Department, LaGuardia
Community College, CUNY, Long Island City, NY, USA
{dpal, gsenchury, pkhethavath}@lagcc.cuny.edu

Abstract. Healthcare using Mobile devices and Cloud Computing is an emergent topic of interest in both industry and research. Though ubiquity of mobile devices and availability of e-Health record plays an important role to enhance health care quality and services, it introduces several challenges too such as, limited power of mobile devices, privacy and anonymity of patient data, the reliability of end user, interoperability and availability of e-Health records, etc. Under HIPAA regulations, healthcare data needs to be secured from unsolicited disclosure when sharing among multiple medical service providers. In this paper, we focus on exchanging healthcare data among doctor and patient's healthcare service providers. We develop an android mobile application and a Cloud service to exchange patient's data among authorized users in a secure and privacy-preserving manner. The cloud service that we developed performs cryptographic computations, communicates with mobile application and also overcomes the limitations of mobile devices. We evaluate our proposed solution using real-world data set. The performance analysis proves the efficiency of our application. Privacy analysis of our proposed solution depicts that our system is secure and protects the privacy of healthcare data.

Keywords: Healthcare · Cloud · Mobile cloud · Privacy · Security

1 Introduction

Healthcare industry is evolving constantly. With emerging digitalization of medical records, health sectors are becoming more inclined towards using cloud computing applications because of its advanced technological infrastructure. Use of Electronic Medical Records (EMR), Electronic Health Records (EHR), Personal Health Records (PHR) is still at its early stages and it is also seen that a very few percent of hospitals even in the developed countries like the United States have comprehensive records systems [1]. There have been many organizations promoting and encouraging EHRs adoption and there is a progress that has been observed [2]. Use of EHRs is very beneficial as it improves patient care, better coordination among doctors, reduced medical errors and prominent financial and operational performances [3]. EHRs are recommended since it provides enhanced healthcare quality and safer care for patients.

Despite the benefits of EHRs, the security and privacy concerns of EHRs are unavoidable. Under Health Insurance Portability and Accountability Act (HIPAA) [19] regulations, protecting privacy and security of EHRs are important when medical service providers exchange data electronically [3, 5, 7, 14]. Nowadays the primary mechanism to guarantee security and privacy of EHRs is access control [14]. Access control mechanism gives access to the appropriate person and maintains a log of all accesses and communications. The problem arises when hospitals, doctors or other Medical Service Providers (MSP) need to share EHRs over the Internet with each other.

There has been a significant increase in the use of mobile devices in the healthcare industry, especially by doctors, medical students and faculty [4]. For prompt and efficient health-related operations, mobile devices act as a valuable medium. Mobile devices still face the challenges of having limited power in terms of processing, battery, storage and bandwidth. With the limited amount of processing power, mobile devices cannot run complex security algorithms. The aforementioned challenges of mobile devices can be solved using the cloud services as it provides on-demand computing resources such as network storage, server time, etc. without the need for human intervention. Mobile Cloud Computing mechanism in Fig. 1, is a disruptive innovation which administers solutions for the problems associated with the mobile devices. Using mobile cloud service, patient's record can be stored in the cloud storage and the necessary complex computation can be performed in the cloud servers instead of mobile devices. However, cloud environment introduces greater privacy and security threats to confidential data in transit and in cloud storage too. [22–24] proposed secure data transmission mechanisms using mobile devices. In this paper, we propose a secure and privacy preserving mechanism to share EHRs among authorized users such as hospitals and doctors using mobile devices. To make our technique more efficient we developed a cloud service which performs all computations such as decryptions and rendering data from storage and makes it available to appropriate user.

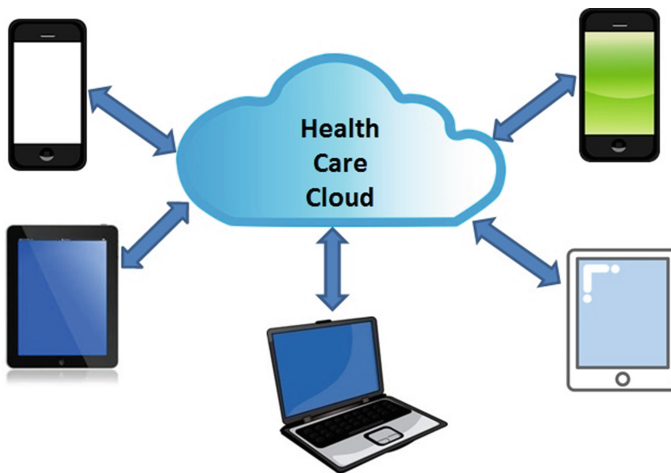


Fig. 1. Mobile cloud computing

Our paper is organized as follows. We discuss Related Work in Sect. 2. Section 3 explains the Problem Statement. Preliminaries and our Proposed Solution has been given in Sects. 4 and 5. We analyze our system in terms of performance and privacy in Sect. 6. In Sect. 7 we present the Conclusions.

2 Related Work

EMR, EHR, PHR needs to be shared across multiple health care providers such as pharmacy, doctors, hospitals, laboratory etc. to enhance health care service quality. Healthcare providers exchange data under different circumstances, such as emergency situations, when patient sees a new doctor, etc. [20] presents a real-time framework for healthcare data exchange. Health care services and exchange of health care information using mobile devices such as tablets, smart phones, laptops, personal digital assistants (PDA) ease the access to health care data especially in emergency situations and improves the health care services [10]. Despite the advantages of Mobile Healthcare (m-Healthcare) it still faces end-user reliability, efficiency, privacy and security challenges [12]. Lu et al. [11] propose a secure and privacy-preserving opportunistic computing framework for mobile-healthcare emergency (SPOC). SPOC is based on attribute-based access control and non-homomorphic encryption based privacy-preserving scalar product computation (PPSPC). It allows the medical user to decide on the participation of other medical users with similar symptoms in opportunistic computing of personal health information. In [13], a cloud-assisted privacy preserving mHealth monitoring system (CAM) has been proposed. To protect service providers' intellectual property and clients' privacy CAM proposes a key private proxy re-encryption technique and outsourcing decryption technique. A secure, privacy-preserving EHR system using attribute based cryptography and public-key encryption with a keyword search in cloud computing environment has been proposed in [21]. However, [20] does not secure the health care data exchange. In [21] user takes responsibility to secure their data in the cloud by providing privacy mechanism. Our proposed solution preserves the privacy of patients' sensitive information and reduces overhead from a user. Moreover in our scheme the healthcare service provider takes responsibility for users' privacy.

Cloud computing plays a significant role in health care industry as it caters all the requirements to ameliorate healthcare service such as availability of patients data irrespective of patient's/doctor's geographic location, large storage facilities, easy and quick access to data, etc. To improve the elasticity of resources and reduce cost overhead [5–9] uses EMR with cloud computing. Under HIPAA regulations preserving patients, privacy is important. Besides all the advantages of cloud computing, cloud services are still vulnerable to security and privacy threats. Different privacy and authentication mechanism in health care cloud has been proposed [5–9]. In [5] the authors propose a novel framework for access control to PHR, within cloud computing environment. Since PHR of a particular patient can be handled by multiple users and owners, to reduce the complexity of key management among a large number of users the system has been divided into multiple security domains (SD). To protect patients' PHR multi authority attribute-based encryption (MA-ABE) techniques has been used.

[6] presents a conceptual design and prototype implementation of a system based on Internet of Things (IoT) Gateways. The proposed system aggregate health sensor data handles security issues using digital certificates and PKI data encryption and uses the cloud as a back-end infrastructure. Privacy and anonymity are two essential requirements to exchange data securely in a peer-to-peer fashion. [7] proposes an anonymous on-the-fly privacy preserving secure data exchange protocol in P2P healthcare cloud environment using pairing-based cryptography. For each session of data exchange, a dynamic temporary ID is generated by the peers to produce a session key using the proposed protocol. In mobile cloud computing Weiwei et al. [8] propose a secure data service mechanism to achieve secrecy and access control in mobile cloud computing. To implement fine-grained access control of data and data privacy the authors have explored identity-based proxy re-encryption scheme in this work. A framework of privacy preserving attribute-based authentication system [9] to authenticate patient/physician has been proposed in mHealth networks. Based on privacy requirements during communication between patient and physician, different levels of privacy and attributed based authentication mechanism has been proposed in [9].

3 Problem Statement

Privacy preserving health care data exchange using the cloud and mobile devices is challenging because of the sensitivity of health information. In a real world scenario, a patient’s hospital (PH) keep all patients’ medical record $R_P(R_{P_1}, R_{P_2}, \dots, R_{P_i}, \dots, R_{P_n})$. When a patient P_i sees a new doctor $D_i(D_i \notin PH)$ or in an emergency situation to diagnose and treat the patient P_i , doctor D_i needs to know all relevant medical record R_{P_i} from patient’s hospital (PH), pharmacy, laboratory, etc.

Figure 2 shows the overall view of the healthcare data exchange scenario. Secure exchange of EHR/EMR/PHR among doctor, patient, hospital, pharmacies, etc. over the

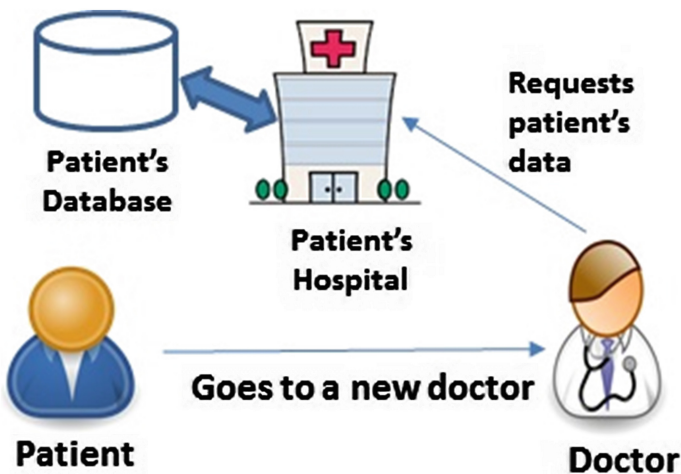


Fig. 2. Overview of the healthcare data exchange

Internet is challenging as healthcare data contain patients' sensitive information and mobile devices have limitations (limited power of processing, complex computations, etc.). The goal of secure exchange of healthcare data is to protect patients' confidential medical record from unauthorized disclosure yet make patient's record available to authorized persons whenever they require it. Healthcare using cloud and mobile devices is an emerging segment of eHealth that provides a solution for faster and better access to patients' data and improved patient care. Besides the faster and better access to patients' data, mobile devices, and the cloud is vulnerable to security and privacy threats.

4 Preliminaries

4.1 Shamir's Threshold Secret Sharing

We use Shamir's (t, n) threshold secret sharing scheme [18] in this paper. Shamir's (t, n) threshold secret sharing scheme exploits polynomial interpolation. Secret S can be shared by n number of users and can be reconstructed if the number of shares for reconstruction exceeds some threshold value t . This secret sharing mechanism uses polynomial function of order $(t - 1)$ which can be constructed as,

$$f(x) = d_0 + d_1x + d_2x^2 + \dots + d_{t-1}x^{t-1}$$

Here d_0 denotes the secret S .

Given any t shares $\langle x_0, f(x_0) \rangle, \dots, \langle x_{t-1}, f(x_{t-1}) \rangle$ the secret S can be reconstructed as well by using Lagrange Interpolation formula as follows:

$$S = \sum_{i=0}^{t-1} \left(\prod_{\substack{j=0 \\ j \neq i}}^{t-1} \frac{x_j}{x_j - x_i} \right) f(x_i)$$

5 Proposed Solution

We focus on exchanging healthcare data between doctor and other MSPs over the Internet using mobile devices and cloud services. eHealth, along with mobile devices, cloud computing, and cryptographic schemes makes a secure and efficient way to provide better patient care and improves health care quality when a doctor D_i needs to know a patient's medical record from PH for better treatment or in an emergency. To access patient's data from PH , we use the mobile device as it offers an easy and fast way to connect and access patient's record R_{p_i} .

We propose our solution in Algorithm1 and Fig. 3 depicts our Algorithm1 step by step. We develop a mobile application MA_D for doctors through which they can get access of R_{p_i} in the healthcare cloud (HCC). Only an authorized doctor D_i can log into the MA_D . When a D_i requests to PH for patient's data PH checks the authenticity of a doctor from Doctor's Hospital (DH) and patient (Algorithm 1, Line 1-5). Our solution

sends encrypted patient’s data $E(R_{P_i})$ from PH to HCC over the Internet. To encrypt patient’s data any standard public key encryption scheme can be used where the private key- public key pair (x, y) is generated by a Trusted Third Party (TTP), e.g., Certificate Authority (CA) and only public key y is distributed to the PH . To enhance security TTP generates a secret S and splits the secret into n number of shares $S_1, S_2, \dots, S_i, S_j, \dots, S_n$, where $n \geq 4$. TTP encrypts all the shares of secret S with public key y , $E_y(S_1), E_y(S_2), \dots, E_y(S_i), E_y(S_j), \dots, E_y(S_n)$. (Algorithm 1 Line 7). TTP sends public key y along $E_y(S_i)$ to PH . To encrypt data at PH we use RSA public key encryption scheme [15]. We use Shamir’s Threshold Secret Sharing mechanism [18] in our proposed solution. To reconstruct the secret S at least t (where $3 \leq t \leq n$) number of shares are required and t is determined by the TTP . PH combines patient’s record R_{P_i} with $E_y(S_i)$, encrypts $(R_{P_i}, E_y(S_i))$ with y and sends $E_y((R_{P_i}, E_y(S_i)))$ to HCC (Algorithm 1, Line 8-9). Doctor D_i log into the MA_D and TTP sends $E_y(S_j)$ to D_i (Algorithm 1, line 9-10). Since we use Shamir’s Threshold Secret Sharing Scheme therefore to reconstruct the secret S at least t numbers of shares are required. Any alteration in any S_k where $k \in \{1, t\}$ or $E_y(S_k)$ will not reconstruct S again and will result in an error. Among t numbers of shares $E_y(S_i)$ belongs to PH and $E_y(S_j)$ to D_i , therefore, rest $(t - 2)$ numbers of secrets are sent by TTP to HCC (Algorithm 1 line 11). From all the t numbers of shares at HCC if it possible to reconstruct S then HCC sends a link to D_i ’s MA_D otherwise to decline the access (Algorithm 1 line 14-17).

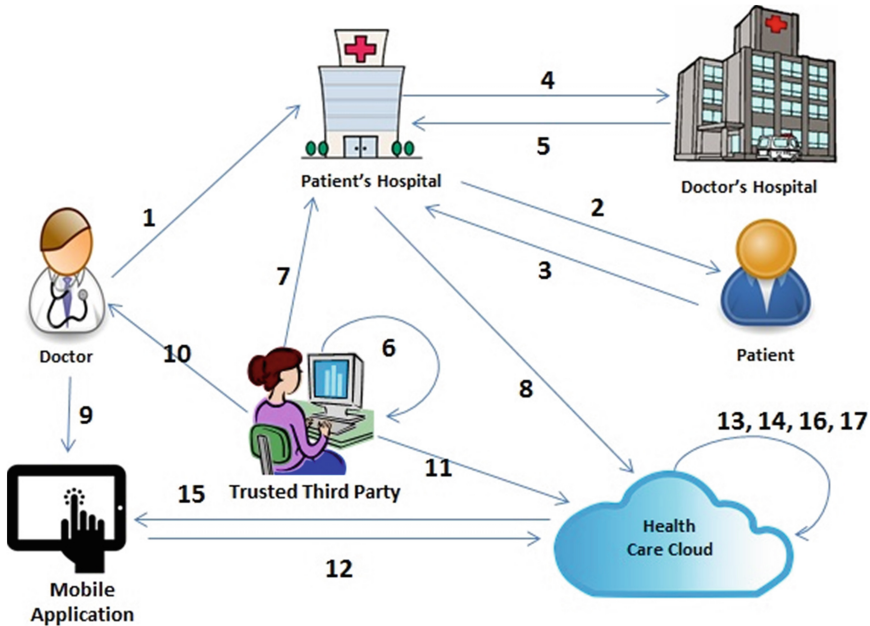


Fig. 3. Privacy preserving health care data exchange using mobile devices and HCC

5.1 Algorithm 1

Input: Patient's medical record R_P . Private key-Public key pair (x, y) and Secret S generated by TTP .

Output: Patient's record R_{P_i} . on doctor's mobile device without losing patient's privacy.

Steps:

1. Doctor D_i contact with Patient's Hospital (PH).
2. PH create a contact with a patient to check doctor's authenticity.
3. Patient sends the acknowledgement ACK_P to PH .
4. PH contact with Doctor's Hospital (DH) to check if the doctor is an authorized doctor.
5. DH sends Doctor's Authorization Acknowledgement ACK_D to PH .
6. If both ACK_P and ACK_D are true then Trusted Third Party (TTP) generates a secret S . TTP splits S into n number of shares $S_1, S_2, \dots, S_i, S_j, \dots, S_n$, where $n \geq 4$. TTP also generates a private key- public key pair (x, y) . TTP encrypts secret shares, $E_y(S_1), E_y(S_2), \dots, E_y(S_i), E_y(S_j), \dots, E_y(S_n)$ with public key y . $E_y(S_i)$ denotes encrypted text of secret share S_i .
7. TTP sends public key y along with $E_y(S_i)$ to PH .
8. PH combines patient's data R_{P_i} with $E_y(S_i)$ and generates $(R_{P_i}, E_y(S_i))$. PH encrypts $(R_{P_i}, E_y(S_i))$ with y and sends $E_y((R_{P_i}, E_y(S_i)))$ to HCC . We assume HCC is secure.
9. Doctor D_i log in to the MA_D to see patient's data.
10. TTP sends $E_y(S_j)$ to D_i .
11. TTP sends private key x, S and $(t-2)$ numbers of shares of secret $S_{(t-2)}$ (where $S_i, S_j \notin S_{(t-2)}$) to HCC . We assume the communication between TTP and HCC is through a secure channel.
12. MA_D sends $E_y(S_j)$, Patient's ID (PID) to HCC .
13. HCC decrypts $E_y((R_{P_i}, E_y(S_i)))$ with private key x and deduct $E_y(S_i)$ from $E_y((R_{P_i}, E_y(S_i)))$. HCC further decrypts $E_y(S_1), \dots, E_y(S_i), E_y(S_j), \dots, E_y(S_t)$ and reconstruct the secret S' .
14. if $S = S'$
15. HCC sends a link to MA_D to see patient's data R_{P_i} . from HCC .
16. else
17. Decline access to see R_{P_i} .

6 Analysis

6.1 Experimental Analysis

Experiment Setup

In our proposed solution we used Amazon EC2 [17] cloud t2.micro instance which provisions Microsoft Windows Server 2008 R2 base instance. We installed WAMP server on our instance and used MySQL to store the data. To create cloud service that

connects to the doctor's mobile application (MA_D) PHP has been used. We created our mobile application using Android Studio. Figure 4 shows the login screen of MA_D . We use a real-world data set, the Pima Indians Diabetes Data Set [16] to implement our system.

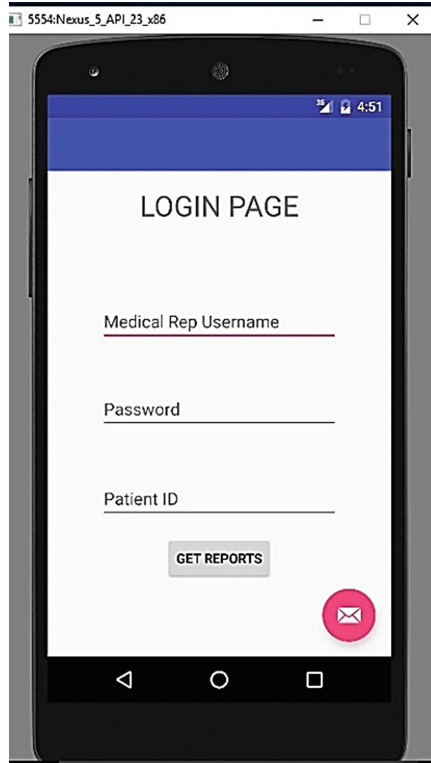


Fig. 4. Login screen of Doctor's mobile application

6.2 Performance Analysis

We test the performance of our system in terms of time efficiency. After doctor log in to the application MA_D and sends $E_v(S_j)$, Patient's ID (PID) to HCC (Algorithm 1, Line 12), the decryption procedure starts in HCC . Figure 5 shows the round trip time T_{RT} between doctor's mobile application MA_D and HCC . T_{RT} includes T_{CF} , T_{RCS} , T_D , T_R , T_{MD} , and T_{PR} . T_{CF} is the time to connect MA_D to HCC and fetching data from HCC . T_{RCS} is the time to run cloud service. Decryption time is T_D and reconstruction time of the secret S is T_R . To deduct patient's data from the combination of (D, S) it takes time T_{MD} . T_{PR} is time to parse data from HCC to mobile application in JavaScript Object Notation (JSON) format, rendering the JSON object and display it on MA_D . Therefore $T_{RT} = T_{CF} + T_{RCS} + T_D + T_R + T_{MD} + T_{PR}$.

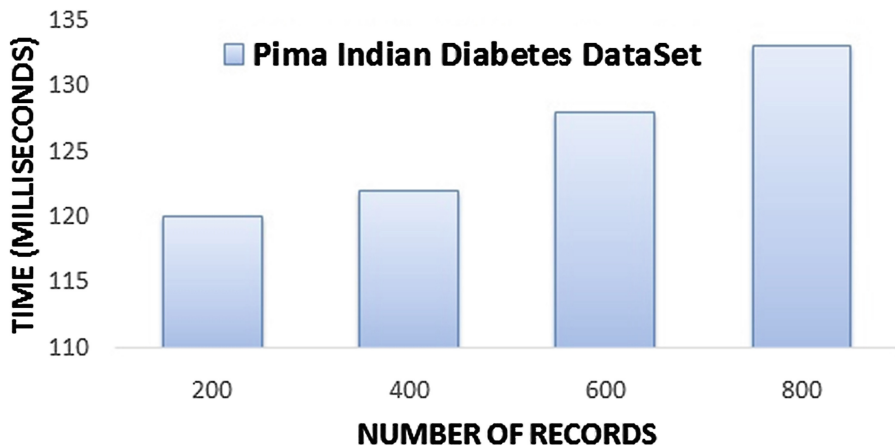


Fig. 5. Round trip time T_{RT} between Doctor's mobile application MA_D and HCC

We evaluate the proposed scheme using a real word data set, the Pima Indians Diabetes Data Set [16]. In this data set we use eight attributes: Number of times pregnant, Plasma Glucose Concentration a 2 h in an oral glucose tolerance test, Diastolic blood pressure (mm Hg), Triceps skin fold thickness (mm), 2-hours serum insulin (mm U/ml), Body mass index (weight in kg/(height in m)²), Diabetes pedigree function, Age (years). Figure 5 shows the round trip time T_{RT} for 200, 400, 600 and 800 rows of records (each record with eight attributes) from the Pima Indians Diabetes Data Set.

From Fig. 5 it is evident that with the increasing size of records stored in HCC the round trip time increments are almost negligible. For the 800 records, our proposed solution takes less than 0.14 s.

After doctor clicks on the link sent by HCC to MA_D (Algorithm 1, Line 15), the doctor can see patient's medical record R_{P_i} from HCC . Figure 6 shows the snapshot of our application when it shows patient's data on MA_D from HCC .

6.3 Privacy Analysis

In our proposed solution we preserve the privacy of patient's health information using cryptographic scheme. At first, we protect patient's privacy by checking doctor's authenticity (Algorithm 1, Line 1–5). Only a valid and authorized doctor can log in to the MA_D . To secure patient's data in transit we send encrypted patient's medical record $E_y((R_{P_i}, E_y(S_i)))$ from PH to HCC (Algorithm1, Line 8). Moreover to enhance security before encrypting patient's record R_{P_i} , we combine it with encrypted secret share $E_y(S_i)$. We assume that HCC is secure. As mobile devices are prone to be stolen or lost, even if the doctor is logged in to the system doctor cannot see patient's actual data unless it provides the encrypted secret share $E_y(S_j)$ to HCC .

To reconstruct the original secret S , t numbers of shares secrets $E_y(S_1), \dots, E_y(S_i), E_y(S_j), \dots, E_y(S_t)$ are required (Algorithm 1, Line 13). Any alteration of any of

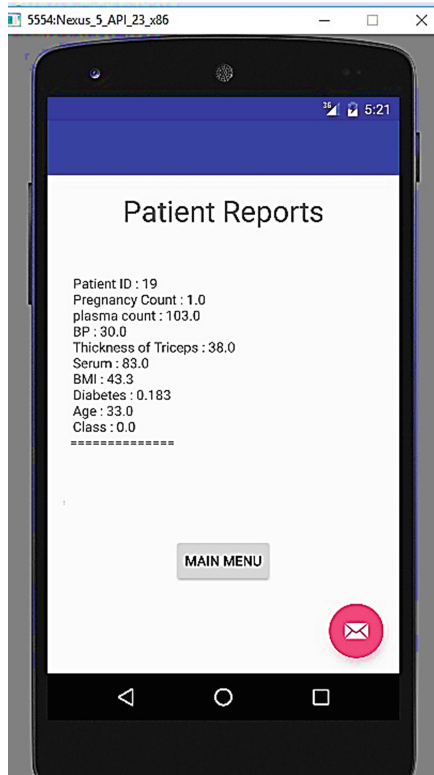


Fig. 6. Patient’s data displayed on MA_D

the shares of secret S will not reconstruct S again. Therefore reconstruction of correct S ensures doctor D_i and PH 's authenticity and protects patient’s data from unauthorized use. Moreover, for a particular patient P_i the secret S and threshold number t is determined by the TTP and S, t changes every time doctor D_i requests for patient P_i 's record R_{P_i} to PH . HCC sends a link to see patient data to MA_D (Algorithm 1 Line 15) only after it successfully reconstructs the actual secret S . We use TTP to generate the key pair (x, y) to avoid any physical security concerns and insider attacks in PH .

7 Conclusions

Exchanging health care data plays an important role for improving health care quality and services. We proposed a secure and privacy-preserving health care data exchange scheme using mobile devices and the health care cloud in this paper. We implemented an android application to exchange data between doctor and patient’s hospital. To secure data exchange we used public key encryption scheme and secret sharing mechanism in our proposed solution. The experimental result shows that the data exchange time (using cloud and mobile device) between doctor and patient’s hospital is

very less. The analysis of our system in terms of efficiency and privacy demonstrate that our solution is efficient and protect patients' sensitive health information from unwanted disclosure.

References

1. Jha, A., DesRoches, C., Cambell, E., Donelan, K., Rao, S., Ferris, T., Shields, A., Rosenbaum, S., Blumenthal, D.: Use of electronic health records in U.S. hospitals. *New Engl. J. Med.* **360**, 1628–1638 (2009). doi:[10.1056/NEJMsa0900592](https://doi.org/10.1056/NEJMsa0900592)
2. Gans, D., Kralewski, J., Hammons, T., Dowd, B.: Medical groups' adoption of electronic health records and information systems: practices are encountering greater-than-expected barriers to adopting an EHR system, but the adoption rate continues to rise. *Health Aff.* **24** (5), 1323–1333 (2005)
3. Menachemi, N., Collum, T.H.: Benefits and drawbacks of electronic health record systems. *J. Risk Manag. Healthcare Policy* **4**, 47–55 (2011)
4. J. T. Boruff, and Storie, Dale, M.L.I.S., M.A. Mobile devices in medicine: A survey of how medical students, residents, and faculty use smartphones and other mobile devices to find information*(*)(**). *Journal of the Medical Library Association* 102(1), pp. 22–30. 2014
5. Li, M., Yu, S., Ren, K., Lou, W.: Securing personal health records in cloud computing: patient-centric and fine-grained data access control in multi-owner settings. In: Jajodia, S., Zhou, J. (eds.) *SecureComm 2010*. LNICST, vol. 50, pp. 89–106. Springer, Heidelberg (2010)
6. Doukas, C., et al.: Enabling data protection through PKI encryption in IoT m-Health devices. In: *2012 IEEE 12th International Conference on Bioinformatics & Bioengineering (BIBE)*. IEEE (2012)
7. Rahman, S.M.M., et al.: Privacy preserving secure data exchange in mobile P2P cloud healthcare environment. *Peer-to-Peer Networking and Applications*, 1–16 (2015)
8. Jia, W., et al.: SDSM: a secure data service mechanism in mobile cloud computing. In: *2011 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. IEEE (2011)
9. Guo, L., et al.: A privacy-preserving attribute-based authentication system for mobile health networks. *IEEE Trans. Mob. Comput.* **13**(9), 1927–1941 (2014)
10. Wu, J.-H., Wang, S.-C., Lin, L.-M.: Mobile computing acceptance factors in the healthcare industry: a structural equation model. *Int. J. Med. Informatics* **76**(1), 66–77 (2007)
11. Lu, R., Lin, X., Shen, X.: SPOC: a secure and privacy-preserving opportunistic computing framework for mobile-healthcare emergency. *IEEE Trans. Parallel Distrib. Syst.* **24**(3), 614–624 (2013)
12. Baig, M.M., GholamHosseini, H., Connolly, M.J.: Mobile healthcare applications: system design review, critical issues and challenges. *Australas. Phys. Eng. Sci. Med.* **38**(1), 23–38 (2015)
13. Lin, H.: CAM: cloud-assisted privacy preserving mobile health monitoring. *IEEE Trans. Inf. Forensics Secur.* **8**(6), 985–997 (2013)
14. Benaloh, J., Chase, M., Horvitz, E., Lauter, K.: Patient controlled encryption: Patient privacy in electronic medical records. In: *Proceedings of the ACM Cloud Computing Security Workshop*, pp. 103–114 (2009)
15. Rivest, R.L., Shamir, A., Adleman, L.: A method for obtaining digital signatures and public-key cryptosystems. *Commun. ACM* **21**(2), 120–126 (1978)

16. UCI Machine Learning Repository. archive.ics.uci.edu/ml/datasets.html. Accessed 25 Nov 2015
17. Amazon EC2. amazon.com/EC2. Accessed July 2014
18. Shamir, A.: How to share a secret. *Commun. ACM* **22**(11), 612–613 (1979)
19. U.S. Department of Health and Human Services. The privacy rule. hhs.gov/ocr/privacy/hipaa/administrative/privacyrule/. Accessed 25 Nov 2015
20. Woodside, J.M.: EDI and ERP: a real-time framework for healthcare data exchange. *J. Med. Syst.* **31**(3), 178–184 (2007)
21. Narayan, S., Gagné, M., Safavi-Naini, R.: Privacy preserving EHR system using attribute-based infrastructure. In: *Proceedings of the 2010 ACM Workshop on Cloud Computing Security Workshop*. ACM (2010)
22. Kefford, M.G., Vanderstraeten, A.M.E., Houthoof, M.R.L.: Methods and systems for secure transmission of information using a mobile device. U.S. Patent No. 6,880,079, 12 April 2005
23. Nichols, T.J.: Method and apparatus to secure data transfer from medical device systems. U. S. Patent No. 7,039,810, 2 May 2006
24. Kambourakis, G., Maglogiannis, I., Rouskas, A.: PKI-based secure mobile access to electronic health services and data. *Technol. Health Care* **13**(6), 511–552 (2005)

Secure Certificateless Signature Scheme with Batch Verification from Bilinear Pairings

N.B. Gayathri^(✉) and P. Vasudeva Reddy

Department of Engineering Mathematics,
Andhra University, Visakhapatnam, AP, India
gayatricrypto@gmail.com, vasucrypto@yahoo.com

Abstract. In view of simplifying certificate management complexities in the traditional Public Key Cryptography (PKC) and to abolish the key escrow problem in identity based PKC (ID-PKC), concept of Certificateless Public Key Cryptography (CL-PKC) was introduced. Batch Cryptography emphasizes new developments in information security and communication networks. It has been developed to enhance the efficiency of signatures verification, by verifying a batch of n message, signature pairs in a single instance. Batch Verification (BV) can be used in various areas where many clients interact with a single server. Mail servers, Sensor Networks, e-commerce are the best examples for BV. In this paper, we present a certificateless signature (CLS) scheme that supports BV using pairings. The proof of security is presented in Random Oracle Model (ROM) under the assumption of Computational Diffie-Hellman (CDH) Problem is intractable. More over the security proofs are made without using forking lemma [20] to achieve tight security. The efficiency analysis shows that our CLS scheme is more secure and efficient than the existing schemes.

Keywords: Public key cryptography · CLS scheme · Batch verification · Bilinear pairing · ROM · CDH problem

1 Introduction

Digital Signatures play a very important role in information security and communication networks by providing message authentication, data integrity and non-repudiation. The concept of CL-PKC was first proposed by Al-Riyami et al. [1], in 2003. Unlike the traditional PKC and ID-PKC, CL-Public key cryptographic schemes allow the verifier to verify the signatures without certificate and resolve the certificate management problem. Moreover, this cryptosystem completely abolishes the key escrow problem in ID-PKC by taking user's secret key as a combination of partial secret key generated by Key Generation Centre (KGC) and secret value chosen by user.

The basic idea of BV is to amortize the computational cost and time in verification process. In BV process, different signatures of different users on different messages can be batched to verify the signatures in a single instance instead of verifying them one after the other. This feature enables us to achieve high efficiency by reducing computational cost and time. BV can be used in many applications such as banking transactions, mail server, traffic control, military applications, wireless sensor networks etc. where many members interact with a single server. The concept of BV was first presented by Fiat [6] in 1990, based on RSA signatures. In 1994, Naccache et al. [9] presented the first efficient batch verifying scheme for DSA signatures. In 1998, Bellare et al. [3] presented a pioneer work for BV and explained three standard methods for batching modular exponentiations. In 2005, Yoon et al. [13] proposed the first ID-based signature scheme with BV. Later many batch verifying schemes were proposed in traditional and ID-based setting. But there is no considerable work in certificateless setting.

Combination of BV technique with certificateless signatures integrates the advantages of both. The first batch verifying CLS scheme was presented by M. Geng et al. [7] in 2009. This scheme uses small exponent test to achieve efficient and secure BV. However, it achieves only Girault's Level 2 security [5]. Later, in 2014, C. I. Fan et al. [5] proposed a strongly secure CLS scheme supporting BV. It also uses small exponent test. This scheme achieves Girault's Level 3 security [5]. But BV cost is more due to more number of map to point hash functions. Moreover these two schemes use Forking Lemma [10] in their proof of security. Since the reductions using forking lemma are not tight, these schemes do not achieve tight security. To the best of our knowledge, these are the only schemes in certificateless setting which supports BV.

In this paper, to improve the efficiency of verification process and to achieve tight security we develop a CLS scheme which supports BV. This scheme is designed using bilinear pairings over elliptic curves. The security of this scheme is proved in ROM under the assumption that the CDH problem is hard. Moreover the security proofs are made without using forking lemma [10] to achieve tight security.

Structure of the Paper. Remainder of this paper is structured as follows. In Sect. 2 we presented some notations. In Sect. 3 we presented our CLS scheme with BV and its security proof. In Sect. 4 we presented the efficiency analysis of our scheme. Finally Sect. 5 deals with conclusion.

2 Preliminaries

We omit the Preliminaries, Definition of BV, Syntax and security model of CLS scheme in batch verification due to space constraint. Please refer to [5, 11] for details. Some notations are used throughout this paper for our convenience and are represented in Table 1.

Table 1. Notations and their meanings

Notation	Meaning
l, s	Security parameter & master secret key of the system generated by KGC
τ	System Parameter
z_q^*	The group with elements $1, 2, \dots, q-1$ under addition modulo q
G_{Adt}, G_{Mlt}	Additive & Multiplicative cyclic groups of same prime order q
H_1, H_2, H_3	Cryptographic one way hash functions
ID	User Identity
$UPSK_{ID}, USK_{ID}, UPK_{ID}$	User partial secret key, User secret key & User public key of the identity respectively
ADV_1, ADV_2	Type-I & Type-II adversaries respectively
ξ	An algorithm to solve CDH problem by using adversaries
$e : G_{Adt} \times G_{Adt} \rightarrow G_{Mlt}$	An admissible bilinear map
Ω	Signature on a message

3 New CLS Scheme with BV and Security Analysis

In this section we present an efficient CLS scheme with BV and its formal security analysis.

3.1 Proposed CLS Scheme with BV

Master Key Gen: KGC run this algorithm by taking security parameter $l \in \mathbb{Z}^+$ as input and performs the following.

- Choose additive and multiplicative cyclic groups as G_{Adt} and G_{Mlt} of same prime order q with a bilinear pairing $e : G_{Adt} \times G_{Adt} \rightarrow G_{Mlt}$; and $P \in G_{Adt}$ as a generator of G_{Adt} .
- Select a random $s \in \mathbb{Z}_q^*$ as the master secret key and sets master public key as $Q_{Pub} = sP$.
- Choose three cryptographic hash functions $H_1 : \{0, 1\}^* \rightarrow G_{Adt}$, $H_2 : \{0, 1\}^* \rightarrow \mathbb{Z}_q^*$ and $H_3 : \{0, 1\}^l \rightarrow G_{Adt}$. KGC publishes the system parameters as $\tau = \{q, G_{Adt}, G_{Mlt}, e, P, Q_{Pub}, H_1, H_2, H_3\}$ and keeps s secretly.

Partial Key Gen: KGC runs this algorithm by taking ID as input. KGC computes $K_{ID} = H_1(ID)$ and $UPSK_{ID} = sK_{ID}$ and sends $UPSK_{ID}$ to ID via secure channel.

User Key Gen: User runs this algorithm by choosing $x_{ID} \in \mathbb{Z}_q^*$ randomly and sets $USK_{ID} = x_{ID}$ and $UPK_{ID} = x_{ID}P$.

Signature Generation: Signer runs this algorithm by taking $\tau, ID, UPK_{ID}, USK_{ID}, UPSK_{ID}$, message $m \in \{0, 1\}^*$ as input and generates the signature Ω on a message $m \in \{0, 1\}^*$ by performing the following.

- The signer chooses $r \in Z_q^*$ and computes $R = rP$, $h = H_2(m, ID, R, UPK_{ID})$ and $S = H_3(\Delta)$ where Δ is an arbitrary string of length t .
- The signer computes $T = hUPSK_{ID} + S(x_{ID} + r) + rQ_{Pub}$.

Now $\Omega = (R, T)$ is a signature on a message m .

Signature Verification: The verifier runs this algorithm by taking signature $\Omega = (R, T)$ on a message m with ID and corresponding public key UPK_{ID} as input and checks the validity of a signature as follows.

Compute $h = H_2(m, ID, R, UPK_{ID})$ and $S = H_3(\Delta)$, verifies the equation

$$e(T, P) = e(hK_{ID} + R, Q_{Pub}) e(UPK_{ID} + R, S) \quad (1)$$

If the Eq. (1) holds, verifier accepts the signature $\Omega = (R, T)$; rejects otherwise.

Batch Verification: To verify n signatures $(\Omega_i)_{i=1ton}$ of n individual users $(U_i)_{i=1ton}$ with identities $(ID_i)_{i=1ton}$ on messages $(m_i)_{i=1ton}$ respectively, a verifier performs the following.

- Choose $(\delta_i)_{i=1ton} \in Z_q^*$ randomly.
- Compute $K_{ID_i} = H_1(ID_i)$, $h_i = H_2(m_i, ID_i, R_i, UPK_{ID_i})$ and $S = H_3(\Delta)$, for $i = 1, 2, 3, \dots$
- The verifier accepts the validity of n signatures if the following equation holds

$$e\left(\sum_{i=1}^n \delta_i T_i, P\right) = e\left(\sum_{i=1}^n \delta_i (h_i K_{ID_i} + R_i), Q_{Pub}\right) e\left(\sum_{i=1}^n \delta_i (UPK_{ID_i} + R_i), S\right) \quad (2)$$

3.2 Analysis of the Proposed Scheme

In this section we present the proof of correctness of the presented scheme and its security analysis.

3.2.1 Proof of Correctness of Single Signature

The correctness of the scheme can be verified by verifying Eq. (1) as follows.

$$\begin{aligned} e(T_i, P) &= e(h_i UPK_{ID_i} + S(x_{ID_i} + r_i) + r_i Q_{Pub}, P) \\ &= e(h_i UPK_{ID_i} + r_i Q_{Pub}, P) e(S(x_{ID_i} + r_i), P) \\ &= e(h_i K_{ID_i} + r_i P, S P) e(S, (x_{ID_i} + r_i) P) \\ &= e(h_i K_{ID_i} + R_i, Q_{Pub}) e(x_{ID_i} P + r_i P, S) \\ &= e(h_i K_{ID_i} + R_i, Q_{Pub}) e(UPK_{ID_i} + R_i, S). \end{aligned}$$

3.2.2 Proof of Correctness of Batch Verification

The correctness of BV can be verified by verifying Eq. (2) as follows.

$$\begin{aligned}
e\left(\sum_{i=1}^n \delta_i T_i, P\right) &= e\left(\sum_{i=1}^n \delta_i (h_i \text{UPSK}_{ID_i} + S(x_{ID_i} + r_i) + r_i Q_{Pub}), P\right) \\
&= e\left(\sum_{i=1}^n \delta_i (h_i \text{UPSK}_{ID_i} + r_i Q_{Pub}), P\right) e\left(\sum_{i=1}^n \delta_i S(x_{ID_i} + r_i), P\right) \\
&= e\left(\sum_{i=1}^n \delta_i (h_i K_{ID_i} + r_i P), sP\right) e\left(S, \sum_{i=1}^n \delta_i (x_{ID_i} + r_i) P\right) \\
&= e\left(\sum_{i=1}^n \delta_i (h_i K_{ID_i} + R_i), Q_{Pub}\right) e\left(\sum_{i=1}^n \delta_i (\text{UPK}_{ID_i} + R_i), S\right).
\end{aligned}$$

3.2.3 Security Analysis

We prove the security of our CLS scheme against Type I and Type II adversary [11] using the following theorem.

Theorem 1. *The proposed CLS scheme with BV is existentially unforgeable against adaptive chosen message attacks in the ROM with the assumption that the CDH problem is hard.*

We prove this theorem with the help of the following two lemmas.

Lemma 1. *If there exists a probabilistic polynomial-time bounded Type-I batch forger \mathcal{ADV}_1 who can forge any signature of our batch of signatures under adaptive chosen message attack by asking at most $q_{H_1}, q_{H_2}, q_{H_3}$ questions to random oracles H_1, H_2, H_3 respectively, q_{Cuser} questions to the **Create User** request oracle, q_{Rpsk} questions to the **Reveal Partial Secret Key** extraction oracle, q_{Rsk} questions to the **Reveal Secret Key** extraction oracle and q_{Sign} questions to the **Sign** oracle in ROM then there exists an algorithm ξ that can be used by \mathcal{ADV}_1 to solve the CDH problem in elliptic curve group.*

Proof: Let \mathcal{ADV}_1 is a Type-I batch forger. Suppose that \mathcal{ADV}_1 's target identity is ID^* , and he can forge a valid signature on a message (m^*, ID^*) . Now we prove that anyone can construct an algorithm ξ who can solve the CDH problem using \mathcal{ADV}_1 . Challenger ξ is given $(A = uP, B = vP)$ as a random instance of the CDH problem in G_{Adt} .

Initialization Phase: Algorithm ξ sets $Q_{\text{Pub}} = A = uP$ and runs **Master Key Gen** to generate τ . ξ then gives τ and master public key to \mathcal{ADV}_1 and keeps s secretly.

Queries Phase: In this phase, \mathcal{ADV}_1 performs the oracle simulation and ξ responds to these oracles as follows.

Queries on oracle $H_1(H_1(ID_i))$: ξ maintains a list L_1 , which is initially empty. It contains the tuples of the form (ID_i, l_{1i}, K_{ID_i}) . After receiving a query $H_1(ID_i)$, if there is a tuple $H_1(ID_i, l_{1i}, K_{ID_i})$ on L_1 , ξ returns K_{ID_i} . Otherwise, if $ID_i \neq ID^*$, ξ picks a random l_{1i} , sets $K_{ID_i} = l_{1i}P$ else (if $ID_i = ID^*$) it sets $K_{ID_i} = l_{1i}B = l_{1i}vP$. Finally, ξ returns K_{ID_i} and adds K_{ID_i} to L_1 .

Queries on Oracle $H_2(H_2(m_i, ID_i, UPK_{ID_i}, R_i))$: ζ maintains a list L_2 , which is initially empty. It contains the tuples of the form $(m_i, ID_i, UPK_{ID_i}, R_i, l_{2i})$. After receiving H_2 query on $(m_i, ID_i, UPK_{ID_i}, R_i)$, if a tuple $(m_i, ID_i, UPK_{ID_i}, R_i, l_{2i})$ exists on L_2 , ζ returns l_{2i} . otherwise, ζ picks a random $l_{2i} \in Z_q^*$ and returns l_{2i} . ζ adds $(m_i, ID_i, UPK_{ID_i}, R_i, l_{2i})$ to L_2 .

Queries on Oracle $H_3(H_3(\Delta_i))$: ζ maintains a list L_3 , which is initially empty. It contains the tuples of the form (Δ_i, S_i, l_{3i}) . After receiving a query on $H_3(\Delta_i)$, ζ gives the same answer from L_3 , if the query has been made earlier. Otherwise, ζ picks a random $l_{3i} \in Z_q^*$, computes $S_i = l_{3i}P$ and returns S_i . ζ adds (Δ_i, S_i, l_{3i}) to L_3 .

Reveal Partial Secret Key Oracle ($PSK(ID_i)$): ζ maintains a list L_{PSK} , which is initially empty. It contains the tuples of the form $(ID_i, UPSK_{ID_i})$. After receiving a query on $PSK(ID_i)$, ζ gives $UPSK_{ID_i}$ if the request has been made earlier. If $ID_i \neq ID^*$, ζ recovers the corresponding (ID_i, l_{1i}, K_{ID_i}) from the list L_1 and sets $UPSK_{ID_i} = l_{1i}Q_{Pub} = l_{1i}A$ and returns $UPSK_{ID_i}$ to \mathcal{ADV}_1 and adds $(ID_i, UPSK_{ID_i})$ to L_{PSK} . Otherwise, (if $ID_i = ID^*$), ζ aborts.

Create User Oracle ($Cuser(ID_i)$): ζ maintains a list L_{Cuser} , which is initially empty. It contains the tuples of the form $(ID_i, UPK_{ID_i}, USK_{ID_i})$. After receiving a query on $Cuser(ID_i)$, the current UPK_{ID_i} from the list L_{Cuser} will be given if the request has been made earlier. Otherwise, ζ will choose a random $w_i \in Z_q^*$ and sets $UPK_{ID_i} = w_iP$ and $USK_{ID_i} = w_i$. ζ gives UPK_{ID_i} and adds $(ID_i, UPK_{ID_i}, USK_{ID_i})$ to L_{Cuser} .

Reveal Secret Key Oracle ($RSK(ID_i)$): When \mathcal{ADV}_1 makes this query on $RSK(ID_i)$, if $(ID_i = ID^*)$, ζ aborts. Otherwise (if $ID_i \neq ID^*$), ζ finds the tuple $(ID_i, UPK_{ID_i}, USK_{ID_i})$ in a list L_{Cuser} , and returns USK_{ID_i} to \mathcal{ADV}_1 . If there is no tuple in L_{Cuser} , ζ makes a query on $Cuser(ID_i)$ to generate $(UPK_{ID_i} = w_iP, USK_{ID_i} = w_i)$. ζ saves these values in L_{Cuser} , and returns $USK_{ID_i} = w_i$.

Replace Public Key Oracle ($RPK(ID_i)$): After receiving a query on $RPK(ID_i)$, ζ finds $(ID_i, UPK_{ID_i}, USK_{ID_i})$ in L_{Cuser} . ζ replaces $UPK_{ID_i} = UPK'_{ID_i}$ and $USK_{ID_i} = \perp$

Signing Oracle: When \mathcal{ADV}_1 makes this query on (ID_i, m_i) , ζ chooses $r_i, h_i \in Z_q^*$ and computes $R_i = r_iP - h_iK_{ID_i}$. If the tuples containing h_i already exists in list L_2 , then ζ chooses another $r_i, h_i \in Z_q^*$ and tries again. Set $h_i = H_2(m_i, ID_i, UPK_{ID_i}, R_i)$ and $S_i = H_3(\Delta_i)$. Then ζ computes $T_i = l_{3i}(R_i + UPK_{ID_i}) + r_iQ_{Pub}$. Finally ζ responds to \mathcal{ADV}_1 with $\Omega_i = (R_i, T_i)$. (R_i, T_i) is a valid signature on message m_i as it satisfies Eq. (1).

Forgery: Hence, \mathcal{ADV}_1 forges a valid signature $\Omega^* = (R^*, T^*)$ on messages m^* under ID^* . Suppose ζ can construct n valid signatures $(\Omega_i^*)_{i=1ton}$ on messages $(m_i^*)_{i=1ton}$ of the signers under $(ID_i^*)_{i=1ton}$ and the corresponding $(UPK_{ID_i}^*)_{i=1ton}$ of n users $(U_i)_{i=1ton}$ with a state of information Δ^* by \mathcal{ADV}_1 such that

- i. BV holds.
- ii. There exists $I \in \{1, 2, 3, \dots, n\}$ such that \mathcal{ADV}_1 has not asked the Partial Secret Key queries for ID_I^* and \mathcal{ADV}_1 has not asked the Sign oracle query.

Without loss of generality, we let $I = 1$. In addition, the forged signature must satisfy Eq. (2) i.e.

$$\begin{aligned} e\left(\sum_{i=1}^n \delta_i T_i^*, P\right) &= e\left(\sum_{i=1}^n \delta_i (h_i^* K_{ID_i}^* + R_i^*), Q_{Pub}\right) e\left(\sum_{i=1}^n \delta_i (UPK_{ID_i}^* + R_i^*), S^*\right) \\ &= e(\delta_1 (h_1^* K_{ID_1}^* + R_1^*), Q_{Pub}) e\left(\sum_{i=2}^n \delta_i (h_i^* K_{ID_i}^* + R_i^*), Q_{Pub}\right) \\ &\quad \times e\left(\sum_{i=1}^n \delta_i (UPK_{ID_i}^* + R_i^*), S^*\right) \end{aligned}$$

By our setting $K_{ID_1}^* = l_{1i}^* vP$, $S^* = l_{3i}^* P$, $R_1^* = r_1^* P$, $UPK_{ID_1}^* = w_1^* P$

$$\begin{aligned} e(\delta_1 (h_1^* K_{ID_1}^* + R_1^*), Q_{Pub}) &= e\left(\sum_{i=1}^n \delta_i T_i^*, P\right) \times \\ &\quad \left\{ e\left(\sum_{i=2}^n \delta_i (h_i^* K_{ID_i}^* + R_i^*), Q_{Pub}\right) e\left(\sum_{i=1}^n \delta_i (UPK_{ID_i}^* + R_i^*), S^*\right) \right\}^{-1} \\ \Rightarrow e(\delta_1 (h_1^* l_{1i}^* vP + r_1^* P), uP) &= e\left(\sum_{i=1}^n \delta_i T_i^*, P\right) \times \\ &\quad \left\{ e\left(\sum_{i=2}^n \delta_i (h_i^* K_{ID_i}^* + R_i^*), Q_{Pub}\right) e\left(\sum_{i=1}^n \delta_i (UPK_{ID_i}^* + R_i^*), S^*\right) \right\}^{-1} \\ \Rightarrow \delta_1 (h_1^* l_{1i}^* uvP + r_1^* Q_{Pub}) &= \sum_{i=1}^n \delta_i (T_i^* - (w_i^* + r_i^*) S^*) - \sum_{i=2}^n \delta_i (h_i^* l_{1i}^* + r_i^*) Q_{Pub} \\ \Rightarrow uvP &= \left\{ \sum_{i=1}^n \delta_i (T_i^* - (w_i^* + r_i^*) S^* - r_i^* Q_{Pub}) - \sum_{i=2}^n \delta_i (h_i^* l_{1i}^* Q_{Pub}) \right\} (\delta_1 h_1^* l_{1i}^*)^{-1}. \end{aligned}$$

Lemma 2. *If there exists a probabilistic polynomial-time bounded Type-II batch forger ADV_2 who can forge any signature of our batch of signatures under adaptive chosen message attack by asking at most q_{H_2} , q_{H_3} questions to random oracles H_2, H_3 respectively, q_{Cuser} questions to the **Create User** request oracle, q_{Rsk} questions to the **Reveal Secret Key** extraction oracle and q_{Sign} questions to the **Sign** oracle in ROM then there exists an algorithm ζ that can be used by ADV_2 to solve the CDH problem in elliptic curve group.*

Proof: Let ADV_2 is a Type-II batch forger. Suppose that ADV_2 's target identity is ID^* , and he can forge a valid signature on a message (m^*, ID^*) . Now we prove that anyone can construct an algorithm ζ who can solve the CDH problem using ADV_2 . Challenger ζ is given $(A = uP, B = vP)$ as a random instance of the CDH problem in G_{Adt} .

Initialization Phase: \mathcal{ADV}_2 chooses a random value $s \in Z_q^*$ as master secret key and sets $Q_{Pub} = sP$. \mathcal{ADV}_2 runs **Master Key Gen** to generate τ and master public key and then gives s and master public key to the challenger ζ .

Queries Phase: In this phase, \mathcal{ADV}_2 performs the oracle simulation and ζ responds to these oracles as follows.

Create User Oracle ($Cuser(ID_i)$): ζ maintains a list L_{Cuser} , which is initially empty. It contains the tuples of the form $(ID_i, UPK_{ID_i}, USK_{ID_i})$. After receiving a query on $Cuser(ID_i)$, the current UPK_{ID_i} from the list L_{Cuser} will be given if the request has been made earlier. ζ will choose a random $l_{1i} \in Z_q^*$ and sets $UPK_{ID_i} = l_{1i}P$ if $ID_i \neq ID^*$; otherwise, it sets $UPK_{ID_i} = l_{1i}B = l_{1i}vP$. In both cases, ζ sets $USK_{ID_i} = l_{1i}$. ζ gives UPK_{ID_i} and adds $(ID_i, UPK_{ID_i}, USK_{ID_i})$ to L_{Cuser} .

Reveal Secret Key Oracle ($RSK(ID_i)$): When \mathcal{ADV}_2 makes this query on $RSK(ID_i)$, if $(ID_i = ID^*)$, ζ aborts. Otherwise (if $ID_i \neq ID^*$), ζ finds the tuple $(ID_i, UPK_{ID_i}, USK_{ID_i})$ in a list L_{Cuser} , and returns USK_{ID_i} to \mathcal{ADV}_2 . If there is no tuple in L_{Cuser} , ζ makes a query on $Cuser(ID_i)$ to generate $(UPK_{ID_i} = l_{1i}P, USK_{ID_i} = l_{1i})$. ζ saves these values in L_{Cuser} , and returns $USK_{ID_i} = l_{1i}$.

Queries on Oracle: $H_2(H_2(m_i, ID_i, UPK_{ID_i}, R_i))$: ζ maintains a list L_2 , which is initially empty. It contains the tuples of the form $(m_i, ID_i, UPK_{ID_i}, R_i, l_{2i})$. After receiving H_2 query on $(m_i, ID_i, UPK_{ID_i}, R_i)$, if a tuple $(m_i, ID_i, UPK_{ID_i}, R_i, l_{2i})$ exists on L_2 , ζ returns l_{2i} . otherwise, ζ picks a random $l_{2i} \in Z_q^*$ and returns l_{2i} . ζ adds $(m_i, ID_i, UPK_{ID_i}, R_i, l_{2i})$ to L_2 .

Queries on Oracle: $H_3(H_3(\Delta_i))$: ζ maintains a list L_3 , which is initially empty. It contains the tuples of the form (Δ_i, S_i, l_{3i}) . After receiving a query on $H_3(\Delta_i)$, ζ gives the same answer from L_3 , if the query has been made earlier. Otherwise, ζ picks a random $l_{3i} \in Z_q^*$ and computes $S_i = l_{3i}uP$, returns S_i . ζ adds (Δ_i, S_i, l_{3i}) to L_3 .

Signing Oracle: When \mathcal{ADV}_2 makes this query on (ID_i, m_i) , ζ chooses $r_i, h_i \in Z_q^*$ and computes $R_i = r_iP - h_iK_{ID_i}$. If the tuples containing h_i already exists in list L_2 , then ζ chooses another $r_i, h_i \in Z_q^*$ and tries again. Set $h_i = H_2(m_i, ID_i, UPK_{ID_i}, R_i)$ and $S_i = H_3(\Delta_i)$. Then ζ computes $T_i = l_{3i}(R_i + UPK_{ID_i}) + r_iQ_{Pub}$. Finally ζ responds to \mathcal{ADV}_2 with $\Omega_i = (R_i, T_i)$. (R_i, T_i) is a valid signature on message m_i as it satisfies Eq. (1).

Forgery: Hence, \mathcal{ADV}_2 forges a valid signature $\Omega^* = (R^*, T^*)$ on messages m^* under ID^* . Suppose ζ can construct n valid signatures $(\Omega_i^*)_{i=1ton}$ on messages $(m_i^*)_{i=1ton}$ of the signers under $(ID_i^*)_{i=1ton}$ and the corresponding $(UPK_{ID_i}^*)_{i=1ton}$ of n users $(U_i)_{i=1ton}$ with a state of information Δ^* by \mathcal{ADV}_2 such that

- i. BV holds.
- ii. There exists $I \in \{1, 2, 3, \dots, n\}$ such that \mathcal{ADV}_2 has not asked the Partial Secret Key queries for ID_I^* and \mathcal{ADV}_2 has not asked the Sign oracle query.

Without loss of generality, we let $I = 1$. In addition, the forged signature must satisfy

$$\begin{aligned}
e\left(\sum_{i=1}^n \delta_i T_i^*, P\right) &= e\left(\sum_{i=1}^n \delta_i (h_i^* K_{ID_i}^* + R_i^*), Q_{Pub}\right) e\left(\sum_{i=1}^n \delta_i (UPK_{ID_i}^* + R_i^*), S^*\right) \\
&= e\left(\sum_{i=1}^n \delta_i (h_i^* K_{ID_i}^* + R_i^*), Q_{Pub}\right) e(\delta_1 (UPK_{ID_1}^* + R_1^*), S^*) \\
&\quad \times e\left(\sum_{i=2}^n \delta_i (UPK_{ID_i}^* + R_i^*), S^*\right)
\end{aligned}$$

By our setting $UPK_{ID_1}^* = l_{1i}^* vP$, $S^* = l_{3i}^* uP$, $R_1^* = r_1^* P$, $UPK_{ID_i}^* = l_{1i}^* P$, $R_i^* = r_i^* P$ and $Q_{Pub} = sP$.

$$\begin{aligned}
e(\delta_1 (UPK_{ID_1}^* + R_1^*), S^*) &= e\left(\sum_{i=1}^n \delta_i T_i^*, P\right) \times \\
&\quad \left\{ e\left(\sum_{i=1}^n \delta_i (h_i^* K_{ID_i}^* + R_i^*), Q_{Pub}\right) e\left(\sum_{i=2}^n \delta_i (UPK_{ID_i}^* + R_i^*), S^*\right) \right\}^{-1} \\
&\Rightarrow \delta_1 (l_{1i}^* l_{3i}^* uvP + r_1^* l_{3i}^* uP) = \left\{ \sum_{i=1}^n \delta_i (T_i^* - (h_i^* UPSK_{ID_i}^* + sR_i^*)) - \sum_{i=2}^n \delta_i (UPK_{ID_i}^* + R_i^*) l_{3i}^* u \right\} \\
&\Rightarrow uvP = \left\{ \sum_{i=1}^n \delta_i (T_i^* - (h_i^* UPSK_{ID_i}^* + sR_i^*)) - \sum_{i=2}^n (\delta_i (l_{1i}^* uP + r_i^* uP) l_{3i}^*) - \delta_1 r_1^* l_{3i}^* uP \right\} (\delta_1 l_{1i}^* l_{3i}^*)^{-1}. \square
\end{aligned}$$

4 Efficiency Analysis

In this part we compare our scheme with the relevant schemes [5, 7] in terms of security, signature length, verification cost and computation cost. We consider the experimental results presented in [2, 4, 8, 12] for various cryptographic operations and their conversions. Table 2 presents these conversions and detailed comparison with relevant schemes is presented in Table 3.

From the following Table 3, the communicational and computation cost of our scheme is more efficient than C. I. Fan et al. [5] and almost as efficient as Geng et al. [7] scheme. But our scheme is more secure than all other schemes since the security reduction do not use Forking lemma.

Table 2. Notations and descriptions of various cryptographic operations and their conversions

Notations	Descriptions
T_M	Time required to compute modular multiplication operation
T_E	Time required to compute the elliptic curve point multiplication (Scalar multiplication in G_{Adt}): $T_E = 29T_M$
T_P	Time required to compute the bilinear pairing in G_{Mlt} : $T_P = 87T_M$
T_H	Time required to compute a map to point hash function: $1T_H = 1T_E = 29T_M$
T_A	Time required to compute the elliptic curve point addition (point addition in G_{Adt}): $T_A = 0.12T_M$

Table 3. Comparison of the proposed scheme with the related schemes

Scheme	Signing cost	Batch verification cost	Sign. length	Without forking lemma
M. Geng et al. (2009)	$3T_E + 1T_A + 1T_H$	$3T_P + 4nT_E + (4n - 3)T_A + 1T_H$	$2 G_{Adr} $	No
C.I. Fan et al. (2014)	$5T_E + 3T_A + 1T_H$	$3T_P + 5nT_E + (5n - 3)T_A + (n + 1)T_H$	$4 G_{Adr} $	No
Our scheme	$4T_E + 2T_A + 1T_H$	$3T_P + 4nT_E + (5n - 3)T_A + 1T_H$	$2 G_{Adr} $	Yes

5 Conclusions

In this paper, we have proposed a novel and secure batch verifiable CLS scheme using bilinear pairings over elliptic curves. This scheme verifies the batch of signatures using small exponent test. The presented scheme is unforgeable under CDH assumption. Moreover the scheme is proved without using Forking lemma which results our scheme is tightly secure. Thus we can apply our scheme in practical environments such as Internet of Things environments associated with intelligent transportation systems (ITS) to manage traffic caused by vehicles in a metropolitan area etc.

Acknowledgements. The authors are grateful and sincerely thank the reviewers for their valuable suggestions. This work is supported by WOS-A, DST, Govt. of India under the grant No.SR/WOS-A/PM-1033/2014 (G), WOS-A, DST.

References

1. Al-Riyami, Sattam S., Paterson, Kenneth G.: Certificateless public key cryptography. In: Laihi, Chi-Sung (ed.) ASIACRYPT 2003. LNCS, vol. 2894, pp. 452–473. Springer, Heidelberg (2003)
2. Barreto, Paulo S.L.M., Kim, Hae Y., Lynn, Ben, Scott, Michael: Efficient algorithms for pairing-based cryptosystems. In: Yung, Moti (ed.) CRYPTO 2002. LNCS, vol. 2442, pp. 354–368. Springer, Heidelberg (2002)
3. Bellare, Mihir, Garay, Juan A., Rabin, Tal: Fast batch verification for modular exponentiation and digital signatures. In: Nyberg, Kaisa (ed.) EUROCRYPT 1998. LNCS, vol. 1403, pp. 236–250. Springer, Heidelberg (1998)
4. Cao, X., Kou, W., Du, X.: A Pairing –free Identity Based Authenticated Key Agreement Protocol with Minimal Message Exchanges. Inf. Sci. **180**(15), 2895–2903 (2010)
5. Fan, C.I., Ho, P.H., Tseng, Y. F.: Strongly secure certificateless signature scheme supporting batch verification In: Mathematical Problems in Engineering, vol. 2014, Article ID 854135, 11 pages. Hindawi Publishing Corporation. <http://dx.doi.org/10.1155/2014/854135>. (2014)
6. Fiat, A.: “Batch RSA,” in Advances in cryptology-CRYPTO, pp. 175–185. (1990)

7. Geng, M., Zhang, F.: Batch verification for certificateless signature schemes. In: Proceedings of the International Conference on Computational Intelligence and Security (CIS 2009), pp. 288–292, December. 2009
8. MIRACL Library. <http://certivox.org/display/EXT/MIRACL>
9. Naccache, David, Raihi, DavidM, Vaudenay, Serge, Raphaeli, Dan: Can D.S.A. be improved? In: De Santis, Alfredo (ed.) EUROCRYPT 1994. LNCS, vol. 950, pp. 77–85. Springer, Heidelberg (1995)
10. Pointcheval, D., Stern, J.: Security arguments for digital signatures and blind signatures. *J. Cryptology* **13**(3), 361–369 (2000)
11. Shim, K.A.: Security models for certificateless signature schemes revisited. *Inf. Sci.* **296**, 315–321 (2015)
12. Tan, S-Y., Heng, S-H., Goi, B-M.: Java implementation for pairing-based cryptosystems. In: Taniar, D., Gervasi, O., Murgante, B., Pardede, E., Apduhan, B.O. (eds.) ICCSA 2010, Part IV. LNCS, vol. 6019, pp. 188–198. Springer, Heidelberg (2010)
13. Yoon, HyoJin, Cheon, Jung Hee, Kim, Yong-Dae: Batch verifications with id-based signatures. In: Park, Choon-sik, Chee, Seongtaek (eds.) ICISC 2004. LNCS, vol. 3506, pp. 233–248. Springer, Heidelberg (2005)

System and Network Security

Security Requirements Elicitation and Modeling Authorizations

Rajat Goel^(✉), Mahesh Chandra Govil, and Girdhari Singh

Malaviya National Institute of Technology Jaipur, Jaipur, India
rajatgoel185@gmail.com

Abstract. Today security is almost inevitable for any software. To achieve this, the security requirements of the software ought to be efficiently modeled. However, existing modeling languages like Unified Modeling Language have certain limitations when it comes to modeling non-functional requirements like security. Most of the software of present era are hosted on internet or cloud and involve heavy exchange of crucial information between great multitudes of users. In this backdrop security becomes an obvious prerequisite. This paper proposes a methodology to elicit security requirements from all concerned stakeholders, assess security level required for every software asset and present this security assessment through easy but effective diagrams.

1 Introduction

Security is a prime requirement for almost any software now. To make a software secure researchers [1, 2] have stressed upon considering security within the development life cycle. Any such consideration will pay higher dividends if included during the initial phases i.e. requirements and design [3].

Requirements engineering plays a crucial role in secure software development. Obtaining perfect and precise requirements is not an easy task [4]. According to Haley et al. [5] there has been no satisfactory integration of security requirements engineering into requirements engineering, so far.

For modeling the requirements effectively, an adequate modeling language is necessary. Modeling facilitates an unambiguous communication between stakeholders, especially the users or customers, who give priority to diagrammatic representation over words [6].

Unified Modeling Language (UML) [7, 8] is one such popular language but it is not competent enough for modeling non-functional requirements. Researchers [9–12] have argued the abilities and usefulness of UML and its diagrams.

All this, has motivated a new improved methodology for eliciting and modeling security requirements. This methodology mainly addresses two aspects. Firstly, the level of security for different assets of the system and secondly, the authorizations a particular stakeholder has on the assets. A case study of hotel management system is considered here for the sake of explanation and does not necessarily represent the real-life scenario completely.

2 Methodology

In this methodology, all types of stakeholders [13] of the software rank assets under five security parameters [14] viz. Authentication, Non-repudiation, Confidentiality, Integrity and Authorization. The term –Asset has been limited to data items, in this research work. Stakeholders rank only the assets related to them. The methodology has five steps namely, Elicitation and Resolution, Mapping Stakeholders to Assets, Developing Association Diagrams, Rating and Calculation, and Design. Sections 3 to 7 describe these steps. Section 8 discusses the related works. Section 9 concludes the paper with future directions.

3 Elicitation and Resolution

The process is initiated by the representatives from the development and the client side along with the business expert, who may be independent or belong to a third party. All of these are referred to as Analysts in this work. The analysts sit together to identify more assets and stakeholders. On identification of more stakeholders, all the stakeholders (older and newer) again meet to further identify the stakeholders and assets. This incremental process goes on until identification of new assets and stakeholders ceases.

Analysts carefully analyze the identified assets and stakeholders to resolve any ambiguities or inconsistencies. For example, in the elicitation process two stakeholders ‘Attendant’ and ‘Waiter’ could be identified. If the hotel considers both of them as same, one of these will be discarded and only one is accepted. Similar treatment is required for the assets. If the asset list contains both ‘Bill’ and ‘Invoice’, then any one may be chosen. Generalization and specialization are the other issues to be resolved. For instance, if both ‘Attendant’ and ‘Guard’ are elicited as stakeholders, they may be generalized as ‘Staff Member’ or if ‘Staff Member’ exists, it could be specialized vice-versa. Such decisions are specific to the client’s requirements.

At the end of this phase, the final sets for stakeholders and assets are obtained denoted by S and A respectively. For the case study considered, sets S and A are:

S = Manager, Receptionist, Customer, Attendant, Chef, Analyst

A = Complaint Record, Bill, Customer Name, Customer Address, Customer Ph. no., Room No. Room Type, Order

Here, the complaint record is a registered maintained by the hotel to note down the feedback and/or complaints of the customers. Room type means the type of the room i.e. single, double etc. Order is any food item(s) ordered by the customer.

An advisory group is formed that includes one stakeholder of each type along with the analysts. It is formed with an intension, to provide regular inputs to the development team whenever necessary throughout the process of development.

4 Mapping Stakeholders to Assets

The analysts map the stakeholders to assets. Mapping for the hotel management case study is shown in Table 1. The shaded cells denote that the stakeholder uses or is affected by the corresponding asset. For example, asset ‘Bill’ is related to customer and not with the chef.

Table 1. Stakeholders mapped to assets

Assets	Stakeholders				
	Manager	Receptionist	Customer	Attendant	Chef
Complaint Record					
Bill					
Customer Name					
Customer Address					
Customer, Ph. no.					
Room No.					
Room Type					
Order					

5 Developing Association Diagrams

According to Table 1, association diagrams are drawn that graphically denote the association between the stakeholders and assets, pictorially. Figure 1 shows the association between a ‘Receptionist’ and the ‘Room No.’ i.e. stakeholder and asset respectively.

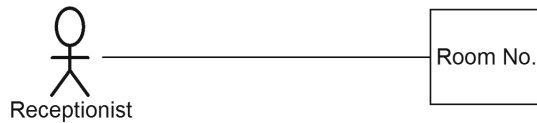


Fig. 1. Stakeholder-asset association diagram

The association diagram can be extended to show multiplicity. A stakeholder may be associated with more than one asset, as seen in Fig. 2. It is apparent in the figure that associated assets are relevant to the receptionist as per Table 1. Conversely, Fig. 3 depicts the association of an asset with its related stakeholders. It shows the stakeholders related to asset ‘Room No.’. In other words, Fig. 2 is stakeholder-centric while Fig. 3 is asset-centric.

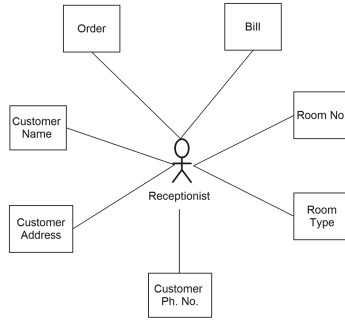


Fig. 2. Association between a stakeholder and multiple assets

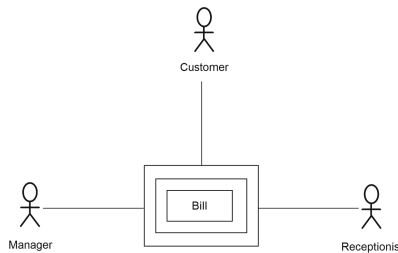


Fig. 3. Association between an asset and multiple stakeholders

6 Rating and Calculation

A Rating Table is developed. As every stakeholder may not be conversant with all the assets and may not be technically-aware, he is allowed to rank only the assets relevant to him and those too on the parameters selected for him by the analysts. Prior to ranking, the participants are made to understand the parameters, the intended use of software and the assets.

6.1 Rating Table

The stakeholder rates the assets relevant to him/her under the parameters through a rating table. There is one rating table for each type of stakeholder. It is derived from Table 1. It may be possible, in some certain critical applications that a particular stakeholder may be refrained from rating an asset under a particular parameter. The values of the authorization parameter, for every asset with every stakeholder, are inserted by only the analysts. Table 2 is the table for the receptionist. It shows that a Receptionist can rank assets over the 'Confidentiality' parameter. NA means 'Not Applicable'.

Table 2. Rate table for receptionist

Assets	Parameters				
	Authenticition	Confidentiality	Integrity	Non-repudiation	Authorization
Complaint record	NA	1	NA	NA	1
Bill	NA	1	NA	NA	2
Customer name	NA	1	NA	NA	1
Customer address	NA	2	NA	NA	1
Customer ph. no	NA	3	NA	NA	1
Room no	NA	1	NA	NA	3
Room type	NA	2	NA	NA	1
Order	NA	2	NA	NA	3

6.2 Rating Values

The stakeholders provide values 1, 2 and 3 to the assets. Final rating of each asset is calculated by the average of ratings submitted for that asset by all concerned stakeholders under all applicable parameters. The range of average values and corresponding ratings are shown in Table 3. In some cases, value of a parameter for all assets may be fixed as high. For example, value of integrity in the banking applications. NA is not considered in any calculation.

Table 3. Rating values

Rank	Value	Range
Low	1	1 to 1.66
Medium	2	>1.66 to 2.33
High	3	>2.33

6.3 Calculating Individual Parameter Ratings

Equation (1) is used for the calculation of averages of Authentication (P_1), Confidentiality (P_2), Integrity (P_3) and Non-repudiation (P_4) parameters individually denoted by $Average_{atn}$, $Average_{cnf}$, $Average_{int}$ and $Average_{nrp}$ respectively. P is taken as the set of five parameters (Authorization parameter is denoted by P_5). For any $P_q \in P$ for k^{th} asset a_k and n stands for the number of members in the consultation or the number of rank matrices. Authorization parameter is dealt in a different manner, which will be discussed in Subsects. 6.5 and 6.6. It can be said that $Average_p$ signifies the average of all values of an asset k in all the rank matrices under parameter p.

$$Average_p = value(s_i a_k p_q) / n \quad \text{where, } a \in A, s \in S, p \in P \quad (1)$$

6.4 Calculating Security Ratings

Security average of an asset is denoted by $Average_{sec}$ and calculated through (2) for each asset. It is the average of averages of authentication, integrity, non-repudiation and confidentiality parameters, whichever are applicable, for that asset.

$$Average_{sec} = Average(Average_{atn}, Average_{cnf}, Average_{int}, Average_{nrrp}) \quad (2)$$

From all the rating tables, filled by all stakeholders, the $Average_{sec}$ is calculated for all assets and accordingly the ranks are assigned. This is shown in Table 4 for the case study considered.

Table 4. Asset security ranks

Asset	$Average_{sec}$	Rank
Complaint record	1.12	Low
Bill	2.83	High
Customer name	2.25	Medium
Customer address	2.49	High
Customer ph. no	2.78	High
Room no	1.34	Low
Room type	1.10	Low
Order	1.89	Medium

6.5 Authorization Parameter

The authorization parameter needs a different treatment. Authorization rights are determined for a stakeholder with his/her relevant assets. So, ratings furnished by other stakeholders are immaterial. Hence, average calculation like other parameters will be misleading. For this, the rating tables must be considered individually for each stakeholder. The Authorization rank of each asset in a particular rank matrix will signify the authorization rights of the stakeholder to that asset. Significance of Authorization ratings is discussed in next subsection.

6.6 Implications of Ranks

Once the ratings are identified for security, adequate measures can be taken. Expenditure of time, effort and money can be prioritized and managed well. Accordingly security techniques can be selected. A ‘Low’ security rating will stand for adopting security measures like username and password. In case of

‘Medium’ one can use a smart card and ‘High’ would suggest the use of biometric mechanism. The choice of these measures will also be affected by the cost to be incurred.

An Authorization rating of an asset will expound the access rights its relevant stakeholder(s) possesses for it. A ‘Low’ rating means Read (R) Permission and ‘High’ means ‘Write’ (W). ‘Medium’ rating signifies ‘Write with Permission’ (WP) i.e. Write with permission of another stakeholder (usually, facilitator of the service). WP is an innovation introduced in this methodology. For instance, through Table 2 it can be derived that a Receptionist has Authorization value as 1 for Complaint Record, meaning he/she has low rights or can only ‘Read’ this asset.

7 Design

Based on the asset ranks the Asset-rank diagrams and their extension, the ‘Authorization and Security’ diagrams are built. The ‘Authorization and Security’ diagrams are supplemented by textual templates for clarity and avoidance of ambiguities.

7.1 Asset-Rank Diagram

The rating of assets is shown diagrammatically by the number of concentric rectangles around them. Figures 4, 5 and 6 show assets with low, medium and high ratings respectively. As seen in Table 4, only confidentiality and authorization parameters are applicable to the receptionist. So Figs. 4 and 5 denote the confidentiality ranks. Authorization ranks is dealt in next subsection.



Fig. 4. Asset with low confidentiality rank

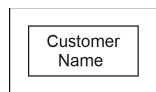


Fig. 5. Asset with medium confidentiality rank

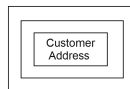


Fig. 6. Asset with high confidentiality rank

7.2 Authorization-Security Diagram

Authorization-Security Diagram has two flavors.

Stakeholder-Oriented. Figure 7 is a stakeholder-oriented Authorization-Security Diagram, which supplements the association diagram shown in Fig. 2. Assets are shown with their security ratings. The authorization right of the stakeholder to a particular asset is written on the connector. In this diagram, it is seen that the Receptionist has a right to ‘Write with Permission’ on ‘Bill’. The permission will be granted by the Manager. Bill is a high security asset. Such diagrams are made for every stakeholder type. Table 5 is the template for Fig. 7.

To avoid cluttering, assets with same authorization rights and same security rating can be clubbed. Figure 8 is equivalent to Fig. 7 but clubs similar assets.

Asset-Oriented. Figure 9 is an Asset-oriented Authorization-Security Diagram, which is a supplementation of association diagram shown in Fig. 3.

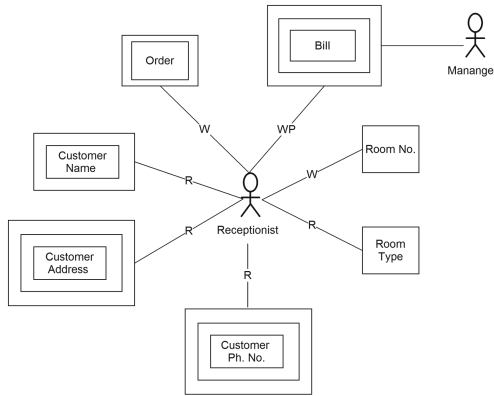


Fig. 7. Stakeholder-oriented Authorization-Security Diagram

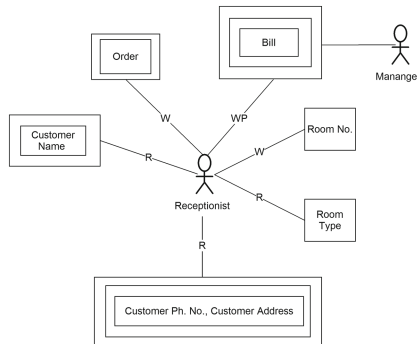


Fig. 8. Stakeholder-oriented Authorization-Security Diagram with clubbed assets

An Asset is taken at the centre with its security rating and connected to all of its related stakeholders. The authorization right possessed by a particular stakeholder on the asset is written on the respective connector. In this diagram, it is seen that a Customer has only ‘Read’ right on the Bill. Table 6 is the corresponding template.

Table 5. Stakeholder-oriented authorization-security template

Stakeholder: Receptionist			
Assets	Security rank	Authorization rights	Permitting stakeholders
Order	Medium	W	NA
Bill	High	WP	Manager
Room no.	Low	W	NA
Room type	Low	R	NA
Customer ph. no.	High	R	NA
Customer address	High	R	NA
Customer name	Medium	R	NA

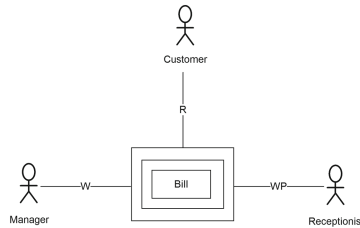


Fig. 9. Asset-oriented authorization-security diagram

Table 6. Asset-oriented authorization-security template

Asset: Bill		
Security Rating: High		
Stakeholders	Authorization rights	Permitting stakeholders
Customer	R	NA
Manager	W	NA
Receptionist	WP	Manager

8 Related Works

Other works in the direction of improving requirement elicitation and modeling are Security-aware Software Development Life Cycle (SaSDLC) [15], ADORA [16], addition of context view [17], formalization of UML [18], Proliferation of UML [19] and a text-heavy activity diagram [20]. Misuse Cases [21] improve use cases to represent security threats. CORAS [22] is a model-based security analysis method that also includes the concepts of assets and stakeholders. There is a sort of core group consisting of user, developer, security expert and decision maker. This method also advocates an initial meeting of stakeholders to identify assets. In contrast to SecREAD, CORAS doesn't rank assets on security parameters but on the likelihood of threats upon them. However, Controlled requirement specification (CORE) [6] is another method that considers views of all stakeholders but non-functional requirements don't get due importance. Analysts have a limited role which inhibits proper requirement elicitation. Issue-based Information System (IBIS) is a formal elicitation process but it doesn't resolve conflicts and is devoid of graphics and iteration [23]. UMLsec [24] is an attempt to extend UML for security but it lacks an elicitation and assessment procedure. The above techniques and more have been discussed at length in Goel et al. [3].

9 Conclusion

The proposed methodology is unique in several ways. It gives due importance to stakeholders' views. Ranking is performed by all stakeholders on more number of parameters. The notion of relevance has been introduced for the first time. The design is conducted in accordance with a well-structured requirement elicitation and assessment procedure. The methodology includes all the good practices advocated by researchers to achieve a better software product like iteration in requirement elicitation [25,26], client's representation in the development team [27] and involvement of stakeholders in the development of software [28]. The rank-based Authorization and Security Diagram, proposed in this methodology, evolves gradually from its constituent elements. It is easier to understand and is able to precisely state the aspirations of all stakeholders of the software. The templates avoid any ambiguity.

In future, the methodology will be applied on more complex real-world scenarios to check its applicability. New diagrams will be proposed for other security parameters. More options for obtaining security rankings like worst case scenarios and threat modeling can be explored.

References

1. Shreyas, D.: Software engineering for security - towards architecting secure software. In: ICS 221 Seminar in Software Engineering, University of California, Irvine, pp. 1–12 (2001)

2. Lindvall, M., Basili, V.R., Boehm, B.W., et al.: Empirical findings in agile methods. In: XP Universe and Agile Universe Conference on Extreme Programming and Agile Methods, pp. 197–207 (2002)
3. Goel, R., Govil, M.C., Singh, G.: Imbibing security in software development life cycle: a review paper. In: Afro - Asian International Conference on Science, Engineering and Technology, pp. 593–599 (2015)
4. Van Lamsweerde, A.: Goal-oriented requirements engineering: from system objectives to UML models to precise software specifications. In: 25th International Conference on Software Engineering (2003)
5. Haley, C.B., Laney, R., Moffett, J.D., et al.: Security requirements engineering: a framework for representation and analysis. *IEEE Trans. Softw. Eng.* **34**(1), 133–153 (2008)
6. Mullery, G.P.: CORE-a method for controlled requirement specification. In: 4th International Conference on Software Engineering, pp. 126–135 (1979)
7. Booch, G., Rumbaugh, J., Jacobson, I.: Unified Modeling Language User Guide. Pearson Education India, Noida (2005)
8. Booch, G., Rumbaugh, J., Jacobson, I.: Unified Modeling Language User Guide. Addison Wesley, Boston (2015)
9. Choppy, C., Reggio, G.: Requirements capture and specification for enterprise applications: a UML based attempt. In: Australian Software Engineering Conference, pp. 19–28 (2006)
10. Konrad, S., Goldsby, H., Lopez, K., Cheng, B.H.C.: Visualizing requirements in UML models. In: International Workshop Visualization Requirements Engineering, p. 1 (2007)
11. Dobing, B., Parsons, J.: How UML is used. *Commun. ACM* **49**(5), 109–113 (2006)
12. Dobing, B., Parsons, J.: Dimensions of UML diagram use: a survey of practitioners. *J. Database Manag.* **19**(1), 1–18 (2008)
13. Pressman, R.S.: *Software Engineering a Practitioner's Approach*. McGraw-Hill, New York (2001)
14. Forouzan, B.A.: *Data Communications and Networking*. McGraw-Hill, New York (2007)
15. Talukder, A.K., Maurya, V.K., Santhosh, B.G., et al.: Security-aware software development life cycle (SaSDLC)- processes and tools. In: IFIP International Conference on Wireless Optical Communications Networks, pp. 1–5 (2009)
16. Glinz, M.: Problems and deficiencies of UML as a requirements specification language. In: International Workshop on Software Specification and Design, pp. 11–22 (2000)
17. Woods, E.: Harnessing UML for architectural description: the context view. *IEEE Softw.* **31**(6), 30–33 (2014)
18. Chanda, J., Kanjilal, A., Sengupta, S., Bhattacharya, S.: Traceability of requirements and consistency verification of UML use case, activity and class diagram: a formal approach. In: International Conference on Methods Models in Computer Science, pp. 1–4 (2009)
19. Kobryn, C.: UML 3 and the future of modeling. *Softw. Syst. Model.* **3**(1), 4–8 (2004)
20. Samuel, B.M., Watkins III, L.A., Ehle, A., Khatri, V.: Customizing the representation capabilities of process models: understanding the effects of perceived modeling impediments. *IEEE Trans. Softw. Eng.* **41**(1), 19–39 (2015)
21. Sindre, G., Opdahl, A.L.: Eliciting security requirements with misuse cases. *Requir. Eng.* **10**, 34–44 (2005)

22. Stolen, K., Braber, F.D., Dimitrakos, T., et al.: iTrust Workshop (2002)
23. Kishore, S., Naik, R.: Software Requirements and Estimation. Tata McGraw-Hill Education, New York (2001)
24. Jürjens, J.: UMLsec: extending UML for secure systems development. In: Jézéquel, J.-M., Hussmann, H., Cook, S. (eds.) UML 2002. LNCS, vol. 2460, pp. 412–425. Springer, Heidelberg (2002)
25. Sabahat, N., Iqbal, F., Azam, F., Javed, M.Y.: An iterative approach for global requirements elicitation: a case study analysis. In: International Conference on Electronics and Information Engineering, pp. 361–366 (2010)
26. Kasirun, Z.M., Salim, S.S.: Focus group discussion model for requirements elicitation activity. In: International Conference on Computer and Electrical Engineering, pp. 101–105 (2008)
27. Wäyrynen, J., Bodén, M., Boström, G.: Security engineering and extreme programming: an impossible marriage? In: Zannier, C., Erdogmus, H., Lindstrom, L. (eds.) XP/Agile Universe 2004. LNCS, vol. 3134, pp. 117–128. Springer, Heidelberg (2004)
28. Kamata, M.I., Tamai, T.: How does requirements quality relate to project success or failure? In: Requirements Engineering Conference, pp. 69–78 (2007)

Two Level Signature Based Authorization Model for Secure Data Warehouse

Anjana Gosain¹ and Amar Arora^{2(✉)}

¹ USICT, Guru Gobind Singh Indraprastha University, Delhi, India

² National Informatics Centre, DeitY, Government of India, Delhi, India
amar.arora@nic.in

Abstract. Data Warehouse (DW) security has emerged as a crucial aspect since for the sake of high availability data warehouses started connected to internet. In order to comply with the security requirements, the authentication of legitimate users by verification of user credentials like username, password, etc. has become a standard. On successful verification, different variations of Role Based Access Control (RBAC) techniques are being used restricting user access to the facts and dimensions. But these RBAC's can only restrict the user access as per their respective roles and there is no check on the behavior pattern of the user access. In this paper, a two level signature based behavior analysis model has been introduced to keep a check on the user's access pattern. At the first level, the user provides its authentication credentials. On successful verification of these credentials, the user has been allowed to access elements as per its role. Once the user tries to access the DW elements his access pattern will be recorded to form usage access signature. Over the period of time user access profile is created which is used to match the signature of the user. If in case, the user's signature does not fit in the user access profile created over the period of time, the second level of verification will be performed in a form of secret question etc. The user query will be processed only on successful clearing of the second authentication level; else the current query will be suspended with regret message from the system. This further strengthens the security of the DW even on the compromise of the user's initial entry credentials.

Keywords: Data warehouse security · Signature based authorization · Two level authorization

1 Introduction

Data Warehouse is a collection of subject-oriented, time variant, integrated, non-volatile data which facilitates the management for making a knowledgeable decision (Inmon 1991). It needs special task of extracting, transforming and load procedures on historical data to help decision makers to improve their business process (Becker et al. 2008). Its global reach and web accessibility has made confidentiality as one of the major issue (Dhillon 2000). To deal with confidentiality and authentication, user credentials such as username, password, etc. are being used over the time. Once the users authenticate themselves by providing essential security information Audit Rules (AR) (Belén et al.

2012) are used to provide enhanced security features. It prevents misuse of authorization by logging the frustrated attempts over several multidimensional elements. But these audit logs are only used to identify the attacker who has tried to exploit the system by attempting to access the elements for which the user is not authorized. Here our intention is to develop extra security level for those cases, where the user is accessing its authorized areas but it is deviating from the user access pattern which has been developed over the period of time. It prevents the unauthorized access for most of the DW elements, even though the security parameters such as username, password for accessing the system stands compromised.

2 Background and Related Work

In this section the background and related work is organized into the following subsections: (1) DW Security Review, (2) Access Control in Data Warehouse.

2.1 DW Security Review

In an attempt to make data warehouse more secure, an adapted mandatory access control for OLAP-cubes based security approach is presented (Kirkgoze et al. 1997). The primary advantage of using this approach is its flexibility of assigning roles to different virtual sub-cubes. Metadata, which describes the contents of the data warehouse (Berson and Smith 1997) can also be used to data warehouse environment's security mechanism. Here, metadata is composed of access rules along with corresponding information about security subjects & objects. Over the period of time, steady growth in the number of OLAP necessitates the requirement of proper access control mechanisms, which ensures the confidentiality of the sensitive data (Santos et al. 2011). Some commercial systems (Cognos 1998; Oracle 1998; Chase et al. 1999; Microsoft 1999; MicroStrategy 2000) do provide mechanisms to cope with these requirements; but these approaches are highly proprietary. The solutions for confidentiality problems regarding DW's are also being discussed and an extension of UML for the secure DW is provided (Fernández-Medina 2004). It allows designers to specify main security aspects in the conceptual MD modeling, thus resulting in the design of a secure DW system. Various other approaches which uses UML extension for the DW Security (Fernandez-Medina et al. 2005; Villarrol et al. 2006; Eduardo et al. 2006; Eduardo et al. 2007; Emilio et al. 2009; Salem et al. 2012) have also been proposed. One of the solutions on DW security (Lopes et al. 2014) investigates the method for encrypting and querying a DW hosted in a cloud, but it leads to high amount of computation overhead at the server side.

2.2 Access Control in Data Warehouse

The Role Based Access Control (RBAC) (Sandhu et al. 1996) is one of the best ways to control the access of DW entities among the variety of users and the privileges associated with them. Once all the user-roles are populated into the database, the formulation of role-based rules are performed, followed by implementation of workflow engine

modules. Then through these elements, role-based privileges can be quickly entered and updated across multiple systems, platforms, applications and geographic locations. Here RBAC provides companywide control process for managing data and resources (Iyer et al. 2007). Although RBAC is widely used and is capable of handling the entire system, there are some issues like the unclear definition of groups and user, and no mention of duties along with roles. As a solution an Extended RBAC for secure data warehousing has been proposed (Iyer et al. 2007). Another extension to the RBAC has been proposed as Temporal RBAC (TRBAC) which allows temporal restrictions on roles themselves, user-permission assignments (UA), permission-role assignments (PA) and role hierarchies (RH) (Uzun et al. 2014). In recent work another Role Based Access Technique is introduced which is applied to summarize data (Ali et al. 2014). The restrictions on the basis of summarized data classify the user access on the basis of a level of summarization they are authorized to. In all the above research studies RBAC is in place to make sure that user is accessing only those sections of the DW for which it has been authorized. But there is no check on its behavior once the initial verification has been performed and the user accesses DW elements as per its role. So, if any intruder enters the system with an authorized user credentials, RBAC will not be able to restrict the user access. In order to overcome this issue, we propose a two-level authorization system where the user access pattern will be tracked and monitored over a period of time once the initial verification of authentication has been done. When the user deviates from its normal access behavior, then the second level of authorization will be activated to verify its authentication by some different method like secret question etc. In a recent proposal, an intrusion detection mechanism for data warehouse has been proposed (dos Santos 2014). In this research work DW-DIDS (Data Warehouse Database Intrusion Detection system based on the analysis of user actions at the SQL command level etc., but in this article also there is no double check of the authenticity of the user if it's actions deviates from role profile.

3 Motivating Example

To illustrate our work let us consider an example, where the authentication details of the Managing Director of a banking DW which is authorized to access most of the facts and dimension at all the summarization levels has been compromised. Now this intruder who enters the system through compromised details will be able to access the entire banking DW elements including all financial transactions throughout the time span. Here, the RBAC can have very little impact as the user's privilege enjoys the maximum access. So the intruder can access most of the financial secrets of the bank irrespective of the existing RBAC protection. This creates a different kind of threat to the entire DW; it becomes more crucial when the cost of leakage of information is too high. Thus, if the user access signature has been maintained over a time interval and analyzed, any suspicious behavior can be detected and the user can be asked to provide the second level of authentication in the form of a secret question. So the user credentials which stand compromised would not be a threat to entire DW information for which the user is authorized and damage will only be limited to the DW elements which matches the

user’s accessed signature profile. The proposed system prevents most of the role wise accessible information, even if some user credential has been compromised. The second level of authorization also helps in reducing the false positive thus providing more robustness to the role profile created over the time (Fig. 1).

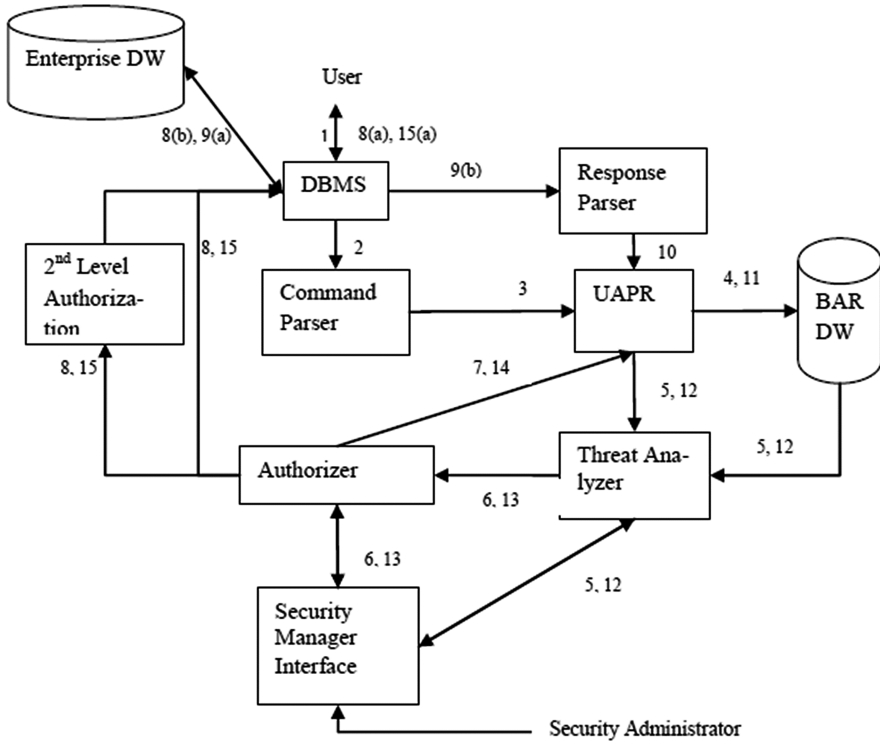


Fig. 1. Two Level Authorization Model for Accessing Data Warehouse

4 Authorization Model

Here two-level authorization model has been proposed which allows system to ask for second level authorization whenever a suspicious activity is detected. The working of the model is explained below:

1. At the first instance, user initiates a DW query to the DBMS after verification of its initial authentication details.
2. Once the DBMS has authorized the initial entry into the system, DBMS passes query to the command parser for converting it into the query tokens.
3. The details of the query and the details like username, CPU time, response size, processed rows, processed columns, etc. forms a signature in UAPR (User Access Pattern Recorder).

4. The UAPR updates the signature in the BAR (Behavioral Analysis Repository) to form a user access behavior profile which gets refreshed by completion of every request and response.
5. Then UAPR passes the control to threat analyzer to analyze the current request on the basis of updated signature available in BAR DW and some external inputs if any from security manager via security manager interface. The calculated similarity between the current and user access profile classifies query as authorized or unauthorized.
6. The threat analyzer sends its output to the authorizer for the decision whether to initiate the 2nd level authorization or not, which is decided on the basis of threshold on similarity set by the security administrator.
7. On the basis of input from threat analyzer, Authorizer decides whether to initiate the 2nd level authorization or allow the same command to continue. This decision has also been logged in BAR with the help of UAPR. This decision has been taken on the basis of input from threat analyzer and also the policy rules formulated by security administrator via security manager interface.
8. The DBMS asks for another authorization in the form of some secret question on the basis of the input from the authorizer before moving further for results, if the same has been initiated by the authorizer. Otherwise, the control has been passed to the DBMS without it.
 - a. If 2nd level authorization has been initiated, then the user would be asked to prove its identity by answering the additional secret question. If user unable to answer it correctly then the current request would be suspended and the result would be sent to response parser for updating the user profile through UAPR, else the appropriate result sets would be sent to response parser.
 - b. Once the previous stage has been passed the DBMS communicates Enterprise DW to provide the response to the DBMS accordingly.
9. The Enterprise DW processes the request from DBMS and provides the result back to the DBMS.
 - a. The response containing the result set passes to the DBMS for further processing.
 - b. Response from the user passes to the response parser for conversion of response data into the response tokens.
10. The response parser passes its tokens to UAPR which updates the user signature on the basis of response tokens to update the user profile in BAR. The response details like number of processed rows, processed columns, etc. further strengthens the user signature in UAPR (User Access Pattern Recorder). The response details also contain the details of the response of user on the basis of answer of 2nd level authorization, if the same has been initiated by the authorizer.
11. Then the same cycle will follow similar to the input query from steps 11 to 15 in order to finally take a decision to show the results to the user.

Here, 2nd level of authorization further strengthens the authorization process. Whenever the user access deviates from its long historical signatures maintained over the time, this mechanism provides a 2nd layer of security to ensure the authenticity of the user by some security question etc. This mechanism allows another chance to user to prove its

authenticity in case of deviation. It also helps in the creation of more robust signature base over a period of time which in turn creates better user profiles.

5 Conclusion and Future Work

The mechanism of two level authorizations strengthens the security of the system in the following two ways:

1. The 2nd level of authorization itself bring robustness in the signature and user profile as some of the odd signatures which don't come in the acceptance ambit can be included in the accepted signature list by the verification of 2nd level of authorization.
2. It also act as a 2nd level challenge to the user, if the user's action doesn't match the user historical profile created over a period of time. So it also acts as an additional security level even if the initial security credentials stand compromised.

In future, a comparative study would be performed to support the model on the basis of various parameters such as no. of false positive, no. of true negative etc.

References

- Inmon, W.H.: *Building the Data Warehouse*. Wiley, Hoboken (1991)
- Becker, B., Kimball, R., Mundy, J., Ross, M., Thorthwaite, W.: *The Data Warehousing Lifecycle Toolkit*. Wiley, Hoboken (2008)
- Kirkgoze, R., Katic, N., Stolba, M., Tjoa, A.: A security concept for OLAP. In: *Proceedings Eighth International Workshop on Database and Expert Systems Applications (DEXA)*. IEEE (1997)
- Berson, A., Smith, J.S.: *Data Warehousing Data Mining & OLAP*. Series on Data Warehousing and Data Management. McGraw-Hill, New York (1997)
- Santos, R., Bernardino, J., Vieira, M.: A survey on data security in data warehousing: issues, challenges and opportunities. In: *EUROCON - International Conference on Computer as a Tool (EUROCON)*, pp. 1–4. IEEE (2011)
- Cognos Incorporated: *Schrittweise Anleitungen for Transformer*. Cognos Power-Play Version 6.0 (1998)
- Microsoft Corporation: *Microsoft SQL Server OLAP Services Cell-level*. Security White-paper (1999)
- MicroStrategy Incorporated: *MicroStrategy. 7 Administrator Guide* (2000)
- Oracle Corporation: *Oracle Express Database Administration Guide*. Release 6.2, Part No. A59962-01 (1998)
- Chase, D., Spofford, G., Thomsen, E.: *Microsoft OLAP Solutions*. Wiley, New York (1999)
- Fernández-Medina, E., Trujillo, J., Villarroel, R., Piattini, M.: Extending UML for designing secure data warehouses. In: Atzeni, P., Chu, W., Lu, H., Zhou, S., Ling, T.-W. (eds.) *ER 2004*. LNCS, vol. 3288, pp. 217–230. Springer, Heidelberg (2004)
- Fernandez-Medina, E., Piattini, M., Trujillo, J., Villarroel, R.: A UML profile for designing secure data warehouses. *Latin Am. Trans.* **3**(1), 40–48 (2005). IEEE
- Villarroel, R., Soler, E., Fernández-Medina, E., Trujillo, J., Piattini, M.: Using UML packages for designing secure data warehouses. In: Gavriloa, M.L., Gervasi, O., Kumar, V., Tan, C., Taniar, D., Laganá, A., Mun, Y., Choo, H. (eds.) *ICCSA 2006*. LNCS, vol. 3982, pp. 1024–1034. Springer, Heidelberg (2006)

- Eduardo, F., Juan, T., Rodolfo, V.: A UML 2.0/OCL extension for designing secure data warehouses. *J. Res. Pract. Inf. Technol.* **38**(1), 31–44 (2006)
- Eduardo, F., Juan, T., Rodolfo, V., Mario, P.: Developing secure data warehouses with a UML extension. *Inf. Syst.* **32**(6), 826–856 (2007). Elsevier
- Emilio, S., Eduardo, F., Juan, T., Mario, P.: A UML 2.0 profile to define security requirements for Data Warehouses. *Comput. Stand. Interfaces* **31**(5), 969–983 (2009). Elsevier
- Salem, A., Triki, S., Ben-Abdallah, H., Harbi, N., Boussaid, O.: Verification of security coherence in data warehouse designs. In: Fischer-Hübner, S., Katsikas, S., Quirchmayr, G. (eds.) *TrustBus 2012. LNCS*, vol. 7449, pp. 207–213. Springer, Heidelberg (2012)
- Dhillon, G.: *Information Security Management: Global Challenges in the New Millennium*. IGI Global, Hershey (2000)
- Iyer, S., Kantarcioglu, M., Thuraisingham, B.: Extended RBAC-based design and implementation for a secure data warehouse. *Int. J. Bus. Intell. Data Min. (IJBIDM)* **2**(4), 367–382 (2007)
- Belén, V., Carlos, B., Eduardo, F., Esperanza, M.: A practical application of our MDD approach for modeling secure XML data warehouses. *Decis. Support Syst.* **52**(4), 899–925 (2012). Elsevier
- Lopes, C.C., Times, V.C., Matwin, S., Ciferri, R.R., Ciferri, C.: Processing OLAP queries over an encrypted data warehouse stored in the cloud. In: Bellatreche, L., Mohania, M.K. (eds.) *DaWaK 2014. LNCS*, vol. 8646, pp. 195–207. Springer, Heidelberg (2014)
- Ali, S., Rauf, A., Khusro, S., Zubair, M., Farman, H., Ullah, S.: An authorization model to access the summarized data of data warehouse. *Life Sci. J.* **11**(6 s) (2014)
- Sandhu, R.S., Coyne, E.J., Feinstein, H.L., Youman, C.E.: Role-based access control models. *IEEE Comput.* **29**(2), 38–47 (1996)
- Uzun, E., Atluri, V., Vaidya, J., Sural, S., Ferrara, A.L., Parlato, G.: Security analysis for temporal role based access control. *J. Comput. Secur.* **22**, 961–996 (2014)
- dos Santos, R.J.R.: Enhancing data security in data warehousing. Ph.D. thesis submitted at Department of Informatics Engineering, Faculty of Sciences and Technology, University of Coimbra (2014)

Nonlinear Tracking of Target Submarine Using Extended Kalman Filter (EKF)

S. Vikranth, P. Sudheesh^(✉), and M. Jayakumar

Department of Electronics and Communication Engineering,
Amrita School of Engineering, Coimbatore Amrita Vishwa
Vidyapeetham University, Coimbatore 641112, India
viswak.nawin.ks@icloud.com,
{p_sudheesh,m_jayakumar}@cb.amrita.edu

Abstract. This paper presents the effective method for submarine tracking using EKF. EKF is a Bayesian recursive filter based on the linearization of nonlinearities in the state and the measurement system. Here the sonar system is used to determine the position and velocity of the target submarine which is moving with respect to non moving submarine, and sonar is the most effective methods in finding the completely immersed submarine in deep waters. When the target submarines position and velocity is located from the reflected sonar, an extended Kalman filter is used as smoothening filters that describes the position and velocity of the ship with the noisy measurements given by sonar that is reflected back. By using the algorithm of extended Kalman filter we derived to estimate the position and velocity. Here the target motion is defined in Cartesian coordinates, while the measurements are specified in spherical coordinates with respect to sonar location. When the target submarine is located, the alert signal is sent to the own ship. This can be excessively used in military applications for tracking the state of the target submarine. Prediction of the state of the submarine is possible, with Gaussian noise to the input data. The simulation results show that proposed method is able to track the state estimate of the target, this was validated by plotting SNR vs MSE of state estimates. Here in this algorithm regressive iteration method is used to converge to the actual values from the data received.

Keywords: Extended Kalman Filter (EKF) · Active sonar · Passive sonar · Non-linear filters · Prediction methods

1 Introduction

One of the major applications of sonar system is in marine military applications. Sonar is used to find the target submarines location which includes the position and velocity in which target travels. Modern marine applications are becoming very quite and sophisticated. Sonar is an acronym meaning Sound Navigation and Ranging used to identify the potential threats and also to determine their own position. Types of sonar used in marine applications are of two types. They are Active sonar and Passive sonar. In Active sonar, system emits a pulse and where the operator waits for the echoes [1]. Active sonar uses transmitter and receiver. A pulse of sound is called ping. It is electronically generated

using sonar. Then the sound wave is recorded and by the difference of time from which the pulse emitted to time when it returns back we can locate the location of the object. In Passive sonar, the submarine listens to the sound around the Submarine instead of transmitting the signal and by comparing the recorded sound to the library sound, it will be able to find the position of the approaching submarine [2]. We can use Active sonar to measure the position and velocity of approaching submarine. It is due to Doppler Effect. When a Vehicle approaching moves away, the pitch of the sound decreases. This happens because the crest of the sound waves moves far away as the source moves. This is called Doppler effect. In a similar concept submarines work, when they emit a known frequency and it returns back with different frequency they can calculate the position and velocity of that object which is intercepted by the frequency [3]. Our primary objective is tracking of target submarine position and velocity.

Here the position and velocity of the submarine can be precisely attained by the Extended Kalman Filter. The extended Kalman filter linearizes about current mean and covariance [16]. Extended Kalman filter is the novel method to estimate the state of the system subjected to noise. Extended Kalman filter linearize the estimation using the partial derivatives of process and measurement functions around the current estimate to compute in non linear conditions [4, 5]. Since it is non linearized version of Kalman filter we can use it for many applications unless like Kalman filter. Firstly, prediction strategy is used where the data required for the filter gets predicted. In the second stage Correction of errors will be made [6]. People can design the filter depending on the application. Due to its capacity to handle non linear dynamic problems, Extended Kalman filter is used widely in various fields. This method is used extensively in missile guidance and location tracking systems applications. Here, the application, the own Ship is assumed to be stationary at one place from which any submarine which enters in to the patrol region covered by the own ship will be detected. In this application, the Active sonar is used. Here the Integrated sonars have been placed all along the length of the submarine. Integrated sonar includes both passive and active sonars. During war time only passive sonar is on because of the advantage of the stealth characteristics, it has over active sonar. Active sonar when used, searcher cannot identify the target without revealing its own position. Here the Extended Kalman Filter is used for a particular range and the results are plotted with their respective errors.

2 Tracking Basics

2.1 Algorithm

The Extended Kalman filter is the Non optimal estimator for the sequence of states produced by the model. It has two parts Time update and measurement update. Before going to the next step we have to initialize the values of position and velocity in a range. First Time update is done which gives the initial values. The general predicted state estimate equation is,

$$x_{k+1} = f(x_k) + w_k \tag{1}$$

Measurement model is,

$$z_k = h(x_k) + v_k \tag{2}$$

Where

- x_k - State vector representing a particular model at the time instant k,
- $f(x_k)$ - State transition function,
- w_k - Random Gaussian noise with mean zero with covariance matrix q_k
- $h(x_k)$ - Transformation matrix that maps the state vector parameters to measurement vector domain,
- z_k - Measurement vector,
- v_k is the random Gaussian noise vector with mean zero and covariance matrix r_k . Here for this application the EKF algorithm approximates (2) by taking the linear terms in the function at the predicted state estimate. In this algorithm the State transition matrix and measurement vector are modelled depending upon the application situation,

The state vector in (1) is made up of three components of position and velocity.

$$x_k = [Xp_k, Yp_k, Zp_k, \dot{X}v_k, \dot{Y}v_k, \dot{Z}v_k]^T \tag{3}$$

$$x_{k+1} = F_k x_k + w_k$$

Where F_k is the state transition matrix. This F_k can be linearized about the particular instant k.

Where Xp_k, Yp_k, Zp_k are the coordinates of the submarine at the time instant k and $\dot{X}v_k, \dot{Y}v_k, \dot{Z}v_k$ are the component of velocity of the submarine at time instant k.

We know state transition matrix can be modelled by taking Jacobian of the dynamic equation,

$$F_k = \left. \frac{\partial f}{\partial x} \right|_{\hat{x}_k} \tag{4}$$

and the state transition matrix can be modelled as,

$$F = \begin{bmatrix} 1 & 0 & 0 & T & 0 & 0 \\ 0 & 1 & 0 & 0 & T & 0 \\ 0 & 0 & 1 & 0 & 0 & T \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{5}$$

Here $T = t - \Delta t$, where dt is the time taken by the consecutive sonar ping to get reflected back to the sonar. In other words, time elapse between measurements.

The consecutive sonar pings can be reflected back to onboard sonar sensor on different time interval depending on the ocean condition.

Measurement equation is given by the matrix

$$h(.) = \begin{bmatrix} \sqrt{(Xp - Xp_0)^2 + (Yp - Yp_0)^2 + (Zp - Zp_0)^2} \\ \tan^{-1} \frac{(Zp - Zp_0)}{(Xp - Xp_0)} \\ \tan^{-1} \frac{(\sqrt{(Xp - Xp_0)^2 + (Yp - Yp_0)^2})}{(Zp - Zp_0)} \end{bmatrix} \quad (6)$$

Here the measurement equation is nonlinear. So it is necessary to take the Jacobian for the h(.)

$$H_k = \left. \frac{\partial h}{\partial X} \right|_{\hat{x}_k} \quad (7)$$

Here the measurement matrix is consisting of Range, Bearing angle and Azimuthal angle. Figure 1 shows the position of the target as in h(.).

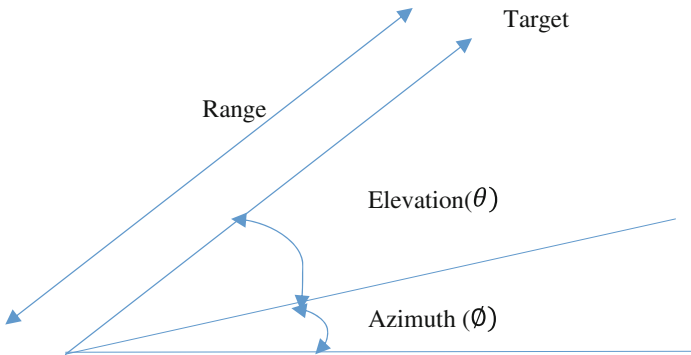


Fig. 1. Location of target submarine.

Range is the area in which the submarine can be found from the own submarine. Bearing angle is the angle in degrees with respect to the true east. Elevation angle gives the altitude of the target submarine from the altitude of the own ship. Here the position and velocity of the target submarine is given with respect to own submarines position were the sonar receiver is placed. That is what is used in the above measurement equation were the Cartesian coordinates are subtracted by values. These x_0, y_0, z_0 denote the location of the receiver in the submarine [7–9]. However, this nonlinear measurement equation can be linearized [10, 11]. Position and velocity of target at time k can be calculated by linearizing it.

$$z_{k+1} - z_k = \Delta(z_k) = H\Delta x_k + v_k \quad (8)$$

From Eq. (8) it is clear that it can be linearized.

In next step project the error covariance ahead,

$$P_{k+1} = F_k P_k F_k^T + q_k$$

Other equations which are used for estimation and are used in measurement update equations are,

(a) Kalman Gain

$$k_k = P_{k+1} H^T (H P_{k+1} H^T + r_k)^{-1} \quad (9)$$

(b) Update error with estimate

$$x_{k+1}^{up} = x_{k+1} + k_k (z_k - H(x_{k+1})) \quad (10)$$

(c) Update error covariance

$$P_{k+1}^{up} = (I - k_k H) P_{k+1} \quad (11)$$

All these steps are repeated to get the fine tuned values.

Once these values are formulated next step is to use it to the next set of vector received from the sensor.

2.2 Modelling Noise

Any unwanted signal in the desired signal is called as ‘noise’. All the practical system has some noise. All the noise from nature is modelled as white noise because it has a constant power spectral density for all frequency bands. Here we also add the Gaussian noise to the already present signal which has noise present in the system. So system can be modelled as real world applications with noise present in it. We know Gaussian distribution has Zero mean and variance which is finite. Given random process $Y \sim N(\mu, \sigma^2)$ where Y is continuous random process with mean μ and variance σ^2 . The probability density function of Y is,

$$f(y) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(y-\mu)^2}{2\sigma^2}} \text{ for } -\infty < y < \infty \quad (12)$$

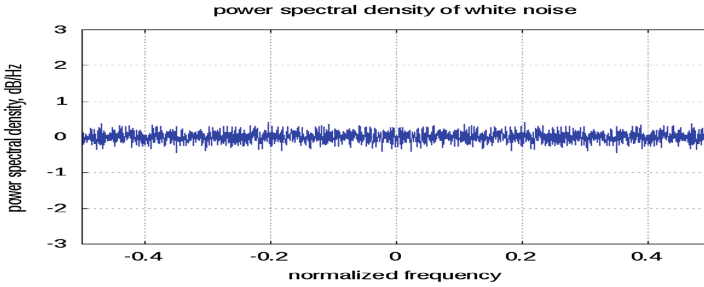


Fig. 2. Power spectral density of white noises

Hence both the process and measurement noise are considered as additive white Gaussian noise [13, 17] (Fig. 2).

Here in the Submarine target location, we have factors like P_k and q_k and r_k . The $P(k)$ is the covariance associated with the state vector. The terms along the main diagonal of the P matrix gives the variance associated with the corresponding terms in state vector.

$$P(k) = 10^{-5} \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{13}$$

Where noises cannot be modelled completely because it depends on complete randomness. Here the values of measurement noise and process noise are assumed.

Here $t1$ and $t2$ are sampling time interval.

The covariance matrix w_k is given by q_k , is given by the main diagonal elements of matrix.

$$q_k = \begin{bmatrix} t1\sigma_x^2 & 0 & 0 & 0 & 0 & 0 \\ 0 & t1\sigma_y^2 & 0 & 0 & 0 & 0 \\ 0 & 0 & t1\sigma_z^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & t2\sigma_x^2 & 0 & 0 \\ 0 & 0 & 0 & 0 & t2\sigma_y^2 & 0 \\ 0 & 0 & 0 & 0 & 0 & t2\sigma_z^2 \end{bmatrix} \tag{14}$$

$$t1 = \frac{\Delta t^2}{2}, \quad t2 = \Delta t^2$$

The covariance matrix of V is given by,

$$r_k = \begin{bmatrix} \sigma_r^2 & 0 & 0 \\ 0 & \sigma_\theta^2 & 0 \\ 0 & 0 & \sigma_\varphi^2 \end{bmatrix} \tag{15}$$

The values of variances in above matrices can be changed depending upon the application. In the following application the values taken are depending upon the scenario as mentioned. Here Δt is assumed as 1 and the values are incorporated in the algorithm. Δt is the sampling rate. The values of q_k and r_k can be varied to match the real world environment conditions.

2.3 Results and Discussions

In the Graphs Following, we note that the submarine is assumed to be found from the own ship in a particular distance and velocity. From that we can interpret the efficiency of Tracking algorithm by comparing the tracking value to the original value and the error obtained by following method is also proposed. In the case of position, the values are considered to be detected in a particular range. The Velocity is approximated to the normal scenario of submarines in which it usually travels. In Figs. 3, 4, 5, 6, 7, 8 Original values and True value are plotted for both position and velocity which shows that error values are much reduced, in which the maximum value of error was found to be limited to 0.25 km. In case of Velocity also it is reduced to 0.3 km/h.

In Fig. 9 MSE vs Iterations is plotted. It is evident from the graph that the SNR value increases from 5 to 35 in step of 10 MSE vs SNR curve becomes so smooth. This shows that, as SNR increases the error decreases. The value of the error decreases as the iteration goes on increasing [14]. When the noise in the surroundings is less the error is minimized to minimum value. This also shows Fig. 10 that the filter is able to adapt to the surroundings. We are going see about NMSE VS SNR. Normalized mean square error is calculated by the following formula,

$$NMSE = \frac{\sum \text{square of error}}{\text{Average of estimate} \times \text{Average of True Value}} \tag{16}$$

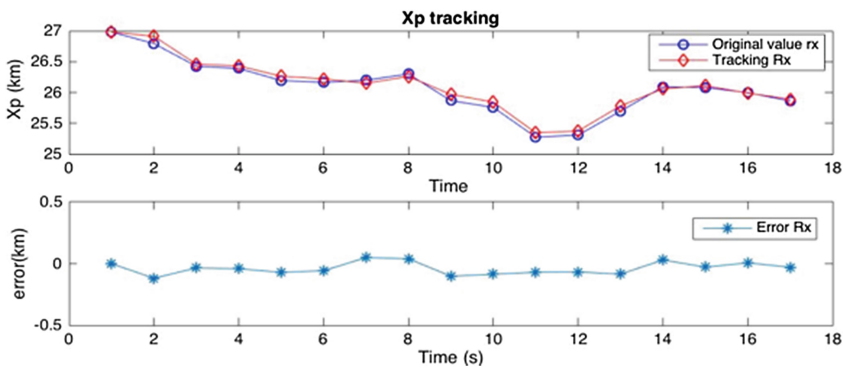


Fig. 3. Position of (x coordinate) of submarine and the corresponding error

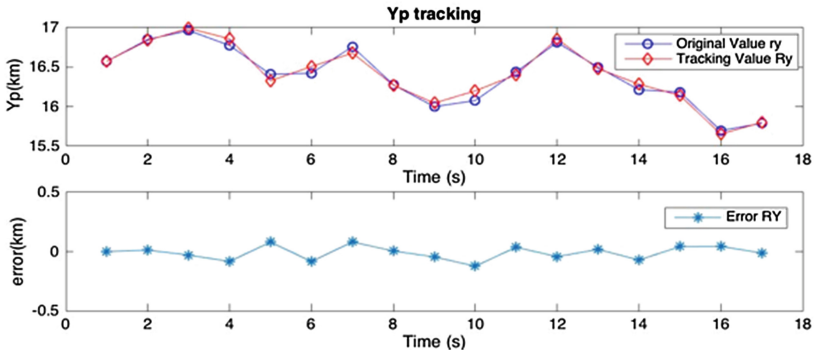


Fig. 4. Position of (y coordinate) of submarine and the corresponding error

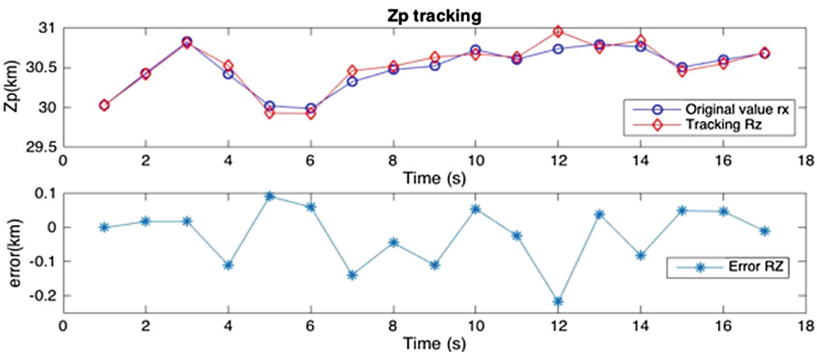


Fig. 5. Position of (z coordinate) of submarine and the corresponding error

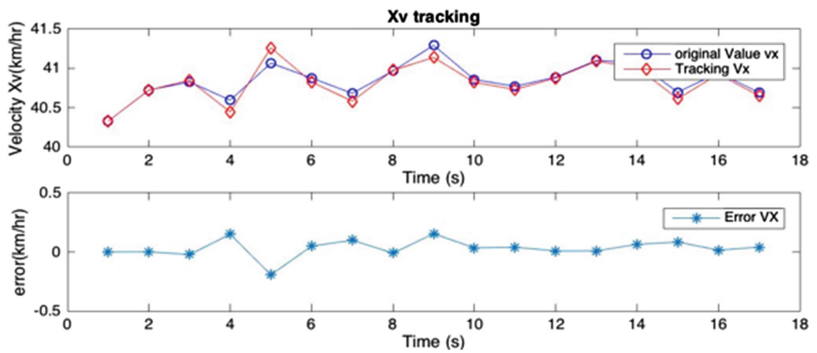


Fig. 6. Velocity of (x coordinate) of submarine and the corresponding error

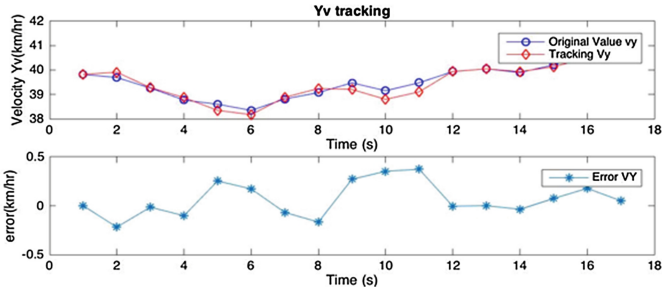


Fig. 7. Velocity of (y coordinate) of submarine and the corresponding error

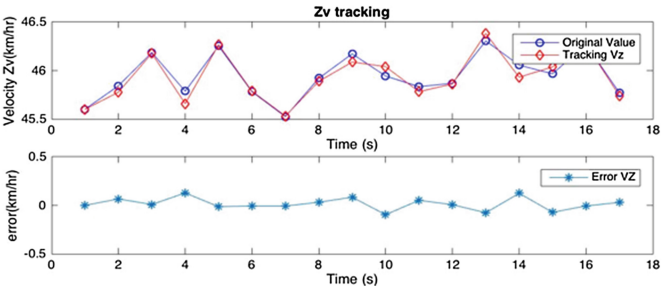


Fig. 8. Velocity of (z coordinate) of submarine and the corresponding error

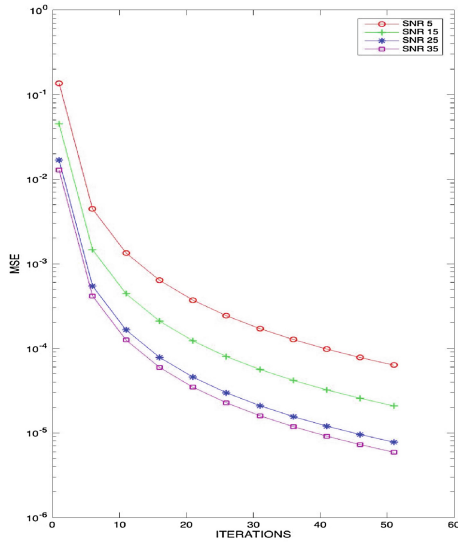


Fig. 9. Plot of mean squared error vs iteration

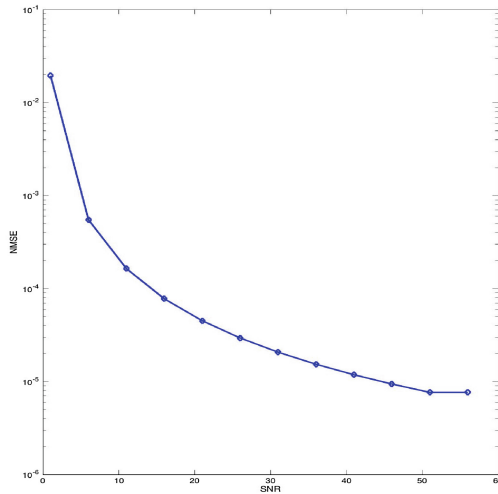


Fig. 10. Plot of NMSE vs SNR for position and velocity (Iterations = 30)

Table 1. Prediction accuracy for different SNR

Case no.	No. of samples	SNR	Accuracy
Case 1	501	5	92.5
Case 2	501	15	95.66
Case 3	501	25	96
Case 4	501	35	97.5
Case 5	501	45	99.35

To verify the extended Kalman filter for the given algorithm, a graph between normalized mean square error and signal to noise ratio over a range of 0 to 60 is plotted. The values in Table 1 which provide necessary details on increase in accuracy as SNR increases.

Table 1 represents the accuracy value increasing over the changes in SNR. Figure 10 shows how precise the extended Kalman filter tracks the position and velocity of the submarine as signal becomes more precise. The continuous drop in the value shows that filter is able to adapt to surroundings.

3 Conclusion

In this paper we presented the idea of finding the position and velocity of submarine using EKF and how to get the knowledge out of the data received from the sonar waves. An extended Kalman filter is successfully applied in determining State estimates in spite of noise present in the system. We also see that in the SNR vs MSE graph the estimated error is in the order of 0.1 m/s and as we increase the SNR the error in the system reduces to the minimum value of 0.00001 m/s which shows that

algorithm becomes more regressive as the SNR increases. The important factor which the determination of state estimates position and velocity is meticulously analyzed, verified and plotted. The idea can be extended to the Unscented Kalman Filter (UKF) where the equations are calculated without linearization [15]. It gives more accurate information about the target but it can be achieved with certain tradeoff conditions like increase in the computational complexities.

References

1. Keller, A.C.: Submarine detection by sonar. *AIEE* **66**, 1217–1230 (1947)
2. Zhou, S., Willett, P.: Submarine localization estimation via network of detection only sensors. *IEEE Trans. Sig. Process.* **55**(6), 3104–3115 (2007)
3. Wang, X., Musicki, D., Ellem, R., Fletcher, F.: Efficient and enhanced multi-target tracking with doppler measurements. *IEEE Trans. Aerosp. Electron. Syst.* **45**(4), 1400–1417 (2009)
4. Lerro, D., Bar-Shalom, Y.: Interacting multiple model tracking with amplitude feature. *IEEE Trans. Aerosp. Electron. Syst.* **29**(2), 494–509 (1993)
5. Julier, S., Uhlmann, J., Durrant-Whyte, H.F.: A new method for the nonlinear transformation of means and covariance's in filters and estimators. *IEEE Trans. Autom. Control* **45**(3), 477–482 (2000)
6. Kulikov, G.Y., Kulikov, M.V.: The accurate continuous-discrete extended Kalman filter. *IEEE Trans. Sig. Process.* **64**(4), 948–958 (2016)
7. Farina, A.: Target tracking with bearings-only measurements. *Elsevier Sig. Process.* **78**(1), 61–78 (1999)
8. Sadhu, S., Srinivasan, M., Mondal, S., Ghoshal, T.K.: Bearing only tracking using square root sigma point Kalman filter. *IEEE India Annual Conference 2004*, pp. 66–69. *INDICON* (2004)
9. Kirubarajan, T., Lerro, D., Bar-Shalom, Y.: Bearings-only tracking of maneuvering targets using a batch-recursive estimator. *IEEE Trans. Aerosp. Electron. Syst.* **37**(3), 770–780 (2001)
10. Pachter, M., Chandler, P.R.: Universal linearization concept for extended Kalman filter. *IEEE Trans. Aerosp. Electron. Syst.* **29**(3), 946–962 (1993)
11. Lerro, D., Bar-Shalom, Y.: Tracking with debiased consistent converted measurements versus EKF. *IEEE Trans. Aerosp. Electron. Syst.* **29**(3), 1015–1022 (1993)
12. Athans, M., Wishner, R.P., Bertolini, A.: Suboptimal state estimation for continuous-time nonlinear systems from discrete noisy measurements. *IEEE Trans. Autom. Control* **13**(3), 504–514 (1968)
13. Salmond, D.J., Parr, M.C.: Track maintenance using measurements of target extent. *IEEE Proc.-Radar Sonar Navig.* **150**(6), 389–395 (2003)
14. Nordsjo, A.E., Dynamics, S.B.: Target tracking based on Kaman-type filters combined with recursive estimation of model disturbances. *IEEE International Radar Conference*, pp. 115–120 (2005)
15. Gustafsson, F., Hendeby, G.: Some relations between extended and unscented Kalman filters. *IEEE Trans. Sig. Process.* **60**(2), 545–555 (2012)
16. Nair, N., Sudheesh, P., Jayakumar, M.: 2-D tracking of objects using Kalman filter. In: *International Conference on Circuit, Power and Computing Technologies (ICCPCT 2016)* (2016)
17. Seshadri, V., Sudheesh, P., Jayakumar, M.: Tracking the variation of tidal Stature using Kalman filter. In: *International Conference on Circuit, Power and Computing Technologies (ICCPCT 2016)* (2016)

Tracking Inbound Enemy Missile for Interception from Target Aircraft Using Extended Kalman Filter

T.S. Gokkul Nath, P. Sudheesh^(✉), and M. Jayakumar

Department of Electronics and Communication Engineering,
Amrita School of Engineering, Coimbatore,
Amrita Vishwa Vidhyapeetham, Amrita University, Coimbatore, India
p_sudheesh@cb.amrita.edu

Abstract. Breakthrough developments in missile guidance technology have made interception of inbound enemy missiles very difficult. Thus, it poses a huge risk and critically puts defensive capability of fighter aircrafts under test. This paper addresses the usage of Extended Kalman Filter (EKF) algorithm to estimate and track the location of inbound missile for interception by firing countermeasures. In this respect, prediction of the missile's location and trajectory is essential to enable the countermeasures fired to intercept the Enemy's missile accurately. Further, the proposed method can be used to alert the pilot regarding the inbound enemy missile and can be guided with various approaches to avoid or intercept it. EKF has been the best forecaster of the missile's location and trajectory since it has been extensively used to track objects in 3-Dimensions and in Missile guidance. EKF developed in this paper provides satisfactory results with a miss rate of 2.1 % and with localization error of 1.2 %. Thus the proposed method can be used in fighter jets for interception of inbound enemy missiles. It can further be used to track enemy aircraft's activity within the observed range.

Keywords: Extended Kalman filter · Missile interception · Proportional navigation guidance law · Anti-ballistic missile

1 Introduction

Missile Interception and avoidance have been an essential and challenging area of research in the past few decades. Modern missile interceptors need to be designed to engage and destroy a variety of highly maneuverable targets. Trajectory estimation of inbound missiles and guidance of anti-tactical ballistic missiles are two challenges in the research field of missile interception and evasion. Challenges that has been addressed already in the past are estimation of evasive acceleration required to dodge inbound missile [1], control system based on second-order sliding mode control to robustly enforce hit-to-kill guidance strategy [2], Interception of non-manoeuvring aircraft which follows pure proportional navigation [3], Estimation of Line of Sight using image sequence analysis [4], Target estimation using Sliding Mode observer/differentiator [5], Two Step adaptive method to detect and identify target [6], Tracking Missile using Modified Spherical Coordinates [7], Design and analysis of Missile interception system [8], Decentralized auction method for cooperative interception [9], Triangle interception

method for target detection and guidance [10], Scenarios for intercepting and defending missile that follow non-smooth guidance laws [11], Miss model and maneuver model for anti-ship missile and intercept model for surface to air missile was discussed [12] and various methods for interceptor guidance has been discussed with simulation [13]. The Proposed method discusses the tracking of inbound missile (Pursuer) with respect to an interceptor (Countermeasures) that has to be fired in order to protect the host aircraft from enemy missile attack. We assume that the inbound missile follows proportional navigation model as most of the missiles follow PN Guidance law at its terminal phase. The Objective of the interceptor is to maneuver suitably such that the distance between pursuer gets reduced and rendezvous condition is achieved at a safe distance far away from the host aircraft. The associated guidance law is developed for the interceptor such that interceptor acceleration command is generated based on the state estimate of the inbound missile. Magnitude, bank angle and heading angle are essential for estimation of the acceleration commands in a 3-dimensional problem. EKF linearizes the non-linear system dynamics by approximation using Taylor's expansion about the previous estimates. Since the assumed inbound missile model is non-linear, the best optimal estimate can be obtained by using Extended Kalman filter as it has been widely used in non-linear estimation and navigation systems. The Miss distance and Miss rate was calculated based on the localization error and corresponding error discrepancies in Closing Velocity [12]. Interceptors have airborne radars inside them which are used to guide them from the own aircraft to the intended inbound enemy's missile. The receiver radar is tuned to the same frequency as the inbound missile. Reflected radar energy from the inbound enemy missile is detected and autopilot on the interceptor is used to control suitably according to the deflection and guide the missile towards the inbound missile by an estimated trajectory.

The paper is organized as follows, Sect. 2 reviews the proportional guidance law that the target missile follows during the terminal phase. Section 3 reviews the Extended Kalman Filter Algorithm. Section 4 explains how the target inbound missile is modeled for interception. Section 5 discussed the results obtained from simulation experiments. Section 6 concludes the proposed solution and highlights the results obtained.

2 Proportional Navigational Law

Proportional navigation is a commonly used pursuit guidance technique based on guidance commands that are proportional to the rate at which the line-of-sight (LOS) changes between the pursuing interceptor and the inbound missile [14]. This technique is basically an implementation of the constant-bearing trajectory whereby a constant velocity pursuer flies a straight line trajectory to intercept a constant velocity target. The proportional navigation system attempts to null out perturbations to this nominal pursuit triangle by controlling the changes in rotation of the line-of-sight.

Figure 1 illustrates this condition, where \vec{r} is the LOS vector between the aircraft and the predicted target missile location; the displacement error \vec{e} , can be decomposed into two components: one along (\vec{e}_{\parallel}) and one perpendicular (\vec{e}_{\perp}) to the predicted target LOS [14].

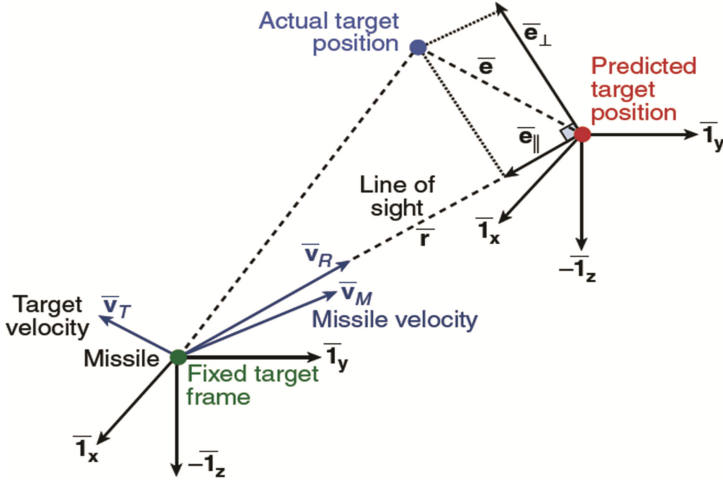


Fig. 1. Proportional navigation model [14]

$\vec{r}_{LOS} = \frac{\vec{r}}{\|\vec{r}\|}$ Is the unit vector along the LOS;

$\vec{v}_R = \vec{v}_A - \vec{v}_M$ Where \vec{v}_R is the relative velocity vector.

Proportional navigation (PN) law is one of the commonly used in guidance law due to its simplicity, effectiveness, and easier to implementation. This law is based on maintaining the line of sight angle θ_L constant as the range r decreases. Interception is ensured if the bearing and heading angle does not change as range decreases. These angles are measured with respect to a fixed coordinate axis (in this case x-axis). Rate of change in bearing and heading angle and closing velocity are needed to implement the PN guidance law.

$$\text{Closing velocity } v_c = -\dot{r}_{LOS}.$$

3 Extended Kalman Filter (EKF)

Kalman Filter is conventionally used for tracking applications [15, 16]. The Extended Kalman filter implements a Kalman filter for a nonlinear system that result from the linearization of non-linear system dynamics. This linearization is done using Taylor series expansion about the previous estimate. EKF gives only an approximation of the optimal estimate.

3.1 System Modelling

State Space Model:

$$x_t = F_t x_{t-1} + B_t u_t + w_t \quad (1)$$

Where,

- \mathbf{x}_t - N dimensional state vector at time instant t.
- \mathbf{u}_t - P dimensional control inputs vector.
- \mathbf{F}_t - $N \times N$ state transition matrix.
- \mathbf{B}_t - $P \times N$ control input matrix.
- \mathbf{w}_t - vector containing the process noise terms.

Measurement Model:

$$z_t = H_t x_t + v_t \tag{2}$$

Where,

- \mathbf{z}_t - measurement vectors.
- \mathbf{H}_t - transformation matrix.
- \mathbf{v}_t - vector containing the measurement noise.

3.2 Extended Kalman Filter Algorithm

The Kalman filter algorithm consists of two stages:

Time Update Equations (Prediction):

Prediction of the state ahead:

$$\hat{x}_k^- = f(\hat{x}_{k-1}, u_{k-1}, w_k) \tag{3}$$

Estimation of Error Covariance ahead:

$$P_k^- = F_k P_{k-1} F_k^T + Q \tag{4}$$

Measurement Update Equations (Correction):

Computation of Kalman Gain:

$$K_k = P_k^- H_k^T (H_k P_k^- H_k^T + R)^{-1} \tag{5}$$

Updation of estimates using measurement z_k . obtained:

$$\hat{x}_k = \hat{x}_k^- + K_k (z_k - H \hat{x}_k^-) \tag{6}$$

Updation of error covariance matrix:

$$P_k = (I - K_k H) P_k^- \tag{7}$$

Where,

- P - State variance matrix.
- Q - Process noise covariance matrix (Due to Navigation).
- H - Measurement matrix of the system.
- K - Kalman gain of the Filter.
- R - Measurement noise covariance matrix (Due to Sensor Imperfections).

Process noise and measurement noise is assumed to be a zero mean Gaussian white noise with covariance Q and R respectively.

4 Modelling of Inbound Target Missile

The inbound enemy missile can be modelled as follows:

$$\text{State Vector: } X_k = [x_{mk} \ y_{mk} \ z_{mk} \ \dot{x}_{mk} \ \dot{y}_{mk} \ \dot{z}_{mk}]^T$$

Here, $x_{mk} \ y_{mk} \ z_{mk}$ respectively are the locations of the missile in rectangular coordinates. This gives the position of the inbound target missile obtained from the radar in regular intervals of time. $\dot{x}_{mk} \ \dot{y}_{mk} \ \dot{z}_{mk}$ are the velocities of the missile in respective directions.

$$\text{Observation Vector: } Z_k = [r_k \ \theta_k \ \varphi_k \ \dot{r}_k \ \dot{\theta}_k \ \dot{\varphi}_k]^T$$

Here, $r_k \ \theta_k \ \varphi_k$ respectively are the measured location of the missile in polar coordinates. This gives the position, Pitch angle (Bearing Angle) and Azimuthal angle (Heading Angle) of the inbound target missile obtained from the radar in regular intervals of time. $\dot{r}_k \ \dot{\theta}_k \ \dot{\varphi}_k$ are the closing velocity of the missile. $\dot{\theta}_k$ and $\dot{\varphi}_k$ are computed for ensuring rendezvous between the countermeasures and inbound enemy missile. The State Transition for estimating the state variables is reperesented as:

$$F(k) = \begin{bmatrix} 1 & 0 & 0 & dt & 0 & 0 \\ 0 & 1 & 0 & 0 & dt & 0 \\ 0 & 0 & 1 & 0 & 0 & dt \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

The Measurement matrix can be obtained by computing Jacobian over the dynamic measurement equation $h(.)$ of the system:

$$h(.) = \begin{bmatrix} \sqrt{(x_m - x_{AC})^2 + (y_m - y_{AC})^2 + (z_m - z_{AC})^2} \\ \tan^{-1} \frac{y_m - y_{AC}}{\sqrt{(x_m - x_{AC})^2 + (z_m - z_{AC})^2}} \\ \tan^{-1} \frac{z_m - z_{AC}}{x_m - x_{AC}} \\ \dot{r} \\ \dot{\theta} \\ \dot{\varphi} \end{bmatrix}$$

Approximation of nonlinear function by utilizing the first term in a Taylor expansion of the nonlinear function The EKF approximates $h(.)$ to

$$r = \sqrt{(x_m - x_{AC})^2 + (y_m - y_{AC})^2 + (z_m - z_{AC})^2}$$

$$\theta = \tan^{-1} \frac{y_m - y_{AC}}{\sqrt{(x_m - x_{AC})^2 + (z_m - z_{AC})^2}} \varphi = \tan^{-1} \frac{z_m - z_{AC}}{x_m - x_{AC}}$$

$$H_k = \begin{bmatrix} \frac{\partial x_m}{\partial x_m} & \frac{\partial y_m}{\partial y_m} & \frac{\partial z_m}{\partial z_m} & \frac{\partial v_x}{\partial v_x} & \frac{\partial v_y}{\partial v_y} & \frac{\partial v_z}{\partial v_z} \\ \frac{\partial r}{\partial x_m} & \frac{\partial r}{\partial y_m} & \frac{\partial r}{\partial z_m} & \frac{\partial r}{\partial v_x} & \frac{\partial r}{\partial v_y} & \frac{\partial r}{\partial v_z} \\ \frac{\partial \theta}{\partial x_m} & \frac{\partial \theta}{\partial y_m} & \frac{\partial \theta}{\partial z_m} & \frac{\partial \theta}{\partial v_x} & \frac{\partial \theta}{\partial v_y} & \frac{\partial \theta}{\partial v_z} \\ \frac{\partial \varphi}{\partial x_m} & \frac{\partial \varphi}{\partial y_m} & \frac{\partial \varphi}{\partial z_m} & \frac{\partial \varphi}{\partial v_x} & \frac{\partial \varphi}{\partial v_y} & \frac{\partial \varphi}{\partial v_z} \\ \frac{\partial \dot{r}}{\partial x_m} & \frac{\partial \dot{r}}{\partial y_m} & \frac{\partial \dot{r}}{\partial z_m} & \frac{\partial \dot{r}}{\partial v_x} & \frac{\partial \dot{r}}{\partial v_y} & \frac{\partial \dot{r}}{\partial v_z} \\ \frac{\partial \dot{\theta}}{\partial x_m} & \frac{\partial \dot{\theta}}{\partial y_m} & \frac{\partial \dot{\theta}}{\partial z_m} & \frac{\partial \dot{\theta}}{\partial v_x} & \frac{\partial \dot{\theta}}{\partial v_y} & \frac{\partial \dot{\theta}}{\partial v_z} \\ \frac{\partial \dot{\varphi}}{\partial x_m} & \frac{\partial \dot{\varphi}}{\partial y_m} & \frac{\partial \dot{\varphi}}{\partial z_m} & \frac{\partial \dot{\varphi}}{\partial v_x} & \frac{\partial \dot{\varphi}}{\partial v_y} & \frac{\partial \dot{\varphi}}{\partial v_z} \end{bmatrix}$$

5 Results and Observations

EKF is applied, assuming that the inbound missile can be detected by the early warning radar with a range of 5 km. Time interval between successive measurements from sensors is assumed to 0.1 s.

Figure 2 illustrated the relationship between the mean squared error and number of iterations that are performed for fine tuning the estimated for various Signal to Noise (SNR)

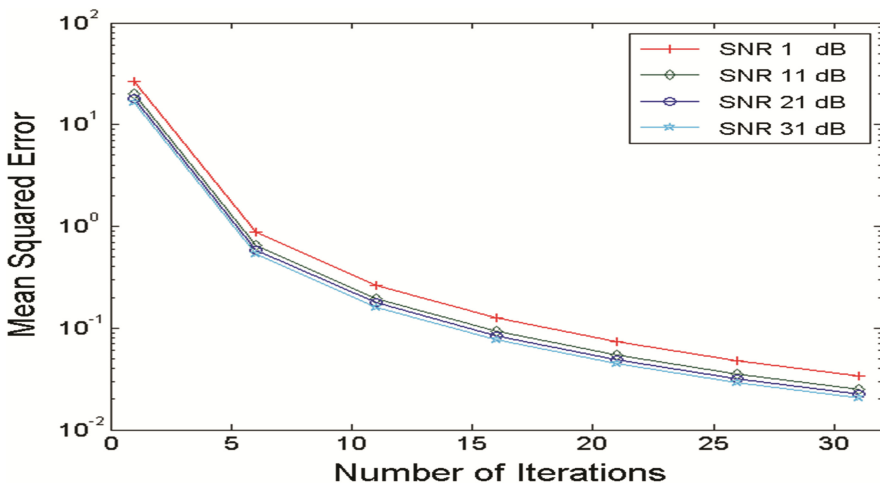


Fig. 2. Mean squared error vs. number of iterations for estimating the state variables.

Values. It can be observed that the mean squared error decreases exponentially and becomes almost constant after 25 iterations. This signifies that estimates are properly fine-tuned.

Figures 3, 4 and 5 shows the tracking by the Extended Kalman Filter and its corresponding error. It can be observed that the error in position is less than 20 cm along z and y axes and 30 cm along x axis for an object located in a space of 5000 m which is almost 98.8 % accurate. Figures 6, 7 and 8 shows the velocity tracking and its corresponding. It can be observed that the error in velocity is less than 2 m/s for an object velocity in a range of 200 m/s which is almost 99 % accurate. Figure 9 illustrates the rate at which the Normalized Mean Squared Error (NMSE) decreases as the SNR value increases. It can be observed that higher SNR ensures very less NMSE Value.

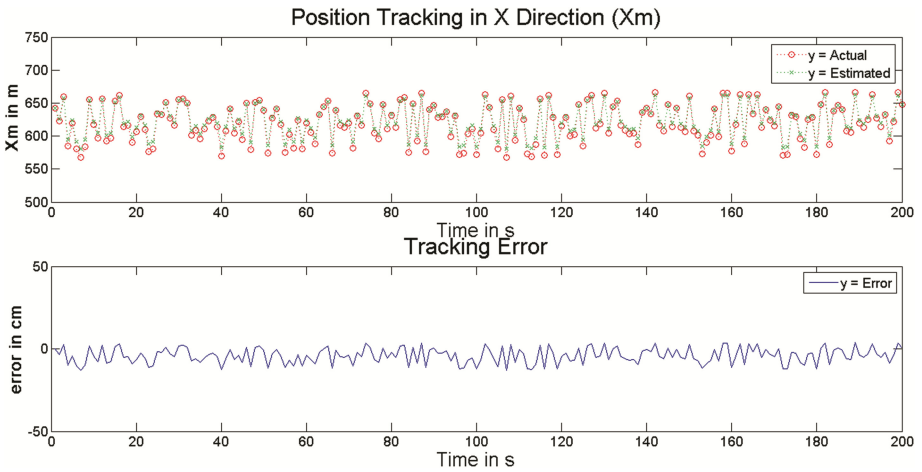


Fig. 3. Tracking of Position along x-axis and corresponding error

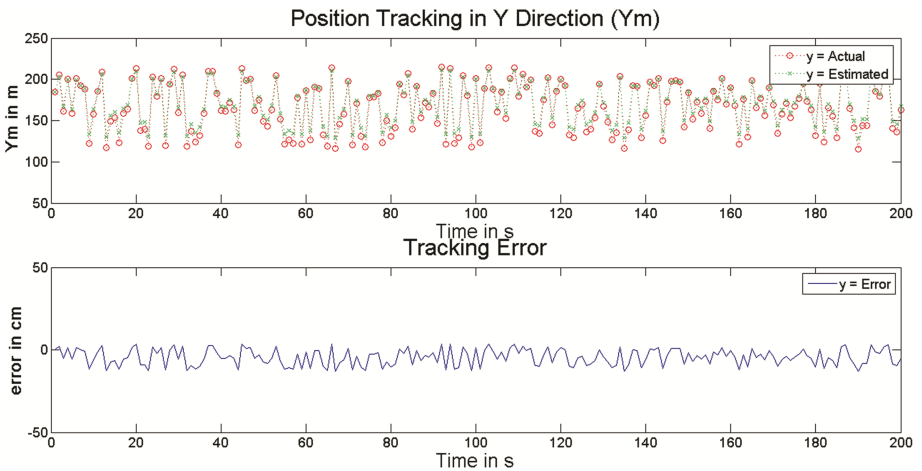


Fig. 4. Tracking of Position along y-axis and corresponding error

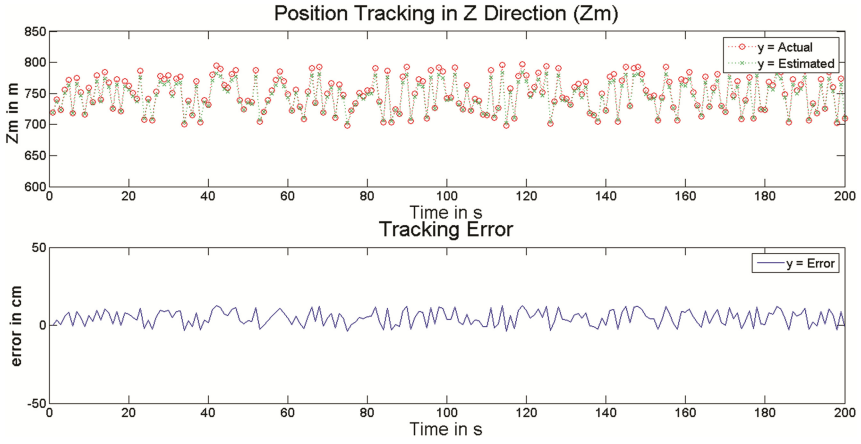


Fig. 5. Tracking of Position along z-axis and corresponding error

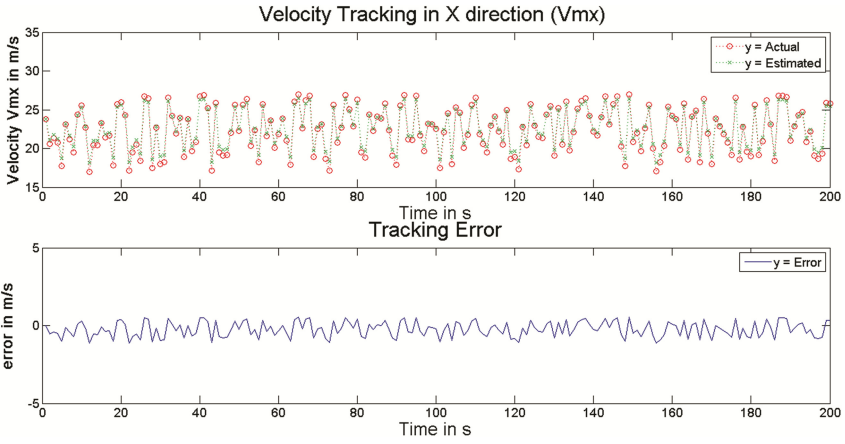


Fig. 6. Tracking of Velocity along x-axis and corresponding error

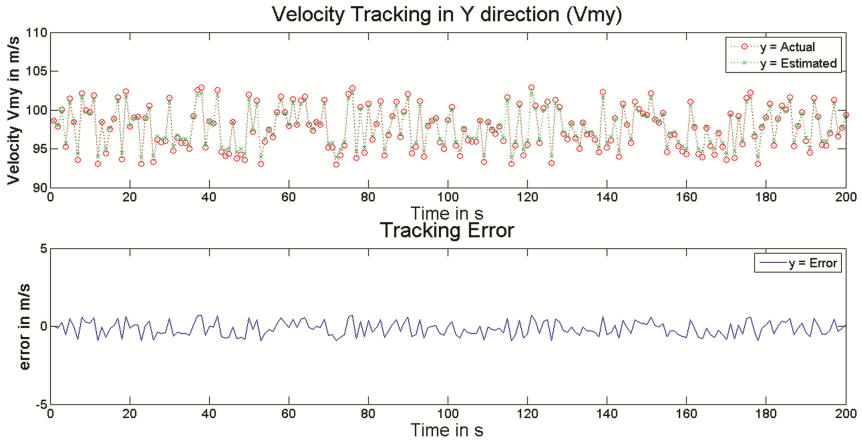


Fig. 7. Tracking of Velocity along y-axis and corresponding error

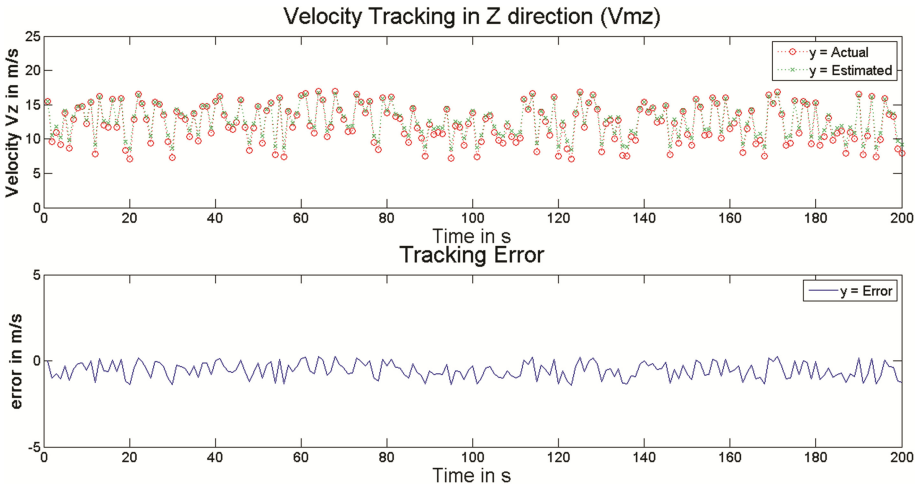


Fig. 8. Tracking of Velocity along z-axis and corresponding error

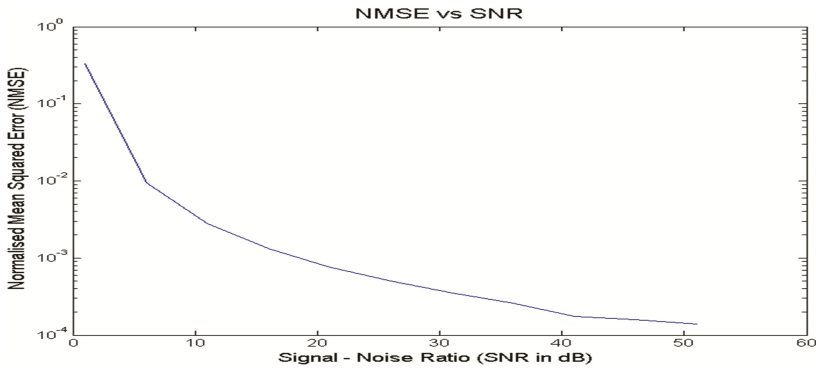


Fig. 9. Normalized mean squared error vs. SNR (Iterations = 25)

6 Conclusion

Inbound Enemy missile was assumed to follow Proportional Navigational law at its terminal phase and the countermeasures were guided suitably by tracking it using Extended Kalman filter. Thus, the proposed method has demonstrated that EKF can be employed successfully to intercept inbound enemy missile. The performance of the proposed method is better than previous works and was verified by simulation. It was observed that the miss rate was 2.1 % and accuracy of locating the missile was 98.8 %. The future scope of the proposed method involves extension to intercept of multiple missiles and reduction of miss rate. Further, there are many other types of Kalman Filter that solve missile guidance and interception problems which can be used and their results can be compared and analyzed.

References

1. Lin, Y.-p., Lin, C.-l., Suebsaiprom, P., Hsieh, S.-l.: Estimating evasive acceleration for ballistic targets using an extended state observer. *IEEE Trans. Aerosp. Electron. Syst.* **52**(1), 337–349 (2016)
2. Shtessel, Y.B., Shkolnikov, I.A., Levant, A.: Guidance and control of missile interceptor using second-order sliding modes. *IEEE Trans. Aerosp. Electron. Syst.* **45**(1), 110–124 (2009)
3. Ravindra, V., Bar-Shalom, Y., Gottesman, S.: Aim identification of missiles with a minimal parameter set. *IEEE Trans. Aerosp. Electron. Syst.* **45**(1), 405–414 (2009)
4. Zhu, Z., Xu, D., Liu, J., Xia, Y.: Missile guidance law based on extended state observer. *IEEE Trans. Ind. Electron.* **60**(12), 5882–5891 (2013)
5. Zhurbal, A., Idan, M.: Effect of estimation on the performance of an integrated missile guidance and control system. *IEEE Trans. Aerosp. Electron. Syst.* **47**(4), 2690–2708 (2011)
6. Yuzhe, W., Xiaoping, S.: Adaptive two step method and its application in the interception of hypersonic random maneuvering target. In: 3rd International Symposium on Systems and Control in Aeronautics and Astronautics (ISSCAA-2010), pp. 512–517, 8 June 2010

7. Sadeghi, H., Poshtan, J., Montazeri, A.: A modified proportional guidance law for homing missiles by using of nonlinear filters. In: 5th International Symposium on Mechatronics and Its Applications, ISMA 2008, pp. 1–6, 27 May 2008
8. Li, L.Y., Liu, F.X., Mei, Y.Y.: Modeling and simulation system design of tactical ballistic missile interception based on UML. In: Fourth International Conference on Computational and Information Sciences (ICCIS), pp. 53–56, 17 August 2012
9. Jia, J., Peng, Z.: Modeling and optimization of cooperative interception and guidance allocation in multi-platform air defense. In: 34th Chinese Control Conference (CCC), pp. 2710–2714, 28 July 2015
10. Yang, G., Qinhe, G., Jian, X., Yukun, Q., Xiaoxiang, H.: Hypersonic vehicles against a guided missile: a defender triangle interception approach. In: IEEE Chinese Guidance, Navigation and Control Conference (CGNCC), pp. 2506–2509, 8 August 2014
11. Poznyak, A.: Non-smooth missiles guidance: interceptor-defender scenario with uncertainties. In: 13th International Workshop on Variable Structure Systems (VSS), pp. 1–6, 29 June 2014
12. Fan, P.F., Liu, J.Q., Ouyang, Z.H.: Miss distance algorithm of terminal ship-to-air missile based on vector operation. In: IEEE Chinese Guidance, Navigation and Control Conference (CGNCC), pp. 2572–2576, 8 August 2014
13. Johnson, P.A., Brien Jr., R.T.: An investigation of interceptor guidance methods through modelling and simulation. In: Thirty-Ninth Southeastern Symposium on System Theory (SSST'07), pp. 258–262, 4 March 2007
14. Palumbo, N.F., Blauwkamp, R.A., Lloyd, J.M.: Basic principles of homing guidance. Johns Hopkins APL Tech. Dig. **29**(1), 25–41 (2010)
15. Nair, N., Sudheesh, P., Jayakumar, M.: 2-D tracking of objects using Kalman filter. In: International Conference on Circuit, Power and Computing Technologies (ICCPCT 2016) (2016)
16. Seshadri, V., Sudheesh, P., Jayakumar, M.: Tracking the variation of tidal stature using Kalman filter. In: International Conference on Circuit, Power and Computing Technologies (ICCPCT 2016) (2016)

**Steganography/Visual
Cryptography/Image Forensics**

A Secure One-Time Password Authentication Scheme Using Image Texture Features

Maitreya Maity¹, Dhiraj Manohar Dhane^{1(✉)}, Tushar Mungle¹,
Rupak Chakraborty², Vasant Deokamble³,
and Chandan Chakraborty¹

¹ School of Medical Science and Technology,
Indian Institute of Technology, Kharagpur, India
dmdhane@smst.iitkgp.ernet.in

² Department of Information Technology, DIT University, Dehradun, India

³ Department of E & TC, Marathwada Mitra Mandal's College of Engineering,
Pune, India

Abstract. The internet is a big giant in the today's world playing the backbone for information and communication technologies. Most of the network application needs a very reliable and one-way authentication process to ensure system resources are securely accessed by authorised users over the Internet. Password-based authentication is the most common mechanism and is easy to implement for authentication purpose. However, such general scheme will be vulnerable to attackers to steal the static credential combination with ease using eavesdropping, brute forcing, password replay, etc. One-time password (OTP) is the promising solutions to overcome from such attacks. It generates the different unique passwords each time for the user login into the system. This paper proposes a novel method of OTP authentication using image features. Here the system calculates different features from randomly selected areas of an image and applies symmetric key cryptography to generate the random OTP for the user. The proposed approach is convenient and computationally less expensive offering high-level security.

Keywords: One-time password · Texture feature · Authentication · Feature extraction · Encryption

1 Introduction

The password based authentication is a most frequently used authentication protocol in today's era. Since its induction till recent, the password-based authentication has not been toppled by any other approach. The superiority of a network system depends on upon a number of attacks resisted. The attack may be from inside the system or from outside external entity [1]. This paper deals with the prevention of external attacks using one-time password (OTP) approach. OTP system helps in preventing a replay of password attack.

S/Keys is one of the most popular examples of OTP authentication. Here, the user will select one seed value (k) and a random number (N) for computing the hash value.

Client and server both share a one-way hash function among themselves. The client uses the hash function and applies total N number of times on k to get the final hash value. The server only stores the final hash value and next attempt sequence number ($n < N$) for a particular user. When a user tries to log in with his login information, the server will send the sequence number to client and client apply $(N-n)$ time hash function on the seed value k . The generated hash value will be sent to the server for validation. The server also applies the hash function to the user's hash value and matches with the stored hash value. If both values match, then server authenticates the user successfully and update the new hash value and increments the next one attempt sequence. Therefore, no password is shared over the network. Every time client sends a new unique password which has no resembles with either previous or next generated password. Consequently, many other OTP authentication mechanisms has been found in literature, where they try to improve the security at reduced cost and increase in efficiency [2–6]. Most of the schemes follow some mathematical model to compute hash password from previous or using time stamp and location of the client, or some challenge and counter again techniques. Besides making some soft mathematical tools for password hash generation, a new approach is also introduced, where several hardware systems have used for generating a password or supporting authentication process directly. Smart card and token based one-way authentication process are also among the very familiar approach [7]. Biometric-sensor-based authentication process gave a new direction for non-sharable, unique, and notable authentication process [8]. However, hardware-based authentication process solely depends on the smart card. Therefore, loss of card makes huge trouble for authentication [8, 9]. Biometric scanning result is dependent on the client's physical response and varies with client behaviour. Fingerprint, iris, voice, face are four main categories for user authentication in biometric-based approach. If any of the categories is compromised, then altering not the best option.

The authors [10, 11] have used square image patterns for dynamic password generation without any extra hardware and complex mathematics. The size of pattern square and image can be varied adaptively. Torres et al. [12] have also suggested the theft prevention mechanism in automated teller machine using one-time usable patterns as a password which will expire after being used once. The pattern based password tool allocates a password to a user. The password indicates a spatial location of minimum one element of a plurality of elements on a matrix thereby generating the character. Finally, characters are authenticated based on spatial location assigned in the matrix. Hussain [13] has proposed multilevel validation scheme dividing the system into thoughtful sublevels and making use of different authentication protocol for each level for testing. However, the system takes more time to authenticate ultimately slowing down. Hussain et al. [14] in their recent article have evaluated the usability of proposed methods using a pre-test and post-test questionnaires. The evaluation metrics were efficiency, effectiveness, memorability, and user satisfaction of the new scheme.

This paper introduces a new approach to authentication for multimedia and data mining based on OTP authentication methodology. The methodology section elaborates the proposed approach, and later we have illustrated the scheme with an example. The workflow of the proposed method is depicted in Fig. 1.

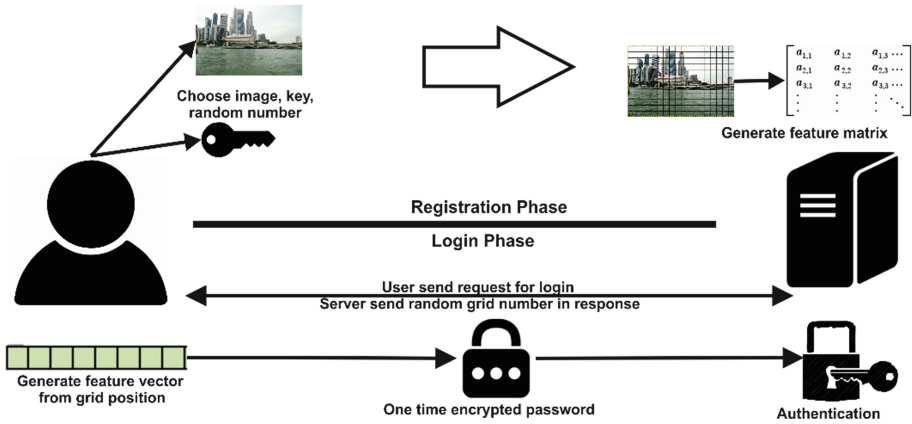


Fig. 1. Workflow of the proposed security scheme.

2 Methodology

Kindred to general password authentication framework, the proposed scheme contains three steps: registration step, login phase, and password update phase. Each step of the authentication framework is described in the following sub-sections.

2.1 Registration Phase

The guest user makes a request to remote server for granting access to the system through registration. Registration is a one-time process where the user has to follow preconfigured steps and provide some general and some confidential information into the system. Each user independently selects his login ID (U) and a KEY (K). However, the user will be instructed to choose a key as the password of length minimum six characters long. The server will then process the registration request. If the user’s information doesn’t conflict with the registered user ID, then the server will grant the access to the user. Otherwise, the server asks the user to repeat the process until the user doesn’t provide a unique ID.

After successful registration, the user will be asked to choose a standard sized random image (I) given by random choice of server and has to choose a random number (r) in the range $30 \leq r \leq 200$. The information will take the form (r, \bar{F}_I, I) , where \bar{F}_I is a feature of an image. The server will divide the image into r parts. Succeeding an image splitting process, each segmented image part will be of equal size (same width and height) excluding the last fragmented part $[I = \sum_{n=1}^r i_n]$. Forthcoming, the server will apply the pre-developed image feature extraction algorithm to extract a set of feature integer values from each fragmented image.

The feature extraction algorithm developed here extracts different textural features. Each feature set for each image part will contain similar feature vector.

$$\overline{F}_I = \sum_{n=1}^r f_n; f = \text{feature set for each image part} \tag{1}$$

and, $f_1 = \{x'_1, x'_2, \dots, x'_k\}; k = \text{length of feature vector}$
 $f_2 = \{x''_1, x''_2, \dots, x''_k\}; k = \text{length of feature vector}$
 \vdots
 $f_r = \{\tilde{x}_1, \tilde{x}_2, \dots, \tilde{x}_k\}; k = \text{length of feature vector}$

After mathematical computation, the server will store the complete information $(U, K, r, \overline{F}_I)$, into registered user entry. The server will also send the same information to the client. The client will receive the server response and stores the content of (r, \overline{F}_I, I) into client agent program.

2.2 Login Phase

In this phase the registered user will request to the server with login credential and in response server will authenticate the user based on his sent message. The user will use his client agent program (AP) locally to generate a password for accessing the system. The client program will interact with the user to generate a one-time password. The password generation process is as follows.

AP stores the information of (r, \overline{F}_I, I) . AP send a request to the server with user-id for login. The server checks the user-id and chooses an unused grid number in response. AP shows the images in a panel to the user, generates a grid line box of size r , and superimposes it on an image. Each grid box is equal to each fragmented image part processed in server previously. The AP will show the retrieved grid number (server response) on the screen. When the user chooses the particularly highlighted grid for login, AP will extract the feature vector (f) for the selected image part. Subsequently, AP will ask for the K from the user and use it to generate a standard symmetric key cryptography to encrypt the feature vector preceding with selected grid box number $(V = n + f_n)$. The encryption will generate an alphanumeric sequence which will be the password for the user.

Image Panel: An image I is in the background and the grid box panel of size r is superimposed on I . The retrieved grid position will be highlighted in the panel. The user can choose the highlighted grid number from the panel as shown in Fig. 2.

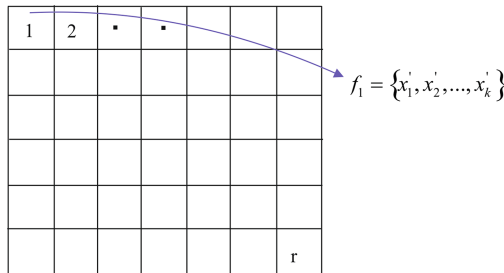


Fig. 2. Superimposed grid panel of size r on the background image.

Encryption: When the user chooses the selected grid, AP will retrieve the vector (f) from the same location of the vector set \bar{F} . Thus, AP will use the vector and format it for input string (V) of the encryption process as shown in Fig. 3. Since, grid box number always lies between 10 and 99, therefore, the selected grid will always contain two integers. Likewise, for each feature vector f_n , an integer value lies between 0 and 999. Accordingly, every feature takes three integer values. If we use feature extraction algorithm for counting K different features, the length of input integer sequence will be $2 + 3 * K$ and this length will be equal to each point.

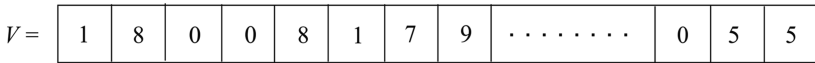


Fig. 3. Encryption input vector (V).

Consider that the user chooses the 18th block of the grid. Therefore, the first two digits show the value as 18. Later each three-digit is the value of the nth feature of the 18th fragment of an image I . An encryption algorithm will use vector string V and different features K to generate the one-time password (P) as shown in Fig. 4. However, it may create a problem when the number of features increased, the size of the P will also increase. Therefore, an encoding-cum-compression block is also introduced to generate the fixed length password for all possible case.

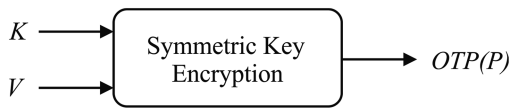


Fig. 4. OTP generation from symmetric key encryption.

Authentication: An authentication is an important login step. Here, the Client will enter user ID (U) and P while making the login request. Once the server receives the request, it will make query of the key K for the user U . Server will use the same symmetric cryptography technique to decrypt the message P using the key K . If decryption is successful, then server will read first two digits from the message as an image fragment number and matches each three successive digits with the stored information referencing the fragment number for the distinct U . Authentication result will become true, if all the matches found valid otherwise authentication will be false. In the end, the server responds back to the user with the authentication result.

2.3 Password Update Phase

When the client provides a correct login ID and password credentials, then the server makes successful authentication and server will keep a log of the selected grid box number. Once the grid box number is authenticated, then the user will not be able to

choose the same grid box for authentication again. Therefore, a total number of successful login can be attempted by a particular user will depend on the random number r chosen for splitting an image. Hence, the user needs to upload a new image and choose the same image at the time of successful login.

2.4 Feature Extraction

Image features play the key role in generating a unique password. Here, we have considered three different kinds of image texture features, which are popular in many research works [15, 16], that can be extracted from image grid. The textural features considered here are co-occurrence matrix, run-length matrix, and local binary pattern.

GLCM Based Textural Features: One of the primary problems in an image analysis is the assessment of complicated dissimilarities in the image texture. These dissimilarities are frequently owing to the relative location of different intensity pixels. These changes in the spatial interconnections of pixels are described by a gray-level co-occurrence matrix (GLCM). Here, the second-order distributions of gray value of pixels are investigated [17]. Based on GLCM, Haralick [18] defined a set of 19 measures of textural features based on average measurement of matrix function of the angles ($\theta = 0^\circ, 45^\circ, 90^\circ, 135^\circ$).

Let, a gray level image; $I = [f(x,y)]_{M \times N}$. So that, normalised gray-tone spatial-dependence matrix is obtained as $p(i,j) = P(i,j)/R$. Therefore, $p_x(i) =$

$\sum_{j=1}^{N_g} p(i,j)$ and $p_y(j) = \sum_{i=1}^{N_g} p(i,j)$. The sum of spatial dependence matrix $P_x(i)$ and $P_y(j)$ is given as,

$$p_{x+y}(k) = \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} p(i,j), \quad i+j = k, \quad k = 2, 3, \dots, 2N_g \quad (2)$$

and the difference between $P_x(i)$ and $P_y(j)$ is given as,

$$p_{x-y}(k) = \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} p(i,j), \quad |i-j| = k, \quad k = 0, 1, \dots, N_g - 1 \quad (3)$$

where, N_g is the number of distinct gray levels in the image. The list of 19 measures of GLCM textural features used in this study is tabulated in Table 2.

Run Length Matrix Features: In Gray Level Run-Length Method (GLRLM), texture is distinguished by run-length primitives. A run-length primitive is connected a set of pixels having an identical gray level in an angular orientation of run. The run-length descriptors are derived from the run-length matrix M . Each component of $M(a, r|\theta)$ constitutes the number of runs with pixels of gray level intensity a and length r along the angular orientation of run θ . Ordinarily, the orientations of the run are $0^\circ, 45^\circ, 90^\circ$ and

135°. The run-length encoding for each direction will produce four run-length matrices M of L by N_r , where L is the number of gray levels of the each image grid and N_r is the possible maximum run-length in the corresponding image. Here, we have considered total 11 different textures (see Table 1.) for each orientation (0°, 45°, 90°, 135°).

Local Binary Pattern: Local Binary Pattern (LBP) is an important textural operator which labels the image pixels by thresholding the neighbourhoods of each pixel. The results in a binary number. Let us consider a monochrome image $I(x, y)$ with g_c as a gray level of a randomly selected pixel (x, y) . Let g_p denote the gray value of a sampling point P in an evenly spaced circular neighbourhood of P and radius R around pixel (x, y) .

$$g_p = I(x_p, y), p = 0, \dots, P - 1 \text{ and} \tag{4}$$

$$x_p = x + R \cos(2\pi p/P), \tag{5}$$

$$y_p = y - R \sin(2\pi p/P), \tag{6}$$

The basic LBP is derived by summing the thresholded difference weighted by power of 2 and is defined as,

$$LBP_{P,R} = \sum_{p=0}^{P-1} s(g_p - g_c) 2^p; \text{ where } s(x) = \begin{cases} 1, z \geq 0 \\ 1, z < 0 \end{cases} \tag{7}$$

Ojala et al. [19] introduced the concept of uniformity in texture analysis to map LBP labels. The uniformity U and the rotation invariant texture description is computed as,

$$LBP_P^{riu2} = \begin{cases} \sum_{p=0}^{P-1} s(g_p - g_c), & \text{if } U(LBP_{P,R}) \leq 2 \\ P + 1 & \text{otherwise} \end{cases} \tag{8}$$

and,

$$U(LBP_{P,R}) = |s(g_{P-1} - g_c) - s(g_0 - g_c)| + \sum_{p=0}^{P-1} |s(g_p - g_c) - s(g_{p-1} - g_c)| \tag{9}$$

Here, we have considered radius ($R = 1, 2, 3$) and corresponding pixel count P as 8, 16, 24 to compute the mean, variance, and entropy respectively from LBP image as feature descriptor.

Table 1. Run-Length textural features based of run orientation (0°, 45°, 90°, 135°).

Feature name	Expression	Feature name	Expression
Short run emphasis	$\frac{1}{n_r} \sum_{a=1}^L \sum_{r=1}^{N_r} \frac{M(a,r)}{r^2}$	Long run emphasis	$\frac{1}{n_r} \sum_{a=1}^L \sum_{r=1}^{N_r} M(a,r)r^2$
Gray level non-uniformity	$\frac{1}{n_r} \sum_{a=1}^L \left(\sum_{r=1}^{N_r} M(a,r) \right)^2$	Run length non-uniformity	$\frac{1}{n_r} \sum_{r=1}^{N_r} \left(\sum_{a=1}^L M(a,r) \right)^2$
Run percentage	$\frac{n_r}{M(a,r)L}$	Low gray level run emphasis	$\frac{1}{n_r} \sum_{a=1}^L \sum_{r=1}^{N_r} \frac{M(a,r)}{a^2}$
High gray level run emphasis	$\frac{1}{n_r} \sum_{a=1}^L \sum_{r=1}^{N_r} M(a,r)a^2$	Short run low gray level emphasis	$\frac{1}{n_r} \sum_{a=1}^L \sum_{r=1}^{N_r} \frac{M(a,r)}{a^2 r^2}$
Short run high gray level emphasis	$\frac{1}{n_r} \sum_{a=1}^L \sum_{r=1}^{N_r} \frac{M(a,r)a^2}{r^2}$	Long run low gray level emphasis	$\frac{1}{n_r} \sum_{a=1}^L \sum_{r=1}^{N_r} \frac{M(a,r)r^2}{a^2}$
Long run high gray level emphasis	$\frac{1}{n_r} \sum_{a=1}^L \sum_{r=1}^{N_r} M(,r)a^2 r^2$		

Table 2. GLCM textural features based on average measurement of matrix function of the angles ($\theta = 0^\circ, 45^\circ, 90^\circ, 135^\circ$).

Feature name	Expression	Feature name	Expression
Cluster shade	$\sum_i \sum_j \{i+j - \mu_x - \mu_y\}^3 \times p(i,j)$	Energy	$\sum_i \sum_j \{p(i,j)\}^2$
Inverse difference moment	$\sum_i \sum_j \frac{1}{1+(i-j)^2} p(i,j)$	Homogeneity	$\sum_i \sum_j \frac{p(i,j)}{1+(i-j)^2}$
Sum variance	$\sum_{i=2}^{2N_g} (1 - SumEntropy)^2 p_{x+y}(i)$	Variance	$\sum_i \sum_j (i - \mu)^2 p(i,j)$
Entropy	$-\sum_i \sum_j p(i,j) \log(p(i,j))$	Sum average	$\sum_{i=2}^{2N_g} i p_{x+y}(i)$
Sum entropy	$-\sum_{i=2}^{2N_g} p_{x+y}(i) \log\{p_{x+y}(i)\}$	Auto-correlation	$\sum_i \sum_j (ij)p(i,j)$
Prominence	$\sum_i \sum_j \{i+j - \mu_x - \mu_y\}^4 \times p(i,j)$	Inertia	$\sum_i \sum_j \{i-j\}^2 \times p(i,j)$
Correlation	$\frac{\sum_i \sum_j (ij)p(i,j) - \mu_x \mu_y}{\sigma_x \sigma_y}$	Angular second moment	$\sum_i \sum_j \{p(i,j)\}^2$
Contrast	$\sum_{n=0}^{N_g-1} n^2 \left\{ \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} p(i,j) \right\}$	Difference variance	variance of p_{x-y}
Difference entropy	$-\sum_{i=0}^{N_g-1} p_{x-y}(i) \log\{p_{x-y}(i)\}$	Information measure of correlation	$\frac{HXY - HXY1}{\max\{HX, HY\}}$
Information measure of correlation 2	$(1 - \exp[-2.0(HXY2 - HXY)])^{\frac{1}{2}}$		

Where, HX and HY are individual entropies of p_x and p_y . $HXY = -\sum_i \sum_j p(i,j) \log(p(i,j))$, $HXY1 = -\sum_i \sum_j p(i,j) \log\{p_x(i)p_y(j)\}$, and $HXY2 = -\sum_i \sum_j p_x(i)p_y(j) \log\{p_x(i)p_y(j)\}$

3 Discussions

The proposed approach contains several advantages in the domain of OTP authentication. Firstly, the user doesn't need to send the password for authentication, since the passwords are shared between the users and the server. The user uses the password for encrypting the message, which will be the login password for him. The server will use the same key for decrypting the password. Therefore, no security key is passed over the network while making the request of login. Secondly, the splitting size is not shared over the network. In the worst case scenario, even if the attacker gets the image he is unable to identify the region in an image from which the feature set is generated. This happens because the same image can generate several sub-images for the different splitting number. The feature extraction algorithm will compute different textural features which will vary from an image to image fragment. Therefore, the occurrence of the similarly extracted feature set for two different images is relatively impossible. For example, suppose, we choose colour intensity as the feature, it might be possible to get the same result for two images. However, if we select some gray-level texture or LBP feature, then it is impossible to get the same result for two images. Furthermore, if the attacker eavesdrops the communication and steal the password, he can decrypt the password but cannot use it to re-log in. Unlike other OTP approach, the method used here is simple and time-saving. The only time is consumed in our method is for encrypting the message.

Our study has few limitations. One of the limitations is the password re-use and data overhead. The limitations are because the server and AP store the feature vector set for every user. We address the above limitations in our next study of expanding the expiration and reuse of the stored password. This study is on the verge of publication. The trivial task in the authentication process is a selection of an image with varying texture and high contrast. The use of the single textured image may generate same feature information on different grids. This leads to a vulnerability to possible security attacks, i.e., reply, forgery, impersonation, and denial of service attack [20–22].

In a replay attack, the attacker intercepts the authentication communication and replaces the credential after the authentication session. However, in the proposed protocol, attackers don't know about user's next grid choice and are also unaware of the number of blocks chosen for splitting. Therefore, it is impossible to generate a similar feature vector from unknown image part. In forgery attack, the attacker silently steals continuous authentication sessions and later changes the credential using assorted authentication information. However, in our protocol, the image is not communicated over authentication process; thus, it is neither easy to get the original image nor he is aware of the feature extraction algorithm. Therefore, he cannot generate authentication password. In an impersonation attack, the attacker gets possible verifier using a replay attack or a forgery attack. Therefore, replay attack or forgery attack is not popular approach for the attacker to get in. In Denial of Service attack, the genuine users are denied by the server due to mass offensive actions by attackers. Attackers cannot use the same verifier for authentication in the proposed system. However, the user may be denied by the system for next authentication session for some time.

Time Complexity Analysis: In our protocol, we have calculated the number of attempts required for authentication and found that they are equal to the splitting size of an image. Here, we have also computed the total time required to split the image, search selected grid and generate the encrypted password. Table 3. summarises the time complexity of the proposed method for OTP generation.

Table 3. Time complexity of the proposed protocol for OTP generation.

Image size	Splitting size	Time for 5 features	Time for 10 features	Time for 15 features	Authentication attempts
600×600	30	2 ms	3 ms	5 ms	30
600×600	100	0.9 ms	1.4 ms	2 ms	100
1600×1600	200	3.2 ms	5.1 ms	7.6 ms	200

An Example: The example explained in Fig. 5 gives more insight about the communication of proposed scheme. Suppose, a user (U) tries to make the registration then he will enter a unique user ID (u) and waits for the server response. Server (S) checks for u conflict. If the server grants u , then the user U will enter one alphanumeric key of minimum length 6. Let the key $K = (ABC123)$. Now U will upload an image of dimension (600×600) and choose a random number (say, $r = 30$). After that, S receives an image and r , and will split the image into 30 blocks of the dimension of 20×20 . Accordingly, S will implement the feature extraction algorithm on 30 blocks each. Let's say that the feature extraction algorithm computes five different features. Then, after the mathematical computation, S will get a dataset \bar{F}_1 of 30 vectors where each vector contains five numeric values. Finally, S will keep the complete information of the registration process [u, K, r, \bar{F}_1] and his side and at the same time, client agent program (AP) will also keep [\bar{F}_1, r, I] at the client side. The registration process is now complete. Now, U will try to login into the system. U will run the AP and ask server for login. The server will respond with a random grid number (must be less than or equal to r). AP will highlight the retrieved random grid box point from the image panel. Let say; Server sends the grid point 21. Grid number 21 will be highlighted on the screen AP will fetch the 21st vector from \bar{F}_1 and generate the string $V = [21014256005089105]$. AP will ask the user for the key and implement the symmetric key cryptography algorithm on V . After the encryption process new OTP P will be generated. Then, U will send the [u, P] to S for login verification. S will search for the key K of the user U in his stored database. Then S will decrypt P by the key K using same symmetric key cryptography algorithm. Finally, S interprets the message and looks for the 21st entry in his database for the user U . If each entry matches with the message, then S will send true authentication result to U . After that, U will not be able to choose the same grid point (i.e. 21) for re-log in. In this case, U will be able to log in maximum 30 times. When the successful login attempt reaches the maximum value, then U needs to upload the new image and choose a new random number.

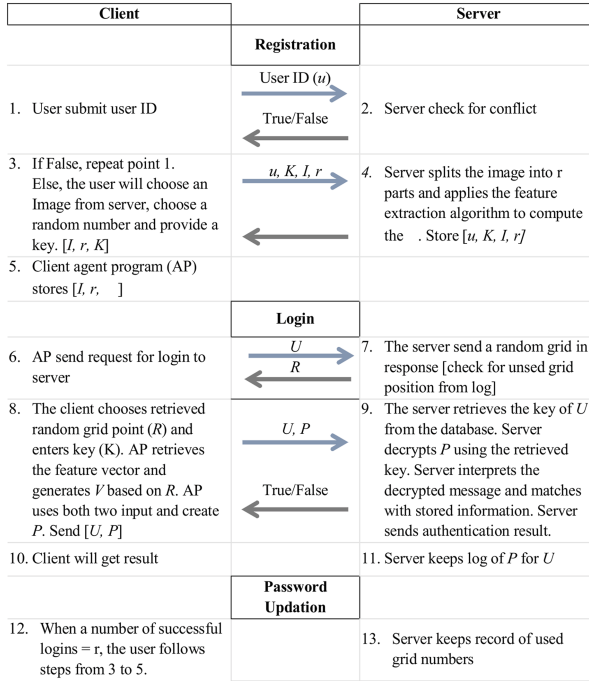


Fig. 5. The step-wise communication protocol between client and server

4 Conclusion

Most of the OTP authentications suffer from the stolen-verifier problem in which the attacker steals the user’s verifier from the server and impersonates the user. The proposed method is unique, intelligent and insusceptible to the attacks towards the OTP authentication. The method is advantageous as it does not require any external hardware or biometric sensor for computing new password every time. It is also computationally less expensive. It seems clear that this strong method will be applied to remote control systems. We are of the belief that, our proposed work has explored a new dimension towards OTP authentications where the multimedia object and data mining methodology can be used for the security. Future research will examine the problem of data expiration and overhead.

Acknowledgments. The first author sincerely acknowledges Shamik Sural, professor of Department of Computer Science and Technology, Indian Institute of Technology Kharagpur, India.

References

1. Haller, N.: The S/KEY one-time password system. In: Internet Society Symposium on Network and Distributed System Security, pp. 151–158 (1995)
2. Lamport, L.: Password authentication with insecure communication. *Commun. ACM* **24** (11), 770–772 (1981)
3. Eldefrawy, M.H., Khan, M.K., Alghathbar, K.: One-time password system with infinite nested hash chains. In: Kim, T.-h., Fang, W.-c., Khan, M.K., Arnett, K.P., Kang, H.-j., Ślęzak, D. (eds.) *SecTech/DRBC 2010*. CCIS, vol. 122, pp. 161–170. Springer, Heidelberg (2010)
4. M'Raihi, D., Machani, S., Pei, M., Rydell, J.: TOTP: Time-based one-time password algorithm. *Internet Request for Comments* (2011)
5. Kim, M., Lee, B., Kim, S., Won, D.: Weaknesses and improvements of a one-time password authentication scheme. *Int. J. Future Gener. Commun. Netw.* **2**(4), 29 (2009)
6. Shamir, A.: An efficient identification scheme based on permuted kernels. In: Brassard, G. (ed.) *CRYPTO 1989*. LNCS, vol. 435, pp. 606–609. Springer, Heidelberg (1990)
7. Xu, J., Zhu, W.-T., Feng, D.-G.: An improved smart card based password authentication scheme with provable security. *Comput. Stand. Interfaces* **31**(4), 723–728 (2009)
8. Ratha, N.K., Connell, J.H., Bolle, R.M.: Enhancing security and privacy in biometrics-based authentication systems. *IBM Syst. J.* **40**(3), 614–634 (2001)
9. ISO 7816-1: Physical characteristics. http://www.cardwerk.com/smartcards/smartcard_standard_ISO7816-1.aspx
10. Kumar, T., Raghavan, S.V.: PassPattern System (PPS): a pattern-based user authentication scheme. In: Das, A., Pung, H.K., Lee, F.B.S., Wong, L.W.C. (eds.) *NETWORKING 2008*. LNCS, vol. 4982, pp. 162–169. Springer, Heidelberg (2008)
11. Raghavan, S.V.: Methods and devices for pattern-based user authentication. Google Patents (2012)
12. Torres, R.J., Brown, D.S., Moghazy, J., Rudd, J.R.: Providing pattern based user password access. Google Patents (2011)
13. Hussain, A.: Enhanced authentication mechanism using multilevel security model. *Int. Arab J. e-Technol.* **1**(2), 49–57 (2009)
14. Alsaiani, H., Papadaki, M., Dowland, P., Furnell, S.: Graphical one-time password (GOTPass): a usability evaluation. *Inf. Secur. J. Glob. Perspect.* **25**, 1–15 (2016)
15. Maity, M., Sarkar, P., Chakraborty, C.: Computer-assisted approach to anemic erythrocyte classification using blood pathological information. In: *International Conference on Emerging Applications of Information Technology (EAIT)*, pp. 116–121. IEEE (2012)
16. Maity, M., Maity, A.K., Dutta, P.K., Chakraborty, C.: A web-accessible framework for automated storage with compression and textural classification of malaria parasite images. *Int. J. Comput. Appl.* **52**(15), 31–39 (2012)
17. Sebastian, V., Unnikrishnan, A., Balakrishnan, K.: Gray level co-occurrence matrices: Generalisation and some new features. *arXiv preprint* (2012). [arXiv:12054831](https://arxiv.org/abs/12054831)
18. Haralick, R.M., Shanmugam, K., Dinstein, I.H.: Textural features for image classification. *IEEE Trans. Syst. Man Cybern.* **6**, 610–621 (1973)
19. Ojala, T., Pietikäinen, M.: Unsupervised texture segmentation using feature distributions. *Pattern Recogn.* **32**(3), 477–486 (1999)
20. Atkinson, R., Haller, N.: On internet authentication (1994)
21. Chun-Li, L., Hung-Min, S., Hwang, T.: Attacks and solutions on strong-password authentication. *IEICE Trans. Commun.* **84**(9), 2622–2627 (2001)
22. Tsuji, T., Shimizu, A.: An impersonation attack on one-time password authentication protocol OSPA. *IEICE Trans. Commun.* **86**(7), 2182–2185 (2003)

Analyzing the Applicability of Bitsum Algorithm on LSB Steganography Technique

Bagga Amandeep^{1(✉)} and G. Geetha²

¹ School of Computer Applications, Lovely Professional University,
Phagwara, Punjab, India

amandeep.bagga@gmail.com

² Division of Research and Development,
Lovely Professional University, Phagwara, Punjab, India

gitaskumar@yahoo.com

Abstract. This paper is mainly focused on the study done to analyze the effect of Bitsum Algorithm on the LSB steganographic technique. The first section of the paper explains about the background of the LSB steganography as well as the research contribution of this paper. The second section concentrates on experimentation done to analyze the applicability of Bitsum Algorithm on LSB Steganography. This experimentation was done to check the correlation between Bitsums of the original image and stego-image, the correlation between Bitsum of the secret message and the value of R (Correlation Coefficient), and the correlation between type of image and the correlation coefficient.

Keywords: Bitsum · Steganography · LSB substitution · Correlation

1 Introduction

Steganography is the process of hiding a data into a covering medium making detection during process of communication almost impossible/very difficult. This class of methods differ from cryptology as it deals with securing privacy of data not through encryption but by ensuring that the data does not even arouse any suspicion of the third party that some data is being transferred/transmitted. This technique involves hiding a text in the data of especially image files taking care to retain the quality as well as the size of the image. The simplest method in Steganography is the substitution of the Least Significant Bit (LSB) in an image that acts as a vehicle for the hidden message or text. By using up to 4 least significant bits in each pixel, the hiding capacity can be increased to an extent that makes detection quite hard.

A number of prevalent approaches on this aspect of data hiding have been identified. Some of the processes have been reviewed in paper [1]. A mathematical analysis of the LSB Steganography is undertaken in [2]. Paper [3], shows a Spatial Domain technique in which the difference between the consecutive pixels and mean of median values is determined to embed payload in 3bits of LSB and 1bit of MSB in a chaotic manner. The approach in paper [4] makes use of a modified LSB method by combining cryptography and steganography which provides security at three levels: At the first level, the text file is compressed and zipped; followed by encryption at the second level

using the proposed algorithm; and at the final level, a secret key is used to ensure the protection of the hidden text against Stego-attacks. Paper [5] presents a novel steganography technique that combines two methods i.e. Discrete Cosine Transform (DCT) and LSB. This combination, with minimal modification to the cover image (at most k -bits per block), ensures optimization of the capacity and invisibility of the secret image which uses DCT to transform to frequency domain. The approach put forth in paper [6] is based on arithmetic progression with LSB. The algorithm of encoding a message is given as: Find the LSBs of each grey pixel in the cover image. This goes to each byte. Should the LSB be not the bit of the message position, flip it, else do nothing. Apply a progression scheme on height and width for getting the position. The results obtained through the proposed approach are better as compared to the classical LSB. Generally, the last bit of the carrier image bytes is modified to include the message bit. This does not yield to a high resistance capability of message concealing. To overcome this, the authors in paper [7] have suggested a way to modify the last 4 bits in the LSB. They have implemented the technique on Bitmap and Wave file formats. LSB steganography, based on bit inversion, has been shown in paper [8]. This technique improves the quality of stego-images in 24-bit colour image. The inversion is carried out on some of the pixels of the LSBs of the cover image on getting input of specific patterns of some bits related to the pixels. In doing so, lesser number of pixels are modified as compared with the standard LSB method. Paper [9] uses a newer version of LSB steganography in which Extended Substitution Algorithm is used to encrypt data. The cipher text so obtained is concealed at two or three LSB positions in the carrier image. Another recent approach [10] is use of polynomials in LSB steganography. In this approach, the original image as cover image and the text file that need to be embedded into original image are inputted. To generate the stream of bits, binary conversion is done by considering the conversion of ASCII value of the character into binary format. Message bits, taken sequentially, are then placed in LSB bit of image byte. The polynomial equation, given in the key, controls the index number of the image byte where replacement of LSB is to be done. A combination of LSB steganography, LZ compression, and RSA algorithm is shown by the authors in [11]. They have shown that the embedding process of LSB steganography replaces the values of the LSB plane with messages, which alters the pixel values of the LSB plane of the stego-medium in comparison to those of the original medium. V. Lokeswara Reddy et al. [12] have shown the application of a genetic algorithm based LSB steganography in JPEG images. This improved adaptive LSB steganography can achieve high capacity while preserving the first order statistics.

In this paper, LSB substitution method is discussed and is used to conduct the experimentation. Section 2 discusses the details of the methodology and experimentation. Three cases are developed to study the effect of bitsum algorithm on LSB steganography. Section 3 details the data and the results obtained from the experiments conducted for three cases. A comparison of different types of images on the basis of value of R (correlation coefficient) is also shown as a bar graph in the later part of Sect. 3. Last section i.e. Section 4 is the summary of the results and the observations.

2 LSB Substitution Method and Methodology for Experimentation

Substitution of the least significant bits of the pixel intensity values of the cover image with the secret data bits is the most common technique for image steganography. For example, using an image with 8-bit pixel depth, one can write one bit (the LSB) of each pixel by XORing it with 1 bit of the secret data bit. This would yield in strong 3 bits of secret data per pixel. Evidently, the stego image generated from a cover image of 100×100 pixels can make room to embed a secret data of total 10,000 bits in it.

As far as the quality of the stego image is concerned, there will almost be no perceptible difference from the colour quality of the cover image. In practical terms, 8 bits can represent 256 levels of intensity for a colour component. The maximum change in colour intensity will be $1/256$ (i.e. 0.39 %) per colour component and this can not be perceived by human eye even after keeping both the cover image and the stego image in front together.

2.1 Bitsum Effect on LSB Steganography

The motivation for this experimentation is taken from the results obtained by implementing Bitsum algorithm on XOR cipher [13]. Bitsum attack poses a threat on XOR cipher [8]. Since LSB substitution method uses XOR to hide the data bits, so this method must be inspected against Bitsum attack.

The data has been generated by hiding the messages into the pictures with LSB substitution method. Images with the 8 – pixel depth had been used for this experimentation. The image of $m \times n$ can be represented in the following equation:

$$\text{Img} = p(i, j) \text{ where } 0 \leq i \leq m \text{ and } 0 \leq j \leq n$$

Secret data of length l can be represented as:

$$D = \{ d(i) | 0 \leq i \leq l, d(i) \in \{0, 1\} \}$$

The stego image of length $m \times n$ can again be represented in the following equation:

$$\text{Stego_Img} = p(i, j) \text{ where } 0 \leq i \leq m \text{ and } 0 \leq j \leq n$$

2.2 Methodology

Different types of images were chosen to perform the experiment by hiding the secret messages into these images by using LSB substitution method. The value of the pixels was converted into binary form and their Bitsums were taken. Bitsums of pixel values (binary form) were again taken and stored. Bitsum values were added row-wise.

Then these Bitsum values of the rows of original image and stego image were put to a correlation test. The method is explained below:

1. Select a Secret Message
2. Choose a Cover Image to embed the secret message
3. Generate a Stego Image by using LSB method
4. Convert the pixel values into the binary form and calculate their Bitsum
5. Take Bitsums of all the rows of the Original and Stego-Image
6. Calculate value of correlation coefficient these values.
7. Repeat the process for different images as well as for different messages and keep a track of the values of the correlation coefficient.

This methodology is based on Bitsum Algorithm [13] and LSB steganographic technique. Different images would show different behavior under same circumstances. This behavior will be analyzed to find the correlation.

2.3 Cases to Study

Case 1: To check the correlation between Bitsum of the original image and that of stego-image.

Case 2: To check the correlation between Bitsum of secret message and value of R (Correlation Coefficient).

Case 3: To check the correlation between type of image and the correlation coefficient.

The methodology and the cases are formulated to analyze the specific correlation between original image and stego image, if exists. The methodology stated in Sect. 2.2 is used to find the results for the cases under study. The following section explains the results of the experimentation done to analyze these cases.

3 Results of Experiments

3.1 Results of the Experimentation for Case 1

Different types of images were chosen to conduct this study. To explain the conduct of this experiment, the image in Fig. 1 (i.e. Penguins) is taken. This figure contains both the original as well as the stego images.

Binary values of original and stego image are shown in Fig. 2.

Bitsum values of the original and stego images are shown in Fig. 2. The binary values of all the 8 – bit pixels is shown in the cells. The number of 1 s in each cell were counted and written as Bitsum of the pixel.

Figure 3 shows the Bitsum values for the original image as well as for the stego image. To generate the Bitsum of each row, all the values in each row were added.

After adding these Bitsum values row-wise, correlation test was conducted.

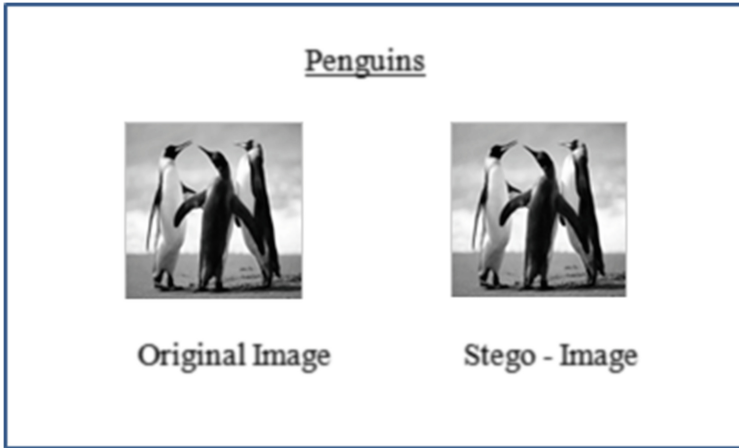


Fig. 1. Data hidden in penguins image

orgbits										
orgbits <100x100 cell>										
	1	2	3	4	5	6	7	8	9	10
1	10100010	10100011	10100100	10100100	10100101	10100111	10101000	10101000	10100111	10100111
2	10100011	10100011	10100010	10100100	10100110	10101000	10100111	10100111	10100110	10100110
3	10100010	10100001	10100001	10100100	10100110	10100111	10100111	10100111	10100110	10100111
4	10100010	10100000	10100010	10100101	10100111	10100111	10100111	10100111	10100111	10101000
5	10100000	10100010	10100011	10100110	10100111	10100111	10100111	10100111	10100111	10101000
6	10100010	10100011	10100011	10100101	10100110	10100111	10100111	10100111	10100111	10101000
7	10100010	10100010	10100011	10100100	10100110	10100111	10100111	10100111	10100111	10101001
8	10100010	10100011	10100011	10100100	10100101	10100110	10100111	10100111	10101000	10101010
9	10100011	10100011	10100011	10100011	10100100	10100110	10100110	10100110	10100110	10101001
10	10100011	10100011	10100011	10100011	10100100	10100110	10100110	10100110	10100110	10101001

finbits										
finbits <100x100 cell>										
	1	2	3	4	5	6	7	8	9	10
1	10100011	10100010	10100101	10100101	10100100	10100110	10101001	10101001	10100110	10100110
2	10100010	10100010	10100011	10100101	10100111	10101001	10100110	10100110	10100111	10100111
3	10100011	10100000	10100000	10100101	10100111	10100110	10100110	10100110	10100111	10100110
4	10100011	10100001	10100011	10100100	10100110	10100110	10100110	10100110	10100110	10101001
5	10100001	10100011	10100010	10100111	10100110	10100110	10100110	10100110	10100110	10101001
6	10100011	10100010	10100010	10100100	10100111	10100110	10100110	10100110	10100110	10101001
7	10100011	10100011	10100010	10100101	10100111	10100111	10100110	10100110	10100110	10101000
8	10100011	10100010	10100010	10100101	10100100	10100111	10100110	10100110	10101001	10101011
9	10100010	10100010	10100010	10100010	10100101	10100100	10100111	10100100	10100110	10101000
10	10100010	10100010	10100010	10100010	10100101	10100100	10100111	10100100	10100110	10101000

Fig. 2. Binary values of the pixels of original image (orgbits) and stego-image (finbits)

The value of R, i.e. correlation coefficient, is 0.9979. This means that a strong positive correlation exists between the Bitsum value of the original image and that of the stego-image. The correlation coefficients for different images for a specific message with Bitsum 72 is tabulated below:

It is visible from Table 1 that correlation is there since all the values are showing positive value of R.

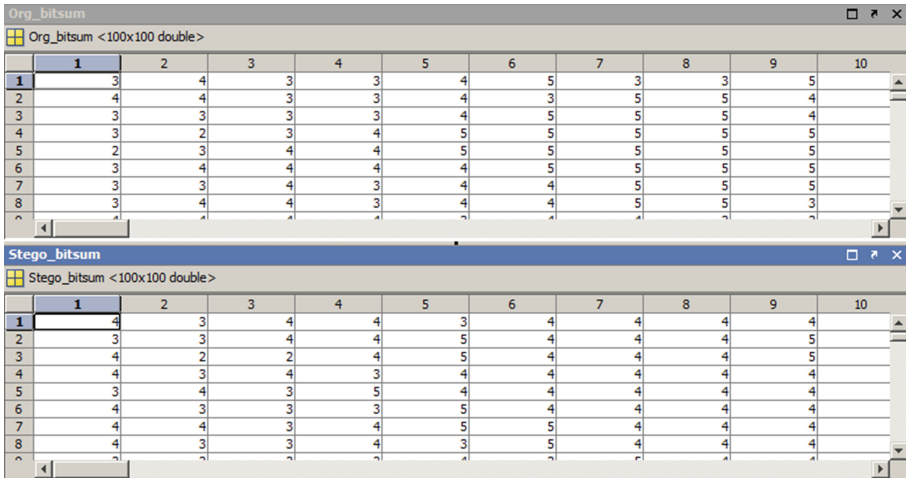


Fig. 3. Bitsum values of the pixels of original image (Org_Bitsum) and Stego-Image (Stego_Bitsum)

Table 1. Correlation coefficients of different images

Image Name	Value of R
Chrysanthemum	0.9784
Penguins	0.9868
Desert	0.9868
Lighthouse	0.9784
Sunil-Gavaskar	0.9824
Red	0.9236
Green	0.7907

3.2 Results of the Experimentation for Case 2

The correlation between Bitsums of the original image and the stego image for a particular message is persistent. The purpose of this study is/was to check correlation between the Bitsum of different data (secret messages) and the value of R. The messages with different Bitsum were taken to analyze this. Images of size 30 X 20 were taken i.e. these images represent 600 pixels of 8 – bits each. Now the data was hidden in some of these pixels. The number of pixels needed to hide the data directly depend on the number of bits in the secret message since we were using the single bit LSB steganographic technique. The values in Table 2 are the result of experiment done on Penguins image. Messages with different Bitsums were taken to analyze their effects on the Value of R.

From the values in Table 2, it can be analysed that increase in the Bitsum of the secret message decreases the value of R. The same experiment was conducted on the other images listed in Table 1. It was decided to choose and test variable types of images e.g. flowers, animals, human face, desert, buildings, and the single coloured plane images.

Table 2. Values of R for penguins

Bitsum of the secret message	Value of R (correlation coefficient)
16	0.9949
60	0.9893
72	0.9868
300	0.9323
400	0.8942
600	0.8314

Table 3. Value of R for different secret messages with different images

Bitsum of the secret message	Value of R (Correlation Coefficient) for						
	Chrysanthemum	Penguins	Desert	Lighthouse	Sunil-Gavaskar	Red	Green
16	0.9962	0.9949	0.9949	0.9962	0.9964	0.9853	0.9698
60	0.9774	0.9893	0.9893	0.9774	0.9836	0.9483	0.8191
72	0.9784	0.9868	0.9868	0.9784	0.9824	0.9236	0.7907
300	0.9555	0.9323	0.9323	0.9555	0.9506	0.6105	0.6659
400	0.9513	0.8942	0.8942	0.9513	0.9433	0.4653	0.6056
600	0.8655	0.8314	0.8314	0.8655	0.8349	0.1809	0.156

Eight images were analyzed on six messages with different Bitsum values. The resulting values for the correlation coefficients for these images are summarised in Table 3. These values strengthen the result obtained from Table 2.

3.3 Results of the Experimentation for Case 3

The intention for this study was to check the value of the correlation coefficient for different types of images. The different images taken for this analysis were of flowers, buildings, human face, planes etc. The image and the table for the values for the correlation coefficient are given below.

The value of the correlation coefficients for different messages is tabulated in Table 3. Again it can be seen from the values in the table that increase in the Bitsum of the secret message, decreases value of R. Similarly, this experiment was conducted on other images also. These images are shown below (Figs 4, 5, 6, 7, 8 and 9):

The variation in the values of the correlation coefficient R can be seen from Table 3.

The following bar chart in Fig. 10 explains this variation. It can be easily analyzed from Fig. 10 that the variation in the values of R is quite minimal in case of flowers, buildings, and human face. But when it comes to the simple images having single colour, it shows more variation in the value of R with respect to the Bitsum of the secret message. The legends in this bar graph are depicting the bitsum of the secret message. Each image is used hide all the six messages and the value of R was observed and plotted here on the graph.



Fig. 4. Desert



Fig. 5. Lighthouse



Fig. 6. Sunil Gavaskar



Fig. 7. Chrysanthemum

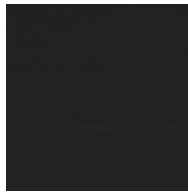


Fig. 8. Red colour

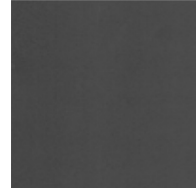


Fig. 9. Green colour

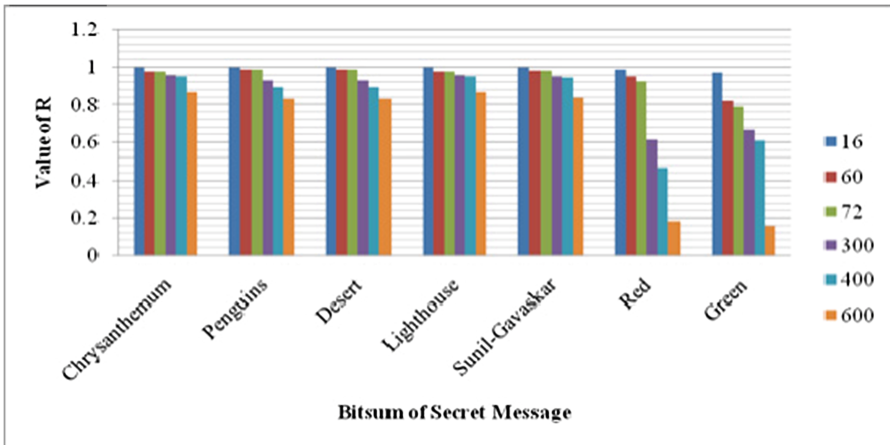


Fig. 10. Bar graph for the values of R for different images with different secret messages

4 Results and Discussions

This paper was dedicated to LSB steganography and the impact of Bitsum attack on this technique. In particular three cases were under analysis. The observations for these three cases are:

- (a) During the experimentation, it was found that there is a correlation between Bitsum of the original image and the stego-image.

- (b) The correlation between the Bitsum of secret message and the value of correlation coefficient was also found. The results showed that whenever Bitsum of the secret message was increased, the value of R decreased.
- (c) The third observation was regarding the correlation coefficient and the types of images. The results have shown that simple/plain images show more variation on changing the Bitsum of the secret message, whereas the variation is quite less in the case of complicated images.

References

1. Prashanti, G., Sandhyarani, K.: A new approach for data hiding with LSB steganography. In: Satapathy, S.C., Govardhan, A., Raju, K.S., Mandal, J.K. (eds.) *Emerging ICT for Bridging the Future - Volume 2. AISC*, vol. 338, pp. 423–430. Springer, Heidelberg (2014)
2. Chandramouli, R., Memon, N.: Analysis of LSB based image steganography techniques. In: *Proceedings. International Conference on Image Processing*, vol. 3, pp. 1019–1022. IEEE, Thessaloniki (2001)
3. Sathisha, N., Madhusudan, G.N., Bharathesh, S., Babu, S.K., Raja, K.B., Venugopal, K.R.: Chaos based spatial domain steganography using MSB. In: *IEEE International Conference on Industrial and Information Systems (ICIIS)*, pp. 177–182. IEEE, India (2010)
4. Boopathy, R., Ramakrishnan, M., Victor, S.P.: Modified LSB method using new cryptographic algorithm for steganography. In: Babu, B.V., Nagar, A., Deep, K., Pant, M., Bansal, J.C., Ray, K., Gupta, U. (eds.) *Proceedings of the Second International Conference on Soft Computing for Problem Solving (SocProS 2012)*, pp. 591–600. Springer, New Delhi (2012)
5. Mohd, B.J., Abed, S., Al-Naami, B., Alounch, S.: Image steganography optimization technique. In: Das, V.V., Ariwa, E., Rahayu, S.B. (eds.) *SPIT 2011. LNICST*, vol. 62, pp. 205–209. Springer, Heidelberg (2012)
6. Goel, S., Gupta, S., Kaushik, N.: Image steganography – least significant bit with multiple progressions. In: Satapathy, S.C., Biswal, B.N., Udgata, S.K., Mandal, J.K. (eds.) *Proceedings of the 3rd International Conference on Frontiers of Intelligent Computing (FICTA) 2014. AISC*, vol. 328, pp. 105–112. Springer, Heidelberg (2015)
7. Jasril, J., Marzuki, I., Rahmat, F.: Modification four bits of uncompressed steganography using least significant bit (LSB) method. In: *IEEE International Conference on Advanced Computer Science and Information Systems (ICACSIS)*, pp. 287–292. IEEE, Depok (2012)
8. Bit Colour: An improved LSB image steganography technique using bit-inverse in 24 bit colour image. *J. Theor. Appl. Inf. Technol.* **80**(2) (2015)
9. Gutte, R.S., Chincholkar, Y.D., Lahane, P.U.: Steganography for two and three LSBS using extended substitution algorithm. *ICTACT Journal on communication technology.* **4**, 685–690 (2013)
10. Prasad, T.J., Giriprasad, M.N.: LSB based image steganography using polynomials and covert communications in open systems environment for DRM. In: *Proceedings of the International Conference and Workshop on Emerging Trends in Technology*, pp. 593–597. ACM, India (2011)
11. Reddy, V.L., Subramanyam, A., Reddy, P.C.: Implementation of least significant bit steganography and statistical steganalysis. In: *Proceedings of the Second International Conference on Computational Science, Engineering and Information Technology*, pp. 671–675. ACM, India (2012)

12. Yu, L., Zhao, Y., Ni, R., Li, T.: Improved adaptive LSB steganography based on chaos and genetic algorithm. *EURASIP J. Adv. Signal Process.* **32** (2010)
13. Bagga, A., Geetha, G.: Implications of bitsum attack on XOR. In: *Proceedings of 2nd National Conference on Emerging Trends in Computer Application*, Chennai (2012)
14. Duric, Z., Richards, D., Kim, Y.H.: Minimizing the statistical impact of LSB steganography. In: Kamel, M.S., Campilho, A.C. (eds.) *ICIAR 2005. LNCS*, vol. 3656, pp. 1175–1183. Springer, Heidelberg (2005)
15. Neeta, D., Snehal, K., Jacobs, D.: Implementation of LSB steganography and its evaluation for various bits. In: *1st International Conference on Digital Information Management*, pp. 173–178. IEEE, India (2005)
16. Singh, A., Singh, H.: An improved LSB based image steganography technique for RGB images. In: *IEEE International Conference on Electrical, Computer and Communication Technologies (ICECCT)*, pp. 1–4. IEEE, India (2015)

Extreme Learning Machine for Semi-blind Grayscale Image Watermarking in DWT Domain

Ankit Rajpal¹(✉), Anurag Mishra², and Rajni Bala¹

¹ Department of Computer Science, Deen Dayal Upadhyaya College,
University of Delhi, Delhi, India

ankit.cs.du@gmail.com, r_dagar@yahoo.com

² Department of Electronics, Deen Dayal Upadhyaya College, University of Delhi,
Delhi, India

anurag_cse2003@yahoo.com

Abstract. In this paper, an Extreme Learning Machine (ELM) for semi-blind grayscale in DWT domain is proposed. Low frequency LL4 sub-band is used for watermark embedding. ELM is iteratively tuned and used for training and predicting DWT coefficients. The quantized and desired LL4 sub-band coefficients of the DWT domain are used in the input dataset to train the ELM. A random key decides the starting position of the coefficients where the watermark is embedded. Both binary and the random sequence are used as watermark. This process enhances the robustness towards common image processing attacks. Experimental results show that the extracted watermark from watermarked and attacked images are similar to the original watermark. Computed time spans for embedding and extraction are of the order of seconds which is suitable for the real time processing of signed images.

1 Introduction

Digital Image watermarking is primarily used to prevent copyright infringement and content authentication. In recent past, a necessity has arisen for copyright protection of important digital media against its illegal copying. The rapid growth of internet and multimedia applications has increased the consideration paid to protecting copyright on multimedia material [1]. Mishra et al. [2] have proposed an informed watermarking technique using ELM. The fact that watermark is distributed irregularly over the entire image after the host image is transformed makes it difficult for the attacker to remove watermark or modify the signed content. Among the transform domain watermarking techniques, the one based on Discrete Wavelet Transform (DWT) has gained more popularity as they end up giving better results in terms of visual imperceptibility and robustness against common image processing attacks [3]. Presently, the problem of watermarking of images has now been converged to be an optimization problem wherein the twin requirements namely visual quality of the signed image and robustness of the embedding algorithm must be balanced out. Many soft computing techniques have been applied to accomplish this mandatory requirement. A number of research groups have proposed various machine learning

techniques to develop robust watermarking algorithms. Specifically, Artificial Neural Networks (ANNs) have been successfully employed to embed and extract watermarks. Piao et al. [4] proposed a blind watermark embedding and extraction algorithm using RBF Neural Network. Lou et al. proposed new healthcare image watermarking technique implemented on human visual model and back-propagation neural networks. The experimental results show that this technique could survive several image processing attacks including JPEG lossy compression [5]. Yang et al. [6] proposed a color image oblivious watermarking scheme based on BPN and DWT. On the other hand, the training model constructed by the common neural networks such as BPN is based on gradient descent optimization which usually suffers from various shortcomings of long training time, multiple local minima, etc. The fuzzy inference based schemes do not suffer from these problems but they are not adaptive in nature [7]. Hybrid variants such as neuro-fuzzy alternatives have also been tried and developed but they are proved to be costly in terms of embedding and extraction time [8]. Dey et al. [9] have proposed a Cuckoo Search based medical image watermarking scheme.

An algorithm called extreme learning machine (ELM) is newly developed and gained popularity to train Artificial Neural Networks (ANNs) [10–12]. The ELM is a Single hidden Layer Feed forward Neural Network (SLFN) architecture. Unlike traditional approaches such as Back Propagation Neural Networks (BPN) which may face difficulties in manual tuning control parameters and local minima, the computations of ELM are extremely fast with reasonably good accuracy. The training and prediction of ELM is reported to have been completed within milliseconds.

This research paper is organized as follows. Section 2 gives the review of Extreme Learning Machine. Section 3 presents research focus and contribution. Section 4 gives the proposed algorithms for embedding and extraction for watermark in a grayscale image. Section 5 discusses the observed results and their analysis. Section 6 gives the conclusion to the proposed work.

2 Review of Extreme Learning Machine Algorithm

Huang [11] has proposed a novel and fast neural network with single hidden layer called Extreme learning machine (ELM). The simplicity of this algorithm over traditional algorithms is due to the absence of any control parameters like stopping criterion, learning rate and number of iterations, etc. In this scheme, the input weights and the bias of hidden nodes are generated randomly and the output weights are analytically computed using simple generalized inverse operation of the hidden layer output matrices [11–13].

2.1 Extreme Learning Machine

Given the N arbitrary training examples $(x_i, t_i)_{i=1,2,\dots,N}$, where $x_i \in \mathbf{R}^d$ and $t_i \in \mathbf{R}^m$, the output of the neural network with single layer containing L hidden nodes can be represented by

$$o_i = \sum_{j=1}^L \beta_j g(a_j, b_j, x_i) = \sum_{j=1}^L \beta_j h_{ij}, i = 1, 2, \dots, N \tag{1}$$

where $a_j \in \mathbf{R}^d$ and $b_j \in \mathbf{R}$ ($j=1,2,\dots,L$) are learning parameters of the j^{th} hidden node, respectively. $\beta_j \in \mathbf{R}^m$ is the link connecting the j th hidden node to the output node. $g(a_j, b_j, x_i)$ is the output of the j th hidden node with respect to the input sample x_i . ELM states that SLFNs with L hidden nodes each with activation function $g(\cdot)$ can approximate these N examples with zero error means that $\sum_{i=1}^N \|o_i - t_i\| = 0$, i.e. there exist β_j, a_j and b_j such that

$$H\beta = T \tag{2}$$

where $H = \{h_{ij}\}$ is the hidden-layer output matrix and $h_{ij} = g(a_j, b_j, x_i) \cdot \beta = (\beta_1\beta_2 \dots \beta_L)$ is the output weight matrix and $T = (t_1 t_2 \dots t_N)$ is the target output.

The input weights and biases do not need to be tuned by human and it is therefore easier to compute the hidden-layer output matrix and the output weights. The network is constructed with very few steps and very low computational cost. Under the constraint of minimum norm least square, i.e. $\min\|\beta\|$ and $\min\|H\beta - T\|$, a simple representation of the solution of the system (2) is given explicitly by Huang et al. [11] as

$$\hat{\beta} = H^\dagger T \tag{3}$$

where H^\dagger is the Moore-Penrose generalized inverse [13] of the hidden-layer output matrix H .

Overall, the ELM algorithm is summarized as follows.

Algorithm 1: ELM

Given the training set $(x_i, t_i), x_i \in \mathbf{R}^d, t_i \in \mathbf{R}^m$, an activation function $g: \mathbf{R} \rightarrow \mathbf{R}$, and the number of hidden nodes L :

1. Randomly assign hidden node parameters $(a_j, b_j), j = 1, 2, \dots, L$;
 2. Calculate the hidden-layer output matrix H ;
 3. Calculate output weight vector $\beta = H^\dagger T$
-

3 Research Focus and Contribution

This research work focuses on optimizing the trade-off between the twin parameters of image watermarking: visual quality of signed/attacked images and the issue of robustness. The third issue of payload or capacity of the watermark is ignored. This is because the size of watermark is very less as compared to that

of the host image. We, thus, propose a novel semi-blind grayscale image watermarking scheme using the DWT-ELM architecture to achieve these results. Two different watermarks binary and random sequence have been embedded and extracted from three different gray-scale host images of size 512×512 . These images are Lena, Baboon and Girl. The proposed work assumes more significance particularly because it is widely believed that a DWT based watermarking scheme gives better results in terms of imperceptibility of watermarks within the signed images. Moreover, the ELM algorithm has been used to avoid time lapses to train the neural network. This is contrary to the performance of other gradient descent based neural architectures such as BPN. This is done with an objective to further extend the proposed scheme to real time moving multimedia data such as video both in compressed and uncompressed form. The processing time spans are found to be of the order of millisecond which makes this algorithm fit for developing real time image processing applications. Besides obtaining signed images of good visual quality, we carry out four different image processing operations over signed images as attacks to examine the robustness of the embedding scheme. These attacks are described in detail in Sect. 5. Perceptible quality of the watermarked and attacked images is quantified by PSNR and SSIM_Index. The watermarks have been extracted by using values predicted by the ELM algorithm. For this purpose, the supplied data only belongs to the signed or attacked images, as the case may be. Thus, the watermark recovery is carried out in a semi-blind manner. The robustness of the embedding scheme is evaluated by Normalized Correlation, $NC(W, W')$. It is found that the embedding and extraction processes are well optimized and the proposed DWT-ELM based watermarking scheme is robust enough against the selected attacks.

4 Experimental Details

A semi-blind watermarking scheme using the ELM in DWT domain is implemented in this work. For this purpose, LL4 sub-band coefficients are used to carry out embedding and extraction processes. Cox et al. and others [1, 2] have concluded that robust watermark embedding is possible if the embedding process is executed in low frequency coefficients in transform domain.

In this work, the ELM is trained with the LL4 sub-band coefficients. The LL4 sub-band is used to identify the most appropriate low frequency coefficients. A random key is used to determine the initial location of watermark embedding. Three standard grayscale host images of size 512×512 : - Baboon, Girl and Lena are used to embed a watermark after training the ELM using the quantized values of the LL4 sub-band coefficients. The size of the input dataset is 1024×2 while it produces an output sequence of size 1024×1 whose coefficients are close to the desired LL4 sub-band coefficients. Two different watermarks are tested in this experimental work. These are: - (a) 32×32 size binary image and (b) a 1024×1 size normally distributed random number sequence. The watermarked or signed images are tested for visual quality by computing two full reference metrics:- PSNR and SSIM_Index [14]. The signed images are also

subject to selected image processing attacks to verify the issue of robustness. These attacks are: - (a) JPEG (QF = 25, QF = 50 and QF = 75), (b) Gaussian Noise (5 % and 10 %), (c) Salt and pepper (0.1 % and 0.5 %) and (d) Scaling (resized to half and then restored to original size). Semi-blind extraction of the watermarks from the signed images is done before and after executing image processing attacks. Both embedding and extraction are done using the same key and the same ELM model. In this scenario, only the signed or the attacked image is required to recover the watermark (semi-blind extraction) by predicting the output of the ELM. A comprehensive analysis of the results obtained in this simulation is given in Sect. 5.

Watermark Embedding Algorithm. Listing 1 gives the sequence of steps used to carry out the embedding process.

Listing 1: Embedding

1. Transform the cover image (512 × 512 size) using the 4-level DWT transform. Select LL4 (32 × 32 size) sub-band coefficients and set C_i to it.
2. Quantize C_i by Q, as the input value of ELM and consider the desired output as C_i . The dataset supplied to the ELM is of size 1024 × 2. This ELM constructs the model and predicts the output of size 1024 × 1 as given by Equation (4):

$$P'_i = ELM \left(Round \left(\frac{C_i}{Q} \right) \right) \tag{4}$$

This output is close to the desired output included in the dataset used to train the ELM. The optimized numerical value of Q is 32 for all practical computations.

3. Select the starting location of watermark embedding coefficient C_i using the random secret key.
4. Embed the watermark according to the Equation (5) which uses the predicted output of the ELM (P'_i):

$$C'_{i+key} = P'_{i+key} + \alpha \times w_i \tag{5}$$

where w_i is the watermark which is either binary or random sequence, α is the embedding strength and C'_i are the modified LL4 sub band coefficients obtained after watermark embedding. 5. Perform Inverse DWT to generate signed image.

Several numerical values of α are tested in our experiment. The best results in terms of balancing out visual quality and robustness are obtained for $\alpha = 0.1$. Therefore, $\alpha = 0.1$ is used throughout the experiment.

Watermark Extraction Algorithm. Listing 2 gives the sequence of steps used to extract the watermark from signed and attacked images in a semi-blind manner.

Listing 2: Extraction

1. Transform the signed image (512×512 size) using the 4-level DWT transform. Select LL4 (32×32 size) sub-band coefficients and set C_i'' to it.
2. Quantize C_i'' by Q , and use the already trained ELM model to predict the output:

$$P_i'' = ELM \left(\text{Round} \left(\frac{C_i''}{Q} \right) \right) \quad (6)$$

3. Extract the watermark w_i' using the Equation (7) below, using the output of the ELM in Equation (6) and C_i'' and the secret key.

$$w_i' = \left(P_i'' - C_i'' \right) \times (1/\alpha) \quad (7)$$

where α is the embedding strength.

The watermark extraction is carried out using the algorithm given in Listing 2 and the normalized correlation $NC(W, W')$ is computed between the embedded and the extracted watermarks. This formulation is given in Eq. (8).

$$NC(W, W') = \frac{\sum_{i=1}^n W(i)W'(i)}{\sqrt{\sum_{i=1}^n W(i)^2} \sqrt{\sum_{i=1}^n W'(i)^2}} \quad (8)$$

5 Results and Discussion

Figure 1(a–c) represents three grayscale host images Baboon, Girl and Lena. Figure 1(d) shows the first 32×32 sized binary watermark. The second watermark is a random sequence of size 1024×1 .

The chosen watermarks are subsequently embedded within these images to obtain signed images depicted in Fig. 2(a–c) using formula given in Eq. (5). Their respective PSNR and SSIM_Index values are mentioned below these images. High computed PSNR and SSIM_Index values near unity indicate that the visual quality of these images is very good. Figure 3(a–c) depicts three binary watermarks recovered from images shown in Fig. 2(a–c) respectively. Figure 4(a–c) depicts the signed images - Baboon, Girl and Lena obtained after embedding the random sequence watermark into images depicted in Fig. 1(a–c) respectively.

Table 1 compiles the NC values between original and recovered random watermarks for the three signed images depicted in Fig. 4(a–c). It is clear that NC values are quite high thereby indicating good recovery process.

Table 2 gives a comparison for the parameters: PSNR (dBs) and NC (W, W') as computed by Liu et al. [15] and Huang et al. [16].

The ELM training time, embedding and extraction time spans are computed in this simulation. These are respectively given as 15.6 ms and 312.5 ms and 281.75 ms. The embedding time interval is in milliseconds wherein the ELM training time is included. These small processing time spans clearly indicate that

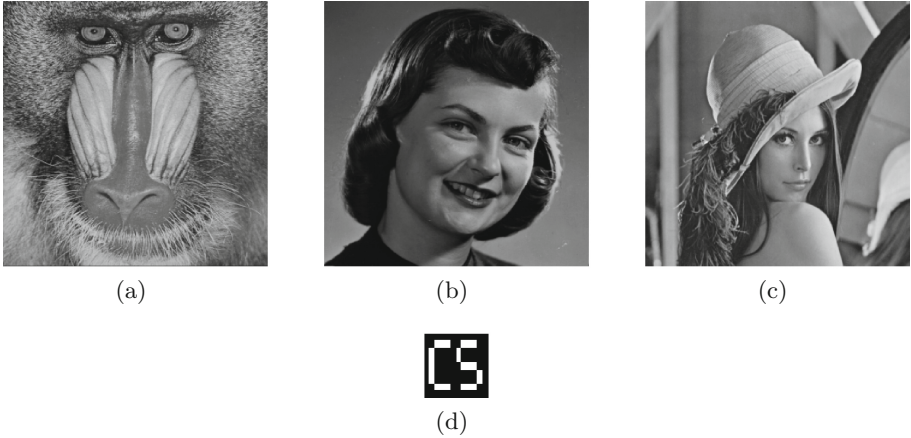


Fig. 1. Original host images - (a) Baboon, (b) Girl, (c) Lena and (d) Original watermark

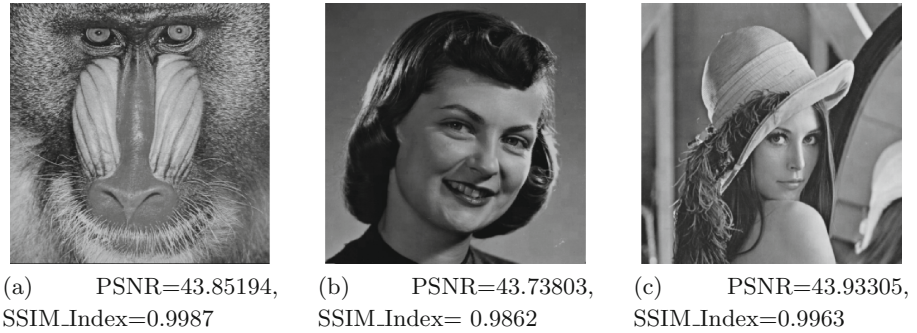


Fig. 2. Signed images using CS watermark - (a) Baboon (b) Girl and, (c) Lena

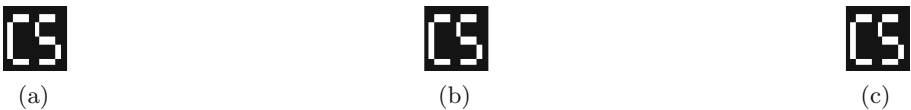
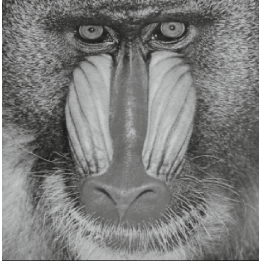


Fig. 3. Extracted watermarks from images of Fig. 4(a-c) respectively (NC = 1.0 in each case)

the proposed watermarking scheme is well suitable to satisfy a real time constraint which is one of the most important requirements of multimedia research especially for videos. We report that extraction time span is almost identical both for the signed and the attacked images. As the watermark recovery does not require ELM model to be created again and rely only on the predicted output values, its numerical component is smaller than that of embedding. The signed images are subject to selected image processing attacks. A brief description of these attacks is given in Sect. 3. The results are presented here:

(a) PSNR=41.90727,
SSIM_Index=0.9984(b) PSNR=42.03708,
SSIM_Index=0.9834(c) PSNR=41.88098,
SSIM_Index=0.9953**Fig. 4.** Signed images using random sequence watermark (a) Baboon, (b) Girl and (c) Lena**Table 1.** NC (W, W') of extracted and original watermarks

Baboon	Girl	Lena
0.9006	0.9014	0.9051

Table 2. Grayscale image (Lena.bmp): comparison of PSNR and NC (W, W') for our method with those of Liu et al. [15] and Huang et al. [16].

Attacks	Liu et al. [15]		Huang et al. [16]		Our method	
	PSNR (dBs)	NC (W, W')	PSNR (dBs)	NC (W, W')	PSNR (dBs)	NC (W, W')
No attack	NA	NA	43.55	1	43.85	1
Scaling	29.12	0.76	NA	0.7851	33.67	0.9945
JPEG (Q = 75)	32.67	0.93	NA	0.9587	36.84	0.9972
JPEG (Q = 50)	29.11	0.85	NA	0.9154	35.14	0.8738
Gaussian noise (5%)	22.63	0.70	NA	NA	36.41	0.9675

5.1 Image Noising

Gaussian noise with noise amount = 5% and 10% respectively is added to the signed images.













1. Binary Watermark of Size 32×32 : The recovered watermarks are still recognizable and shown in Table 3.

2. Random Sequence Watermark of Size 1024×1 : Higher NC (W, W') values indicate that the random sequenced watermarks are successfully recovered as shown in Table 4.

5.2 JPEG Compression

JPEG Compression with Quality factor (Q.F.) = 25, 50 and 75 is applied.

Table 3. The experiment results after adding noise

Noise Type	Image	PSNR(dB)	SSIM_Index	NC(W,W')	Extracted Watermark
Gaussian Noise 5%	Baboon	36.43	0.9699	0.9669	
	Girl	36.52	0.8887	0.9047	
	Lena	36.44	0.9128	0.9517	
Gaussian Noise 10%	Baboon	30.35	0.8867	0.6595	
	Girl	30.58	0.6780	0.6044	
	Lena	30.37	0.7183	0.7142	
Salt and Pepper 0.1%	Baboon	34.44	0.9825	0.8418	
	Girl	34.72	0.9595	0.8964	
	Lena	35.39	0.9729	0.8934	
Salt and Pepper 0.5%	Baboon	28.23	0.9213	0.5551	
	Girl	27.73	0.8522	0.6022	
	Lena	28.17	0.8648	0.5844	

1. Binary Watermark of Size 32×32 : The recovered watermarks are depicted in Table 5.

2. Random Sequence Watermark of Size 1024×1 : The NC (W, W') values indicate that the random sequence watermarks are successfully extracted as compiled in Table 6.

5.3 Image Scaling

We resize the signed image to the size 256×256 and then restored it to the original size i.e. 512×512 .

Table 4. The experiment results after adding noise

Noise type	Image	PSNR (dB)	SSIM_Index	NC (W,W')
Gaussian noise 5 %	Baboon	36.01	0.9696	0.8733
	Girl	36.12	0.8876	0.9047
	Lena	36.01	0.9121	0.8719
Gaussian noise 10 %	Baboon	30.27	0.8876	0.7573
	Girl	30.48	0.8887	0.9047
	Lena	30.25	0.7189	0.7663
Salt and pepper 0.1 %	Baboon	34.80	0.9833	0.8468
	Girl	33.90	0.9547	0.8386
	Lena	34.63	0.9678	0.8522
Salt and pepper 0.5 %	Baboon	28.53	0.9273	0.6739
	Girl	27.78	0.8498	0.6124
	Lena	28.31	0.8685	0.6902

Table 5. The experiment result after JPEG compression













Attack	Image	PSNR(dB)	SSIM_Index	NC(W,W')	Extracted Watermark
JPEG(QF=25)	Baboon	25.82	0.8096	0.5678	
	Girl	33.90	0.8859	0.5614	
	Lena	33.29	0.8826	0.5790	
JPEG(QF=50)	Baboon	28.11	0.8797	0.9060	
	Girl	35.75	0.9166	0.8699	
	Lena	35.16	0.9147	0.8737	
JPEG(QF=75)	Baboon	31.10	0.9289	1.0000	
	Girl	37.42	0.9364	0.9626	
	Lena	36.86	0.9373	0.9972	

Table 6. The experiment result after JPEG compression

Attack	Image	PSNR (dB)	SSIM_Index	NC (W, W')
JPEG (QF = 25)	Baboon	25.79	0.8094	0.6987
	Girl	33.69	0.8830	0.6650
	Lena	33.08	0.8820	0.6936
JPEG (QF = 50)	Baboon	28.04	0.8794	0.8444
	Girl	35.41	0.9134	0.8279
	Lena	34.83	0.9137	0.8494
JPEG (QF = 75)	Baboon	30.97	0.9285	0.8816
	Girl	36.95	0.9336	0.8808
	Lena	36.37	0.9364	0.8895

Table 7. The experiment result after scaling and resizing

Attack	Image	PSNR(dB)	SSIM_Index	NC(W,W')	Extracted Watermark
Scaling	Baboon	23.58	0.7129	0.8558	
	Girl	34.40	0.9225	0.9652	
	Lena	33.68	0.9158	0.9917	

1. Binary Watermark of Size 32×32 : The recovered watermarks are recognizable and compiled in Table 7.

2. Random Sequence Watermark of Size 1024×1 : The high values of NC (W, W') show that the random sequence watermarks are successfully extracted as compiled in Table 8.

Table 8. The experiment result after scaling and resizing

Attack	Image	PSNR (dB)	SSIM_Index	NC (W, W')
Scaling	Baboon	23.56	0.7127	0.8308
	Girl	34.16	0.9200	0.8906
	Lena	33.44	0.9151	0.8966

It is clear from the robustness studies that the proposed DWT based watermarking scheme using ELM network is capable of handling the selected image

processing attacks. The watermark recovery is good as indicated by high computed NC values. The visual quality of the attacked images is good in nearly all attacks. The PSNR and SSIM_Index values are high after executing attacks. Thus, the proposed scheme is capable to optimize the twin criteria of robustness and visual quality. This is carried out successfully in the shortest time span and therefore it is concluded that the watermarking scheme satisfies the real time constraints typically applicable to real time moving multimedia data such as videos. Any video sequence composed of thousands of frames will be finished with watermark embedding and extraction tasks within few minutes.

6 Conclusions

A novel semi-blind watermarking scheme using a newly developed single layer feed-forward neural network (SLFN), commonly known as Extreme Learning Machine (ELM) is proposed. The ELM is trained by using quantized LL4 sub-band coefficients of the host image by taking its 4 level DWT transform. Desired LL4 sub-band coefficients are also part of the 1024×2 size dataset supplied to train the ELM. The network produces a sequence of size 1024×1 which is used to carry out embedding. The random key decides the starting position of the coefficients where the watermark is embedded. Two different watermarks are used in this work. These are - binary image of size 32×32 and a random sequence of size 1024×1 . Extraction of watermarks from signed images is carried out in a semi-blind manner. It is found that the proposed DWT-ELM based watermarking scheme is quite efficient in terms of visual quality of the signed images. This process is also found to enhance robustness towards common image processing attacks. Experimental results show that the extracted watermark from signed and attacked images are similar to the original watermark. Computed time spans for embedding and extraction are of the order of milliseconds which is suitable for the real time processing of signed images. Overall, the proposed watermarking scheme is well optimized both for the visual quality of images on one hand and the recovery of watermarks from signed and attacked images on the other.

References

1. Cox, I.J., Kilian, J., Leighton, T.F., Shamoon, T.: Secure spread spectrum watermarking for multimedia. *IEEE Trans. Image Process.* **6**(12), 1673–1687 (1997)
2. Mishra, A., Goel, A., Singh, R., Chetty, G., Singh, L.: A novel image watermarking scheme using Extreme Learning Machine. In: *International Joint Conference on Neural Networks (IJCNN)*, pp. 1–6 (2012)
3. Zhenfei, W., Guangqun, Z., Nengchao, W.: Digital watermarking algorithm based on wavelet transform and neural network. *Wuhan Univ. J. Nat. Sci.* **11**(6), 1667–1670 (2006)
4. Piao, C., Beack, S., Woo, D., Han, S.: A Blind Watermarking Algorithm Using BPN Neural Network for Digital Image, pp. 285–292. Springer, Berlin, Heidelberg (2006)

5. Lou, D., Hu, M., Liu, J.: Healthcare image watermarking scheme based on human visual model and back-propagation network. *J. C.C.I.T.* **37**(1), 151–162 (2008)
6. Yang, Q., Gao, T., Fan, L.: A novel robust watermarking scheme based on neural network. In: International Conference on Intelligent Computing and Integrated Systems (ICISS), pp. 71–75 (2010)
7. Agarwal, C., Mishra, A., Sharma, A.: Digital image watermarking in DCT domain using Fuzzy Inference System. In: 24th Canadian Conference on Electrical and Computer Engineering (CCECE), pp. 822–825 (2011)
8. Agarwal, C., Mishra, A.: A novel image watermarking technique using fuzzy-BP network. In: Proceedings of 6th International Conference on Intelligent Information Hiding and Multimedia Signal Processing, pp. 102–105 (2010)
9. Dey, N., Samanta, S., Yang, X., Das, A., Chaudhuri, S.: Optimisation of scaling factors in electrocardiogram signal watermarking using cuckoo search. *2013 Int. J. Bio-Inspired Comput.* **5**(5), 315–326 (2013)
10. Huang, G.: The Matlab code for ELM (2004). <http://www.ntu.edu.sg/home/egbhuang>
11. Huang, G., Zhu, Q., Siew, C.: Extreme learning machine: theory and applications. *Neurocomputing* **70**, 489–501 (2006)
12. Lin, M., Huang, G., Saratchandran, P., Sudararajan, N.: Fully complex extreme learning machine. *Neurocomputing* **68**, 306–314 (2005)
13. Serre, D.: *Matrices: Theory and Applications*. Springer, New York (2002)
14. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: from error visibility to structural similarity. *IEEE Trans. Image Process.* **13**(4), 600–612 (2004)
15. Liu, Q., Jiang, X.: Design and realization of a meaningful digital watermarking algorithm based on RBF neural network. In: 2005 International Conference on Neural Networks and Brain, Beijing, pp. 214–218 (2005)
16. Huang, S., Zhang, W., Feng, W., Yang, H.: Blind watermarking scheme based on neural network. In: 7th World Congress on Intelligent Control and Automation 2008 (WCICA 2008), Chongqing, pp. 5985–5989 (2008)

A Passive Blind Approach for Image Splicing Detection Based on DWT and LBP Histograms

Mandeep Kaur^(✉) and Savita Gupta

University Institute of Engineering and Technology, Panjab University, Chandigarh, India
mandeep@pu.ac.in, savita2k8@yahoo.com

Abstract. Splicing is the most generic kind of forgery found in digital images. Blind detection of such operations has become significant in determining the integrity of digital content. The current paper proposes a passive-blind technique for detecting image splicing using Discrete Wavelet Transform and histograms of Local Binary Pattern (LBP). Splicing operation introduces sharp transition in the form of lines, edges and corners, which are represented by high frequency components. Wavelet analysis characterizes these short-time transient by measuring local sharpness or smoothness from wavelet coefficients. After first level wavelet decomposition of the image, texture variation is studied along the detailed and approximation coefficients using local binary pattern (LBP), since tampering operations disrupts the textural microstructure of an image. Feature vector is formed by concatenating the LBP histogram from the four wavelet sub bands. The classification accuracy of the algorithm was determined using svm classifier using 10-fold cross validation. The method gives maximum accuracy for the chrominance channel of YCbCr color space, which is weak at hiding tampering traces. It is tested on four different kinds of standard spliced image dataset and its performance is compared with some of the latest methods. The method offers accuracy up to 97 % for JPEG images present in the spliced image dataset.

Keywords: Passive-blind approach · Image forensics · Tamper detection · LBP histograms

1 Introduction

The digitization of data becomes imperative with the increase in quantum of data to be handled. Once digitized the benefits are enormous like easy storage, transmission, processing, analysis, automation etc. One predicament which is difficult to ignore is the ease with which multiple copies can be generated. Adding to this problem is identification of the original. The issue becomes more sensitive in case of digital images as they need to be highly reliable in many social and legal issues like insurance claims, medical, research, cyber crimes and forensics. It has thus raised a concern as the manipulated images can be used to gain illegal advantages. There is serious need of methods that can accurately discriminate a forged image from a legitimate one, as creating realistic forgeries has become much easier with the presence of sophisticated image editing tools.

These forgery detection methods, therefore suffer from very high false positive rates [1]. The quest to overcome these challenges has boosted the research in this direction.

The authenticity of digital images can be determined by using two kinds of approaches: Active and Passive. Active approaches incorporate digital signatures or watermarks, hence suffer from certain constraints as they are intrusive in nature and need the presence of original image for proving authentication. This led to the development of passive-blind approaches, which are non-intrusive (no information like signature is embedded in the image) and blind (do not require original copy to determine the authenticity of digital content). They are based on the fact that authentic digital images have consistent inherent patterns or statistical properties that get disrupted by tampering operations. The technology that detects the integrity of digital content based on the inconsistency in the inherent pattern is multimedia forensics [2]. Digital image forensics, is a subset of multimedia forensics, and is also the focus of this paper.

Tampering in a digital image is performed mostly by cut-paste (splicing) and copy-move operation. To hide the tampering traces, various post processing operations are applied like blurring, adding noise, recompression etc. are performed. A detail review of various passive blind methods is given in [1–4].

The current paper presents a passive-blind technique that can classify a test image as genuine or tampered, using discrete wavelet transforms and LBP as texture feature. Evaluation of the method is done following databases: CASIA v1.0, CASIA v2.0, COLUMBIA spliced, COLUMBIA uncompressed. It shows accuracy up to 97 % on JPEG spliced images from the CASIA2 database. The current paper proposes a method that does not follow a block based processing which is very computationally expensive. The methods works on entire database in reasonable time frame. Processing on entire CASIA1 databases containing 1721 images took 58.36 s, (including training and testing time) giving average accuracy of 93 %.

The organization of the paper is as follows: Sect. 2 covers a brief review of related research in area. The proposed methodology is explained in Sect. 3. Experimental setup and discussion on results is covered in Sect. 4 respectively. The paper is concluded in Sect. 5.

2 Previous Research

Blind image forgery detection techniques mostly strive for features that discriminates an original image from the forged one. These distinguishing features are extracted from spatial and frequency domain that can be used for both forgery detection and forgery localization. Most of the forgery detection methods use block based approach or approach based on feature points. In block based approach, given image is divide into overlapping or non-overlapping blocks, then features are extracted from each block. Dividing into non-overlapping blocks and extracting features is computationally expensive. In feature point approach, descriptors like SIFT are used to identify points that are invariant to geometric transformation. It is mostly used to detect copy-move forgery.

Processing performed in frequency domain being faster and efficient; it has demonstrated wide applications in digital forensics. Wavelet decomposition, owing to its

superior multi-resolution and space frequency analytical capability is extremely significant in highlighting the inconsistencies caused by tampering operations. In [5], Hilbert-Huang transform (HHT) is used for image splicing detection as the cut-paste operation is highly non-linear and non-stationary. Moments of characteristic functions with wavelet decomposition are used to draw a statistical model of natural image. A natural image model by [6], also uses DWT to the image pixel 2-D array, along with multi-block DCT coefficient 2-D arrays, and prediction-error 2-D arrays. Markov features are generated in [7] from the transition probability matrices in DCT domain. It is used to confine the correlation between block DCT coefficients. It also utilizes the features generated in DWT domain to study dependency among wavelet coefficients across orientations and scales. Copy-move forgery is detected in [8] using Dyadic Wavelet transform (DyWT). It is found shift invariant as compared to DWT and is suitable for copy-move attack. It divides the LL and HH further into overlapping blocks for extracting features. Singular value decomposition (SVD) approach is used in [5] that applies different transforms including DCT, DWT and DFT.

Splicing operation when performed on digital images always results in distortion of textural micro-pattern of the image. These inconsistencies in texture can be captured by using textural descriptors such as Local Binary Pattern (LBP), Weber Local Descriptors (WLD), GLCM etc. Multi resolution WLD is used in [10]. This method is later compared with LBP texture in [7]. It gives maximum accuracy of 94.29 % with feature level fusion and 1330 features. Both multi-WLD and multi-LBP give higher performance accuracy for splicing detection than copy-move forgery detection. This is because, copy-move operation results in similar texture micro-pattern in the copied and pasted regions. This reduces the discrimination capability of the classifier.

Many passive-blind methods are proposed in literature but most of the methods are tested on customized database, in order to check the presence or absence of a particular tampering trait. It is desirable to test on standard tamper detection database available for research purpose or realistic tampered images, as it enables better evaluation of the reliability and robustness of a given method. Databases are made public for researcher purpose like the splicing database [8, 9], Copy-Move databases [10].

The current paper uses single level wavelet decomposition. It then studies the texture features in its approximate band and also in horizontal, diagonal and vertical direction for detecting inconsistencies. It is fast and efficient as it does not use overlapping block for extracting features. Evaluation is done on four different kinds of splicing database. The results thus can be easily reproduced and verified.

3 Proposed Method

The methodology of the proposed forgery detection technique is illustrated in Fig. 1. A given colored image first converted to YCbCr color space. It represents a color in the form of luminance component (the Y channel) and chrominance component (Cb or Cr channel). Cb & Cr are the blue and red difference respectively. Luminance channel describes the image content and is strong enough to hide the tampering traces. The Chrominance channels describe the weak signal content of the image like edges. Any inconsistency in these

edges, caused by the tampering operation are emphasized and hence become noticeable. Therefore, further processing is performed on the extracted chrominance channel (Cr) after experimental verification as it was found to give maximum accuracy. Single level discrete wavelet transform is then applied to get the low level coefficients and approximation coefficients. The texture of these [LL, LH, HL, HH] is extracted using local binary patterns. For efficient training testing of features, histogram of these texture images is taken. These lbp histograms are concatenated and fed to SVM classifier for training. The given algorithm was tested on different color models, image types and different databases.

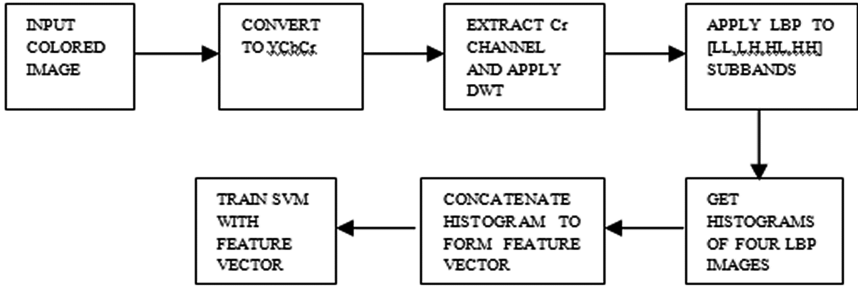


Fig. 1. Block diagram of the proposed methodology

3.1 Wavelet Decompositions of the Image

The discrete wavelet transform (DWT) can provide unique and discriminatory representations that can quantify vital and interesting structures (edges, details) with good resolution by few coefficients. It is also computationally effective. These coefficients can be directly used as features. These features can be directly extracted from the wavelet domain, describing the anomalies in the data. It basically reduces the correlation among wavelet coefficients and provides energy compaction in few wavelet coefficients. Wavelet analysis gives approximations and detail coefficients. The approximations are the high-scale, low-frequency components of the signal. The details are the low-scale, high-frequency component. The wavelet transformation of a given image helps in analyzing it in different scales and orientation. As the splicing process often introduces sharp transition in the image 2-D array in terms of edges, lines and corners which are characterized by high frequency components in the Fourier transform domain. Wavelet analysis characterizes these short-time transient in signals that help in measuring local sharpness or smoothness from wavelet coefficients (Fig. 2).

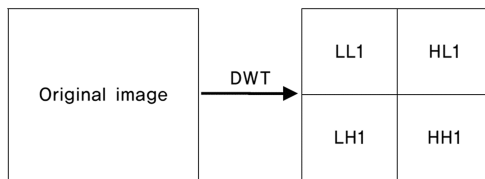


Fig. 2. Single level DWT decomposition of an image

3.2 LBP Histograms

LBP is a commonly used local texture descriptor. Main advantages over other descriptors is that its computational complexity is low and is invariant to monotonic illumination changes [7]. It labels each pixel in the image by thresholding the neighborhood pixels with the centre pixel, giving a sequence of binary numbers. Histogram of these labeled values is used to describe texture. A basic LBP operator is calculated in a rectangular window. LBP can also be extracted in a circular neighborhood (P, R), where P is the number of neighbors and R is the radius of the neighborhood. In this work, results are reported with circular LBP, using $P = 8$ and $R = 1$. The histogram from each sub-band is normalized. The histogram has 256 bins corresponding to 256 gray values. The histogram from four sub-bands is concatenated to forms the feature vector of dimension 1024. The reason behind using LBP is that whenever cut and paste operation is performed in images the consistency in texture in the four sub images gets disrupted. The svm classifier learns these inconsistencies that get highlighted in tampered images as compared to authentic images. It is even more prominent when splicing operation is done used JPEG images due to 8×8 grids mismatching which results in blocking artifacts. Therefore, the texture feature can be an excellent indicator of tampering of digital media (Fig. 3).

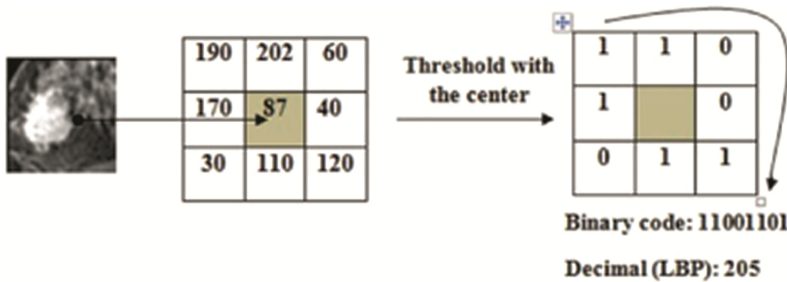


Fig. 3. Process of LBP code generation for a given window size

4 Experiments and Results

The experimental environment for the proposed algorithm is Intel(R) Core i7-4702MQ CPU 2.20 GHz, 4 GB memory and MATLAB R20015. This section first provides brief description of the various databases used for evaluation, followed by the evaluation policy. Finally, the results of the experiment conducted are presented and discussed.

4.1 Evaluation Database

The proposed method was tested on various standard image dataset available for detecting image splicing. The CASIA1, CASIA2 and Columbia spliced and Columbia uncompressed are the commonly used databases that contain authentic (Au) and tampered images (Tp). The details of the various databases are given in Table 1.

Table 1. Standard Database used for evaluation of the Proposed method

Database	Au	Tp	Total	Image type	Size
CASIA TIDE v1	800	921	1721	jpg	384 × 256
CASIA TIDE v2	7491	5123	12614	bmp, tiff, jpg	240 × 16 to 900 × 600
COLUMBIA	933	912	1845	bmp	128 × 128 image blocks
COLUMBIA uncompressed	183	180	363	tiff	757 × 568 to 1152 × 768

4.2 Evaluation Policy

The current application is a two class problem, wherein a given image has to be classified as Authentic (class 1) or Tampered (class 2). For training and testing, SVM classifier was used with 10-fold cross validation. and rbf as kernel function. The performance measures used are Accuracy, Sensitivity, Specificity and Area under ROC curve (AUC).

Accuracy is the percent ratio of correctly classified images to the total number of images and is calculated using the following equation:

$$\text{Accuracy} = (\text{TP} + \text{TN}) / (\text{TP} + \text{TN} + \text{FP} + \text{FN}) \times 100 \quad (1)$$

where, TP, TN, FP, FN respectively, are the numbers of true positive, true negatives, false positives and false negatives. TP in this scenario occurs when a tampered image is detected as tampered and TN is when a original image is detected as authentic.

Sensitivity or True Positive Rate (TPR) is ratio of all correctly classified positive cases divided by all True Positive cases. *Specificity* or False Positive Rate (FPR) is defined as the ratio correctly classified negative cases divided by all True Negative cases.

ROC curve consists of FPR as horizontal axis and TPR as the vertical axis.

$$\text{Sensitivity (true positive): TPR} = \text{TP} / (\text{TP} + \text{FN}) \quad (2)$$

$$\text{Specificity (false positive): FPR} = \text{FP} / (\text{FP} + \text{TN}) \quad (3)$$

4.3 Results and Discussion

The performance of the proposed method based on the four parameters, and tested on different databases is given Table 2.

Table 2. Performance of the proposed method

Database	Accuracy (%)	Specificity (%)	Sensitivity (%)	AUC
CASIA TIDE v1	92.62	95.55	89.25	.9824
CASIA TIDE v2	94.09	97.35	91.87	.9885
CASIA2 JPEG	97.34	92.20	98.76	.9935
COLUMBIA	75.93	72.26	79.53	.8039
COLUMBIA uncompressed	87.05	83.89	90.16	.9191

Performance on Spliced Image Dataset: When run on entire database, maximum accuracy of 94.09 % is achieved for CASIA2 dataset containing 12614 images. Minimum accuracy is attained for the Columbia spliced dataset of 75.93. It is due to the small size of authentic and spliced image blocks (128×128) present in the database, which is not sufficient enough to extract features based on edge inconsistencies. The ROC curve of the proposed method for the spliced dataset is given in Fig. 4. The method follows very simplified approach and is efficient as it executed on entire CASIA1 databases containing 1721 in 58.36 s, giving average accuracy of 93 %.

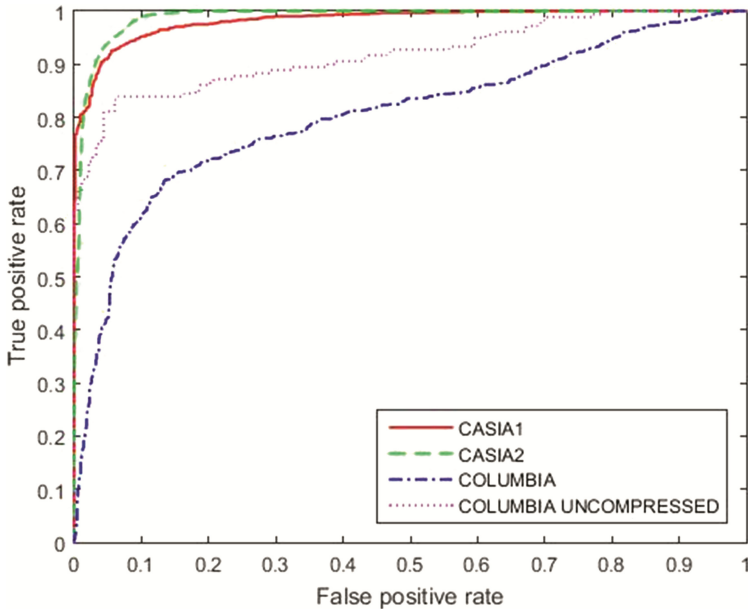


Fig. 4. ROC curves of the proposed method on Splicing databases

Performance on JPEG Images: The proposed method was tested on the JPEG images available in standard database. CASIA v1, CASIA v2. On JPEG images it exhibited maximum accuracy of 97.34 % and AUC value is 99.35. Such high accuracy is attained because JPEG compression algorithm involves block based processing, where image is divided into 8×8 blocks on which DCT is applied and quantization is performed. Whenever a portion from JPEG images is cut and/or pasted onto a JPEG image it mostly results in block mismatching i.e., one JPEG block may not completely align with other blocks or their neighbor. It generates some extraneous edges. Such inconsistency in edges helps in discriminating tampered images from authentic one.

4.4 Comparison with Other Passive-Blind Methods

The current method was compared with some of the latest methods that were tested on the standard databases. However, the results reported in these methods are on only one database, either CASIAv1.0, CASIAv2.0, or Columbia. The proposed method is better than [7] with multi-level LBP descriptor and method in [11] using GLRLM texture descriptors. However, its performance is comparable to the methods in [10] when tested in CASIA1 and multi-level WLD. Also it shows very slight improvement when compared to method in [6] when tested on CASIA2. Accuracy of [7] was improved to 94.19 % with Feature level fusion from both Cb & Cr channel and multi-level WLD. Our method gives comparable performance without feature fusion. Method proposed in [12] is tested only on Columbia database and shows better performance than the proposed approach. The method proposed in [13] takes into account different scales and orientations of steerable pyramid transform (SPT). Our method shows comparable performance with single level decomposition, except in the case of Columbia database where accuracy of 96.39 % is achieved in [13]. The proposed method gives equally good performance when uniform LBP is applied which results in substantial decrease in number of histogram bins, further reducing complexity. Also, when tested on standard copy-move database it gives accuracy of 96 %, results of which are not included in this paper (Fig. 5 and Table 3).

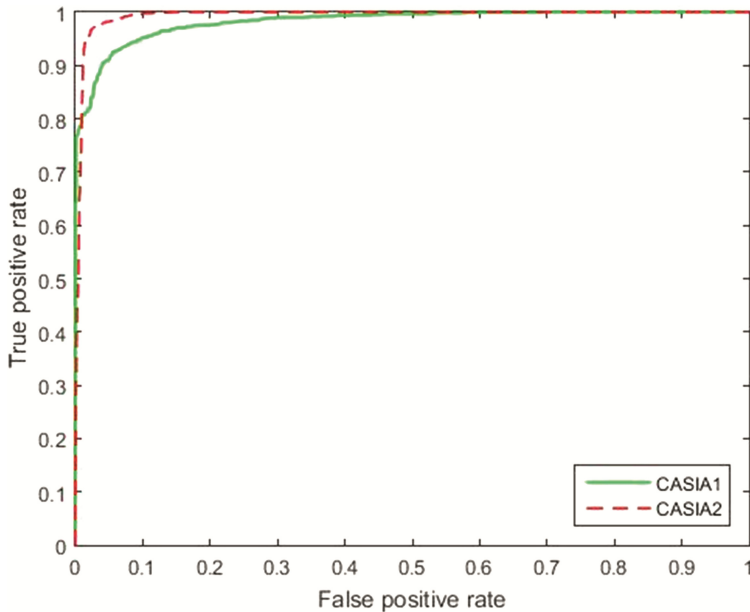


Fig. 5. ROC curves of the proposed method for JPEG images

Table 3. Comparison of the proposed technique is with other approaches

Method	Approach Used	CASIA1	CASIA 2	COLUMBIA	COLUMBIA uncompressed
[7]	Markov feature in DCT and DWT domain.	–	93.42	–	–
[7]	Multi WLD Multi LBP (Cr Channel outcome)	92.44 85.93	–	–	–
[12]	Edge gradient matrix and DWT	–	–	80.58	–
[11]	GLRLM Texture features	80.71	–	–	–
Proposed Method	DWT and LBP Histogram	92.62	94.09	75.93	87.05

5 Conclusion

A passive-blind method based on single-level DWT and LBP is proposed for splicing detection in digital images. Splicing operation introduces sharp transitions characterized by high frequency components. These localized changes in digital images can be captured by reducing the correlation among wavelet coefficients. Further, inconsistencies in the textural microstructure of the wavelet decompositions are studied using LBP as texture descriptor. It is tested on four different kinds of standard tampered image datasets. It is concluded that it gives maximum accuracy of 97.34 % when the splicing is done using JPEG images. One limitation is that its performance gets affected when size of forged image is very small like in Columbia spliced database and thus will be included in the future work.

References

1. Babak Mahdian, S.: A bibliography on blind methods for identifying image forgery. *J. Sig. Process. Image Commun.* **25**(6), 389–399 (2010)
2. Birajdar, G.K., Mankar, V.H.: Digital image forgery detection using passive techniques: a survey. *Digital Invest.* **10**(3), 226–245 (2013)
3. Farid, H.: A Survey of Image Forgery Detection. *Signal Process. Mag.* **26**(2), 16–25 (2009)
4. Piva, A.: An overview on image forensics. *ISRN Signal Process. J.* **2013**, 22 (2013). Article ID 496701
5. Fu, D., Shi, Y.Q., Su, W.: Detection of image splicing based on hilbert-huang transform and moments of characteristic functions with wavelet decomposition. In: Shi, Y.Q., Jeon, B. (eds.) *IWDW 2006. LNCS*, vol. 4283, pp. 177–187. Springer, Heidelberg (2006)
6. Yun, Q., Shi, C.: A natural image model approach to splicing detection. *ACM*, 12–13 (2007)
7. He, Z., W, Lu, et al.: Digital image splicing detection based on Markov features in DCT and DWT domain. *Pattern Recogn.* **45**, 4292–4299 (2012)

8. Ghulam Muhammad, M.: Passive copy move image forgery detection using undecimated dyadic wavelet transform. *Digital Investigation*, 49–57 (2012). Elsevier
9. Zahra Moghaddasi, H.: A comparison study on SVD-based features in different transforms for image splicing detection. In: *International Conference on Consumer Electronics-Taiwan (ICCE-TW)*, Taiwan, pp. 13–14 (2015)
10. Muhammad Hussain, G.: Image forgery detection using multi-resolution weber local descriptors. In: *EuroCon 2013, Zagreb, Croatia*, pp. 1570–1577 (2013)
11. Muhammad Hussain, S.: Comparison between WLD and LBP descriptors for non-intrusive image forgery detection. (2014)
12. Ng, T.T., Chang, S.F.: A dataset of authentic and spliced image. In: *Columbia Image Splicing Detection Evaluation Dataset, DVMM, Columbia University*. <http://www.ee.columbia.edu/dvmm/>. Accessed 2004
13. Dong, J., Wang, W.: CASIA tampered image detection evaluation (TIDE) database v1.0 and v2.0 (2011). <http://forensics.idealtest>
14. Amerini, I.: A SIFT-based forensic method for copy-move attack detection and transformation recovery. *IEEE Trans. Inf. Forensics Secur.* **6**(3), 1099–1110 (2011)
15. Saba Mushtaq, A.: Novel method for image splicing detection. In: *International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, pp. 2398–2403 (2014)
16. He, Z., Sun, W.: Digital image splicing detection based on approximate run length. *Pattern Recogn. Lett.* **32**, 1591–1597 (2011)
17. Muhammad, G., Al-Hammadi, M.H., Hussain, M., et al.: Image forgery detection using steerable pyramid transform and local binary pattern. *Mach. Vis. Appl.* **25**, 985–995 (2014)

An Image Forensic Technique for Detection of Copy-Move Forgery in Digital Image

Ashwini Malviya^(✉) and Siddharth Ladhake

Sipna College of Engineering and Technology, Amravati, India
ash.malviya@gmail.com, sladhake@yahoo.co.in

Abstract. Image morphing is a common practice nowadays. To validate a digital image is considered as a perplexing task in the field of image forensic. With numerous kind of tampering been carried out on a digital image, the paper focuses on a detection of common forgery referred to as copy-move forgery or cloning, which is nearly untraceable. The paper contemplates on the color content of the forged image and employs three different methods of feature extraction to aid the detection of forgery. The experimental results show that the feature extraction methods employed detects the forged region accurately and are also effective to rotation and scaling. A performance analysis in detection of forgery for the three methods in terms precision and recall is also presented in the paper, along with a comparison with other state-of-the-art detection methods.

Keywords: Forgery detection · Digital Forensics · copy-move forgery detection

1 Introduction

In the arena of image forensic, the confirmation of a digital image has been a thought-provoking assignment. With thriving technology, the usage of refined cameras and photo editing software has risen. A smart phone itself has different user-friendly options to alter a picture captured, as desired. It becomes tedious to identify an altered image from the real image as it has become very easy to tamper an image. Scrutiny of all kind of evidence is an inevitable part of Forensic science. The legal system banks on the forensic examination.

A digital image can be forged, by adding, deleting or changing an object in the image, or it can be spliced with other image. The forgery taken into consideration here is the copy-move forgery also referred as cloning. In this kind of tampering, any undesired content of the image can be hidden by pasting a portion copied from the same image on the object. The image contents also can be enriched by cloning.

Though this cloning is not traceable by a common viewer, it creates glitches in the statistics of the pixels of the image. In forensic pertaining to digital image, the pixel plays a crucial role. The changes occurring in these pixel values on tampering can aid the detection of the forgery. The digital forensics investigation is carried out by active and passive approaches. Unlike the active detection approach, the paper presents a passive approach for detection of forgery which works in absence of watermark. The approach explores the HSV histogram, color moment and auto color Correlogram for

feature extraction. These extraction methods are extensively employed in image retrieval systems. In the rest of the paper we discuss the different state-of-the-art techniques for copy-move tampering detection, followed by the proposed approach and the experimental results followed by conclusion.

2 Related Work

Extensive research has been carried out in past few years for copy-move forgery detection. In this segment we probe into different techniques used for the forgery detection. The traditional methodology for the detection scheme starts with usually dividing the image into overlapping blocks, with specific block size. The block based methods implemented by [2, 3] use DCT coefficient and PCA for extracting features from each block respectively. Further the matrix formed is lexicographically sorted for identifying the duplicated region.

A robust detection technique proposed by W. Luo et al. [4], derived the block characteristics of each block and compared it with other blocks to get the match. These methods were computationally less complex but failed in detection at event of variation arising due to noise and compression. A Keypoint based method [6, 10] uses Scale Invariant Feature Transform algorithm for detection of copy-move forgery.

DWT and SVD based detection technique proposed by G. Li et al. [5], gives a reduced dimension representation. The sorting is simpler and the technique is robust to retouching and compression. N. Myna et al. [7] also presented reduced dimension representation by determining the wavelet transform of the image. The detection of forgery is based on extracting log polar coordinates and phase correlation.

Extraction of features by calculating the Fourier-Mellin Transform of the blocks of the image was proposed by Bayram et al. [9]. The attributes of Fourier Mellin transforms were analyzed in this method, also counting bloom filter was preferred over lexicographically sorting.

Jing-Ming Guo et al. [12] proposed an efficient detection technique using improved daisy descriptor. The method also incorporates adaptive non-maximal suppression for matching of keypoints. This descriptor used is rotation invariant and therefore can detect duplicated region which have been subjected to any kind of transformation. Leida Li et al. [13] used Polar Harmonic transform of circular block for feature extraction. The match was detected on post processing the image.

A detection scheme which involved feature extraction of blocks by histogram of Gabor magnitude was proposed by Chen-Ming Hsu et al. [14]. Like [13] the match was detected on applying post processing operations. Recently Cheng-Shian Lin et al. [15] reduced the computational time by 10% by introducing the cluster expanding block algorithm for detection of duplicated region. In a recent prior work [16] analyzed the color contents of the image for feature extraction which efficiently reduces the complexity.

3 Proposed Scheme

The proposed scheme as illustrated in Fig. 1, first preprocesses the input image, followed by dividing the image into blocks. Often to make the tampering imperceptible the copied region is rotated or flipped or subjected to some other transformation before pasting, therefore we subject each block to transformations. Further the features are extracted from each block. The extracted features form an array, features of each block is matched with every other block to identify the cloning. The match detection is done by using suitable similarity measure.

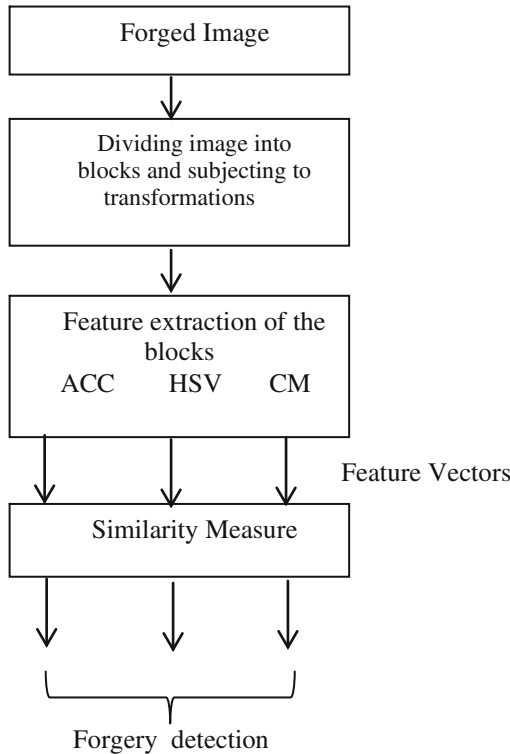


Fig. 1. Functional Flow diagram for proposed scheme

Preprocessing the image involves filtering the noise from the image and dividing the image into blocks. Dividing the image into blocks has been traditional in cloning detection methods. We divide the image of size $m \times n$, into overlapping blocks of size $m/4 \times n/4$. Each block is subjected to 8 transformations, mainly rotation, flipping and transpose. Features are extracted from each block. Features are extracted by computing the HSV histogram, color moments and auto color Correlogram of each block.

HSV Histogram. As we focus on the pixel for forgery detection, the HSV color space can be well analyzed for varying values of the Hue, Saturation and Intensity of pixel.

The image is first split into H, S, and V plane and then each plane is quantized, specifying the number of quantization levels. A final histogram is created and then normalized.

Color Moments. Color feature vectors are derived from the RGB model, the scheme takes into account two color moments. The first color moment namely mean and the second moment namely standard deviation are computed for each channel. The mean and the standard deviation are defined as follows:

$$\mu = \frac{1}{M} \sum_{i=1}^M f_i \quad \sigma = \sqrt{\frac{1}{M} \sum_{i=1}^M (f_i - \mu)^2} \tag{1}$$

AutoColorCorrelogram (ACC). The AutoColorCorrelogram was introduced by [8] for spatial color indexing. It computes the mean color by taking in consideration all the pixels of a particular color, say C_j at a distance k from all pixels of another color, say C_i . It can be defined as follows:

$$ACC(i, j, k) = Avg\gamma_{C_i, VC_j}^{(k)}(I) = \left\{ Avg\gamma_{C_i, C_{jr}}^{(k)}(I), Avg\gamma_{C_i, C_{js}}^{(k)}(I), Avg\gamma_{C_i, C_{jb}}^{(k)}(I) \middle| C_i \neq C_j \right\} \tag{2}$$

Where the mean colors are formulated as follows:

$$Avg\gamma_{C_i, C_{jx}}^{(k)}(I) = \frac{\prod_{C_i, C_{jx}}^{(k)}(I)}{\prod_{C_i, C_{jx}}^{(k)}(I)} \bigg|_{x = r, g, b} \tag{3}$$

The image here is quantized into m colors. VC_j implies, RGB value of color m .

Forgery Detection. Manhattan distance or commonly referred as L1 norm is used as a similarity measure for comparing the features of each block with every other block. The features of each block are arranged in a row of a matrix. The L1 measure requires less computational time as compared to Euclidean distance. Its simplicity and robustness makes it more appropriate for match detection. If $a=(x_1, x_2, \dots, x_n)$ and $b=(y_1, y_2, \dots, y_n)$ the Manhattan distance is obtained by,

$$MH(a, b) = \sum |x_i - y_i| \quad = \quad \text{for } i = 1, 2, \dots, n. \tag{4}$$

The distance is calculated by taking sum of absolute difference between the considered block feature vectors and all other blocks feature vectors. Then the resultant matrix is sorted to find the least difference row.

4 Visual Results

The detection scheme developed is tested on few images which are created individually and on image database CoMoFoD which is made available online by [17]. The database comprises of set images of small and large categories, where the images are subjected

to various alterations viz. rotation, scaling, distortion, translation and combination of the same. We have considered the image set which is small image category (512×512). Figure 2 consist of three images, wherein the first image is the original image, the second image is forged image, and wherein a portion of image is copied and rotated earlier to pasting in the same image, and a duplicated region is identified by using ACC for feature extraction in the next image. The images in Figs. 3, 4 and 5 are from the CoMoFoD database.



Fig. 2. Original image, forged image and forgery detected using ACC. The forged region has undergone rotation prior to pasting.



Fig. 3. Original image, forged image (cloned region scaled) and forgery detected using ACC.



Fig. 4. Original image, forged image (plain copy move) and forgery detected using Color moments

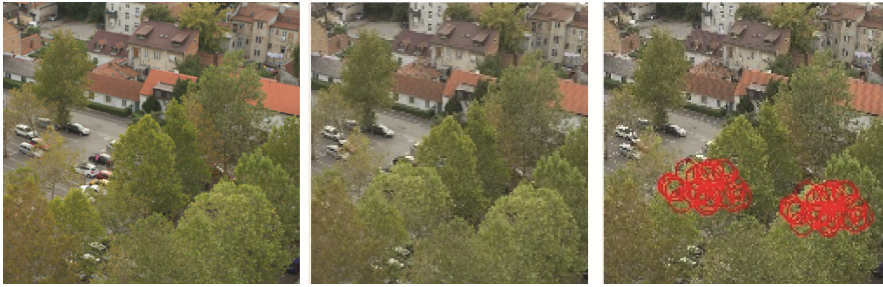


Fig. 5. Original image, forged image and forgery detected by extracting HSV Histogram of each block.

Figure 3 depicts a forgery detection using ACC where the forged part is scaled before pasting.

Figure 4 depicts detection of copy-move forgery using color moments of the blocks for feature extraction.

The forgery is detected for the image in Fig. 5 using HSV Histogram, combinational transformations is used to tamper the image.

5 Metric and Statistical experimental Results

The performance analysis is carried out at image level. The performance assessment is based on the false and true outputs. The precision P and recall R is computed as follows:

$$P = \frac{T_p}{T_p + F_p} \quad R = \frac{T_p}{T_p + F_n} \quad F_1 = \frac{2PR}{P + R} \tag{5}$$

T_p represents the number of forged images detected correctly; F_p represents the number of images incorrectly detected as forged and F_n indicates the falsely missed tampered images. The combinational measure of precision and recall is given by F_1 . Table 1 gives the performance analysis of about 200 images from the CoMoFoD database, which consists of 100 forged and 100 original images. The performance analysis in terms of precision for three different methods employed is shown in figure 6.

Table 1. Cloning detection results at image level.

Methods	Tp	Fp	Fn	P (%)	R (%)	F ₁
ACC	94	6	6	0.940	0.940	0.940
Color Moments	94	8	6	0.922	0.940	0.931
HSV	92	11	8	0.893	0.920	0.906

AutoColorCorrelogram shows effective detection with precision of 94% when the proposed scheme is tested on images from the database. Also a comparison with state-of-the-art methods employed [11] for copy-move forgery detection is presented in graphical form in figure 7.

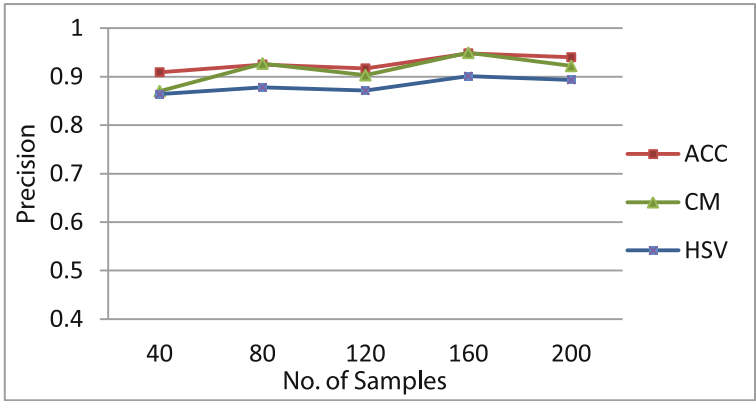


Fig. 6. Comparative analysis based on F1 score.

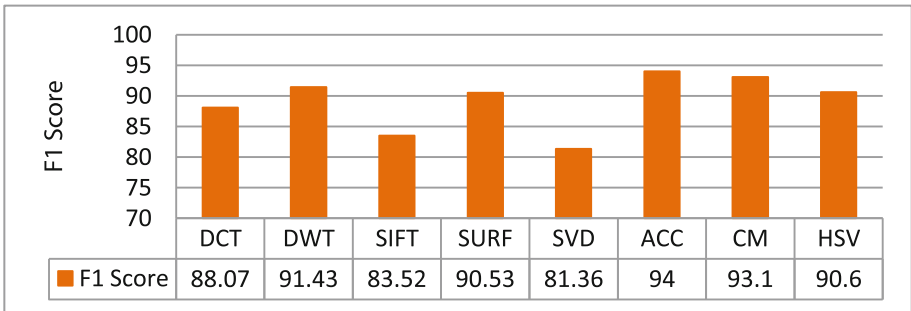


Fig. 7. Comparative analysis based on F1 score.

6 Conclusion

In proposed scheme, an efficient copy move tampering detection technique is developed. The feature extraction methods implemented in the scheme have been employed widely for content based image retrieval earlier. The proposed scheme presents three different detection techniques for copy move forgery detection, which is less complex and provides robustness to transformation and noise. AutoColorCorrelogram shows effective detection with highest precision when the proposed system is confirmed on images from the database. The detection method is also effective in detection of forgery on event of scaling and multiple cloning in the same image.

References

1. Farid, H.: A Survey of Image Forgery Detection. *Signal Process. Mag.* **26**(2), 16–25 (2009)
2. Fridrich, J., Soukal, D., Lukáš, J.: Detection of copy-move forgery in digital images. In: *Proceedings of Digital Forensic Research Workshop*, August 2003
3. Popescu, A., Farid, H.: Exposing digital forgeries by detecting duplicated image regions. Department of Computer Science, Dartmouth College, Technical report TR2004-515 (2004). www.cs.dartmouth.edu/farid/publications/tr04.html
4. Luo, W., Huang, J., Qiu, G.: Robust detection of region-duplication forgery in digital images. *Int. Conf. Pattern Recogn.* **4**, 746–749 (2006)
5. Li, G., Wu, Q., Tu, D., Sun, S.: A sorted neighborhood approach for detecting duplicated regions in image forgeries based on DWT and SVD. In: *IEEE International Conference on Multimedia and Expo*, pp. 1750–1753, July 2007
6. Huang, H., Guo, W., Zhang, Y.: Detection of Copy-Move Forgery in Digital Images Using SIFT Algorithm. In: *Pacific-Asia Workshop on Computational Intelligence and Industrial Application*, vol. 2, pp. 272–276, December 2008
7. Myna, A.N., Venkateshmurthy, M.G., Patil, C.G.: Detection of region duplication forgery in digital images using wavelets and log-polar mapping. In: *IEEE International Conference on Computational Intelligence and Multimedia Applications*, pp. 371–377, December 2007
8. Tungkasthan, A., Intarasema, S., Premchaiswadi, W.: Spatial color indexing using ACC algorithm. In: *Seventh International Conference on ICT and Knowledge Engineering*, pp. 113–117 (2009)
9. Bayram, S., Sencar, H., Memon, N.: An efficient and robust method for detecting copy-move forgery. In: *IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 1053–1056, April 2009
10. Amerini, I., Ballan, L., Caldelli, R., Del Bimbo, A., Serra, G.: A SIFT-based forensic method for copy-move attack detection and transformation recovery. *IEEE Trans. Inf. Forensics Secur.* **6**(3), 1099–1110 (2011)
11. Christlein, V., Riess, C., Jordan, J., Riess, C.: Angelopoulou, E: An evaluation of popular copy-move forgery detection approaches. *IEEE Trans Inf. Forensics Secur.* **7**(6), 1841–1854 (2012)
12. Guo, J.M., Liu, Y.-F., Wu, Z.-J.: Duplication forgery detection using improved DAISY descriptor. *Expert Syst. Appl.* **40**, 707–714 (2013). Elsevier International Journal
13. Li, L., Zhu, S., Xiaoyue, H.W.: Detecting copy-move forgery under affine transforms for image forensics. *Comput. Electr. Eng.* **40**(6), 1951–1962 (2014). Elsevier Ltd.
14. Hsu, C.-M., Lee, J.-C., Chen, W.-K.: An efficient detection algorithm for copy-move forgery. In: *10th Asia Joint Conference on Information Security*, pp 33-36, May 2015
15. Lin, C.-S., Chen, C.-C., Chang, Y.-C.: An efficiency enhanced cluster expanding block algorithm for copy-move forgery detection. In: *International Conference on Intelligent Networking and Collaborative Systems (INCOS)*, pp. 228–231, September 2015
16. Malviya, A.V., Ladhake, S.A.: Pixel based image forensic technique for copy-move forgery detection using auto color correlogram. *Procedia Computer Science* **79**(2016), 383–390 (2016). In: *7th International Conference on Communication, Computing and Virtualization 2016*. Elsevier Ltd.
17. Tralic, D., Zupancic, I., Grgic, S., Grgic, M.: CoMoFoD - new database for copy-move forgery detection. In: *Proceedings of 55th International Symposium, ELMAR-2013*, pp. 49–54, September 2013

Secure Authentication in Online Voting System Using Multiple Image Secret Sharing

P. Sanyasi Naidu¹ and Reena Kharat^{1,2(✉)}

¹ Department of Computer Science and Engineering, GITAM University, Vishakhapatnam, India
snpasala@yahoo.com, reenakharat@gmail.com

² Department of Computer Engineering, Pimpri Chinchwad College of Engineering, Pune, India

Abstract. In the democratic country, all eligible citizens of the country are involved in decision making through voting. Right to vote allows voters to choose their candidate and government. In order to increase the number of voting, there is a need to design a secure e-voting system. The important part of the online voting system is authentication of an individual as a legitimate voter. Our system provides secure authentication based on non-transferable personal credentials like biometric features. Visual cryptography is used to provide confidentiality to the biometric database. Hash of the PIN is embedded into image shares using cryptography and steganography. The proposed system provides biometric and password authentications at a time.

Keywords: Online Voting · Authentication · Multiple Secret Sharing · Steganography · Biometrics

1 Introduction

In current scenario, a voter has to carry a paper based voter identity card to authenticate him/her at the time of voting. This authentication process is manual and paper based. Fake voter may create fake voter ID card with his/her own photo and name of eligible voter. For authentication, the officer matches photo in Voter ID with present voters face. As voter ID card is paper based, fake voter passes through this authentication as authentic voter. He votes in lieu of other eligible voters. Another problem is fake voter votes more than one time by using multiple voter ID cards. Therefore, there is need to replace manual paper based authentication with digital authentication.

At the same time, trustworthy remote voting system is today's requirement. It will provide access to overseas people, military people and people not present in state on the day of voting to exercise their vote. Voter will believe in the system if it is highly secure. Confidentiality, integrity, identification and authentication are the main security considerations for remote voting system.

Our proposed system gives solution for digital authentication in online voting system. This method is based on non-transferable credentials like biometrics.

2 Related Work

In digital world of today, there is need for secure online voting system. Online voting system has four phases as registration, authentication, vote casting and vote counting. Here, we concentrate on registration and authentication.

Different authentication scheme has been proposed. In [1], author has proposed registration and authentication scheme using PIN. During registration, PIN is generated. Voter accesses voting system and authenticates himself by using login Id and PIN. In [3], Kerberos mechanism is used to implement complete online voting system. Here, voter authentication is done through his ID and password. In [1, 3], transferable credentials are used. Authentic voter may transfer these credentials to others with or without his will. Anyone having these credentials can login and vote. So authentication in voting system based on non-transferable credentials is essential.

Every individual has unique biometric features. Biometrics is biological characteristics which includes fingerprints, retinal and iris scanning, hand geometry, voice patterns, facial recognition, DNA and other techniques. Digital authentication using biometric features helps us to avoid fake voter to pass through authentication. Due to biometric authentication, one voter can cast one vote at a time. So, multiple votes casting by single voter are prevented.

Biometric is non-transferable feature which is used for authentication in online voting system [10]. Secret voting password provided during registration is used along with their captured biometric identification [12]. Smart card is provided to voter which contains voters fingerprint image. During authentication, fingerprint from smart card is matched with fingerprint from database. Drawback of this system is that it does not check whether the person holding smart card is the legitimate voter.

If cryptography and steganography is combined then security of embedded data is enhanced [2]. In [13], LSB technique is used for hiding data in cover image. If color image is used, each pixel contains 24 bits of RGB value. So, total 3 bits can be embedded in each pixel of color image [4]. In LSB technique of steganography, pixels are selected sequentially for embedding information. Malicious user can easily extract secret information from stego-image by reading LSB of each pixel.

In [11], cryptography and steganography is used at the same time to provide password and biometric security to voter accounts. Algorithm uses fingerprint as biometric measure and secret key. Pixel selection is done randomly and these random numbers are generated using password as seed. Therefore, it is difficult for malicious user to extract secret information from stego-image. Fingerprint image stored in database is used to retrieve password from stego-image. System does not take live fingerprint of voter. System authenticates voter using password which is again transferable.

Biometric device like fingerprint scanner is costly and not available with every user. As microphone is cheaper and commonly available device, voice is used for authentication [9]. Fingerprint has less false acceptance rate and false rejection rate than that of voice. Therefore, fingerprint is more preferred than voice.

In [7], visual cryptography is used to hide password. It provides low cost mutual authentication for voters and election servers. The drawback of system is that it does not check the person entering password is the legitimate voter or not. So, anyone intercepting

mail can cast vote in lieu of legitimate voter. In [6], identification using card reader followed by authentication using live fingerprint is used to enhance security. Also use of Kerberos in authentication provides protection against eavesdropping and replay attacks.

For secure and robust authentication, we can have to use combination of cryptography and steganography followed by visual cryptography.

3 Background

3.1 Visual Cryptography

Naor and Shamir [5] proposed visual cryptography first time in 1994. The scheme divides an image into two shares. When these two shares are overlapped with each other, original image is revealed. Without true shares, the original image is not revealed. Single share doesn't reveal any secret.

For white pixel two identical shares are chosen. For black pixel opposite shares are chosen. Figure-1 shows how black and white pixel of an image is divided into two shares. It shows two options for each pixel. Any one of the option is chosen randomly.

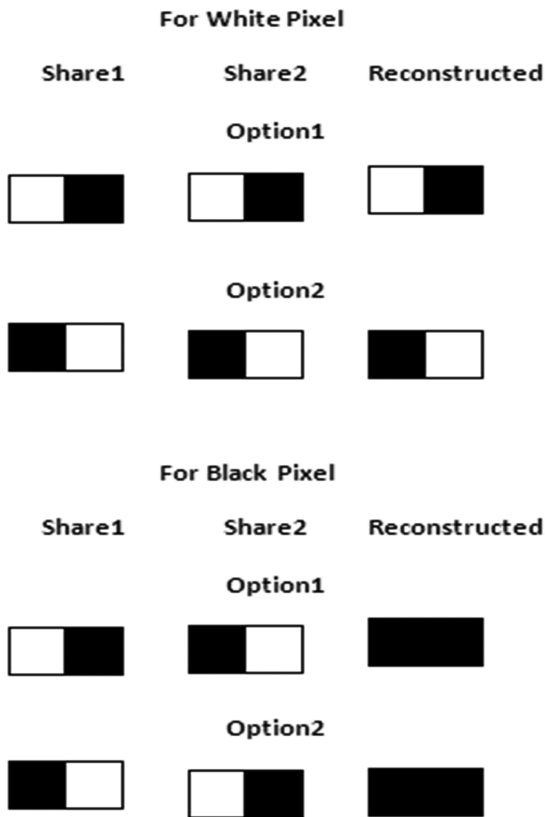


Fig. 1. Visual cryptography

3.2 Multi Secret Sharing Scheme

In [8], authors have proposed a method of sharing two images secretly using secret sharing. Suppose there are two images $image_1$ and $image_2$ to be shared secretly. Two shares of each image are created using secret sharing. $image_1$ is divided into two shares $share_{11}$ and $share_{12}$. $image_2$ is divided into two shares $share_{21}$ and $share_{22}$. One share of $image_1$, $share_{11}$ is rotated by 90° and stacked on $share_{21}$ of $image_2$. Similarly, another share of $image_1$, $share_{12}$ is stacked on $share_{22}$ of $image_2$. This scheme is used in our proposed system to store fingerprint image and photograph of voter secretly.

4 Proposed System

The main aim of authentication module is not to allow fake voter to vote in lieu of eligible voter. Here, we are using photograph and fingerprint image as non-transferable credentials. We will not store photograph, fingerprint and PIN as it is in the database so that tampering will be infeasible. We use multiple image sharing using visual cryptography and steganography. Our proposed system is divided into two phases as registration phase and authentication phase.

4.1 Registration Phase

During registration, election officials will first check if the individual is eligible to vote by checking his/her current voting card and identity proof issued by government of that country. Once it is verified, voter is registered by taking photograph and fingerprint. Voter is also directed to enter secret PIN of four digits. Each voter will be assigned a unique voter identification number (VIN). We use VIN for identification. We use photograph, fingerprint and PIN in authentication phase. This data is very sensitive, so we need to provide confidentiality to the database. We have used multiple secret sharing and steganography for database confidentiality. Figure-2 shows steps to be executed during registration phase.

The steps for registration are as follows:

- (1) Take photo of eligible voter and create two shares $share_P_1$ and $share_P_2$ of photo-image using visual cryptography.
- (2) Take fingerprint of eligible voter and create two shares $share_T_1$ and $share_T_2$ of fingerprint image using visual cryptography.
- (3) Rotate $share_P_1$ by 90° .
- (4) Stack this rotated share $share_P_1$ with $share_T_2$ to generate $share_1$.
- (5) Stack share $share_P_2$ with $share_T_1$ to generate $share_2$.
- (6) Direct user to enter 4-digit PIN.
- (7) Generate 32-bit time stamp (TS).
- (8) Generate encrypted secret message, $ESM = E(PU_{AS}, (\text{Hash}(\text{PIN} \parallel \text{TS}) \parallel \text{TS}))$.
- (9) Embed even bits of encrypted secret message in $share_1$ using LSB technique and store this $stego_share_VIC$ along with voter identification number (VIN) in smart card called voter identification card (VIC).

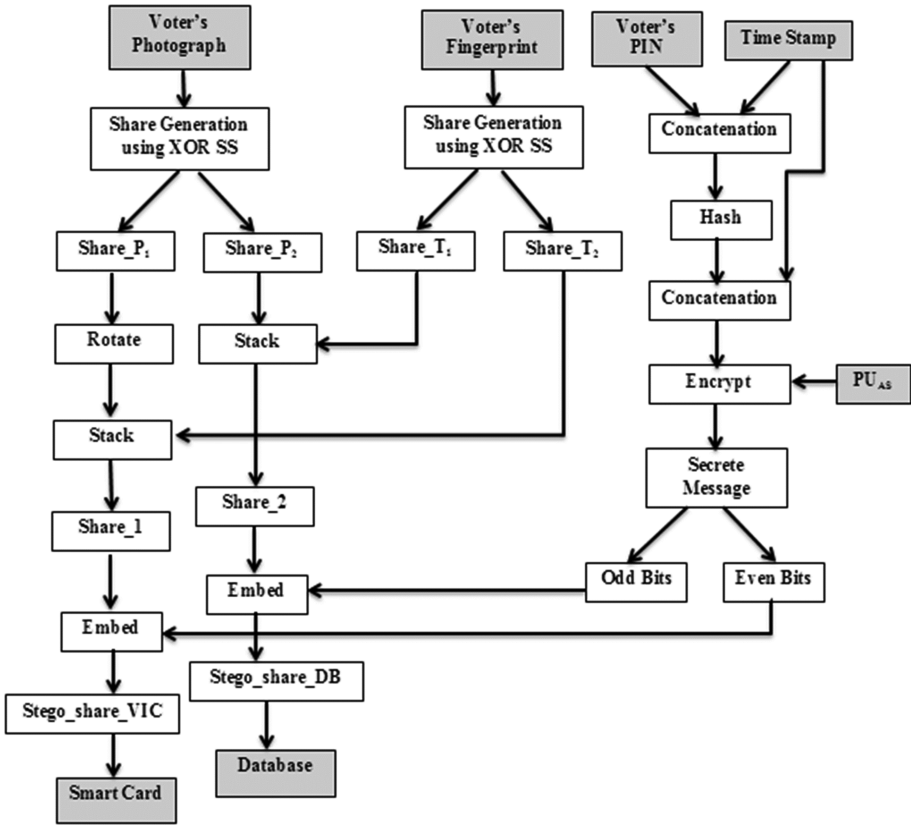


Fig. 2. Registration phase

- (10) Embed odd bits of encrypted secret message in share_2 using LSB technique and store this stego_share_DB in Voter's database.

4.2 Identification and Authentication Phase

During authentication, a voter has to carry VIC. It contains VIN and image share share_VIC. Identification phase uses VIN and authentication phase uses share_VIC. Figure 3 shows steps to be executed during authentication phase.

Steps in authentication phase are as follow.

- (1) Scan VID through smart card reader and read VIN and stego_share_VIC.
- (2) Fetch voter's data from database using VIN and get another share stego_share_DB and STATUS flag for the voter.
- (3) If STATUS flag is FALSE then reject the voter as vote has been casted already. If STATUS flag is TRUE then go to next step.
- (4) Decode stego_share_VID to get even bits of secret message using LSB technique. These bits are stored in the even position of buffer and get share_1.

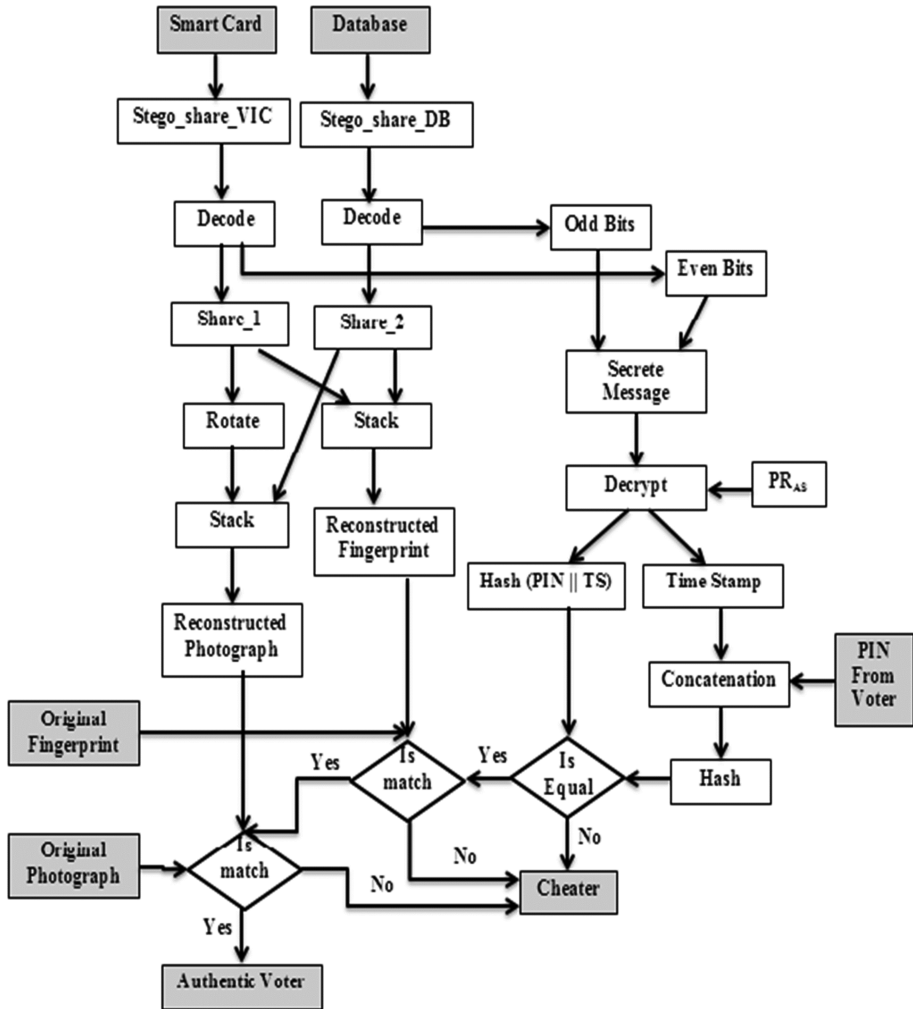


Fig. 3. Authentication phase

- (5) Decode stego_share_DB to get odd bits of secret message using LSB technique. These bits are stored in the odd position of buffer and get share_2.
- (6) Buffer is decrypted using private key of authentication server. This gives secret message $SM = (\text{Hash}(\text{PIN} \parallel \text{TS}) \parallel \text{TS})$.
- (7) Least significant 32 bits of secret message is taken as TS. Most significant 512-bits of secret message are taken as received hash.
- (8) User is directed to enter 4-digit PIN.
- (9) Concatenate entered PIN with received TS and generate its hash using SHA-512.
- (10) Compare received hash from step 7 with generated hash from step 8. If it matches then go to step 10, otherwise reject him/her as unauthentic voter.
- (11) Rotate share_1 by 90° and stack it with share_2 to generate photo_image.

- (12) Compare photo_image with voter's face and if matched then go to step 12. Otherwise reject him/her as unauthentic voter.
- (13) Stack share_1 and share_2 to generate fingerprint image.
- (14) Take live fingerprint of voter. If it matches with fingerprint image then allow him to vote. Otherwise give three chances to user to enter fingerprint again. After three chances if fingerprint is not matched then reject him/her as unauthentic voter.

5 Security Discussion

System requires share of finger print image and photo image from VIC. When these shares overlap with shares from database, finger print and photo image is reconstructed. Creation of fake VIC is not possible. This is because a system stores even bits of encrypted hash of PIN in VIC. PIN is encrypted using private key of Authentication Server. Remaining odd bits of encrypted hash of PIN is stored in database of Authentication Server. In order to create fake VIC, one needs private key of Authentication Server and access to Authentication Server to change share.

In authentication phase, live finger print and photo image of face is captured. It is compared with reconstructed images of finger print and photo. If both matches then only voter is allowed for voting. So, even if voter gives his voting card to another person, that person can't cast the vote as everyone has unique and non-transferable biometric features.

If VIC is lost, no information is leaked as original images of fingerprint and photo are not stored in the VIC. Also VIC contains even bits of encrypted PIN. Even if fake voter creates fake voting card for legitimate voter with his biometric feature, he will not pass through our authentication process. This is because one original share is stored in database of government.

System maintains STATUS flag for each voter. Initially, STATUS flag is set to TRUE for all voters. In authentication phase, first system checks the STATUS flag. If it is TRUE then system performs authentication for the voter. If voter is authentic then it will allow him to cast the vote. After authentication and successful vote casting, STATUS flag is set to FALSE. If STATUS flag is FALSE then system rejects the voter as flag indicates vote has been casted already. Therefore, system guarantees one vote per voter.

6 Conclusion

Biometric features are non-transferable. Therefore, use of biometric features during authentication gives more security compare to password. Proposed system provides secure authentication through biometric features and PIN so that only authentic voter cast the vote. Confidentiality of the voter registration database and voter identity card is maintained by using multiple secret sharing and steganography. As multiple secret sharing is used to store two images, share size is same as image size. So, there is no increase in space to store shares. Integrity of voter registration database and voter identity card is checked through hash code.

References

1. Al-Anie, H.K., Alia, M.A., Hnaif, A.A.: e-Voting protocol based on public kry cryptography. *Int. J. Netw. Secur. Appl. (IJNSA)* **3**(4), 87–98 (2011)
2. Abdulzahra, H., Ahmad, R., Noor, N.M.: Combining cryptography and steganography for data hiding in images. In: *Applied Computational Science*, pp. 128-135 (2014). ISBN: 978-960-474- 368-1
3. Abd-alrazzq, H.K., Ibrahim, M.S., Dawood, O.A.: Secure internet voting system based on public key kerberos. *IJCSI Int. J. Comput. Sci. Issues* **9**(2), 428–435 (2012). No 3
4. Johnson, N.F., Jajodia, S.: Exploring steganography: seeing the unseen. *IEEE Comput. Mag.*, 26–34 (1998)
5. Naor, M., Shamir, A.: Visual cryptography. In: De Santis, A. (ed.) *EUROCRYPT 1994*. LNCS, vol. 950, pp. 1–12. Springer, Heidelberg (1995)
6. Khasawneh, M., Malkawi, M., Al-Jarrah, O., Hayajneh, T.S., Ebaid, M.S.: A biometric-secure e-Voting system for election processes. In: *Proceeding of the 5th International Symposium on Mechatronics and its Applications (ISMA08)*, Amman, Jordan, 27–29 May 2008
7. Paul, N., Evans, D., Rubin, A., Wallach, D.: Authentication for remote voting. workshop on human-computer interaction and security systems, Ft. Lauderdale, FL, 6 April 2003
8. Dastanian, R., Shahhoseini, H.S.: Multi secret sharing scheme for encrypting two secret images into two shares. In: *International Conference on Information and Electronics Engineering*, vol. 6. IACSIT Press, Singapore (2011)
9. Saini, S., Dhar, J.: An eavesdropping proof secure online voting model. In: *2008 International Conference on Computer Science and Software Engineering*. IEEE (2008)
10. Vermani, S., Sardana, N.: Innovative way of internet voting: secure on-line vote (SOLV). *IJCSI Int. J. Comput. Sci. Issues* **9**(6), 73–78 (2012). No 3
11. Katiyar, S., Meka, K.R., Barbhuiya, F.A., Nandi, S.: Online voting system powered by biometric security using steganography. In: *2011 Second International Conference on Emerging Applications of Information Technology*. IEEE (2011)
12. Sridharan, S.: Implementation of authenticated and secure online voting system. In: *4th ICCCNT 2013*. IEEE, 4–6 July 2013
13. Wayner, P.: *Disappearing Cryptography*. AP Professional Books, Boston (1996)

Applications Security

Touch and Track: An Anti-theft and Data Protection Technique for Smartphones

Sohini Roy¹✉, Arvind Kumar Shah², and Uma Bhattacharya¹

¹ Indian Institute of Engineering Science and Technology Shibpur, Howrah, India
roysohini266@gmail.com, ub.cs@iiests.ac.in

² Microsoft Global Delivery Services India Pvt. Ltd., Hyderabad, India
arsha@microsoft.com

Abstract. At present, the use of smartphones has reached a boom. With the increase in the use of smartphones, the need for protecting such devices from theft has also come up. A number of anti-theft mechanisms are designed so far to protect the smartphones from being stolen away. However, only anti-theft mechanisms are not enough. A good data protection technique should be coupled with a strong anti-theft mechanism such that, even if the device is stolen, the confidential data does not go into the wrong hands. In this paper, a new anti-theft mechanism combined with a robust data protection technique is designed for smartphones. In this work, context-awareness is used as a decision-making technique to automatically determine a theft of the device and as a consequence data protection technique, tracking mechanism and recovery system are initiated. So that, not only the device is recovered in a re-usable form but also the thief is caught easily. The work proposed in this paper has been compared with Where's My Droid app and AndroidLost Free. The outcome of the comparison shows the novelty of our approach in terms of automatic theft detection, device tracking (including indoor location detection) and data protection.

Keywords: Smartphones · Anti-theft application · Data protection · Data backup · Device tracking · Theft-detection · Wi-Fi interface

1 Introduction

Smartphones have become an integral part of the daily life of people nowadays. Statistics of the usage of smartphones in India shows a sudden rise from 2010 to 2015. Only 3 % of the Indians used smartphones during the year 2010. Current records [1] show that the usage of smartphones in India has risen from 3 % to 25 %. It is also predicted that by 2017 this percentage will increase to 32 %. So, the present time can be referred as an era of smartphones.

Due to the light-weight, cheaper rate and easy portable nature of smartphones, it has even surpassed the popularity of other portable devices like laptops, among the common mass. Accessing Internet, playing advanced games, watching videos, mobile banking, online shopping, social networking etc. does not need a laptop to support them. Smartphones are there to act as a substitute. Together with a large number of such features,

the smartphones are equipped with large internal memory space and have provision for memory cards of huge storage. Thus, currently smartphones are used not only for accessing the Internet but also for storing documents. These documents can also be sensitive documents like passwords, private photographs, videos and SMSs. However, with the increase in use of smartphones, loss and stealing of them has also increased. Thus, securing such important and confidential data from going into the hands of unauthenticated users is a must in smartphones.

As the theft and loss of smartphones have become very common, a new door for research on smartphones has opened. This research area focuses on designing anti-theft mechanisms for smartphones and also aims in providing measures for data protection, even if the phone is lost or stolen. In this paper a context-based theft detection technique is proposed. The work named as ‘Touch and Track’ uses biometric authentication system and also considers other contexts to determine a theft. As a theft is determined by Touch and Track, a tracking application together with a data protection method is initiated. Thus, Touch and Track offers a fair chance of recovery of the lost smartphone with the help of the tracking application.

The rest of the paper is structured as follows: Sect. 2 reviews some related works and mentions the scope of this work. Section 3 mentions about the scope of the work and the assumptions are stated in Sect. 4. Section 5 describes the proposed scheme. Section 6 provides a comparison of the proposed scheme with two popular existing anti-theft applications, namely— **Where’s My Droid app** [7] and **AndroidLost Free** [8]. The paper is concluded in Sect. 7 and the future works are stated in Sect. 8.

2 Related Works

Different anti-theft mechanisms are nowadays inbuilt into the smartphones of different companies. Theft and loss of smartphones have become so common that anti-theft mechanisms are now an inevitable feature of smartphones. Yet researches are still going on to improve such features. **Anti-theft Application for Android Based Devices** [2] aims in improving the SMS based anti-theft mechanisms by replacing it by MMS. It points out the drawback of GPS based tracking systems which lacks the ability to identify the thief appropriately. The scheme proposed by this paper requires the use of front and back camera of the smartphones and also support for multimedia messages. The application after installation works in the background and stores the SIM number. Whenever it recognizes a SIM change, it starts the front and back camera of the smartphone to record a video clip of the surroundings without taking any permission from the user. The application then sends an MMS and a number of snapshots to an alternate mobile number and an email id that was provided during installation. **Biometric Anti-theft and Tracking System for mobiles – BATS** [3] integrates mobile phones with a fingerprint reader as a method of anti-theft. The paper assumes that a smartphone is provided with five fingerprint readers. The application requires the user to register his/her fingerprint to work in the admin mode in which the user is allowed to access the phone completely. The admin can permit some other users to have access to the smartphone as in the guest mode or protected mode in which the users can have access to some of the services of

the phone only. A fingerprint is scanned and is processed by means of image processing technique to find a match with the registered fingerprints. If the scanned fingerprint is an unknown one then the phone profile automatically changes to unauthenticated mode and all the services of the phone are blocked. As an unauthenticated access is observed by the application, a voice call is also made to a register mobile number at regular intervals. The alert message includes the IMEI number, information of time and place the mobile has been used, so that the phone can be recovered. The same alert is also sent to the saved contact numbers in the phone. The application uses A-GPS to accurately determine the position of the current user or the thief.

Device tracking is a location aware method of locating the lost device and thereby recovering it. Various in-built features of the smartphones are used in tracking them. **SAPt: A Stolen Android Phone Tracking Application** [4] uses SMS to enable the user to track the stolen or lost android smartphone. GPS and GSM networks are used to perform the tracking operation. In SAPt the stolen device acts as the server and another phone acts as the client whose number is hardcoded in the stolen device initially. The application is provided with feature like SIM card detection which detects whether an authorized SIM card is inserted into the phone and sends a notification to the registered number mentioning the number of the new SIM. Camera of the stolen phone can be turned on and off via SMS from the client and the smartphone can capture pictures and mail them to the user's registered mail ID. It is assumed that the new SIM is also GPRS enabled. The application can also track the thief's incoming and outgoing calls and SMS their number to the client phone. Though, the application is highly useful, the features supported by it are time consuming and are constrained with a large number of assumptions like lac of options like resending of SMS when it is not delivered, need for GPRS enabled SIM etc. It also cannot guarantee the recovery of the smartphone since calling the thief's acquaintances or knowing only the IMEI and IMSI number of the new SIM via SMS may not help. **Location Aware Device Discovery for Physically Constrained Environments** [5] has two major components namely—the server, which is provided with the building floor plan and is capable of storing and mapping device capability information; and the devices that listen to the server to get server's connectivity information. This information helps them to get registered to the server. It is also mentioned that the devices move between two modes namely—registration and discovery. In the registration mode each device forwards location and its capabilities to the location server such that each device becomes a visible part of the system. In the discovery phase each device acts as a client and forwards queries to the location server to get the list of devices near to it and their location information. A mapping process is carried on by the location server on the basis of the client's query. It is regardless of the underlying location model and the location sensing technology. **Track ME!** [6] proposes a web based Mobility Analysis System which can collect location data from cell phone users via opportunistic Internet connections that is either by 802.11 capacity of smartphones or 3G internet connection. The system then converts these low level location data to high level mobility information and also it adds a temporal dimension. TRACK ME! is provided with a Query Engine that provides an interface for other applications for executing and querying different filters over mobility profiles of cell phone users. Another important framework associated with TRACK ME! is the location prediction which interacts with

the application services and uses the current mobility window to predict the next location of the cell phone user.

Google Play Store now consists of a large number of anti-theft and data protection applications. Each of the applications has some unique pros and cons. Some of the applications help the users to get the lost phone back and keep the data safe. Some other applications focus in tracking the missing phone and thereby track the thief. **Where's My Droid app** [7] helps in tracking a lost or stolen phone from anywhere, and all the features can either be initiated with a text messaged attention word or through the online control center. The ring feature supported by this app makes the phone ring at the maximum volume even if the phone is kept at silent mode after stealing it. GPS flare is another feature of this app that gives the last known location of the phone till the power is switched off. The camera feature of this app helps in taking pictures and thereafter the picture is saved in the SD card of the phone and is finally uploaded to the server wherefrom only the actual user of the phone can view it. Another feature of this app is the passcode feature that locks the main menu of this app. The lock feature of this app enables the user to remotely lock and unlock the device. The wipe feature helps in remotely deleting all the data stored in the phone so that they remain inaccessible to the unauthorized user. The uninstall defense feature of this app engages the device screen lock if the app is tried to be uninstalled. The SIM feature is another important feature of this application that notifies the user about any change of SIM or phone number. In spite of having a large number of features the main drawback of this app is that it is of no use if the phone is switched off right after it is stolen. Moreover, initiating each and every feature of the app by sending attention words via texting or online commander is a time consuming procedure. By the time the user identifies the phone is stolen and tries to recover it, the thief may move to a distant place. Even if data is wiped or phone is locked that chances of recovery is very poor for this application.

In **AndroidLost Free** [8] the user of the lost phone can remotely activate several features in the phone having this app installed, either by SMS or by the web. This app lets the phone to read SMSs and send them to the email address of the original user. The user of the phone can activate the alarm to a ring along with a flashing, enable and disable the GPS, data and Wi-Fi connection, remotely wipe the SD card, and get the latest call list on the phone. AndroidLost helps the user of the phone to locate the lost phone on a map and even if the phone is at indoor location, a nearby location can be obtained. SMSs can be sent from the lost phone with the help of this app's online portal. However, the replies will still go to the lost phone. The user can also remotely lock and unlock the phone from the web. AndroidLost will not use any battery since it does not poll any server to find out if it is lost; rather it uses the latest technology from Google to send messages to the phone. This feature will save battery but it does not take into consideration the time for determining whether the phone is lost. The user of the phone can check the remaining battery life, IMEI and SIM details after the phone is lost. The app sends an email to the original user's email address when the SIM card of the lost phone is changed. Hidden SMSs can be sent to the lost phone to activate its features and these messages will not display any icon or play any sound. The app allows the phone to take pictures using front and back camera and send them to the owner. If the phone is lost in

an indoor area, text-to-speech facility of AndroidLost can be used and the phone can be made to say loudly “I am Lost, please pick me up”.

3 Scope of the Work

The major objectives of the research work have been listed below.

- Intelligently determine that the mobile is actually stolen.
- Locate the mobile and track it: The mobile location can be tracked using the proposed approach.
- Erase the critical data that has been stored in the mobile or protect the sensitive data in the phone from unauthorized access.
- Recover the lost device.

4 Assumptions

- Initially, each mobile phone will have to register its ICC ID (Integrated Circuit Card ID) and IMEI number to the centralized cloud.
- Corresponding to each ICC ID and IMEI number, the device profile is stored in the cloud. The device profile includes the user information like name, address, e-mail id and fingerprint of the user.
- The smartphones are equipped with a fingerprint scanner and also it stores the fingerprint of the user.
- Some local clouds or clone clouds are placed at of the public places like shopping malls, traffic signals etc.

5 Proposed Work

The total work is divided into three major sections namely—context aware theft detection technique, location aware device tracking and recovery application.

5.1 Context Aware Theft Detection Technique

This module deals with theft detection as well as data protection.

5.1.1 Theft Detection

Setting alphanumeric passwords or patterns have become an old concept. Such kind of anti-theft mechanisms are also not free from a large number of security breaches. Once a pattern input or password input of the smartphone, provided by the user is followed by a third person, he or she can easily gain access to the device. All the android smartphones are touch sensitive. Thus, in this application, touch of the user is used as context information to determine the theft of the device.

In this application, whenever the user or anyone else touches the phone-screen for unlocking it, the fingerprint of the person touching the phone is captured as a context data using fingerprint scanner. It is then matched by the smartphone with the fingerprint, previously stored in it. If match is found, the user is identified and then only the phone is unlocked. Otherwise, if match is not found, the phone remains locked and a strong alarm is generated to notify the user about the theft. The tone of the alarm is a special one reserved only for theft detection. This alarm tone cannot be set as a ringtone or for any other purpose. However, this can only alert the user if he is nearby. So, only the alarm generation is not a sufficient anti-theft mechanism and thus a suspected theft notification is sent to the cloud (or local cloud).

Just as the cloud gets a probable theft notification, the application sends an email to the stored email address of the user, asking the user to confirm about the theft. It may happen that the user has left his smartphone at home or at any other place wherefrom he is sure to get back his smartphone later. In such cases, any acquaintance of the user may touch the phone. Although the application doesn't allow unlocking of the phone if the user is not identified, it tries to avoid false alarms indicating theft and also subsequent device tracking and recovery methods. Now, if the user confirms the theft by sending an acknowledgement to the cloud, the cloud informs the smartphone about the theft and all stored data of the smartphone starts getting deleted. The anti-theft application then starts the tracking application. However, if network access is not available from the smartphone, neither the phone will be able to send the suspected theft notification to the cloud, nor the cloud will be able to send any acknowledgement to the smartphone. If the phone is left at home or any safe place, then the current user will be asked to provide an authentication password which can be provided by asking the user of the phone and so, no tracking system will be started. If the password is not provided or wrongly provided and either the SIM is removed or the phone is switched off then the smartphone itself comes to the conclusion that the phone is stolen and therefore initiates the subsequent mechanisms to track the device and protect the stored data.

5.1.2 Data Protection

It is very easy to identify a theft if the device is stolen by a naïve thief. He will have no idea about the security systems of the phone and thus the device can easily come to the conclusion that it is stolen. On the other hand, if the device is stolen by a professional person having good knowledge of the security systems, he will not take the risk of touching the phone to unlock it. Still he can access the stored data in the external storage device and even the internal memory of the phone by connecting it to another PC/laptop. As the smartphone is connected to some other device by the thief, the smartphone asks for unlocking it first. In order to unlock, the thief must touch the phone and thus the anti-theft mechanism detects the theft. Even if the phone is still not touched and file transfer is tried by the thief to carry on by some means, our theft detection application will be able to detect the theft and encrypts all its stored data. These encrypted data is then compressed into a zip file and forwarded to the owner of the phone as an email attachment.

The flow chart of theft detection technique of Touch and Track is given in the following Fig. 1:

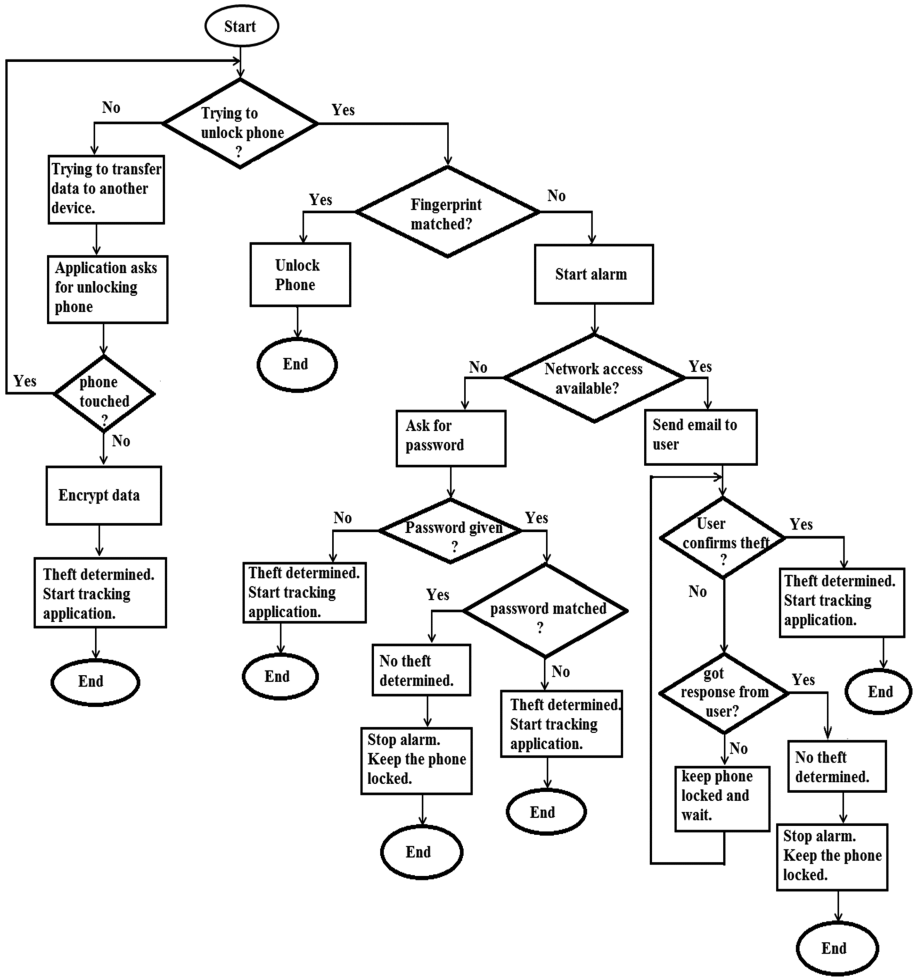


Fig. 1. Flowchart of theft-detection technique of smartphone

5.2 Location Aware Device Tracking

As soon as the theft is determined by the context-aware anti-theft application, the location-aware tracking application is initiated. The device tracking application is divided into two portions as—sender’s application and reader’s application. These two portions of the device tracking application run independently in every smartphone having this application installed. The reader’s application keeps on running as a background application whenever a smartphone is switched on. The sender’s application is also a background application and is initiated only when a theft is determined. When a theft is determined, automatically the reader’s application is closed to save the energy of that particular smartphone. The working of both the applications is described in details.

5.2.1 Sender's Application

When the context-aware anti-theft application determines a theft, it sends a request to the sender's application to initiate itself. The sender's application first checks whether WIFI is on. If not, the application starts the WIFI. The WIFI then detects whether there is any Wi-Fi enabled local cloud within its range. If a local cloud is found then, the stolen device forwards a HELP_ME message including its IMEI number to it via Wi-Fi interface and waits for an acknowledgement. Otherwise, if no local cloud is detected within the range of the device, then it broadcasts the HELP_ME message to the nearby android devices. It is the responsibility of the reader's application to send back an acknowledgement to the sender device. Once an acknowledgement is received from any receiver's end, the sender's application is stopped for that time to minimize energy drainage and the recovery application is started. When the current location of the stolen device is changed, the sender's application is re-initiated.

5.2.2 Reader's Application

The reader's application runs on every android device having this application installed and also in the local clouds. After receiving the HELP_ME message, the receiver's application sends back an acknowledgement to the device from which the HELP_ME message came. If a local cloud receives the HELP_ME message, it forwards a LOCATED message to the central cloud, which includes the IMEI number of the stolen phone, a time stamp of when it received the HELP_ME message and its own location. The local cloud keeps on tracking the stolen device by measuring the distance of the phone from itself until the device goes beyond its range. When the local cloud finds that the device is far away from itself and it has very limited access to the device, the device is informed that its location is changed. Thus, the sender's application can be restarted. A LOCATED message is again forwarded by the local cloud to the central one to make the central cloud know that the device is no longer within the local cloud's range.

On the other hand, if the HELP_ME message is broadcasted to a number of mobile devices, each of the devices receiving the message will try to send back an acknowledgement to the sending device. This will drain a lot of energy of the stolen device. Thus, to conserve the battery power of the stolen device, once an acknowledgement is received the sender's application is closed for the time being. The device sending back the first acknowledgement to the stolen phone, is chosen as the MONITOR by sending BE_MONITOR message by the lost phone. This BE_MONITOR message consists of the IMEI number of the lost phone, its receiving signal strength value from the MONITOR. The MONITOR will now initiate the formation of a Mobile Ad Hoc Network (MANET) [9], by forwarding a JOIN_REQ to all the devices within its Wi-Fi range other than the stolen device. The devices receiving the JOIN_REQ send back an acknowledgement to the monitor and join it to form a network. The monitor stores the IDs (IMEI number) of all those devices in its neighbor list. Each of the devices now starts calculating the distance of the stolen device from them with the help of the receiving signal strength from the stolen device and periodically sends a LOCATED message to the central cloud through Wi-Fi interface. When any of the devices leaves the network i.e. it goes away from the stolen device, it sends a notification to the monitor. In that case, the monitor removes the ID of that device from its list of

neighbors. If the monitor goes away, then the role of the monitor is handed over to any of the devices which sent back acknowledgement to the JOIN_REQ of the monitor and is currently within the network. In order to hand over the duty of the monitor, the old monitor forwards the neighbor list to the new monitor. The new monitor in turn declares itself as the present monitor by broadcasting its own ID to all the devices in the network. If the monitor is about to leave the network and it finds no device within its range to hand over the task of the monitor, then, the stolen device is notified that its current position is changed. Then, the sender’s application is started again.

5.3 Recovery Application

The cloud stores the fingerprint of all the android users. Now, if the smartphone is stolen and the current user tries to unlock the phone, his fingerprint will be captured by the stolen smartphone’s fingerprint scanner and will be included in a message named as FOUND_THIEF message and will be sent to either the local cloud or the MONITOR. Now, this FOUND_THIEF message will be forwarded to the central cloud. The central cloud already has the fingerprints of all the android users who once have registered. Now, if the thief is an android user then the cloud can identify who has stolen the device. This fingerprint can also be used by the police to identify and catch the thief.

Now, if the stolen phone has enough residual battery power, it opens its front and/or back camera just as the theft is determined. The smartphone then takes a video clip of the surroundings and stores it as a compressed file. This file is forwarded to either the local cloud or the device whose acknowledgement was received, only when good access to the device or the local cloud is obtained.

6 Comparison of Touch and Track with Where’s My Droid App and AndroidLost Free

Name of the Application	Anti-Theft Features		Device Tracking Features		Data Protection Features				Other Special Features Or Remarks
	Ring Feature	Automatic Theft Detection	Current Location Detection and Notification	Indoor Location Detection	Remote Locking	Data Wiping	Data / Features accessible to the User	Data Backup	
Where’s My Droid app	Yes (enabled by text)	No	Yes (till switched off)	No	Yes	Yes	User gets to know about SIM change	No	Chances of recovery is poor
AndroidLost Free	Yes (Ring and text to speech)	No	Yes(till switched off)	Yes	Yes	Yes	Sending SMS from the lost phone, latest call list	No	Not only tracks the phone but also torments the thief.
Touch and Track	Yes (enabled automatically)	Yes	Yes (till switched off and reactivated on when switched on again.)	Yes (with the help of available Wi-Fi connections)	Yes	Yes	Camera, voice recorder, SMS features can be used remotely.	Yes.	The chance of recovery is quite high and he thief can also be caught easily.

7 Conclusion

Most of the present smartphone anti-theft applications in Google Store lack the ability to automatically determine a theft. This drawback not only wastes time but also lessens the chance of recovery of the device. Thus, Touch and Track is featured to automatically determine a theft by means of the fingerprint scanner. The tracking application of Touch and Track tries its best to minimize the battery power consumption of the lost device by stopping its reader's application and periodically discontinuing its sender's application. This device based tracking system does not require the GPS and indoor location of the phone can also be tracked by this. Moreover, the application works using Wi-Fi interface. Thus, even if the SIM of the phone is removed, the application will not face any hindrance. It is shown by comparing the Touch and Track application with other existing anti-theft applications that it is more efficient in securing the device by not only protecting it from a theft but also in terms of data protection, data back up and device recovery.

8 Future Works

The proposed anti-theft mechanism is solely dependent upon the fingerprint scanner feature of a smartphone. However, a major drawback of this application is that most of the present smartphones are not equipped with a fingerprint scanner. In order to use this application and secure the smartphones, the smartphone developing companies should include a fingerprint scanner in the device. Fingerprint matching being the most authentic form of user authorization, will not allow any form of security loopholes.

The memory cards of the phones should be designed in a manner such that even if they are taken out of the phone and data is tried to be read from it by the thief, a password authentication must be provided. When a memory card is inserted into the phone for the first time, a password for the memory card is needed to be given. It should also be ensured by the memory card developers that each time, the memory card is accessed externally, this password must be provided correctly.

References

1. Chourabi, H., Nam, T., Walker, S., Gil-Garcia, J.R., Mellouli, S., Nahon, K., Pardo, T.A., Scholl, H.J.: Understanding smart cities: an integrative framework. In: Proceedings of 45th Hawaii International Conference on System Sciences, pp. 2289–2297. IEEE (2012)
2. Khan, A.U.S., Qureshi, M.N., Qadeer, M.A.: Anti-theft application for android based devices. In: 2014 IEEE International Advance Computing Conference (IACC), pp. 365–369. IEEE (2014)
3. Kumar, C.L., Arunachalam, P., Sandhya, S.: Biometric anti-theft and tracking system for mobiles–BATS. *Int. J. Recent Trends Eng.* **1**(1), 237–242 (2009)
4. Sonia, C.V., Aswatha, A.R.: SAPt: a stolen android phone tracking application. *ITSI Trans. Electr. Electron. Eng.* **1**(6), 111–115 (2013)

5. Malkani, Y.A., Dhomeja, L.D.: Location aware device discovery for physically constrained environments. In: Proceedings of 2nd International Conference on Computer, Control and Communication, pp. 1–5. IEEE (2009)
6. Bayir, M.A., Demirbas, M., Cosar, A.: Track ME! a web based location tracking and analysis system for smart phone users. In: Proceedings of 24th International Symposium on Computer and Information Sciences (ISCIS 2009), pp. 117–122. IEEE, September 2009
7. [online]. <http://wheresmydroid.com/features.html>
8. [online]. <https://play.google.com/store/apps/details?id=com.androidlost&hl=en>
9. Roy, S., Sinha, D.: CBSRP: cluster based secure routing protocol. In: Proceedings of First International Conference on Networks and Soft Computing (ICNSC 2014), pp. 82–86. IEEE (2014)

Enhancement of Detecting Wicked Website Through Intelligent Methods

Tarik A. Rashid¹(✉) and Salwa O. Mohamad²(✉)

¹ Software and Informatics Engineering, College of Engineering,
Salahaddin University-Erbil, Hawler, Kurdistan, Iraq
tarik.rashid@su.edu.krd

² School of Computer Science, College of Science,
University of Sulaimania, Sulaimania, Kurdistan, Iraq
salwaomermohammed@yahoo.com

Abstract. Noticeably, different environments of wicked website include different types of information which could be a threat for all web users such as incitement for hacking sites and encouraging them for spreading notions through learning theft networks, Wi-Fi, websites, internet forums, Facebook, email accounts, etc. The proposed work deals with sites to protect from hacking through designing a method that takes full advantage of machine learning and intelligent systems' capabilities to realize the informative contents. The ultimate goal of this work of research is to understand the system behavior and determine the best solution to secure the vulnerable users, state and society via Random Forest (RF) and Support Vector Machines (SVM) methods instead of traditional methods. Random Forest exhibited Promising Results in terms of accuracy.

Keywords: Arabic websites · Multi-class · Random forest

1 Introduction

Apparently internet is the biggest database in the world, often it provides detailed data in different areas which can help people achieve their goals in gaining information and knowledge. Unusually, websites are regarded as an imperative component of internet. Recently, the distribution of wicked websites through the internet has increased significantly, as a result, substantial numbers of web documents from different wicked websites have become available for users. In view of that, users might spontaneously collect both valuable and wicked information typically through those wicked sites via learning penetration, stealing personal information, etc. Thus, refining the information retrieval process has become essential. The process of classifying text or document is defined as apportioning of various documents to a different class. The process mainly depends on the contents of the documents or texts. Besides, this process of classifying text document is very useful and used to tackle numerous problems and applications like web document mining, filtering email messages, grouping and indexing articles, computational linguistics and human computer interaction [1–4]. Although most new web pages contain keywords that help recover associated documents rapidly and precisely, nonetheless, there are still various and previous document forms without appropriate

keywords. Thus, automatic detecting wicked websites through classification is necessary to be used for classifying these various forms of document depending on their contents.

Generally speaking, there are some prevailing and primary languages that are widely used by users for communication purposes on the internet, these languages are: English, Chinese, Spanish, and Portuguese etc. In view of that, numerous of advanced models are developed to tackle document or text classification task in the mentioned languages [5, 6]. Moreover, there are very some degree of works developed for automatically classifying Arabic documents or texts. Simply because classifying Arabic texts is utterly different from English texts. Since Arabic is exceedingly inflectional and vastly derivational language in which monophonical exploration and its analysis is a challenging and interesting task. This research work is conducted to design an automatic and intelligent Arabic webpage classification system and to achieve this, data is collected from several Arabic websites. The main contributions of the paper are: (1) Recommending a system via applying the latest data mining techniques for security purposes to deal with Arabic wicked websites to protect users from harmful information. (2) Better handling multimedia information when massive web pages are need to be processed. (3) And finally, generating interesting patterns and producing significant improvement in terms of accuracy. This research work is structured as follows: The existing literature works regarding web content classification approaches are described in Sect. 2, next, the proposed method is explained in detail in Sect. 3, and finally, the main conclusion points are outlined.

2 Literature Review

From the literature review, there have been quite a lot of research works that address the issues related to extracting texts and selecting significant information via handling text documents. In this section, a review of past experiments using various feature extraction methods to improve text classification accuracy is explained. Remarkably, Data mining is regarded as one of the most artificial intelligence technologies that has been developed for exploring and analyzing significant patterns and rules in large numbers of data. Additionally, Web Content Mining is considered as an important part of data mining and is generally used for detecting significant information from script, audio, video, and pictures in website pages or documents. It is essentially based on research fields in retrieving information, information extraction and information visualization [7]. It is also called web text mining, since the text content is practically and broadly used [4, 8]. In [9, 10], data mining application approaches for extracting valuable intuitions from the web content, construction and procedure are explained. The scheme of stemming is among various techniques that can be used to improve the capabilities of detecting wicked website pages. Basically, it helps to extract the root of a word which in return increases classification accuracy rate [7]. In [11], feature vector was presented in which the web page textual contents are characterized via a weighted vector. They used four classifiers in their experiments, these are SVM, NBC, RF and KNN.

In [12], a text mining technique which extracted Arabic text from web documents was considered using Classifier K-Nearest Neighbor and Naïve Bayes classifier. In this research work, a special corpus consisted of 1562 documents that belonged to 6

categories was established. They extracted feature set of keywords and terms weighting to increase the performance of their technique. In [13], Ramos observed the outcomes of using TF-IDF to determine all those words that have extra advantageous within the documents for using in a query. He stated that words that have high TF-IDF numbers suggest a robust correlation with a document within which they are contained, this suggests that the document can be of interest to web customers, if the query has that word. They delivered evidently a simple algorithm which effectively classifies relevant words that can improve query retrieval. Both Mathioudakis and Koudas established Queue Burst procedure which can handle real time data. This can discover burst of particular keyword over Twitter [14]. The idea of identifying a relevant or a trending word using a module has been inspired from their work. TF-IDF is regarded as one the most widely used approaches to determine important words from the corpuses [12]. In this paper, Khoja stemmer is used for preprocessing and TF-IDF is used to get extracted features, and finally, the classification task is conducted via RF and SVM to detect Arabic wicked websites. In [15], three methods were used for classifying Arabic text; namely, SVM via SMO, NBC, and J48. They intended to determine the accuracy for all classifiers. They relied on stop word removing to find the best classifier. SMO produced the best accuracy and timing. In [16], a method of linguistic root extraction is presented. They removed prefixes and suffixes relying on the word's length. They also used morphological pattern right after reasoning to eliminate infixes. Their root extraction method has been enhanced.

3 Proposed Method

The main steps for performing document classification are as follows: (1) the data must be collected from deferent web documents. (2) Then, nominated features are selected for signifying the class labels and classification algorithms that need to be trained and tested using the collected data. The processes of suggested approach of web document classification for identifying wicked websites can be illustrated in Fig. 1.

3.1 Data Collection

As a first step, a large quantity of various documents was collected from Arabic websites. This is considered as one of the most difficult issues that came across this work for splitting wicked and non-wicked texts. The corpus of 1283 documents from 160 different websites prepared. Only 142 of these websites provided detail descriptions and analysis about data. The collection of the data was based on documents of different lengths. These data collected from website documents, then converted to sentences that contained both wicked and not wicked terms (yes and no labels). These sentences are then preprocessed and filtered and then converted into vectors of words with their frequencies. These vectors of words eventually will help users get their accounts protected from hacking. Actually web document or text classification depends on its contents. An enormous amount of keywords can be found in wicked websites. Table 1, shows the collection of data for two classes of document.

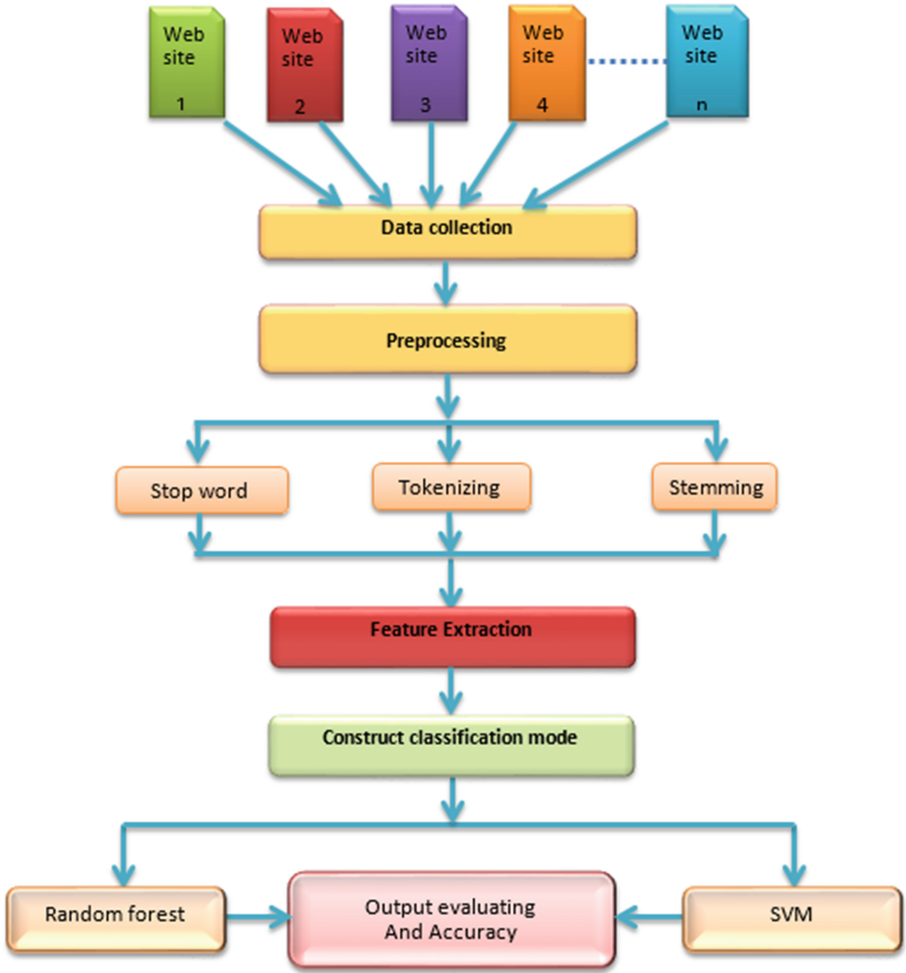


Fig. 1. Shows the process of suggested approach for detecting wicked websites.

Table 1. Number of documents for each category.

Class	# of document
Wicked website	641
Non-wicked website	642
Total	1283

3.2 Preprocessing

The preprocessing step can play a substantial role in text mining techniques. Both training and testing documents must move through this step. It eliminates all frequent

terms and irrelevant data that might diminish the document classification accuracy. It can remove punctuation marks, stop words and special marks which are characterized as non-letters such as @, #, \$, %, &, ‘,’ , etc. Several techniques have been developed in this regard such as stemming, root extraction, and the text encoding which is used to avert potential errors of characters during text reading process. In this work of research the Unicode (UTF-8) is used to encode each document [6]. Clearly, eliminating the stop words shrinks the dimensionality of term space as they do not contribute any meaning to the documents. Accordingly, they do not negatively affect the classification accuracy [6]. The most common words in text documents are articles, prepositions, pronouns, etc. Table 2, shows examples of the stop words.

Table 2. Examples of stop words in Arabic language.

(Arabic) Stop word	Reading in English	Meaning in English
من	Min	From
عن	Ean	About
في	Fee	At
الى	Illa	To me
لكن	Lakin	But
حتى	Hatta	Even
ليس	Laysa	Not
لاسيما	Lasyama	Especially
بما	Bima	Including
ماذا	Madha	What
هناك	Hunak	There
اللتان	Alltan	Who
على	Ealaa	On
نحن	Nahn	We

Basically, the text documents are scanned to detect, normalize or remove keywords, stop words, and other individual words which are occurring in the text are treated as tokens. For example, removing or normalizing the punctuation marks in Arabic alphabet like *Hemzze* ‘□’ or ‘□’ or ‘□’ in every shape to just a character A ‘*elef*’ written as ‘ا’, also ‘ى’ can be substituted via ‘ي’, and ‘ة’ is changed to ‘ه’. Stemming is another approach which is used to identify the root of a word and eventually reduce different grammatical wording ways of the word like its verbs, nouns, adverbs, adjectives, etc. Good stemmers include eliminating both suffixes and prefixes from the word. Ultimately, it reduces the word vectors’ dimensionality which helps enhance the performance of the system. In this work, Khoja stemmer is used to find the requested keywords [7]. In Table 3, some examples of root words are shown.

Table 3. Illustrates derivation forms of the word “write” in Arabic language.

Root in Arabic	Reading in English	Meaning in English
يكتب	Yakut	Write
اكتب	Aktub	I write
كتاب	Kitab	Book
مكتوب	Maktub	Written
كاتب	Katib	Writer
كتابة	Kitaba	Writing

3.3 Features Extraction

Generally speaking, document classification performance relies enormously on the document representation, feature extraction and classification algorithms. Accordingly, correct features are extracted from the words of the text document. Even after stop word elimination and performing stemming, the document is still very massive. All these words are not useful for the classification process. Therefore, in this work, significant words are extracted via using the following features extraction techniques:-

3.3.1 Weighting Assignment

In web documents, the Term Frequency Inverse Document Frequency (TFIDF) provides information about a given word within the fairly amounts of document, it is the most common used model [8]. The inverse document frequency can be defined as the amount of information that can be provided by the word or suggests if the word can be enormously or infrequently found through documents. It can be expressed by using Eq. (1) [12].

$$W_{t,d} = F_{ij} \times \log \frac{\# \text{ of documents}}{\# \text{ of documents containing word}_i} \quad (1)$$

Where F_{ij} , is the word i frequency in the j document.

3.3.2 Term Frequency (TF)

This suggests how important some certain t term found in d document. It is always utilized for text classification task. It can be computed using Eq. (2) [7, 17, 18]:

$$TF_{t,d} = \log (1 + f_{ij}). \quad (2)$$

3.4 Classification

There are two types of approach for tackling document classification; physically or automatically. Usually human knowledge are used for the physical approaches, on the other hand, machine learning techniques are used automatic approaches [17–21]. The automatic approach is widely used as it can handle huge amount of data that need to

be classified. Techniques such as RF and SVM have effectively performed to tackle document classification task [11, 18]. Details of each classifier are in the next subsections.

3.4.1 Random Forest

RF is one of the most popular classifiers which is depended on a group of decision trees; each of which is formed through a subset of features. Each decision tree produces a vote for a class when a sample is fed for classification. The ultimate result for classification is decided by voting among the group of decision trees for the best prevalent class. Essentially, decision tree cannot solve overfitting problem, whereas, RF has the ability to do so [23–27]. The error rate in RF is decided by: (1) Correlation which is a measure of relationship and dependency concerning to some extent two random trees within the forest, obviously, errors can get increased when correlations are increased. (2) Strength which means the individual strength for every tree within the forest. Certainly, a tree that has low error is considered as a strong tree. In other words, the forest error rate drops as the decision trees strength increases. Thus, RF produces very accurate results. Contrariwise, overfitting is perceived by RF for some noisy data set [23–27]. Recursive partitioning is used in decision tree to build for training of decision tree. This means that the input data set can be divided into subsets. In addition, the condition of recursion is terminated if similar output targets are found in the tree nodes. The process of reducing decision tree size can be called decision tree pruning. Usually, decision tree size can be reduced via cutting some parts of the tree that provide low voting for classification. Therefore, the pruning approach would decrease the produced tree's complexity and size and also reduce overfitting on occasion.

3.4.2 Support Vector Machines

This is another type of machine learning technique through which good accuracy can be obtained on large data set. SVM considered to be a large margin technique to classify data that are linearly separable [23–27] Overfitting in SVM can be avoided via minimizing the generalization error. SVM is used widely to tackle text classification simply because feature space dimensions are commonly high [23–27]. In this research work, Sequential Minimal Optimization is used with Polynomial kernel. A procedure is designed via which the training data is accepted and theoretically plots each item on a 2-D plane. Literally, the input of a document is received by SVM in which it is placed and then it is plotted like a point. Finally, SVM draws lines to define in which class the input of the test sample fits. It is worth mentioning that SVM for classification of multiple classes must run multiple times and in each time divides the outstanding items to define the neighboring class.

3.4.3 Results Evaluation and Discussions

The system is trained using the training set for building a classifier and it is tested using the testing set. Thus, the data is divided into two parts (75 % training and 25 % testing). Several quantitative measures (for instance, F- measures, Precision, Recall) are used for evaluating the system [6, 19, 22, 23]. Table 4, shows confusion matrix for both classifiers.

Table 4. Confusion matrix for both SVM and RF classifiers.

Classifier	Predicted class	Class (yes)	Class (no)
SVM	Class (yes)	TP(144)	FP(1)
	Class (no)	FN(8)	TN(168)
RF	Class (yes)	TP(141)	FP(4)
	Class (no)	FN(7)	TN(169)

In the framework of web page classification, the confusion matrix is used which has four entries: TP is stand for true positives, these are the total web pages which are acceptably classified samples as belonging to a right class; FP is stand for false positives, these are the total web pages which wrongly classified sample as belonging to a right class; TN is stand for true negatives which correspond to total web pages acceptably classified as not belonging to a right class; FN is stand for false negatives which refer to total web pages wrongly classified as not belonging to the right class. In addition, three evaluation measures can be defined as follows:-

- (a) Precision: Web pages acceptably classified as belonging to the class divided by total web pages predicted for that class.

$$P = \frac{\# \text{ of relevant document retrieved}}{\# \text{ of document retrieved}} \tag{3}$$

- (b) Recall: Web pages acceptably classified as belonging to the class divided by the total number of web pages that actually belong to that category.

$$R = \frac{\# \text{ of relevant document retrieved}}{\# \text{ of relevant documents in the database}} \tag{4}$$

- (c) F-Measure: The F-measure combines both recall and precision through the same weights into a single parameter [6]. It is defined as follows:

$$F - \text{Measure} = \frac{2(\text{Precision} * \text{recall})}{\text{Precision} + \text{recall}} \tag{5}$$

The F-measure usually can be calculated for all classes in the dataset. Table 5, displays the outcomes of the classification algorithms when applied on the dataset.

Table 5. Results based on accuracy and time

Classifier	Correctly classified instances	Percentage of correctly classified Instances%	Incorrectly classified instances	Percentage of incorrectly classified Instances%	Time taken to build model
SVM	310	96.25	11	3.42	1.98
RF	312	97.19	9	2.80	1.8

It is observed that RF gives the highest accuracy which is 97.19 %. On the basis of accuracy and time, RF classifiers are considered to be the best for classification of wicked websites detection dataset. In Table 6, it can be seen the kappa statistics for both RF and SVM are very high, this means that the two observers agreed more than would be expected just by chance (Although RF is relatively higher than SVM). On the hand, SVM classifier algorithm has less error rates compared to RF algorithm, but the performance of SVM in terms of time and accuracy is slightly lesser than RF.

Table 6. Result based on errors

Classifier	Kappa statistics	Mean absolute error	Root mean squared error	Relative absolute error (%)	Root relative squared error (%)
SVM	0.93	0.034	0.18	6.83	36.89
RF	0.94	0.118	0.19	23.66	38.38

4 Conclusions

Remarkably, there are massive amount of users who use internet around the globe and there are numerous wicked websites all over the world. This research work detects wicked websites. The designed system has the ability to take input documents, then preprocesses them. And finally, automatically identifies wicked web site using data mining techniques. Both SVM and RF techniques are used for this purpose. The experimental results show that RF algorithm produces better accuracy and higher Kappa statistics than SVM. This designed system can easily be adapted and expanded to tackle other security applications such as terrorism forecasting, fraud documents, etc.

References

1. Abouenour, L., Bouzoubaa, K., Rosso, P.: Improving Q/A using Arabic Wordnet. In: Proceedings of the Arab Conference on Information Technology (CIT 2008). IBTIKARAT Research Group, Tunisia (2008)
2. Alkhalifa, M., Rodríguez, H.: Automatically extending NE coverage of Arabic WordNet using wikipedia. In: Proceedings of the 3rd International Conference on Arabic Language Processing, Rabat, Morocco, pp. 23–30 (2009)
3. Boudabous, M.M., Kammoun, N.C., Khedher, N., Belguith, L.H., Sadat, F.: Arabic WordNet semantic relations enrichment through morpholexical patterns. In: Proceedings of the 1st International Conference on Communications, Signal Processing and their Applications, pp. 1–6. IEEE Xplore Press, Sharjah (2013)
4. Elberrihi, Z., Abidi, K.: Arabic text categorization: a comparative study of different representation modes. *Int. Arab J. Inform. Technol.* **9**, 465–470 (2012)
5. El-Halees, A.: A comparative study on Arabic text classification. *Egypt. Comput. Sci. J.* **20**, 57–64 (2008)

6. Yousif, S.A., Samawi, V.W., Elkabani, I., Zantout, R.: Enhancement of Arabic text classification using semantic relations of Arabic WordNet article. *J. Comput. Sci.* **11**(3), 498–509 (2015)
7. Duwairi, R.: Arabic text categorization. *Int. Arab J. Inform. Technol.* **4**, 125–131 (2007)
8. Yadav, M., Mittal, P.: Web mining: an introduction. *Int. J. Adv. Res. Comput. Sci. Softw. Eng.* **3**(3), 683–687 (2013)
9. Kumbhar, V.S., Oza, K.S.: Educational web mining using weka. *Int. J. Eng. Trends Technol. (IJETT)* **26**(3), 128–131 (2015)
10. Alahmadi, A., Joorabchi, A., Mahdi, A.E.: Combining bag-of-words and bag-of-concepts representations for Arabic text classification. In: *Proceedings of the 25th IET Irish Signals and Systems Conference 2014 and 2014 China-Ireland International Conference on Information and Communications Technologies*, pp. 343–348. IEEE Xplore Press, Limerick (2014)
11. Zubi, Z.S.: Using some web content mining techniques for Arabic text classification recent advances on data networks, communications, computers, pp. 73–84 (2009)
12. Ramos, J.: Using tf-idf to determine word relevance in document queries. In: *Proceedings of the First Instructional Conference on Machine Learning* (2003)
13. Mathioudakis, M., Koudas, N.: Twittermonitor: trend detection over the twitter stream. In: *Proceedings of the 2010 ACM SIGMOD International Conference on Management of data*, pp. 1155–1158. ACM, Chicago (2010)
14. Zubi, Z.S.: Text classification in deep web mining latest trends. In: *Applied Informatics and Computing*, pp. 55–64 (2010)
15. Al-Shargabi, B., Alromima, W., Olayah, F.: A comparative study for Arabic text classification algorithms based on stop words elimination. In: *Proceedings of the 2nd International Conference on Intelligent Semantic Web-Services and Applications*, Amman, Jordan (2011)
16. Alsaad, A., Abbod, M.: Arabic text root extraction via morphological analysis and linguistic constraints. In: *16th International Conference on Computer Modelling and Simulation*, pp. 125–130. IEEE, Cambridge (2014)
17. Belmouhcine, A., Benkhalifa, M.: Implicit links based web page representation for web page classification. In: *Proceedings of the 5th International Conference on Web Intelligence, Mining and Semantics, WIMS 2015*. ACM, New York (2015)
18. Hammo, B., Abu-Salem, H., Lytinen, S., Evens, M.: QARAB: a question answering system to support the arabic language. In: *Workshop on Computational Approaches to Semitic Languages*, pp. 55–65, Philadelphia (2002)
19. Khorsheed, M.S., Al-Thubaity, A.O.: Comparative evaluation of text classification techniques using a large diverse Arabic dataset. *Lang. Resour. Eval.* **47**, 513–538 (2013)
20. Fodil, L., Sayoud, H., Ouamour, S.: Theme classification of Arabic text: a statistical approach. In: *Proceedings of the Terminology and Knowledge Engineering (TKE 2014)* (2014)
21. Harrag, F., El-Qawasmah E.: Neural network for Arabic text classification. In: *Proceedings of the 2nd International Conference on the Applications of Digital Information and Web Technologies*, pp. 778–783. IEEE Xplore Press, London (2009)
22. Forman, G.: An extensive empirical study of feature selection metrics for text classification. *J. Mach. Learn. Res.* **3**, 1289–1305 (2003)
23. Lodhi, H., Saunders, C., Shawe-Taylor, J., Cristianini, N., Watkins, C.: Text classification using string kernels. *J. Mach. Learn. Res.* **2**, 419–444 (2002)
24. Breiman, L.: Random forests. *Mach. Learn.* **45**(1), 5–32 (2001)

25. Cortes, C., Vapnik, V.: Support vector networks. *Mach. Learn.* **20**(3), 273–297 (1995)
26. Lin, C.-J.: Asymptotic convergence of an SMO algorithm without any assumptions. *IEEE Trans. Neural Netw.* **13**(1), 248–250 (2002)
27. Platt, J.C.: Sequential minimal optimization: a fast algorithm for training support vector machines: advances in kernel methods – support vector learning (1998)

Prediction of Malicious Domains Using Smith Waterman Algorithm

B. Ashwini^(✉), Vijay Krishna Menon, and K. P. Soman

Centre for Computational Engineering and Networking (CEN),
Amrita School of Engineering, Coimbatore, Amrita Vishwa Vidyapetham,
Amrita University, Coimbatore, India
aswiniblue@gmail.com

Abstract. IT security is an issue in today world. This is due to many reasons such as, malicious domains. Predicting the malicious domain in a set of domains is important. Here we have proposed a method for analysing such domains. In this method Wireshark is used for capturing the network packets. These packets are further given to client server machine and store in server database which makes an interface between the wireshark and machine. The data from the server database are then compared with the dictionary to predict the malicious websites. It is identified in such a way that if a word in a domain matches with any one of the dictionary word then it is considered as non-malicious websites others are malicious websites.

Keywords: Domain Name System (DNS) · Data acquisition · Wire-shark · Malicious · Smith-Waterman

1 Introduction

A new big data approach which includes analysing different log files, network packets is becoming more effective in identifying and preventing security issues. This method ensures to provide solutions to different security problems like inside attacker, fraud detection and different persistent attacks. Network security is the important aspect of secure data transmission. Network security includes authentication, authorization and access the data. Browser is the important entry for many attacks. It focuses on taking the other information in public and private information from users. Malware attacks are improving threat in today's security.

Malicious links are origin of distribution channel to generate malware over the web. The information could be accessed by the unknown user for malicious needs [3]. Web surfers are not aware of malwares, spywares, Trojan to their devices. Moreover they are able to be redirected to such internet via extra equipped schemes equivalent to fast flux network. Hence, users are effectively played to reveal private information utilizing phishing and pharming attacks. Browser gathers all the information from vulnerable data such as favorites, caches, history and cookies. Identifying such websites is not an easy task, due

to billions of World Wide Web. Many extensions had been encapsulated into browsers, toolbars and search engines like yahoo and Google.

The client server system is executed using java programming for interface. It is used to recognize network feature and serializable interface method. It is used to store the message for system and import information for procedure. The client send request about the packets, from the server database the packets are given to the client. For the development of protocols, it is vital to have a just strong protocol analyzer. Especially for IP addresses networking protocol can be used over Ethernet and there are numerous devices to change the Ethernet card into promiscuous mode and due to this fact sniff and analyses all packets. Big data analytic platform that can be utilized for storing this huge data, and also to process the data to derive and design new rules and policies for better protection and prediction. In this experiment the DNS packets which are populated in the network is captured at the gateway of the network. This packet is a parsed using pcap libraries. Using the information retrieved from the DNS Query packets an analysis of the traffic in the network is performed. As the number of users increases the total DNS query packets also increases The DNS data can be easily treated as instance of big data problem. There is a lot of analysis and monitoring tools are available for DNS packet analysis, but they cannot be scaled as the size and velocity at which data arrival increases.

The flow of paper is as follows. The literature survey is explained in Sect. 2 which is followed by the tool used in this method by Sect. 3. Section 4 is Dataset, Sect. 5 is about mathematical background of the system, proposed method is discussed in Sect. 6, Experimental Results of the proposed method in Sect. 7.

2 Literature Survey

Aldwairi et al., proposed a light weight system for identifying malicious website in URL lexical and host features and they named it as MALURLs [5]. The above system relies on Navie Bayes algorithm, a probabilistic model for detecting the malicious websites [5]. Park proposed a frame work for generating forensic data which also analyze the DOS attack traffic on WiBro network. Where, the packets are achieved from malicious DOS attack [6]. Tsolmon Otgonbold proposed a technique called ADAPT for detecting malicious fast-flux domain. In the proposed work he developed a prototype of ADAPT, which reads data from domain zone files to detect malicious fast flux domains [7]. Anjali et al., proposed a work which deals with URLs for identifying the malicious website, where a classification model learning method is used [8].

Yajin et al., presented a systematic study for detecting malicious applications on Android markets where, they used a heuristic based filtering scheme for identifying unknown malicious families [9]. Abes et al., presented a system for capturing local area network packets using Wireshark where large packets are generated almost at line rate [10]. Wolf et al., proposed a method for analyzing packets using Wireshark which is based on IEEE 802.15.4 packets [2].

Khonji et al. survey on detecting the phishing attacks aim the unprotected the really in machine due to human factor. Cyber-attacks expand via mechanism

that exploit frailty found in end users, which make people the weakest component in the security chain [13]. Weibo Chu et. Learn the advantages of machine learning used on phishing using only host based and lexical domain features. The data are collected using real live less than 20h. But the data are available in internet without using the webpage, but it bring the doubtful domains bring additional danger since today malicious webpages contain malicious code like Trojan. Lexical and domain feature are used to phishing urls [14].

Jin et al. proposed detecting spam on social media network security. Large post on available on network but identifying spam post manually is difficult. Proposed method uses scalable learning approach verify as many spam as possible [15].

Chu et al. Learn the advantages of machine learning used on phishing using only host based and lexical domain features. The data are collected using real live less than 20 h. But the data are available in internet without using the webpage, but it bring the doubtful domains bring additional danger since today malicious webpages contain malicious code like Trojan. Lexical and domain feature are used to phishing urls.

3 Wireshark

Wireshark is open-source software and adapted to specific needs of applications [2]. Windump is a network analyzer tool for windows; it uses WinPcap, the Windows port of libpcap. Windump prints the contents of network packets. Windump can write packets to standard output or a file. It listens to the specified Ethernet port eth0 and captures all packets towards port no 53. DNS packets communicate through the well-known port 53 which is allocated for DNS. Windump captures all network traffic on port 53 and saves into Winpcap file. It is a protocol for network analyzer application used for networking problems, analysis of software and protocol improvement. It is attached with local network interface in these incoming data packets that are investigated and displayed to the user. A network analyzer is software tool that can obstruct and data passing over a network. As data travels over the network, sniffers “packets captures” each protocol data unit and decode the packets and analyses it according to the specifications. Wireshark provide to identify the different protocols [1]. It supports large number of protocols like IEEE 802.11, IPV4, IPV6and other protocols. For later analysis of data packets is stored in a file.

4 About Dataset

Given the Domain names as an input, the task is to extract the malicious domains. The dataset is composed of 530749 domain names, half are malicious and other is non-malicious. The domain name is chosen from URLblacklist and other multiple websites. The dataset contains the entertainment, games weather, search engines and other domains etc. We have collected some of data using Wireshark. In Wireshark by the local machines using promiscuous mode we

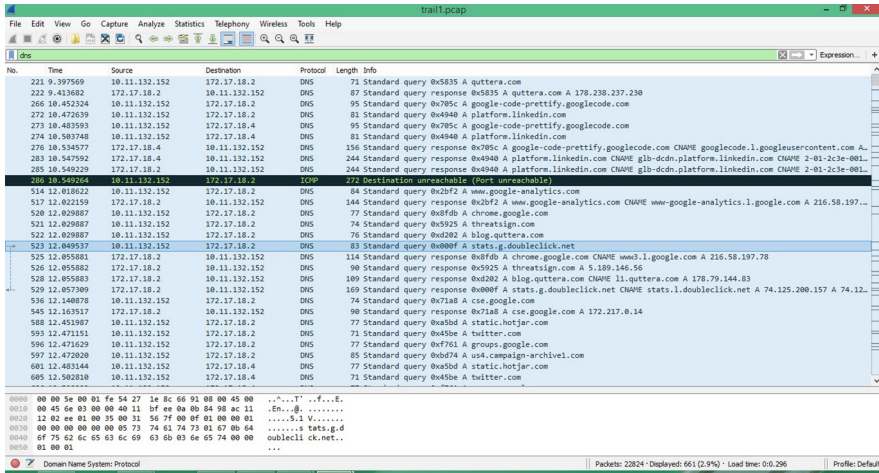


Fig. 1. Capturing packets using wireshark

have collected data (Fig. 1). The DNS packets are extracted for further process. The dictionary contains a 79753 unique words downloaded from internet.

5 Algorithms

5.1 Smith Waterman

In this paper the Smith-Waterman algorithm is used to compare the domain related strings with the words in the dictionary. This algorithm is commonly used to identify the optimal region of similarity in the given sequence [4]. We define $H_{i,j}$ as a cell matrix where $H_{i,j}$ is defined as the following equation where corresponds to the rows and corresponds to the column.

$$H_{i,j} = \max \begin{Bmatrix} 0 \\ H_{i-1,j-1} + S_{i,j} \\ H_{i-1,j} - d \\ H_{i,j-1} - d \end{Bmatrix} \quad (1)$$

Where $S_{i,j}$ denotes the similarity score between two sequences. Here the score between a word and the dictionary words. The symbol d corresponds to the penalty gap for the mismatch in the sequence. The SW algorithm has the following steps

1. Initialization
2. Filling of matrix
3. Trace back (or) back tracking

Here we are interested in forming a matrix with $N_{dw} + 1$ and $N_w + 1$ which corresponds to the rows and columns. Assume, for initial steps there is no gap penalty between the sequences. Hence, fill the first row and first column in the matrix as zero. For filling the rest of the matrix, Eq. (1) is taken into account and a scoring is assumed in order to fill the matrix.

$$S_{i,j} = \begin{cases} 2 & \text{if } N_{dw} = N_w \\ -1 & \text{else} \end{cases} \text{ and } d = 2$$

If a cell gets an answer of -1 then it is filled as 0. Likewise all the cells in the matrix are calculated. For the trace backing, it starts with the cell with largest number and back track until a minimum value number encounters [10].

5.2 Domain Generation Algorithm

Domain Generation algorithm (DGA) are algorithms viewed in lot of families of malware which might be used to periodically generate a tremendous number of domains that can be used as rendezvous aspects with their command and manipulate servers. The tremendous number of skills rendezvous points makes it problematic for law enforcement to simply shut down botnets in view that contaminated computers will attempt to contact some of these domain names every day to receive updates or instructions [11]. By means of using public-key cryptography, it is unfeasible for regulation enforcement and different actors to imitate instructions from the malware controllers as some worms will robotically reject any updates not signed by using the malware controllers.

Our detection methodology is situated on the remark that algorithmically generated domains differ drastically from respectable (human) generated ones in terms of the distribution of alphanumeric characters [12]. The distribution of alphanumeric characters, outlined because the set of English alphabets (a–z) and digits (zero–9) for both reputable as well as malicious domains 1. We derive the next features: (i) First, be aware that both the non-malicious information units show off a non-uniform frequency distribution, in example letters ‘m’ and ‘o’ show up most commonly in the non-malicious ISP information set whereas the letter ‘s’ seems most often within the non-malicious DNS data set. (ii) Even essentially the most sophisticated algorithmic domain generator seen within the wild for Kraken botnet has a relatively uniform distribution, albeit with larger frequencies on the vowels: ‘a’, ‘e’ and ‘i’. (iii) If botnets of future were to evolve and assemble phrases that are pronounceable yet now not in the dictionary, then they might no longer show off a uniform distribution as expected. For example, Kwyljibo displays better frequencies at alphabets, ‘e’, ‘g’, ‘i’, ‘l’, ‘n’, etc. On this regards, techniques which are founded on only the distribution of unigrams (single alphanumeric characters) will not be sufficient.

6 Proposed System

The flow of proposed system is shown in the Fig. 2. The aim of proposed system is to predict the malicious domain in a collection of domains. Network traffic

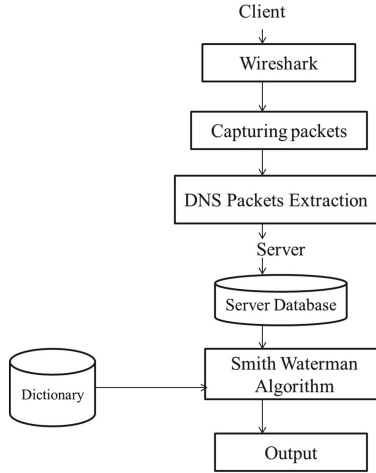


Fig. 2. Flow diagram of proposed system

on local network is inaccessible for you by default. Wireshark is used to capture the new packets which configure the card in promiscuous mode and wait until the necessary packet has been captured. In those packets, the DNS packets are taken into account from where the domain names are extracted. On this system they are important part such as client server software which is being attentive to the whole request of the client and client which can be sending request to server. These extracted features are given to client server machine. The client sends packet to server where the data from the server side are store in database. These data are given as an input to Smith-Waterman algorithm. This algorithm is used to compare the sequence of words with the words in the dictionary. Each domain is compared with entire English dictionary in which the substring in the domain is also compared. Comparing the domain with dictionary is itself is a big data problem where comparing the substring is time consuming process. This is compared in the view that the non-malicious domains can be in found with the words in the dictionary or the words which are having a proper meaning. The words which are not in the dictionary correspond to the malicious domain. Time taken for finding malicious domain in dataset is depicted in Table 1 where the dictionary has 79753 unique words.

7 Experimental Results

The experiment is conducted on windows 64bit machine with i5 core processor and 8 GB RAM. The domain names are collected from various internet resources and wireshark. Wireshark captures Source address, protocol, time, Destination address, number, length, domain names etc., from which the domain names are extracted. These domains are checked manually if duplicates are found it has been removed from the dataset. The requested data from the server are sent from

Table 1. Detailed description of time taken by the System

Domains	Malicious	Non-malicious	Time
100	62	38	5 min
500	290	210	15 min
1000	575	425	1/2 hr
1579	1200	379	1hr
3150	1929	1221	1hr29min
9100	8279	821	1hr50min
243150	119068	124082	2hr13min
530749	368321	162428	3hr45min

the client machine. The server data are given as an input to the Smith Waterman algorithm. The dictionary database is downloaded from internet which is taken by the algorithm for comparing the domain names with dictionary words. If the string and substring in the domain name or matched with any of the dictionary words then confidence score will be high. If the string or substring in the domain name matches with any of the dictionary words then the corresponding confidence score will be comparatively low. If the substring and string of the domain name does not matches with any of the dictionary words then the confidence score will be very low or equivalent to zero. The domain name with very low and zero confidence score, they or generated using domain generation algorithm. The malicious and non-malicious domain can be predicted using our proposed method. Time taken for predicting the malicious websites by the proposed system is depicted in Table 1. It shows that the time taken by the system is growing linearly in increasing the size of the domain names. Hence forth for prediction of malicious websites using smith waterman algorithm depends on the number of domain names it encounters. The non-malicious websites are can also be generated using Domain Generation Algorithm.

8 Conclusion

In this paper, we have predicted the malicious domain based on Smith Waterman algorithm. In this work, we have evaluated domain names using Smith Waterman. In experimental evaluation DNS packets are extracted using wireshark and given as a feature descriptor. Here, domains are given as input for Smith Waterman algorithm and prediction of malicious domains and non-malicious domains are done. The time taken for predicting malicious and non-malicious websites is evaluated.

References

1. Gupta, S., Mamtora, R.: Intrusion detection system using wireshark. *Int. J. Adv. Res. Comput. Sci. Soft. Eng* **2**, 34–36 (2011)
2. Pottner, W.-B., Wolf, L.: IEEE 802.15. 4 packet analysis with wireshark and off-the-shelf hardware. In: *Proceedings of the Seventh International Conference on Networked Sensing Systems (INSS2010)*, Kassel, Germany (2010)
3. Kaushik, S., Singhal, A.: Network security using cryptographic techniques. *Int. J. Adv. Res. Comput. Sci. Softw. Eng.* **2**(12), 105–107 (2012)
4. Saliman, N., Farah, A., et al.: Software Implementation of Smith-Waterman Algorithm in FPGA
5. Aldwairi, M., Alsalmán, R.: MALURLS: a lightweight malicious website classification based on url features. *J. Emerg. Technol. Web Intell.* **4**(2), 128–133 (2012)
6. Park, D.: A study of packet analysis regarding a DoS attack in WiBro environments. *IJCSNS Int. J. Comput. Sci. Netw. Secur.* **8**(12), 398–402 (2008)
7. Otgonbold, T.: ADAPT: an anonymous, distributed, and active probing-based technique for detecting malicious fast-flux domains (2014)
8. Sayamber, A.B., Dixit, A.M.: On URL classification. *Int. J. Comput. Trends Technol. (IJCTT)* **12**
9. Zhou, Y., et al.: Hey, You, Get Off of My Market: Detecting Malicious Apps in Official and Alternative Android Markets (2012)
10. Dabir, A., Matrawy, A.: Bottleneck analysis of traffic monitoring using wireshark. In: *4th International Conference on Innovations in Information Technology, 2007. IIT 2007*. IEEE (2007)
11. Hongke, H., Linhai, Q.: Application and research of multidimensional dataanalysis in power quality. In: *2010 International Conference on Computer Design and Applications (ICCDA)*, vol. 1. IEEE (2010)
12. Liu, L., Han, Z.: Multi-block ADMM for Big Data Optimization in Modern Communication Networks. *arXiv preprint [arXiv:1504.01809](https://arxiv.org/abs/1504.01809)* (2015)
13. Khonji, M., et al.: Phishing detection: a literature survey. *IEEE Commun. Surv. Tutorials* **15**(4), 2091–2121 (2013)
14. Chu, W., et al.: Protect Sensitive Sites from Phishing Attacks Using Features Extractable from Inaccessible Phishing URLs. *Microsoft Research Asia, Beijing* (2011)
15. Jin, X., et al.: Social spam guard: a data mining based spam detection system for social media networks. In: *37th International Conference on Very Large Data Bases, Washington, 29 August 2011* (2011)

Outsource-Secured Calculation of Closest Pair of Points

Chandrasekhar Kuruba, Kethzi Gilbert, Prabhav Sidhaye, Gaurav Pareek^(✉),
and Purushothama Byrapura Rangappa

National Institute of Technology, Ponda, Goa, India
chandusree928@gmail.com, kethzi.gildona@gmail.com,
prabhavsid995@gmail.com, {gpareek,puru}@nitgoa.ac.in

Abstract. Outsourcing data/computation intensive tasks to servers having great computing power and data analytics skills is gaining popularity. While this outsourcing model, due to its cost efficiency, has been widely used by numerous clients, making sure that loss of privacy and integrity of results are not affected remain as challenges, especially in public cloud infrastructure. For addressing these challenges, clients must outsource their data in a privacy-preserving and verifiable manner. The cost of assuring both privacy of data and correctness of results must impose cost marginally less than the cost of actual computation. In this paper, we address the problem of secure outsourcing of Closest Pair of Points computation. Finding Closest Pair of Points is central to many complex applications like clustering. Our scheme involves the client sending encrypted points to the server and receiving the result which is a pair of points (with smallest distance between them) along with a proof of correctness. Data encryption done to ensure privacy of input points must be such that the encrypted points retain the same order as the original points. For this, we designed and used a novel encryption scheme which is additively homomorphic and order-preserving for encrypting input points in our protocol. The protocol requires the server to compute almost all distances to be able to provide the proof of it having computed the results honestly.

Keywords: Verifiable computing · Secure outsourcing · Closest-pair-of points · Cloud security

1 Introduction

In this age of distributed computing devices, cloud computing, more precisely, XaaS (Anything as a Service) is gaining wide acceptance. As a result to this, notion of public clouds has also come into existence. Servers offer services in terms of heavy computations (like complex data mining operations) on big data sets and are paid by the clients which are limited by the computing resources and/or memory storage, for the same. Clients who use any cloud services rely heavily on the fact that the cloud server is not interested in cheating on the computations or breaching the privacy of the client's data. But the client can return results which look like the results client is expecting but the server may not be

calculating them honestly and returning random values in a particular range, for example. In such cases, on one hand, integrity and privacy of client's data is compromised and on the other hand, server saves its computation efforts (in terms of CPU cycles) and memory resources, which is a motivation good enough for server to "cheat". There is motivation for the server to not care about the client's data privacy also. Consider the case where the server which is given the log of web pages visited by a set of users for performing complex data mining operations, is contacted by an advertising agency looking for such logs for user's behavioral analysis. The server can gain significantly by disclosing the logs to the agency. Considering many critical applications for which the data/computations are increasingly being outsourced, one can imagine the threat to his data privacy and reputation. The above sets motivation for secure outsourcing of data and computations. The goals of a secure-outsourcing protocol include securing input and output from disclosure, cheating-proof execution of computations and limited (extra) work on the client's side. A variety of problems have been securely outsourced under two-server model and single server model. Verifiable computing deals with checking the authenticity of the results returned from the server. Effectiveness of a verifiable scheme is the average fraction of the total work a malicious server has to do in order to successfully cheat the client despite the verification algorithm being part of the scheme. This fraction should be very close to 1 for a good verifiable scheme.

In this paper, we consider one of the fundamental geometric problems, computing the closest-pair-of points. Closest Pair of Points deals with identification of the pair with minimum Euclidean distance amongst a given set of points. Since the problem has repeated computation of Euclidean distances between different pair of points, it serves as a primitive step for many data mining applications. The client sends the set of points and the server should compute distance between each pair of points and return the pair of points with minimum distance as output. Secure outsourcing of such a computation requires that the points are encrypted and the server provides the proof of correctness along with the result pair. The encryption we used for encrypting the points is order-preserving so that the relative order of computed distances between the encrypted points is same as for the original points.

Our contributions can be summarized as follows. We addressed the problem of secure outsourcing of computation of Closest Pair of points. Using the proposed scheme, a client delegates the computation in privacy preserving and verifiable manner. The work done by the client for preprocessing the input points and processing the result returned by the server for obtaining the final result amounts to no more than $O(n)$ where n is the number of input points. For achieving this, we have designed and used a novel order-preserving encryption scheme for encryption of input points. A theoretical analysis of correctness and security of the result verification protocol is also presented. We have also proposed a new algorithm for computing Closest Pair of Points which outperforms both brute-force ($O(n^2)$) and divide-and-conquer ($O(n \log n)$) versions of it.

The rest of the paper is organized as follows. Section 2 presents review of important existing literature on privacy preserving and verifiable computation

outsourcing. Section 3 covers all the preliminaries required for understanding a secure outsourcing protocol for any important computation. In Sect. 4, complete description of the proposed scheme for outsourcing closest pair of points is presented which includes proposed order-preserving encryption scheme in Sect. 4.1, outsourcing algorithm in Sect. 4.2, discussion on adaptability of the scheme to the two-server model in Sect. 4.3. Randomized verification for the outsourcing scheme is presented in Sect. 4.4 along with its analysis in Sect. 4.5. A new algorithm for computing closest pair of points that outperforms the existing algorithms existing ones is proposed in Sect. 4.6 along with the experimental results in Sect. 4.7.

2 Related Work

Atallah et al. [8] propose the notion of disguise that can be introduced by the client during the preprocessing phase of a outsource-secured computation. Disguising the computations hides the computation along with the input and output data from the server. Random object generation is considered important in disguising. So, the security and effectiveness of all these disguising schemes rely on the assumption that random component is generated using a “good enough”. The work also encompasses techniques that can be applied for domain and dimension modification.

Certain classes of algebraic and differential computations have also been considered as candidates for secure outsourcing [9]. It is shown that approximating many of the algebraic differential equations can be reduced to solving classical problems like Abstract Equation(AE), Boundary Value Problem(BVP), Initial Value Problem (IVP) with secret parameter etc. Also, non-linear equations can be reduced to linear forms. Next we present the classes of problems and their reduction to the “simpler” problems as in [9] along with basic methodology used for securely outsourcing them.

Hohenberger et al. [10] formally define the notion of outsource security in presence of untrusted server(s). The notion of α -efficient and β -checkable Secure Outsourcing algorithms is also given after which a $(O(\frac{\log^2 n}{n}), \frac{1}{2})$ -outsource secure implementation of modular exponentiation is presented under the two untrusted program model.

Secure outsourcing of product of two large matrices in two server setting appeared in 2008 [1]. The extension for cheating detection for the protocol depends on the infeasibility of predicting a matrix A given a the matrix AR or RA (but not both at the same time), where R is a radom matrix.

Linear Algebra Computations serve as building block for many cryptographic algorithms. Problem of secured outsourcing of characteristic polynomial and eigenvalues of a matrix is addressed by Hu et al. [12] without any cryptographic assumptions. The protocol is non-interactive. Verification of result requires $O(n^2)$ local computations by the client. The protocol is efficient with client needing only $O(n^2)$ multiplications to calculate the characteristic polynomial and eigenvalues. Mohassel et al. in [13] show that if there exists a homomorphic

encryption scheme that is distinctive, associative, additive and multiplicative then the matrix computations can be securely outsourced with client doing at most $O(n^2 \log n)$ work.

To combat limited computing power of client, Wang et al. [14] developed a mechanism for securely outsourcing large-scale system of linear equations along with a robust cheating detection mechanism. One-time setup phase of the protocol takes $O(n^2)$ cost. Finally, $O(n)$ cost is incurred by client to calculate the n variables in the problem. When a key (r) of size 768-bit, the protocol performs well above the computation baseline for the problem. However, using the 1024-bit key leads the protocol to perform badly as compared to what is considered as computation baseline for the LE problem.

Problem of public verification of results for outsourced computations without the verifier needing any key [15] appeared in the year 2012. New protocols to outsource-secured *evaluation of high degree polynomials* and *matrix multiplication* are proposed under stronger adversarial models.

The problem of privacy preserving distributed k-means clustering of arbitrary partitioned data was first addressed by Geetha et al. [16] after which improved versions of outsource-secure k-means clustering appeared in [17, 18]. Bunn et al. [17] proposed a slightly more efficient private multi-party k-means clustering algorithm. Additionally, it also addresses the problem that could arise from dishonest contributing sites. The scheme works under the assumption of existence of semantically secure Homomorphic encryption scheme.

3 Preliminaries

3.1 Outsourcing Models

Outsourcing is a method by which a weak client asks the server to perform computations which cannot be done by the client. In *single-server model*, there is one server and one client. The server does the entire computation by itself. The data on which the computation is to be done is passed on by the client to the server. Since the server does the entire computation by itself it also has the result of the desired computation. In case of an untrusted server, the client has high risk of losing his data privacy and/or integrity of computed output. Whereas in *two-server model*, the computation is done in parts by the two servers present and the partial results are put together either by one of the servers or by the client [1]. We follow a model where the servers are allowed to communicate with one another. However, there exist outsourcing models where there exists no communication between the servers. In our algorithm, we assume that the two servers do not collude, hence there is no need for encryption. The claim that the solution proposed in this paper fits both the models is proved in Sect. 4.3.

3.2 Requirements for Our Encryption Scheme

To ensure data privacy, the input points need to be encrypted before sending to the server. The encryption algorithm we propose for securely outsourcing closest pair of points is order preserving and homomorphic.

Order Preservation. Since we have to identify the closest pair of points using distances the encryption must not change the order of the original points i.e. the relative order of the points must not be disturbed.

For instance, consider two points P_1 and P_2 . Also consider an encryption scheme with the encryption algorithm E . Suppose $P_1 > P_2$ then $E(P_1) > E(P_2)$ where $E(P_1)$ and $E(P_2)$ are encrypted values of P_1 and P_2 respectively.

Homomorphic Property. The encryption scheme we use must have homomorphic property over addition. The encryption scheme is said to be additively homomorphic if decrypting the sum of encrypted values will give us the sum of plaintext values. Consider an encryption scheme which gives the encrypted values of P_1 and P_2 denoted by $E(P_1)$ and $E(P_2)$. An encryption scheme is additively homomorphic if

$$Dec(E(P_1) + E(P_2)) = P_1 + P_2$$

Where $Dec(E(P_x))$ is the decryption of $E(P_x)$.

4 Outsourcing of Closest Pair of Points

Outsourcing of the closest pair of points problem requires that a client provides a server a set of n points and the server computes and returns a pair with minimum distance. We discuss all the steps of the proposed scheme in detail and verify the claims we provide with each step.

4.1 The Proposed Order Preserving Symmetric Encryption Scheme

Order preserving encryption scheme is an encryption scheme that preserves the numerical ordering of its input. It was originally developed for enabling efficient range queries over encrypted database [2]. The construction of this scheme is based on the uncovered relation between the random order preserving function and hyper-geometric probability distribution. The order preserving function is a one-one function from the domain set of cardinality M and Range set of cardinality N such that $N > M$. The function can be defined as selection of M out of N ordered items and simultaneously fulfilling the requirement of a good encryption scheme - “as random as possible”. We have implemented this scheme for sample data points shown in Fig. 1a and the data points after encryption are shown in Fig. 1b. It can be seen from the output of encryption the data points get reflected about the line $x = x_{max}/2$ and $y = y_{max}/2$.

Assumptions. Without loss of generality, we can make the following assumptions on the data points:

1. All the data points lie in the first quadrant of the co-ordinate system i.e. for all points $P(x, y)$ both x and y are greater than 0.

2. None of the data points lie on either x or y axis i.e. no points have either abscissa or ordinate as 0.

Key Generation. We begin by identifying the maximum value that the coordinates of our data points take. Let this maximum value be P .

$$P = \text{Max}(x_1, x_2, x_3, \dots, x_n, y_1, y_2, y_3, \dots, y_n)$$

Randomly choose a number r such that $r \geq 3$ ($r \in \mathbb{Z}_m$) and a random $R \in [1, P - 1]$. Let $Q = r \times P$ and for every point $P_i = (x_i, y_i)$, compute $x_{inorm} = x_i \times R/P$ and $y_{inorm} = y_i \times R/P$

Encryption. Now we encrypt a point $P_i(x_i, y_i)$ as

$$\begin{aligned} \text{Enc}(x_i) &= Ex_i = Q - P - x_i - x_{inorm} \\ \text{Enc}(y_i) &= Ey_i = Q - P - y_i - y_{inorm} \end{aligned}$$

where $\text{Enc}(a)$ =Encryption of data value a . Therefore, the encrypted value of point P_i is

$$\text{Enc}(P_i(x_i, y_i)) = EP_i(Ex_i, Ey_i)$$

Claim. Euclidean distance computation can be directly performed on EP_i and EP_j i.e. $ED(EP_i, EP_j) = (1 + R/P) \times ED(P_i, P_j)$, where ED is the Euclidean distance between points P_i, P_j

Proof.

$$\begin{aligned} Ex_i &= Q - P - x_i - x_{inorm} \\ &= r \times P - P - x_i - R/P \times x_i \\ &= P(r - 1) - x_i((P + R)/P) \end{aligned}$$

Similarly,

$$\begin{aligned} Ey_i &= P(r - 1) - y_i((P + R)/P) \\ D(EP_i, EP_j) &= [(Ex_i - Ex_j)^2 + (Ey_i - Ey_j)^2]^{1/2} \\ &= (P + R)/P \times D(P_i, P_j) \end{aligned}$$

Lemma 1. *If $D(P_i, P_j) > D(P_x, P_y)$ then $ED(EP_i, EP_j) > ED(EP_x, EP_y)$*

Consider,

$$D(P_i, P_j) > D(P_x, P_y) \implies (1 + R/P)D(P_i, P_j) > (1 + R/P)D(P_x, P_y) \quad (1)$$

Since R, P are all positive, Using property proved above on Eq.1 we get, $ED(P_i, P_j) > ED(P_x, P_y)$

Because of the above property we can directly compare the Euclidean distances between all points and identify the closest pair. Our encryption scheme successfully hides actual distances between every pair of points.

4.2 Outsourcing Algorithm

Client

- 1 Has set of data points $P = \{P_1(x_1, y_1), P_2(x_2, y_2), \dots, P_n(x_n, y_n)\}$.
- 2 Encrypts data points in P with an appropriate scheme to obtain the set $EP = \{EP_1, EP_2, \dots, EP_n\}$ such that

$$EP_i = Enc(P_i) = (Enc(x_i), Enc(y_i))$$
- 3 Receives the encrypted closest pair of points from the server EP_x and EP_y
- 4 Decrypts the received pair of points to obtain the required results

$$Dec(EP_x) = Dec(Enc(P_x)) = P_x$$

$$Dec(EP_y) = Dec(Enc(P_y)) = P_y$$

$$P_x \text{ and } P_y \text{ are the closest pair of points.}$$

Server

- 1 Receives encrypted set of points $EP = \{EP_1, EP_2, \dots, EP_n\}$ such that

$$EP_i = Enc(P_i) = (Enc(x_i), Enc(y_i))$$
- 2 Computes closest pair of points EP_x, EP_y by computing Euclidean distances between encrypted points in set EP using any of the existing algorithms.
- 3 Sends the results EP_x, EP_y obtained in step 2 to the client.

Next, we show that the outsourcing algorithm we presented here fits the popular two-server model.

4.3 Two-Server Model

In this model, we divide the coordinates between two servers such that the sum of coordinates with the two servers gives the original coordinates. Each server computes the partial results and one of the servers finally combines them and returns them to the client. In this case, if servers are not assumed to collude, the data encryption is not a necessity because each server only has a part of the sensitive information. In our model, we use the following algorithm:

1. The client divides the points into two parts such that The client randomly subtracts the number (x_{i_1}, y_{i_1}) from the co-ordinates of points $P_i = (x_i, y_i)$ i.e.

$$P_{i_2} = (x_i - x_{i_1}, y_i - y_{i_1})$$

let $x_i - x_{i_1} = x_{i_2}$ and $y_i - y_{i_1} = y_{i_2}$

Therefore,

$$x_{i_1} + x_{i_2} = x_i$$

$$y_{i_1} + y_{i_2} = y_i$$

2. Let $P_{i_1} = (x_{i_1}, y_{i_1})$ and $P_{i_2} = (x_{i_2}, y_{i_2})$ Similarly, $P_{j_1} = (x_{j_1}, y_{j_1})$ and $P_{j_2} = (x_{j_2}, y_{j_2})$

3. P_{i_1}, P_{j_1} is sent to S_1 for all $i, j < n$ and P_{i_2}, P_{j_2} is sent to S_2 for all $i, j < n$
4. S_1 computes $D(P_{i_1}, P_{j_1})$. During this computation S_1 computes $(x_{i_1} - x_{j_1})$ and $(y_{i_1} - y_{j_1})$.
5. S_2 computes $D(P_{i_2}, P_{j_2})$. During this computation S_2 computes $(x_{i_2} - x_{j_2})$ and $(y_{i_2} - y_{j_2})$.
6. S_2 sends $(x_{i_2} - x_{j_2})$ and $(y_{i_2} - y_{j_2})$ as well as $D(P_{i_2}, P_{j_2})$ to S_1
7. S_1 combines the results it has computed and the data it has received to compute $D(P_i, P_j)$

Claim. Distance between P_i and $P_j = D(P_i, P_j)$ can be computed if first Server i.e. S_1 receives $D(P_{i_2}, D_{j_2})$ and $(x_{i_2} - x_{j_2}), (y_{i_2} - y_{j_2})$ from S_2

Proof. S_1 already computes

$$D^2(P_{i_1}, P_{j_1}) = (x_{i_1} - x_{j_1})^2 + (y_{i_1} - y_{j_1})^2 \text{ and } (x_{i_1} - x_{j_1}) \text{ and } (y_{i_1} - y_{j_1})$$

S_2 sends

$$D^2(P_{i_2}, P_{j_2}) = (x_{i_2} - x_{j_2})^2 + (y_{i_2} - y_{j_2})^2 \\ \text{and } (x_{i_2} - x_{j_2}) \text{ and } (y_{i_2} - y_{j_2})$$

Adding information from S_1 and S_2 ,

$$(x_{i_1} - x_{j_1})^2 + (y_{i_1} - y_{j_1})^2 + (x_{i_2} - x_{j_2})^2 + (y_{i_2} - y_{j_2})^2 \tag{2}$$

Since we already have $(x_{i_1} - x_{j_1}), (y_{i_1} - y_{j_1}), (x_{i_2} - x_{j_2})$ and $(y_{i_2} - y_{j_2})$, computing $2(x_{i_1} - x_{j_1})(x_{i_2} - x_{j_2}) + 2(y_{i_1} - y_{j_1})(y_{i_2} - y_{j_2})$ and adding with Eq. 2,

$$(x_{i_1} - x_{j_1})^2 + (y_{i_1} - y_{j_1})^2 + (x_{i_2} - x_{j_2})^2 + (y_{i_2} - y_{j_2})^2 + 2(x_{i_1} - x_{j_1})(x_{i_2} - x_{j_2}) \\ + 2(y_{i_1} - y_{j_1})(y_{i_2} - y_{j_2}) \\ = [x_i - x_j]^2 + [y_i - y_j]^2 \\ = D^2(P_i, P_j)$$

□

4.4 Randomized Verification Scheme

Along with privacy of actual input and output points, verifiability of the results returned by the untrusted server must also be ensured. This verification algorithm has mainly two phases. In the first phase, client pre-processes followed by outsourcing of input data along with a challenge. Second phase involves the server that computes the result and responds to the client with the output along with the response to the challenge. Response sent by the server becomes the basis for accepting or rejecting the result. The algorithm is as follows:

Input: Set of points $P = \{P_1, P_2, P_3, \dots, P_n\}$ where $P_i = (x_i, y_i)$.

Output: (P_i, P_j) such that $\min\{d(P_i, P_j)\}$ for all $i, j \leq n$ and $\delta = (d(P_i, P_j))$.

Assumption: The distances between all pair of points are distinct.

Pre-process and Outsource

1. Client C randomly chooses $i, j \in \{1, \dots, n\}$
2. Compute Euclidean distance between P_i, P_j call it α .
3. Send $P = \{P_1, P_2, \dots, P_n\}$ and α to server S .
4. C removes P from storage and only stores (P_i, P_j) .

Compute and Respond

1. Server computes Euclidean distance between $P_i, P_j \forall i, j \in \{1, \dots, n\}$
2. Finds the closest pair (P_i, P_j) and distance between them. Call it d_{min} .
3. Finds x, y such that $\alpha = d(P_x, P_y)$
4. Sends $\pi = (d_{min}, (P_i, P_j), (x, y))$

Verification $(\pi, (i, j))$

1. Check if $(i = x \text{ and } j = y)$ true
 then accepts d_{min} as minimum distance
2. else rejects d_{min}

4.5 Analysis of the Verification Scheme

A result verification scheme for privately outsourced closest pair of points computation has been provided in the previous section. Verification of results returned by the server involves challenging the server with a distance between any randomly chosen pair of points. The server has to return the result of the desired computations (i.e. closest pair) along with the response corresponding to the challenge. In order to still successfully cheat, it has to find the pair of points the client used for computing the challenge. This method can be analyzed by calculating the average amount of work (Euclidean distance computations) the server has to do in order to find out the pair of points used for computing the challenge. Once the server computes response to the challenge, there is no motivation left for the cheating server to calculate the rest of distances and it returns arbitrary result (pair of points) to the client. Let there be n points, then the probability that server guesses the pair of points in one attempt (i.e. by calculating only one distance) is:

$$P(1) = \frac{1}{\binom{n}{2}}$$

If the server continues working out distances in whatever order it wishes, the probability of it succeeding after computing “ w ” distances is:

$$P(w) = \frac{\binom{1}{1} \times \left(\binom{n}{2} - 1\right)}{\binom{n}{w}} = \frac{w}{\binom{n}{2}} \tag{3}$$

So, expected amount of work (Euclidean distance computations) done by the server in before it correctly obtains the challenge points:

$$E[W] = \sum_{w=1}^{\binom{n}{2}} w \times P(w) = \frac{\left(\binom{n}{2} + 1\right)\left(2\binom{n}{2} + 1\right)}{6}$$

The server can follow another rational approach by which it computes a fixed number of distances (say w') an upon failure in finding the challenge points, guesses from the remaining. In such a case, probability of success for the attacker is:

$$P'(w') = \frac{\binom{1}{1} \times \binom{\binom{n}{2}-1}{w'-1}}{\binom{\binom{n}{2}}{w'}} + (1 - \frac{\binom{1}{1} \times \binom{\binom{n}{2}-1}{w'-1}}{\binom{\binom{n}{2}}{w'}}) \times (\frac{1}{\binom{\binom{n}{2}}{w'} - w'}) = \frac{w' + 1}{\binom{\binom{n}{2}}{w'}} \quad (4)$$

Similarly, average number of distance computations the server has to do in order to find the challenge is:

$$E'[W'] = \sum_{w'=1}^{\binom{n}{2}} w' \times P'(w') = \frac{((\binom{n}{2}) + 1)((\binom{n}{2}) + 2)}{3}$$

It is clear from (4) that this strategy gives the attacker a very limited advantage as compared to (3). Next, we discuss an algorithm for calculating the closest pair of points that performs better than both brute force and divide and conquer algorithm.

4.6 Proposed Algorithm for Computing Closest Pair of Points

We propose an algorithm to compute closest pair from a given set of points. In this algorithm we compute relative distances between points. We follow the principle that two points with close relative distance will be close to one another. As observed in experimental results our algorithm outperforms both the divide and conquer algorithm and the brute force algorithm. The algorithm is as follows:

- 1 Let $P = \{P_1, P_2, P_3, \dots, P_n\}$ be a set of points.
 $x_{max} = \max(x_1, x_2, \dots, x_n)$, $x_{min} = \min(x_1, x_2, \dots, x_n)$, $y_{max} = \max(y_1, y_2, \dots, y_n)$, $y_{min} = \min(y_1, y_2, \dots, y_n)$
 $P_{max} = (x_{max}, y_{max})$, $P_{min} = (x_{min}, y_{min})$
- 2 $R = Dist(P_{min}, P_{max})$
 Calculate $R_1 = 2 \times R$, $R_2 = 3 \times R$
 $Sum = 1D$ Array
 for $i = 1, \dots, n$
 $Sum[i] = R_1 \times Dist(P[i], P_i) + R_2 \times Dist(P[i], P_2)$
- 3 Sort the array Sum .
- 4 (*The closest pair of points will be at most 5 positions apart in Sum .*)
 for $i = 1, 2, \dots, n$
 $P[i]$ = the point that corresponds to $sum[i]$
 Find points which corresponds to $sum[i + 1]$ to $sum[i + 5]$
 find the closest point to $P[i]$ among the above 5 points
 return the Closest pair among all above pairs

4.7 Experimental Results

In Fig. 2b it is observed that the time taken by the proposed algorithm is much less than the existing (and best known) divide and conquer version.

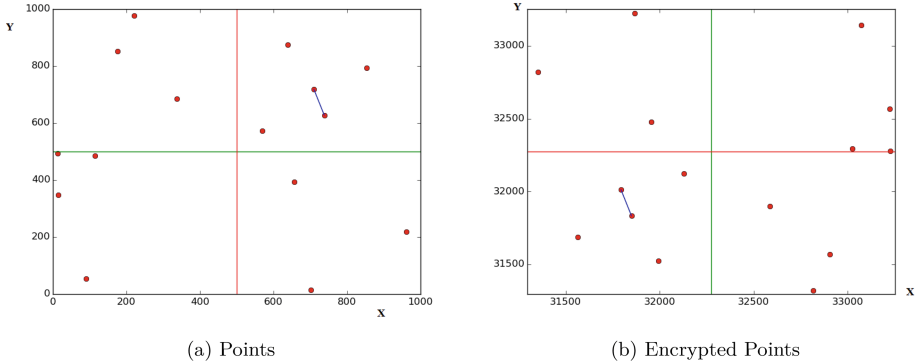
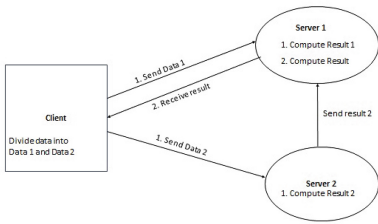
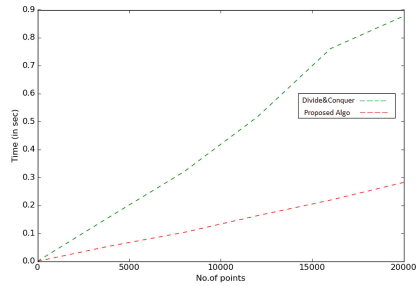


Fig. 1. The points are selected uniformly at random from the range 0–1000 units. Relative orientation and distances between the every pair of points is order preserved using our encryption scheme



(a) The Two-Server model of outsourcing



(b) Comparison between time taken by the divide and conquer and th proposed algorithm for computing closest pair of points

Fig. 2. Model for Two-Server cloud outsourcing and comparative analysis of the proposed scheme with the most efficient algorithm (Divide and conquer)

5 Conclusions and Discussion

Outsourcing of data and computations to the cloud servers with advanced computing resources is an effective application which is attracting numerous cloud users. However, when the server is untrusted, ensuring privacy of data (both input and output) and correctness of results becomes important. While outsourcing computations, it is naturally the case that they are computationally heavy and serve as a primitive step for more complex computations or data analytics. Computing closest pair of points is one such problem which involves repeated computations of Euclidean distances between different pair of points and is therefore a primitive step of many algorithms that involve grouping/clustering the data. A secure outsourcing algorithm for the problem is proposed with input and output privacy being provided by a novel order-preserving encryption algorithm

proposed in this paper. The result verification scheme has also been presented and the accompanying theoretical analysis proves its security. The client does only $O(n)$ work during this process. The adaptability of the proposed secure outsourcing scheme is shown with respect to the two-server cloud model. A novel algorithm for computing closest pair of points that outperforms its best known algorithm is also proposed. The verification algorithm needs to be improved to conform to the existing formal model for verifiable computing. This remains as the ongoing future work.

References

1. Benjamin, D., Atallah, M.: Private and cheating-free outsourcing of algebraic computations. In: 2008 Sixth Annual Conference on Privacy, Security and Trust (2008)
2. Boldyreva, A., Chenette, N., Lee, Y., O'Neill, A.: Order-preserving symmetric encryption. In: Joux, A. (ed.) EUROCRYPT 2009. LNCS, vol. 5479, pp. 224–241. Springer, Heidelberg (2009)
3. Tu, S., Kaashoek, M., Madden, S., Zeldovich, N.: Processing analytical queries over encrypted data. Proc. VLDB Endow. **6**, 289–300 (2013)
4. Ateniese, G., Burns, R., Curtmola, R., Herring, J., Kissner, L., Peterson, Z., Song, D.: Provable data possession at untrusted stores. In: Proceedings of the 14th ACM Conference on Computer and Communications Security (CCS 2007) (2007)
5. Purushothama, B., Amberker, B.: Efficient query processing on outsourced encrypted data in cloud with privacy preservation. In: 2012 International Symposium on Cloud and Services Computing (2012)
6. Vyas, R., Singh, A., Singh, J., Soni, G., Purushothama, B.R.: Design of an efficient verification scheme for correctness of outsourced computations in cloud computing. In: Abawajy, J.H., Mukherjea, S., Thampi, S.M., Ruiz-Martínez, A. (eds.) SSCC 2015. CCIS, vol. 536, pp. 66–77. Springer, Heidelberg (2015). doi:[10.1007/978-3-319-22915-7_7](https://doi.org/10.1007/978-3-319-22915-7_7)
7. Paillier, P.: Public-key cryptosystems based on composite degree residuosity classes. In: Stern, J. (ed.) EUROCRYPT 1999. LNCS, vol. 1592, p. 223. Springer, Heidelberg (1999)
8. Atallah, M.J., Pantazopoulos, K.N., Rice, J.R., Spafford, E.E.: Secure outsourcing of scientific computations. Adv. Comput. **54**, 215–272 (2002)
9. Seitkulov, Y.N.: New methods of secure outsourcing of scientific computations. J. Supercomputing **65**(1), 469–482 (2013)
10. Hohenberger, S., Lysyanskaya, A.: How to securely outsource cryptographic computations. In: Kilian, J. (ed.) TCC 2005. LNCS, vol. 3378, pp. 264–282. Springer, Heidelberg (2005)
11. Tysowski, P.K.: Highly Scalable and Secure Mobile Applications in Cloud Computing Systems (Doctoral dissertation, University of Waterloo) (2013)
12. Hu, X., Tang, C.: Secure outsourced computation of the characteristic polynomial and eigenvalues of matrix. J. Cloud Comput. **4**(1), 1–6 (2015)
13. Mohassel, P.: Efficient and Secure Delegation of Linear Algebra. IACR Cryptology ePrint Archive, 605 (2011)
14. Wang, C., Ren, K., Wang, J., Urs, K.M.R.: Harnessing the cloud for securely solving large-scale systems of linear equations. In: 31st International Conference on Distributed Computing Systems (ICDCS), pp. 549–558. IEEE (2011)

15. Fiore, D., Gennaro, R.: Publicly verifiable delegation of large polynomials and matrix computations, with applications. In: Proceedings of the 2012 ACM Conference on Computer and Communications Security, pp. 501–512. ACM (2012)
16. Jagannathan, G., Wright, R.N.: Privacy-preserving distributed k-means clustering over arbitrarily partitioned data. In: Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery in Data Mining, pp. 593–599. ACM (2005)
17. Bunn, P., Ostrovsky, R.: Secure two-party k-means clustering. In: Proceedings of the 14th ACM Conference on Computer and Communications Security, pp. 486–497. ACM (2007)
18. Liu, D., Bertino, E., Yi, X.: Privacy of outsourced k-means clustering. In: Proceedings of the 9th ACM Symposium on Information, Computer and Communications Security, pp. 123–134. ACM (2014)

Discovering Vulnerable Functions: A Code Similarity Based Approach

Aditya Chandran, Lokesh Jain^(✉), Sanjay Rawat, and Kannan Srinathan

International Institute of Information Technology, Hyderabad, India
lokeshjain@research.iiit.ac.in

Abstract. This paper extends recent work on vulnerability extrapolation. A surge in vulnerability exploits against old and new softwares, urges the importance of detection of vulnerabilities and possible attacks prior to the attacker. How sophisticated an exploit may be, an underlying prerequisite remains to be the presence of at least one memory corruption bug, serving as *entry point* for the exploit. Therefore several rigorous software testing techniques are borrowed to detect and eliminate software bugs as early as possible. Code similarity based bug detection is one of such techniques, which, in the parlance of software security, is also termed as *vulnerability extrapolation*. In this paper, we present a source code similarity based bug identification technique by considering code features that are relevant for security related bugs. Our technique works by enriching (augmenting) abstract syntax trees (ASTs) of functions by considering security relevant properties of the code. We show the effectiveness of the augmented AST based similarity approach over existing methods by evaluating proposed method on real-world applications.

Keywords: Software vulnerability · Abstract syntax tree · Vulnerability extrapolation · Code similarity

1 Introduction

In software, vulnerabilities are constantly discovered and exploited. Vulnerabilities can be identified or detected by analysing if a system contains flaws that could be triggered and exploited by the attacker. These flaws creep into the software mainly due to inadequate system design and/or improper development practices. They severely affect the security of the system and make it vulnerable to attacks.

Several vulnerabilities in the past have led to major security complications. For example, *Duqu malware*, a malware contained a variety of software components that together provided unauthorized services to the attackers which included information stealing capabilities along with injection tools, etc. The malware used the Duqu flaw in the Windows Operating systems to perform malicious activities. The malware *Stuxnet* is an extreme case that featured code for exploiting four unknown vulnerabilities in the Windows operating system, thereby rendering conventional defense techniques ineffective in practice.

The discovery of vulnerabilities in source code remains a critical issue in the study of system security. New vulnerabilities are being continuously discovered in both open and closed source softwares. Empirical analyses of security alert data from intrusion detection systems indicate that open source software vulnerabilities are at a greater risk of exploitation, diffuse more rapidly, and have greater volume of exploitation attempts [12]. Most recent bugs like *Poodle*, *Shellshock*, *HeartBleed* have remained undetected even in major open source softwares. Several techniques have been proposed for discovery and elimination of the vulnerabilities in code. While techniques derived from software testing, such as fuzz testing [15], taint analysis [14] and symbolic execution [14] have all shown promising results, they are often limited to specific conditions and types of vulnerabilities. Therefore, thorough security auditing is necessary to protect the users of software [7]. However, given the sheer size, complexity and different coding environments of any real world application, manual analysis will suffer the same fate as that of automated analysis techniques. However, as advocated in [7], if an analysis can reduce the search space of the code base, to be analyzed by an analyst, by several times, the problem of scalability can be addressed with more reliable results. One possible solution to achieve such a reduction is by using code similarity [13, 17]. Though code similarity has been used in the past to detect vulnerable functions, its adoption from software engineering practice to security bug detection has not been explored much.

Recent work has used code similarity techniques by considering API symbol usage [18] alone, and thereafter has been improved upon by additionally utilizing a cut-down version of the AST of the code [19]. However, it should be noted that the representation used in the work presented in [19] lacks properties that are important for several *security* related bugs. For example, type of function's parameters, return value types, differentiation of pointer declarations and certain light-weight control-flow properties of the code namely- cyclomatic complexity, number of loops, number of nodes, number of edges, provide *hints* on the presence of bugs. By considering these properties when computing code similarity, enhances the chance to detecting functions that may have similar security bugs.

In AST based representation, a function is represented as a tree, with nodes corresponding to declaration types, functions calls and other syntactical constructs of the code base. This representation enables our method to decompose a known vulnerable function into a vector representation and thereby enabling us to apply vector similarity metric, e.g., cosine metric. By applying a similarity metric for a given vulnerable function *vis-a-vis* rest of the functions in a code base, we can find functions structurally similar to the given vulnerable function which are potentially suffering from the same flaw. Such functions can further be analyzed by an analyst for validity. As is elaborated in Sect. 2, the proposed AST based analysis is an extension to the approach presented in [18] with improved results. We augment AST nodes by incorporating several code properties (mentioned above) that are *normally* ignored by AST representation of the code. We name our approach as AST+ (as it extends the original AST based approach). Figure 1 illustrates the workflow of AST+ approach. For a given codebase the AST is extracted. From the AST of the code the subtrees containing API

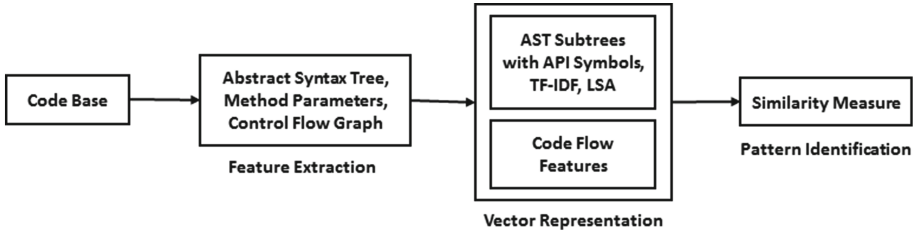


Fig. 1. Schematic diagram of AST+

symbols as leaf nodes are extracted and represented in vector space. The vectors are subjected to dimensionality reduction using Latent Semantic Analysis. The reduced vectors are further augmented with code flow features. These vectors are used to rank the functions based on similarity with a known vulnerable function by finding the cosine similarity of each vector with the vector corresponding to the known vulnerable function.

We evaluate the efficiency of our method using the source code of 6 popular open-source projects: FFmpeg, Pidgin, LibTIFF, mpg123, Tintin++ and ImageMagick. On an average, we are able to produce results with relatively better consistency and accuracy compared to other previous models. We are also able to detect known vulnerable functions that are mentioned in a CVE list but not detect by previous models.

Our contributions can be summarised as follows:

1. **Vulnerability extrapolation using AST+:** We present an approach to extrapolate the vulnerabilities by considering the syntactical properties of code which are represented using the subtrees of the Abstract Syntax Tree of the codebase. We enrich the method by incorporating information targeting those aspects of code which are capable of carrying vulnerabilities such as function parameters, pointer declarations, etc. We also extract the control flow properties such as cyclomatic complexity, no. of loops, no. of nodes and no. of edges from the CFG of the same code and added them to the AST based features to couple code flow based similarity with syntactical similarity.
2. **Evaluation and Case Studies:** We demonstrate that by implementing an abstract syntax tree based code similarity approach, which is further coupled with code flow features, for different open source applications, we are able to identify the existing vulnerabilities mentioned in CVE list.

The rest of the paper is organized as follows. Section 2 deals with Vulnerability Exploitation using AST: AST+. The evaluation of the proposed work is covered in Sect. 3. The related work is covered in Sect. 4. Section 5 deals with Conclusion.

2 Vulnerability Extrapolation Using AST: AST+

Vulnerability detection is one of the main proactive defensive techniques of Computer Security. Vulnerability extrapolation, an approach for vulnerability detection, works on the principle that large source codes very often contain recurring vulnerabilities which are caused by repeated usage of flawed programming pattern and practices. The process of extrapolation is broadly divided into three sub-processes

1. Identification of function that contains the known vulnerability in source code.
2. Based upon the identified function with a known vulnerability we create a list of other functions in the codebase which potentially carry the similar vulnerability.
3. Analyzing for the existence of vulnerability for all functions in the list.

Previous work focuses on extrapolation based on syntactical structure of code using Abstract Syntax Tree [19]. In addition to that, we improve upon the AST based approach presented in previous work by incorporating security specific code properties. AST+ approach is divided into four steps:

1. **Extracting AST for each function in a codebase:** Initially, the AST of the code is extracted for all individual functions. Listing 1.1 and Listing 1.2 shows the function and its AST. AST+ contains a larger number of nodes as compared to AST based representation as described in previous work. This representation provides sufficient information to differentiate between similar API structures but whose variation could lead to the existence or absence of vulnerability. Few such examples include overloaded functions, type of function's parameters and pointers *vs.* non pointer declarations. These differences are handled in detail in the subsequent sections. In this representation, API symbols occur as subtrees rather than individual nodes. In our approach, these sections of the AST are combined to form a single node to represent the API symbol (declarations, function calls, etc.). For each AST corresponding to a function, all subtrees of up to depth 'd' are extracted which contain at least one API symbol node.
2. **Vectorial Representation:** Each function is represented as a vector in order to bring it to a form more suitable for application of Machine Learning techniques. A set of all unique subtrees in the code is created and each function is represented as a feature vector using the subtrees as parameters. The technique of latent semantic analysis (LSA) [4] is applied on the feature vectors to project them in a lower dimensional space.
3. **Identification of Potentially Vulnerable Functions:** The cosine distance between two functions is a measure of their syntactic similarity. It is used to identify functions which are similar to the known vulnerable function. Such functions are likely to be vulnerable.

The above steps are discussed below in detail.

2.1 Extracting AST for Each Function in a Codebase

We parse C code into an AST and utilize this structure for extraction of features. Each function definition is identified by traversing the AST and the subtrees corresponding to function definitions are extracted.

Following mappings are maintained and are used to find the types of a parameter of a function:

1. A mapping of local variables to their datatypes is maintained for each function to assign types for each parameter in a function call. Each variable is stored based on its scope.
2. All global variables are maintained along with their type.
3. A mapping is also maintained to store structural definitions. This maps each element in every structure to its type.

Certain subtrees, which represent an API symbol (declarations, function calls, etc.) are combined to form a single representative node of the API symbol. AST+ Handles API symbols as follows:

1. **Declarations:** The subtree corresponding to a variable declaration node is replaced with the type of the variable being declared. The variable is then added to the local or global variable map as its scope.
2. **Function Definitions:** When a function definition node is encountered, all parameters in the function definition are traversed. The node that represents a function variable is replaced with a node that contains type of corresponding variables. All these variables are added to local variable map.
3. **Function calls:** When a function call node is encountered, the subtree corresponding to the function call is combined to form an api symbol. The variables in the function call are replaced with their types using the global variable map, local variable map and structure definitions list. This is done to differentiate between function calls to different implementations of overloaded functions. Function call parameters are further differentiated for expressions/variables and constants. This is done to differentiate between function calls which use constant parameters and variables as they present an opportunity for existence of a vulnerability.
4. **Pointers:** Pointer declarations are implemented as a variable declaration node with a parent pointer declaration node to indicate a pointer. A variable of this type is mapped to an api symbol which comprises of the base type preceded by pointer to and stored in the corresponding variable list. The subtree corresponding to the pointer declaration node is handled as a single API node.

As mentioned previously, AST+ representation contains a larger set of possible nodes of the Abstract Syntax Tree. This increase in type of nodes is a result of lesser abstraction of statements. This results in a tree with higher information content and directly translates to a superior localised matching of code.

2.2 Vectorial Representation

AST offers a good visual representation of program structure but is not suitable for direct application of ML procedures.

Each function is represented by a vector which allows for flexibility in application of ML techniques. The size of the vector is equal to the number of unique subtrees in all the functions containing at least one api symbol. The number of subtrees can vary according to maximum height of the subtrees. We set the maximum value of height to be 3 in all our experiments as it provides a balanced representation of code. It neither simplifies the subtrees to depend solely on api symbols and simultaneously avoids overly complicated paths which are unlikely to occur more than once.

The value of each feature is the product of number of occurrences of the corresponding subtree in the function with the tfidf weighting of the subtree. LSA is used to analyze the relationship between functions and the subtrees based on the assumption that subtrees which handle similar functionality will tend to occur together in functions. The subtrees which tend to occur together frequently are projected closer in the

new vector space. This results in data with a reduced dimension which is computation friendly and gives more importance to features which help in determining similarity between functions. Experimentally, the best results were obtained with the number of reduced dimensions ranging from 50 to 90. However, we observed that after dimension 60, there was not any significant improvement in the result in any application and as a result, we set the number of reduced dimensions to be 60 for all example codebases.

The vectors are further augmented with four code flow features namely, number of loops, cyclomatic complexity, number of nodes and number of edges. These features are used to measure the similarity of two functions on a code flow basis in addition to syntactical. This augmentation helps in ensuring that the similarity measure is not too heavily dependent only on the basis of occurrence or

Listing 1.1. Example Code

```
int foo(int x) {
    int a = foo2(x);
    int b = foo3(x);
    float f;
    if (a>b) {
        f = foo4(a); }
    else
        f = foo4(b);
    return f;
}
```

Listing 1.2. AST+ of Example Code

```
FileAST:
FuncDef:
  Decl: foo, [], [], []
  FuncDecl:
    ParamList:
      Decl: x, [], [], []
      int
    TypeDecl: foo, []
    IdentifierType: ['int']
  Compound:
    Decl: a, [], [], []
    int
    foo2(int)
    Decl: b, [], [], []
    int
    foo3(int)
    Decl: f, [], [], []
    float
  If:
    BinaryOp: >
    ID: a
    ID: b
    Compound:
      Assignment: =
      ID: f
      foo4(int)
      Assignment: =
      ID: f
      FuncCall:
      foo4(int)
  Return:
  ID: f
```

absence of api symbols and also takes into account the similarity in size, complexity of the two functions. The similarity measure now ensures that functions which have a similar api usage sequence in the code get a much higher similarity measure which directly translates to a better results as the probability of finding a similar vulnerability is higher in such functions.

2.3 Identification of Potentially Vulnerable Functions

By comparing the vectors in the reduced dimensional space, we measured the similarity between two functions using cosine distance. This was chosen to measure similarity as cosine distance generally performs well compared to other metrics for similar problems and for the sake of extending work done previously.

One of the functions which is known to be vulnerable is chosen and the similarity of the rest of the functions with respect to the known function is calculated. The functions are ranked according to their similarity metric with the known vulnerable function. This allows identification of other functions which have a similar API usage pattern which may carry the same vulnerability.

3 Evaluation

We presented a method to extrapolate vulnerabilities based on the syntactical similarity of a vulnerable function to rest of the codebase. This process is used to identify a reduced set of potentially vulnerable functions which are manually analyzed for vulnerabilities. We conduct case studies on codebases of open source libraries like FFmpeg, PIDGIN, MPG123, TinTin++, ImageMagick and LibTiff to empirically show the effectiveness of AST+. We also compare results on FFmpeg, PIDGIN with respect to the results obtained by earlier work [19].

We have implemented the proposed method in python. For each application, we select a known vulnerable function and find k nearest neighbour functions from the application. We experimented with $k = 20, 25, 30, 35$ and in order to select an optimal value of k , we analyzed the results manually to check if targeted vulnerable functions are among these k functions. Based on initial run, we found $k = 25$ to be the best choice. In order to have a fair comparison with results published in [19], we, however, show top 30 neighbors found by AST+.

3.1 Case Study: FFmpeg

Original Vulnerability: In the function *flic_decode_frame_8BPP*, there is no verification that the offsets specified by the video frame refer to locations within the array. This could provide attackers with access to locations outside of the pixel array.

```

1 static int flic_decode_frame_8BPP(AVCodecContext *avctx, void *data, int *data_size,
2 const uint8_t *buf, int buf_size) {
3     int stream_ptr = 0;
4     int pixel_ptr;
5     [...]
6     if (avctx->reget_buffer(avctx, &s->frame) < 0) {
7         av_log(avctx, AV_LOG_ERROR, "reget_buffer() failed\n");
8         return -1;
9     }
10    while ((frame_size > 0) && (num_chunks > 0)) {
11        [...]
12        case FLIDELTA:
13            [...]
14            while (compressed_lines > 0) {
15                stream_ptr += 2;
16                [...]
17            else if ((line_packets & 0xC000) == 0x8000) {
18                pixels[y_ptr + s->frame.linesize[0] - 1] = line_packets & 0xFF;
19            } else {
20                compressed_lines--;
21                pixel_ptr = y_ptr;
22                pixel_countdown = s->avctx->width;
23                for (i = 0; i < line_packets; i++) {
24                    [...]
25                    for (j = 0; j < byte_run; j++, pixel_countdown -= 2) {
26                        [...]
27                    }
28                }
29                default:
30                    av_log(avctx, AV_LOG_ERROR, "Unrecognized chunk type: %d\n", chunk_type);
31                }
32                [...]
33    }

```

```

1 static void rpza_decode_stream(RpzaContext *s)
2 {
3     int pixel_ptr = 0;
4     int stream_ptr = 0;
5     [...]
6     if (s->buf[stream_ptr] != 0x1)
7         av_log(s->avctx, AV_LOG_ERROR, "First chunk byte is 0x%02x instead of 0x1\n",
8             s->buf[stream_ptr]);
9     [...]
10    while (stream_ptr < chunk_size) {
11        [...]
12        case 0xa0:
13            [...]
14            stream_ptr += 2;
15            while (n_blocks--) {
16                block_ptr = row_ptr + pixel_ptr;
17                for (pixel_y = 0; pixel_y < 4; pixel_y++) {
18                    for (pixel_x = 0; pixel_x < 4; pixel_x++) {
19                        [...]
20                    }
21                    block_ptr += row_inc;
22                }
23                ADVANCE_BLOCK();
24                [...]
25            }
26            [...]
27        default:
28            av_log(s->avctx, AV_LOG_ERROR, "Unknown opcode %d in rpza chunk."
29                " Skip remaining %d bytes of chunk data.\n", opcode,
30                chunk_size - stream_ptr);
31            return;
32            [...]
33    }

```

Extrapolation: We apply our methods to extract the top 30 similar functions. As evident from Tables 1 and 2, AST+ finds that all three extrapolated vulnerable functions are detected by AST. Beside this, AST+ is also able to detect a vulnerable function `rpza_decode_stream` (CVE-2013-7009) which was not detected by previous work [19]. The detection of this new function is attributed to the usage of code flow similarity features such as number of loops and cyclomatic complexity. As we can notice in Listing 1.3 and 1.4, the function `rpza_decode_stream`, though not as similar as many other functions in terms

Table 1. FFmpeg (Old AST)

Sim	Function_Name
0.98	flic_decode_frame_15_16BPP
0.92	decode_frame
0.92	decode_frame
0.91	flac_decode_frame
0.90	decode_format80
0.89	decode_frame
0.89	tgx_decode_frame
0.89	vmd_decode
0.89	wavpack_decode_frame
0.88	adpcm_decode_frame
0.88	decode_frame
0.88	aasc_decode_frame
0.88	vqa_decode_chunk
0.87	cmv_process_header
0.87	msrle_decode_8_16_24_32
0.87	wmavoicedecode_init
0.85	decode_frame
0.84	smc_decode_stream
0.84	rl2_decode_init
0.84	xvid_encode_init
0.84	vmdvideo_decode_init
0.83	mjpega_dump_header
0.82	ff_flac_is_extradata_valid
0.82	decode_init
0.82	ws_snd_decode_frame
0.81	bmp_decode_frame
0.81	sbr_make_f_master
0.80	ff_h264_decode_ref_pic_
0.80	decode_frame
0.79	vqa_decode_init

Table 2. FFmpeg (AST+)

Sim	Function_Name
0.949	flic_decode_frame_15_16BPP
0.947	decode_frame
0.945	flic_decode_init
0.920	smc_decode_stream
0.910	decode_frame
0.909	msrle_decode_pal4
0.907	vmd_decode
0.905	vmdvideo_decode_init
0.893	cmv_process_header
0.891	xvid_encode_frame
0.891	vqa_decode_chunk
0.890	rl2_decode_frame
0.884	vqa_decode_init
0.884	vqa_decode_frame
0.882	vmdvideo_decode_frame
0.882	cmv_decode_inter
0.876	rpza_decode_stream
0.873	smc_decode_frame
0.862	vmdaudio_decode_frame
0.859	decode_residuals
0.854	decode_frame
0.851	rl2_decode_init
0.850	cmv_decode_frame
0.829	ff_flac_is_extradata_valid
0.828	aasc_decode_frame
0.826	qdm2_decode_init
0.821	msrle_decode_8_16_24_32
0.819	mjpega_dump_header
0.799	dump_headers
0.779	decode_format80

Table 3. PIDGIN (Old AST)

Sim	Function_Name
1.00	receiveauthgrant
1.00	receiveauthreply
1.00	parsepopup
1.00	parseicon
1.00	generor
0.99	incomingim_ch2_buddylist
0.99	motd
0.99	receiveadded
0.99	mtn_receive
0.99	msgack
0.99	keyparse
0.99	hostversions
0.98	userlistchange
0.98	migrate
0.98	incomingim_ch4
0.98	parse_flap_ch4
0.98	infoupdate
0.98	parserights
0.98	incomingim
0.98	parseadd
0.97	userinfo
0.97	parsemod
0.97	parsedata
0.97	rights
0.97	rights
0.97	uploadack
0.96	incomingim_ch2_sendfile
0.96	rights
0.96	parseinfo_create

Table 4. PIDGIN (AST+)

Sim	Function_Name
1.0	receiveauthgrant
0.997	receiveauthreply
0.979	receiveadded
0.959	incomingim_ch3
0.946	mtn_receive
0.936	evilnotify
0.933	parseadd
0.930	parseicon
0.929	parsedata
0.923	migrate
0.922	keyparse
0.922	clientautoresp
0.921	parse_snac
0.921	aim_chat_readroominfo
0.920	parseinfo
0.919	aim_info_extract
0.918	parsemod
0.916	parse_flap
0.916	incomingim_ch2_buddylist
0.916	parserights
0.915	parse_flap_ch4
0.913	incomingim_ch2
0.910	parseinfo_create
0.910	parseack
0.910	userlistchange
0.909	incomingim_ch4
0.906	infoupdate
0.906	flap_connection_destroy_cb
0.904	missedcall
0.903	aim_locate_adduserinfo

of AST subtrees, does possess many nested loop structures similar to that of `flic_decode_frame_8BPP`. There are also many API symbols common to both functions such as type declarations and multiple usages of the function `av_log` which contributes to syntactical similarity. As both functions have similar loop structure and also common type declaration and function declaration, AST+ is able to capture the similarity more accurately than original AST based method.

Table 5. Results of AST+ on remaining 4 applications. 2nd column shows the example vulnerable function, used for that application; and 3rd column reports the vulnerable functions found by AST+. A * after the function name represents the fact that the function was not listed in CVE list but is potentially vulnerable on the basis of manual analysis

Application name	Example function	Vulnerable function
TinTin++(1.97.9)	add_line_buffer() (CVE-2008-0671)	process_chat_input()(CVE-2008-0672), DO_CHAT()(CVE-2008-0673), buffer_f()*(CWE-119:CWE-120)(CWE-190)
MPG123(0.59r)	readstring() (CVE-2003-0865)	find_next_file(CVE-2004-1284), http_open()(CVE-2007-0578), init_input()*(CWE-120)(CWE-126), url2hostport()*(CWE-120)(CWE-190), getauthfromURL()(CVE-2004-0982)
LibTiff(3.8.1)	TIFFReadDirectory() (CVE-2012-2088)	EstimateStripByteCounts()(CVE-2006-3463), LZWDecode()(CVE-2008-2327), LZWDecodeCompat()(CVE-2008-2327), cvt_whole_image()(CVE-2009-2347), t2p_read_tiff_init()(CVE-2012-3401), readgifimage()(CVE-2013-4243)
ImageMagick(6.2.8)	ReadDIBImage() (CVE-2007-4988)	ReadSGIImage()(CVE-2006-4144), DecodeImage()(CVE-2006-3744), ReadDCMImage()(CVE-2007-4986), WriteDIBImage()(CVE-2007-4986), ReadXBMImage()(CVE-2007-4986), ReadXCFImage()(CVE-2007-4986), ReadXWDImage()(CVE-2007-4986), WriteXWDImage()(CVE-2007-4986), ReadPSDImage()*(CWE-119:CWE-120)

3.2 Case Study: PIDGIN

Original Vulnerability: The function *receiveauthgrant* performs insufficient validation of UTF-8 strings received from network which could result in Denial of Service attacks or execution of arbitrary code.

Extrapolation: We apply AST+ based approach to obtain the top 30 most similar functions from each method. We find that all 8 unknown vulnerable functions mentioned in [19] are detected with better rankings for vulnerable functions as shown in Tables 3 and 4.

As the remaining 5 applications were not analyzed by [19], we summarize the evaluation of AST+ on those 5 applications in Table 5.

4 Related Work

Detecting flaws and bugs in code has always been a crucial aspect of software engineering to ensure a vulnerability-free system i.e. the system which does not leak information and is attack-proof. Many vulnerabilities are being discovered every day and consequently development of tools and techniques to counter

vulnerabilities is also a continuous process. Vulnerability analysis of a system can be done using both static and dynamic tools.

Some static analysis tools are splint [6], cppcheck [1], PScan [3], etc. All these tools help identify several well known vulnerabilities. Though the importance of these tools cannot be denied, they are limited to identifying only specific types of vulnerability and are thus unable to identify complex or subtle vulnerabilities. Apart from these tools, there have been studies on detecting vulnerability using API usage pattern so as to observe the content usage of the code which often plays a vital role in identifying vulnerability. Some static analysis tools are built on such concepts namely Flawfinder [2], ITS4 [16], etc. However, these tools and method are limited to their database. Thus vulnerabilities which have an unknown API usage pattern cannot be detected.

Some studies are based on the hypothesis that most of the code in a system is replicated, thereby increasing the chances of several functions containing a similar if not same vulnerability. Many research papers work based on this hypothesis [8]. There also exist tools and evaluation techniques for detecting such functions in code [5, 9, 10].

A new approach followed the above work where vulnerabilities are extrapolated [18]. In this approach function usage and type names are utilized to find similar functions which are candidates for being vulnerable. This method does not consider the syntax and structural pattern of the code. Thus this method is limited to detect bugs related to particular API symbols.

To overcome the above limitation further research has been done in this area by the same author [19]. This method utilises AST of the code which considers syntax and structural pattern of the code. Thus they are able to detect vulnerabilities that are not only related to API usage but also the structure of the code. With the similar idea of using AST we proposed an improved approach which uses involves usage of AST subtrees, and in turn uses more extensive structure of code (of up to depth d), functional argument types, along with the API usage pattern of code.

Another area of research adopts a more dynamic approach for detecting vulnerability. Methods such as Fuzzing, dynamic taint analysis [11, 14] have been developed to address the limitations of static code analysis. There are several cases in which vulnerabilities cannot be detected within code itself, but get triggered with some types of input during run time. These methods specializes in detecting vulnerability during execution Though the aim is similar i.e. finding vulnerabilities, there are fundamental differences in approaches and therefore, we find it futile to provide any meaningful comparison.

5 Conclusions

Detecting and patching the vulnerabilities in a code base is a key to strengthen the security of computer systems where a small loophole could result in loss or theft of critical data or denial of critical services. Due to large code bases, it has become essential to develop an automated system that can identify and

analyse some vulnerability patterns thereby reducing the huge search space for identifying software vulnerabilities.

Our method utilizes content and structural patterns of the code to extrapolate vulnerability and find similar bugs in an application. Our method identifies the functions which are candidates for being potentially vulnerable functions. Thus assists the security analyst to audit these potentially vulnerable functions.

Overlap in the vulnerable functions identified by our method with the functions listed in CVE database, shows the effectiveness of our method. Our method can also be combined with other vulnerability detection techniques. If a new vulnerability is identified by utilising any of the existing vulnerability detection techniques, it can be extrapolated to the entire code base thus identifying other instances of the same or similar vulnerabilities. By using our technique one can immediately patch similar kind of flaws existing within the application efficiently.

References

1. cppcheck. <http://cppcheck.sourceforge.net/>
2. Flawfinder. <http://www.dwheeler.com/flawfinder/>, d. A. Wheeler
3. Pscan: a limited problem scanner for c source files. <http://deployingradius.com/pscan>, a. Dekok
4. Deerwester, S., Dumais, S.T., Furnas, G.W., Landauer, T.K., Harshman, R.: Indexing by latent semantic analysis. *J. Am. Soc. Inf. Sci.* **41**(6), 391 (1990)
5. Ducasse, S., Rieger, M., Demeyer, S.: A language independent approach for detecting duplicated code. In: *IEEE International Conference on Software Maintenance 1999 (ICSM 1999) Proceedings*, pp. 109–118. IEEE (1999)
6. Evans, D., Larochelle, D.: Improving security using extensible lightweight static analysis. *IEEE Softw.* **19**(1), 42–51 (2002)
7. Heelan, S.: Vulnerability detection systems: think cyborg, not robot. *IEEE Secur. Priv.* **9**(3), 74–77 (2011)
8. Kapser, C., Godfrey, M.W.: Toward a taxonomy of clones in source code: a case study. In: *Proceedings of the Conference on Evolution of Large Scale Industrial Software Architectures (ELISA 2003)*, pp. 67–78 (2003)
9. Kontogiannis, K.A., Demori, R., Merlo, E., Galler, M., Bernstein, M.: Pattern matching for clone and concept detection. In: *Reverse Engineering*, pp. 77–108. Springer (1996)
10. Li, Z., Lu, S., Myagmar, S., Zhou, Y.: CP-Miner: finding copy-paste and related bugs in large-scale software code. *IEEE Trans. Softw. Eng.* **32**(3), 176–192 (2006)
11. Newsome, J., Song, D.: Dynamic taint analysis for automatic detection, analysis, and signature generation of exploits on commodity software. In: *NDSS*. IEEE (2005)
12. Ransbotham, S.: An empirical analysis of exploitation attempts based on vulnerabilities in open source software. In: *WEIS* (2010)
13. Rawat, S., Mounier, L.: Finding buffer overflow inducing loops in binary executables. In: *2012 IEEE Sixth International Conference on Software Security and Reliability (SERE)*, pp. 177–186. IEEE CSP (2012)
14. Schwartz, E.J., Avgerinos, T., Brumley, D.: All you ever wanted to know about dynamic taint analysis and forward symbolic execution (but might have been afraid to ask). In: *2010 IEEE symposium on Security and Privacy (SP)*, pp. 317–331. IEEE (2010)

15. Sutton, M., Greene, A., Amini, P.: *Fuzzing: Brute Force Vulnerability Discovery*. Pearson Education, Upper Saddle River (2007)
16. Viega, J., Bloch, J.T., Kohno, Y., McGraw, G.: ITS4: a static vulnerability scanner for C and C++ code. In: 16th Annual Conference on Computer Security Applications, 2000 (ACSAC 2000), pp. 257–267. IEEE (2000)
17. Williams, C.C., Hollingsworth, J.K.: Automatic mining of source code repositories to improve bug finding techniques. *IEEE Trans. Softw. Eng.* **31**(6), 466–480 (2005)
18. Yamaguchi, F., Lindner, F., Rieck, K.: Vulnerability extrapolation: assisted discovery of vulnerabilities using machine learning. In: *Proceedings of the 5th USENIX Conference on Offensive Technologies*, p. 13. USENIX Association (2011)
19. Yamaguchi, F., Lottmann, M., Rieck, K.: Generalized vulnerability extrapolation using abstract syntax trees. In: *Proceedings of the 28th Annual Computer Security Applications Conference*, pp. 359–368. ACM (2012)

Performance Analysis of Spectrum Sensing Algorithm Using Multiple Antenna in Cognitive Radio

Komal Pawar^(✉) and Tanuja Dhope

G.H.R.C.E.M, Pune, Maharashtra, India

pwrkomal2@gmail.com, tanuja_dhope@yahoo.com

Abstract. There is a rapid growth and development of new wireless devices and applications therefore, there's a need for more wireless radio spectrum. Cognitive radio is used to solve spectrum scarcity and meet the ever increasing demand of the spectrum using dynamic spectrum access. In context to cognitive radio, spectrum sensing is found to be a crucial task. In this paper, we have evaluated the energy detection technique for spectrum sensing using time domain and frequency domain- periodogram method. For performance comparison parameters like the probability of detection, probability of the missing signal detection and the false alarm probability for change in values of signal to noise ratio, sample count are taken into consideration. Multipath and shadowing effects are reduced using multiple antenna techniques which provide better bit error rates.

Keywords: Spectrum sensing · Energy detection · Multiple antenna

1 Introduction

As the number of users, data rates, wireless devices and its applications are growing rapidly, there is a demand for a large amount of the radio spectrum [1]. The availability of the spectrum is a limited, therefore it is necessary to efficiently utilize the whole Spectrum [2]. The static frequency allocation scheme is found to be not effectively utilized and since the spectrum is a limited source, new spectrum allocations are difficult. Cognitive radio (CR) technology was introduced which uses the dynamic frequency allocation scheme and solves the problem of spectrum scarcity [3]. In CR, the spectrum is checked continuously, which is used by licensed user/primary user and if found to be idle then it is given to the unlicensed user/secondary user. Matched filter detection (MFD), Energy detection (ED), and Cyclostationary (CFD) are some of the spectrum sensing technique [2, 4]. Amongst these, the energy detection is a semi-blind detection method used for detection of an unknown signal in additive noise [5]. It is advantageous over the rest of methods as it requires no previous information of the primary received signal also it has less complexity [6]. Matched filter requires accurate synchronization and prior information of the signal. The cyclostationary method requires knowledge of cyclic frequencies of the primary user. It is less prone to noise uncertainty, and provides better detection at low SNR regimes and it requires less signal samples [6, 7]. In wireless channels fading, shadowing makes the spectrum sensing a difficult task.

The Neyman-Pearson criteria states that the spectrum sensing is a two hypotheses sensing problem. Such as given below,

$$\text{Hypotheses 0 : } Y(n) = w(n) \tag{1}$$

$$\text{Hypotheses 1 : } Y(n) = s(n) + w(n) \tag{2}$$

where $s(n) = hr(n)$, h is the channel gain, $w(n)$ is the noise sample with mean zero and variance $2\sigma_w^2$. H_0 = Absence of the user, H_1 = Presence of the user.

Multiple antenna technique is currently used for effective communication and reliable signal transmission. Two-stage sensing method provides efficient utilization of multiple antennas with respect to sensing time and hardware [8]. The performance evaluation of ED using Rayleigh fading channels and the unknown deterministic signal is proposed in [9, 10]. The multiple antenna OFDM scheme along with the square law combining technique for energy detection provides higher performance than a single antenna [10]. Overall, the multiple antenna method increases spectrum efficiency and improves system performance. Below table reflects the comparison of various spectrum sensing techniques (Tables 1 and 2).

Table 1. Comparison of the spectrum sensing techniques

Types	Matched filter	Cyclostationary	Energy detection
Pros	It requires less signal samples	It is more robust to noise variance	Easy to implement
Cons	Requires accurate data of primary signal	Cyclostationary features needs to be associated with primary signal	High false alarm due to the noise uncertainty

Table 2. Related work

Year	Author	Title	Techniques used
2004	Cabric, D., S. Mishra & R. Brodersen [10]	Implementation Issues In Spectrum Sensing For Cognitive Radio	Discusses the implementation challenges for CR design. Identifies two key issues: (1.) Dynamic range reduction (2.) Wideband frequency agility
2007	Digham, F., M.Alouini & M. Simon [11]	On The Energy Detection of Unknown Signals Over Fading Channels	Discusses the ED problem for unknown signals over the multipath channel
2008	Hamed Sadeghi, Paeiz Azmi [12]	A Noval Primary User Detection Method For Multiple Antenna Cognitive Radio	To enhance reliability, multiple antenna receiver with cyclostationary detection are presented

(Continued)

Table 2. (Continued)

Year	Author	Title	Techniques used
2009	Ben Letaif, K. Wei Zhang [13]	Cooperative Communication For Cognitive Radio Networks	Cooperative sensing deals with the hidden terminal problem. Various solutions to address the fading/shadowing effects are presented.
2010	Robert Lopez Valcarce and Gonzalo Vazquez-Vilar & Joseph Sala [14]	Multiantenna Spectrum sensing for Cognitive Radio: overcoming noise uncertainty	Based on approximated GLR, a novel detector is proposed.
2011	Suman Rathi, Rajeshwar Lal Dua, Parmender Singh [15]	Spectrum Sensing In Cognitive Radio Using MIMO Technique	Co-operative and non-cooperative spectrum sensing techniques are discussed. Multiple antennas are described in detail with mathematical calculations.
2013	Atapatta S., Tallambura C., Jiang H [16]	Energy Detection For Spectrum Sensing In Cognitive Radio	Analysis is done with multipath fading and shadowing. Two strategies are used:-data fusion and decision fusion.
2013	Ashish Bagwari, Geetam Singh Tomar, and Shekhar Verma, [17]	Cooperative Spectrum Sensing Based on Two-Stage Detectors With Multiple Energy Detector and Adaptive Double Threshold in Cognitive Radio Networks	The performance comparison of the proposed technique along with the cyclostationary and adaptive scheme is done
2014	Wang Yong-feng, Li Ou [18]	Multi-Channel Coordinated Spectrum Sensing Strategy in Multiple Antennae Based Cognitive Radio Networks	Two methods are used: 1. Branch and Bound 2. Greedy heuristic

2 Energy Detection

One of the most generic and common method for spectrum sensing is ED [11]. Received signal energy is measured over a certain time interval T. Time domain and frequency domain is used for implementation of ED. The frequency domain-ED method requires FFT computations. As seen from Fig. 1, ED requires band pass filter BPF, ADC, squaring law device and integrator.

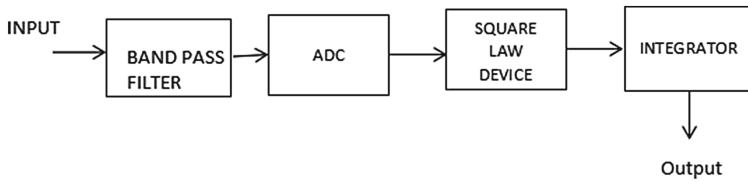


Fig. 1. Energy detection

Initially, the primary signal is a band limited using band pass filter and then squared. The integrated signal is obtained for some time period T. The result is compared against the threshold to determine if the spectrum is vacant [12]. The primary signal S(n) is given as input to the energy detector.

The test static can be represented as,

$$\epsilon_{\text{time}} = \sum_{n=1}^N |S(n)|^2 \tag{3}$$

The result of test static and threshold (λ) are compared to detect availability of primary signal.

$$\lambda \leq \epsilon_{\text{time}} \tag{4}$$

If the threshold is higher than the test static, then the signal is available. Various fading channels are applied to the energy detector. While considering the performance of ED the detection probability and false alarm probability plays a vital role. Formulae for both can be obtained as below [12]: The false alarm probability Pfa is given below

$$P_{fa} = P \left\{ Y > \frac{\lambda}{H_0} \right\} = Q_m \left(\sqrt{2y}, \sqrt{\lambda} \right) \tag{5}$$

The detection probability Pd is given below

$$P_d = P \left\{ Y > \frac{\lambda}{H_1} \right\} = \Gamma \left(n, \frac{\lambda}{2} \right) / \Gamma(n) \tag{6}$$

2.1 Periodogram

This is a discrete fourier transform based method which is used for spectral estimation. Directly from the signal, the power spectral density (PSD) is obtained. PSD of the primary input signal is achieved by taking magnitude square of the corresponding

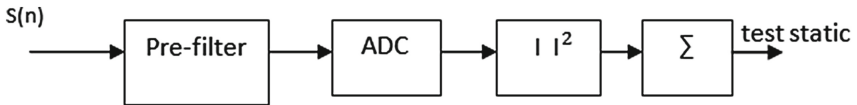


Fig. 2. Energy detection with Periodogram

FFT signal. The periodogram method considers a finite word sequence for estimation of parameters. This technique in time domain multiplies the signal with the rectangular window [12]. Due to the sudden change in signals, frequency response cause, spectral leakage due to undesirable side lobes. The word periodogram is used to determine the hidden periodicities in time series. FFT is used instead of DFT for simulations (Fig. 2).

$$S(k) = \sum_{n=1}^N S(n)e^{-j2\Pi(\gamma-1)(n-1)/N} \tag{7}$$

Where $S(n)$ is the discrete received signal, $N = \text{FFT size}$. Then we apply $S(k)$ to an energy detector as follows

$$\epsilon_{\text{periodogram}} = \frac{1}{N} \sum_{k=1}^N |S(k)|^2 \tag{8}$$

The above equation shows that $S(K)$ is applied to the summation of N component, hence the variance of the statistics varies with respect to FFT size. In order to mitigate this fluctuation, we divide the statistics with the FFT number in order to hold the variance constant.

2.2 Welch’s Periodogram

It is the modified form of periodogram which divides the data sequence into segments along with the windowing technique. The hamming window method is used to calculate periodogram for every segment. In this method these data segments can be either overlapping or non overlapping [13]. In the beginning, the input data sequence is down-converted and passed through the LPF. Later, the data stream is partitioned into segments. These segments are processed using FFT. After FFT, samples are sent to the square-law device. Then L samples are taken from these M segments, followed by a summation of the L samples. Finally, output values are compared with the threshold and corresponding decision for the presence of a signal is done (Fig. 3).

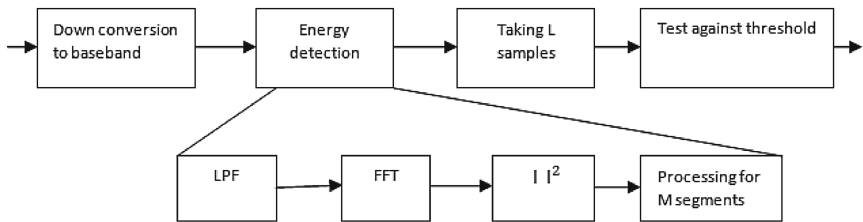


Fig. 3. Energy detection using Welch’s Periodogram

$$w = [w_1(p)w_2(p) \dots w_M(p)] \tag{9}$$

$$w_1(\gamma) = \sum_{p=1}^N w_1(p)e^{-j2(Y-1)(p-1)/N} \tag{10}$$

The welch periodogram is obtained by following equation:

$$\epsilon_{welch} = \sum_{\gamma=1}^L \sum_{l=1}^M \frac{1}{N} |w_l(p)|^2 \tag{11}$$

3 Multiple Antennas

In wireless communication, multiple antenna technique has been a boon to the communication world. As the name suggests there are multiple antennas at transmitter and receiver side which provide better channel capacity and higher data rates as compared to the single antenna. The multiple antennas are usually classified into following three types: SIMO, MISO and MIMO. Multiple antennas in context to MIMO provide improved BER, it increases the folded channel capacity, lowers the sensitivity to fading. The MIMO communication system supports greater throughput by exploring the multiple properties of the channel. In MIMO, multiple channel data traffic is applied through the antennas. It uses signal processing techniques to obtain information of waveform and the corresponding data stream. Incorporation of MIMO significantly improves the data transmission rate, thereby enhancing the performance of a system.

A general block diagram for a MIMO model is shown in Fig. 4. The wireless communication network has more than one transmitter and receiver antennas. Expression for the MIMO channel model is given through its input output relationship as below,

$$Y = HS + W \tag{12}$$

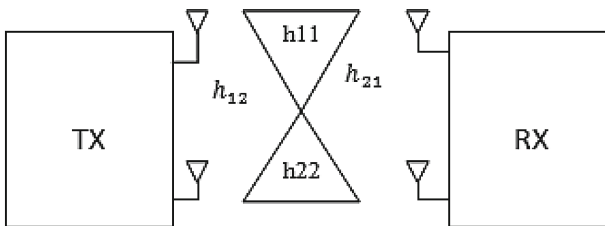


Fig. 4. MIMO system model.

H is channel matrix having independent elements. S is signal and W is the noise that gets added to the signal.

Channel matrix is as follows,

$$H = \begin{pmatrix} h_{11} & h_{12} & \dots & h_{1N_T} \\ \vdots & \vdots & & \vdots \\ h_{N_R 1} & h_{N_R 2} & \dots & h_{N_R N_T} \end{pmatrix} \tag{13}$$

Advantage:

- Higher data rates
- Low interference

4 Simulation Results

In this section, we illustrate numerical results for performance evaluation in the time domain ED. Simulation results show energy detection in the time domain using BPSK modulation under AWGN channel. The simulation study shows comparison of detection probability (Pd), miss detection probability (PMD), and false alarm probability (PFA) with respect to SNR.

Figure 5 shows the missing signal detection versus SNR for varying false alarm probability. It indicates that probability of the missing signal gets lower with high false alarm. We have considered two values of PFA i.e. when PFA = 0.5 and 0.8. Here sample count N, is assumed to be 100.

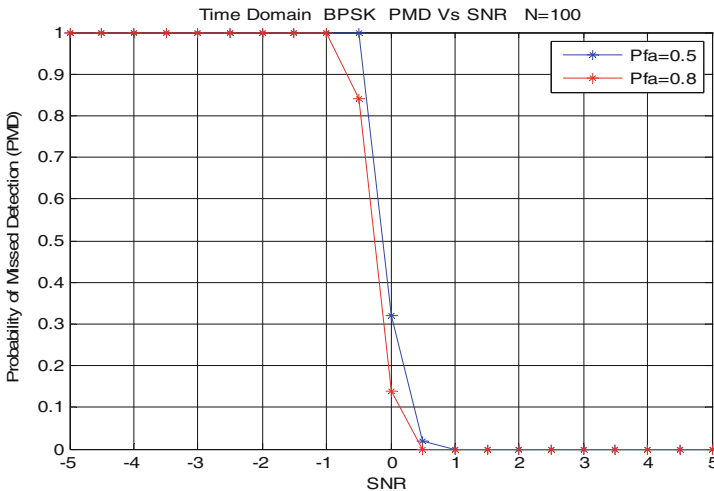


Fig. 5. PMD Vs SNR for varying Pfa

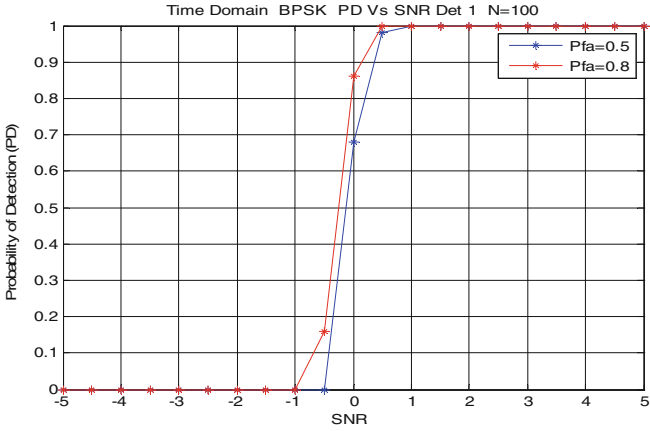


Fig. 6. PD Vs SNR for varying Pfa

Similarly Fig. 6 shows detection probability Vs SNR for changing false alarm probability. Detection probability is measured for each interval of time & plotted against Pfa to observe the effect of Pfa on detection probability.

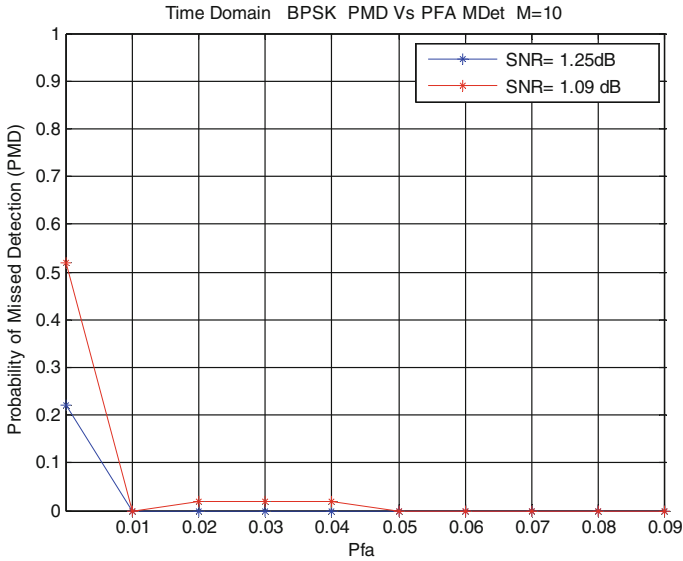


Fig. 7. PMD Vs Pfa for fixed SNR

Figure 7 illustrates the Probability of the missing signal against false alarm probability where SNR and sample count are constant. Values are fixed at SNR = 1.25 db and SNR = 1.09 db, M is no of segments M = 10.

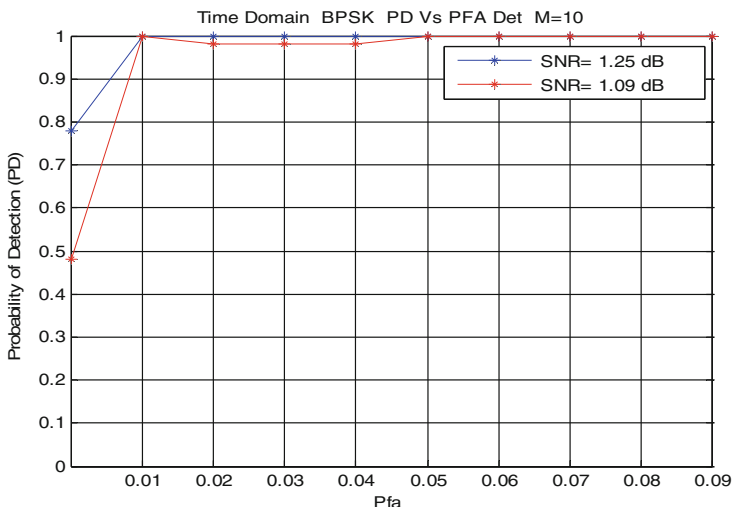


Fig. 8. PD Vs Pfa for fixed SNR

Figure 8 show detection probability versus probability of false alarm at fixed SNR. Similar values are used for simulation. SNR = 1.25 db, SNR = 1.09 db, M = 10.

In Figs. 7 and 8, Pd and Pmd are plotted against Pfa at fixed SNR which indicates that as signal strength increases detection probability increases and missing probability decreases. To achieve better signal detection for the energy detector, detection probability should be high, the probability of missing signal detection be low having low false alarm probability.

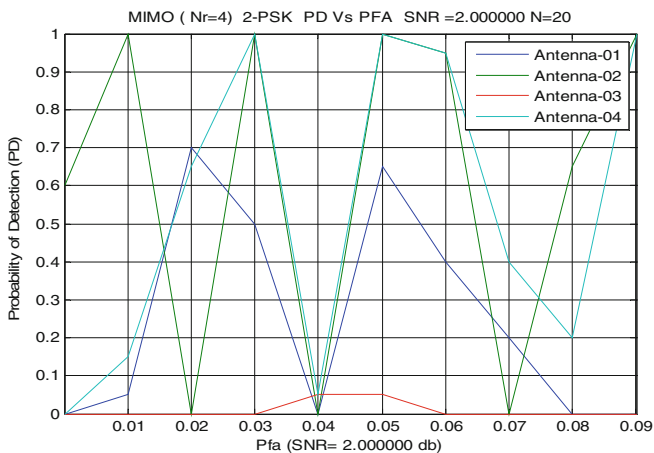


Fig. 9. Multiple antenna selection combining

From Fig. 9 we observe that 4 antennas are present and amongst them, one antenna is chosen for communication purpose. The simulation is carried out with 2-psk for the detection probability versus false alarm probability. Here SNR = 2.0 and N = 20.

Antenna channel Gains are
 0.814724 0.905792 0.126987 0.913376

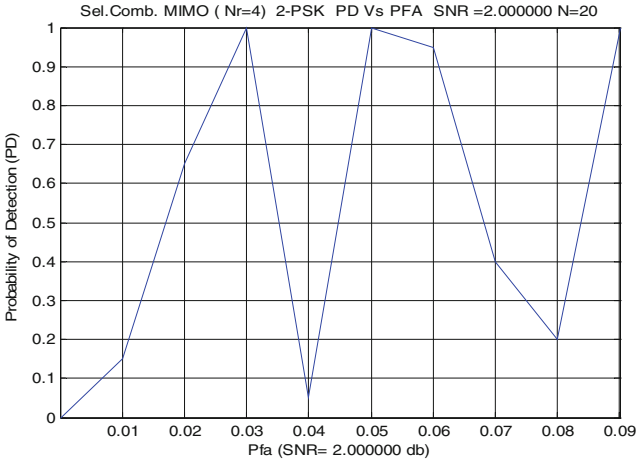


Fig. 10. Multiple antenna selection combining with max gain

The antenna with the maximum channel gain is chosen. The detection probability of this antenna is plotted separately for the observation purpose in Fig. 10.

Selecting Max gain Antenna-04 with gain = 0.913376

5 Conclusion

Cognitive radio is emerging as the need for today’s generation where efficient spectrum utilization and quality of service are in demand. Under this paper, we analyze the performance of ED using spectrum sensing for CR under AWGN. The detection is carried out in the time domain with a single antenna and multiple antennas. Multiple antennas are used to overcome the noise uncertainty. The energy detector is the simplest way for the performance of CR as it does not rely on any previous information of the signal. Simulation results demonstrate considerable detection performance gain.

Acknowledgement. I hereby acknowledge my advisor Prof. Tanuja S. Dhope for guiding me in the right direction and also provide sincere gratitude to G.H.R.C.E.M, PUNE for providing framework to accomplish our work.

References

1. Maleki, S., Pandharipande, A., Leus, G.: Energy-efficient distributed spectrum sensing for cognitive sensor networks. In: IECON (2009)
2. Kaligineedi, P., Khabbaziyan, M.: Malicious user detection in a cognitive radio cooperative sensing system. *IEEE Trans. Wirel. Commun.* **9**(8), 2488–2497 (2010)
3. Haykin, S.: Cognitive radio: brain-empowered wireless communications. *IEEE J. Sel. Areas Commun.* **23**, 201–220 (2005)
4. Dhope, T., Simunic, D., Kerner, A.: Analyzing the performance of spectrum sensing algorithms for IEEE 802.11 of standard in cognitive radio network. *Stud. Inf. Control* **21**(1), 93–100 (2012)
5. Ganesan, G., (Geoffrey) Li, Y.: Agility improvement through cooperative diversity in cognitive radio. In: School of Electrical and Computer Engineering Georgia Institute of Technology, Atlanta, Georgia, pp. 30332–30250
6. Urkowitz, H.: Energy detection of unknown deterministic signals. *Proc. IEEE* **55**, 523–531 (1967)
7. Pandhari, A., Linnartz, J.: Performance analysis of primary user detection in multiple antenna cognitive radio. In: Proceedings of the IEEE International Conference on Communication (ICC 2007), pp. 6482–6486 (2007)
8. Lee, J.-H., Baek, J.-H., Hwang, S.-H.: Collaborative Spectrum Sensing using Energy Detector in Multiple Antenna System. Department of Electronics Engineering, Dongguk University, Seoul
9. Thesis on: Spectrum sensing techniques for Cognitive radio systems with multiple Antennas. by Refik Fatih ÜSTOK Submitted to the Graduate School of Engineering and Sciences of İzmir Institute of Technology, June 2010
10. Cabric, S.D., Mishra, S.M., Brodersen, R.W.: Implementation issues in spectrum sensing for cognitive radios. In: Proceedings of Asilomar conference on Signals, Systems, and Computers, vol. 1, pp. 772–776, 7–10 November 2004
11. Digham, F.F., Simon, M.K.: On the energy detection of unknown signals over fading channels. *IEEE Trans. Commun.* **55**(1), 21–24 (2007)
12. Sadeghi, H., Azmi, P.: A novel primary user detection method for multiple antenna cognitive radio. In: International Symposium on Telecommunications, pp. 188–192 (2008)
13. Letaif, B., Zhang, K.W.: Co-operative communication for cognitive radio networks. *Proc. IEEE* **97**(5), 878–893 (2009)
14. Lopez-Valcarce, R., Vazquez-Vilar, G., Sala, J.: Multiantenna spectrum sensing for cognitive radio: overcoming noise uncertainty. In: IAPR Workshop on Cognitive Information Processing, pp. 310–315. IEEE (2010)
15. Rathi, S., Dua, R.L., Singh, P.: Spectrum sensing in cognitive radio using MIMO technique. *Int. J. Soft Comput. Eng. (IJSCE)* **1**(5), 259–265 (2011). ISSN: 2231–2307
16. Atapattu, S., Tallambura, C., Jiang, H.: Energy Detection for Spectrum Sensing in Cognitive Radio. Springer, New York (2014)
17. Bagwari, A., Tomar, G.S., Verma, S.: Cooperative spectrum sensing based on two-stage detectors with multiple energy detectors and adaptive double threshold in cognitive radio networks. *Can. J. Electr. Comput. Eng.* **36**(4), 172–180 (2013)
18. Yong-feng, W., Ou, L.: Multi-channel coordinated spectrum sensing strategy in multiple antennae based cognitive radio networks (2014)

Diagnosis of Multiple Stuck-at Faults Using Fault Element Graph with Reduced Power

T.S. Gokkul Nath, E.R. Midhila^(✉), Ashwin Swaminathan,
Binitaa Lekshmi, and J.P. Anita

Chennai, India
midhila.er04@gmail.com

Abstract. As the manufacturing processes become more and more advanced as per Moore's law, precise control of silicon process is becoming more and more challenging. This increases the probability of defects and has brought a necessity for testing to ensure fault-free products, making the testing of a chip more complex causing testing challenges.

With large number of transistors, in multiples of thousands being integrated in one chip, multiple stuck-at faults may exist, because of which fault masking and reinforcing effects may come into effect. This may lead to the failure of approaches like Single Location at a Time (SLAT) and restricted single sensitized paths. To counter this, the notion of fault element is used to take into account multiple fault models and use a fault element graph (FEG) to consider fault masking and reinforcing effects among multiple faults. To identify these faults, appropriate test patterns need to be generated that would carry the effect of the fault to the primary output. The test patterns are chosen such that switching power is made to be a minimum.

Keywords: Fault · Fault element graph · Fault location · Test pattern · Failing pattern · Switching power

1 Introduction

Fault diagnosis plays a very important part in the industry, since after each fabrication level there are many irregularities which could occur and with every level, cost of testing increases. A fault can be diagnosed either using Physical failure analysis (PFA) or the software based fault diagnosis tools that can provide definite candidate faults. Single faults have been well studied in the literature [1]. However when multiple faults exist, a fault model fails to accurately locate the actual faults [2]. Results of [3], which use SLAT patterns to detect faults show that almost 41 % of the faults are not diagnosed accurately using single stuck-at fault models.

Previous work on multiple combinational logic faults show that faults can be grouped into two main categories: Diagnostic test-pattern based [4] and Manufacturing test-pattern based [5–15]. The first category deals with finding failing patterns and then further diagnostic test patterns, which distinguishes candidate faults. This process is repeated after each step of identifying the probable candidate locations. Though this method may give high diagnosis quality, it is a costly method to run the test many

number of times. The second category does not generate diagnostic patterns but compares failing responses under test patterns with the failing responses of potential faults and then arrives upon different candidate locations.

In this paper, a manufacturing test pattern based diagnosis method is proposed which also considers the fault masking and reinforcing effect using fault element graphs (FEG). Fault Masking is an occurrence, in which one defect prevents the detection of another. Fault reinforcing effect is when an effect of a fault is nullified because of the presence of another fault. The approach is unique to previous approaches since it considers multiple faults' effect by considering multiple faults at a time rather than single fault at a time to determine multiple faults. After FEG has been constructed, fault elements are iteratively identified with pruning, until all the fault locations are identified and FEG's have been reduced to an empty set or null set.

The next section of the paper discusses in detail about the process of FEG followed by the experimental results.

2 Diagnosing Multiple Faults with Reduction in Switching Power

2.1 Simulator

A true value simulator is constructed by operating the circuit i.e. its gate functions 'L' number of times for each and every possible input patterns. The circuit is then injected with some faults at certain nodes of the circuit. This faulty circuit is again operated the same 'L' number of times as mentioned previously for all input patterns. The outputs received for both these simulators are stored for further processes.

2.2 Test Pattern Generator

The outputs that we receive for the true value and faulty value simulator for each pattern are compared. The patterns, for which the output values for both the simulator are different, will be a test pattern for the injected set of faults. This comparison is done for all the possible input patterns and the ones that can detect a fault are classified as test patterns and are stored separately for its use in further processes.

2.3 Selection of Test Patterns

The test patterns are arranged in all possible combinations and the switching power is found for all of them using Synopsys Primetime power tool. The data received from above is consolidated in ascending order and the first combination of test patterns is chosen for FEG construction.

2.4 FEG Construction

A Fault Element Graph is constructed by back tracing the circuit from the faulty primary output to the primary input. In the FEG, each vertex corresponds to a fault element. A fault element is represented as $l/v/p$ (value v at a location l under a pattern p). The value v inside each node in the graph is the possible faulty value that led to the incorrect output. The directed lines represent the relation between each fault element. For the faulty circuit in Fig. 3, the FEG is constructed by tracing backwards from the faulty output y , z and z' in Fig. 3, x and z' from Fig. 4 and y , z and z' from Fig. 4 for test patterns abcdefg: 0111000, 0000000 and 1110000 respectively. The next level of fault elements in the FEG consists of the nodes that are interconnected with the outputs and the value ' v ' for those fault elements will have the possible faulty value that led to the incorrect machine value at its previous level. In the given example in Fig. 3, for the output node n to have a faulty value 0, either one of the inputs i.e. node e or node f should be 1. Therefore the fault elements at the level following the incorrect output node ' n ' are $e/1/p1$ and $f/1/p1$ (Fig. 1).

In the case of the output NOR gate with output node ' z ', the fault elements present in the next level is only $p/0/p1$. This is because, according to the circuit given in Fig. 2, for the output value ' z ' to have a faulty value of 1, only the value at ' p ' has to be incorrect since it is the only input node to the gate that has the dominant value i.e. 1 (for NOR). This process is continued for the subsequent levels until primary inputs are reached. Once this stage is reached, check if a change in the primary input value to the faulty value given in the FEG, changes the value at any of the primary outputs that has the good machine value i.e. x in Fig. 3, y and z in Fig. 4 and x in Fig. 5. If it does, then the affected good machine-primary output nodes needs to be present in the graph at the same level as the primary input. For example, in Fig. 5, because of the fault element $c/0/p3$, when ' c ' is given the value 0, node ' x ' gets incorrect value 1, which should not happen. Hence, in order to ensure that the correct fault locations are found, fault element $x/0/p3$ is mentioned in the FEG at the same level as $c/0/p3$.

2.5 FEG Scoring

Once the construction of FEG is completed, it is scored. The score of the fault element is mentioned below the fault element label in each node and it represents the contribution of the fault element to the faulty output. For the fault elements at the first level of the FEG, the score is $1/(\text{Total number of faulty outputs})$. In the example given in Fig. 5, the total number of faulty output is 3. Therefore, the score $1/3 = 0.33$ is given to $y/1/p3$, $z/0/p3$ and $z'/1/p3$. The next level of fault elements in the FEG either gets the same score as that of its previous level or a score that is summation of the scores at the previous level or else it can have the score when divided by the number of fault elements at the present level under a faulty output. For example, in the circuit given in Fig. 5, fault element $v/0/p3$ gets the same score as the fault element at the previous level i.e. $y/1/p3$ since they are directly related. The fault element $t/0/p3$ gets the summation of the scores of $v/0/p3$ and $w/0/p3$ because both these fault elements

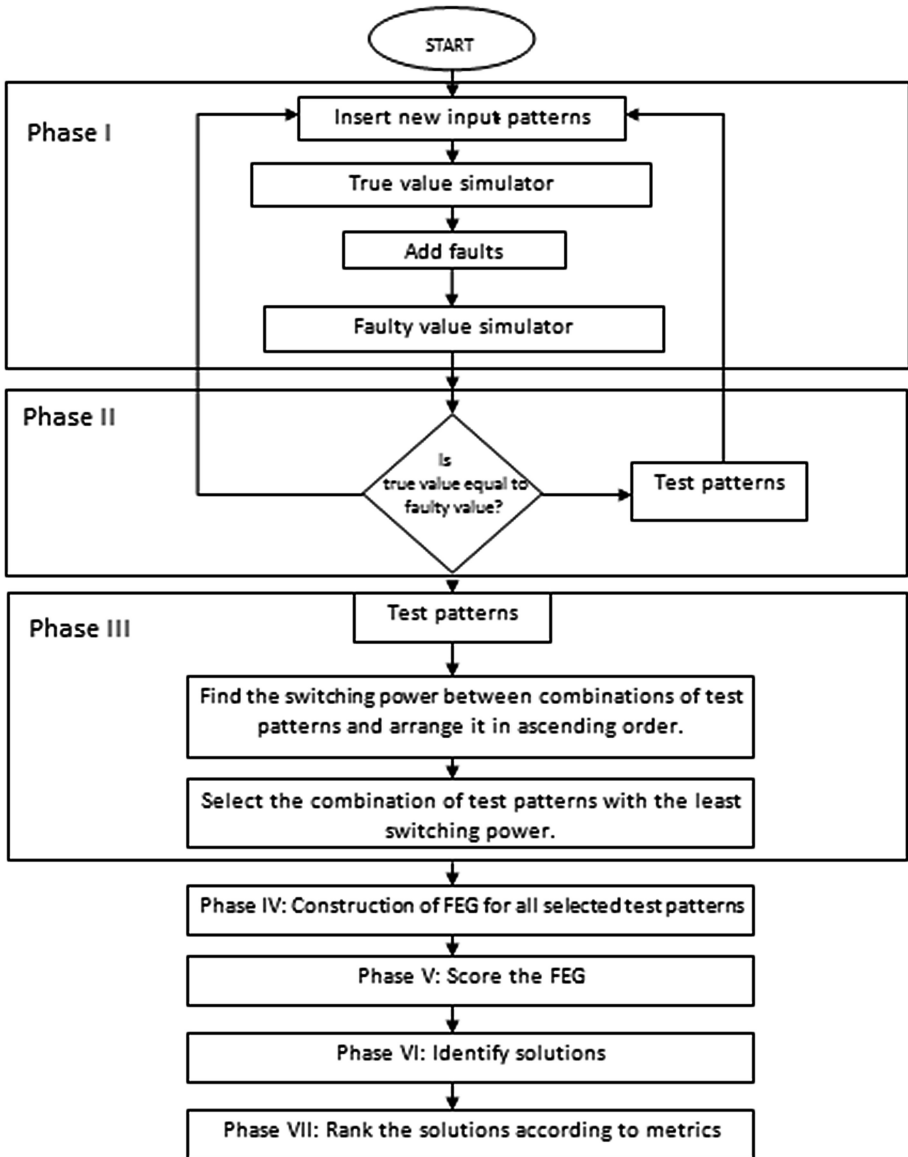


Fig. 1. Flow chart of the proposed method

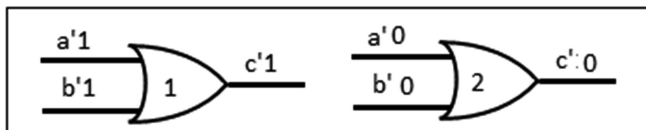


Fig. 2. Scoring for an OR gate

traverses through $t/0/p3$. And the fault elements $s/1/p3$, and $d/1/p3$ in Fig. 5 gets $1/2$ times the score at the fault element $t/0/p3$ i.e. $0.6667/2 = 0.333$. When the score is being given to the input of a gate present in the FEG, it can either be the same score as its output score or divided by the number of inputs from the output score. Consider the gates given in Fig. 2.

In case of gate 1 in Fig. 2, for the output c' to be 1, either one of the inputs must be 1. Since only one input is required to have a certain value, the score given at c' is equal to the score given to a' and the score given to b' .

In case of gate 2 given in Fig. 2, for the output c' to be 0, both the inputs must be 0. Since both the inputs are required to have a certain value, the score at a' and b' is half the score at c' .

The above mentioned steps are followed until the whole FEG is scored. An example is given in Figs. 3, 4 and 5.

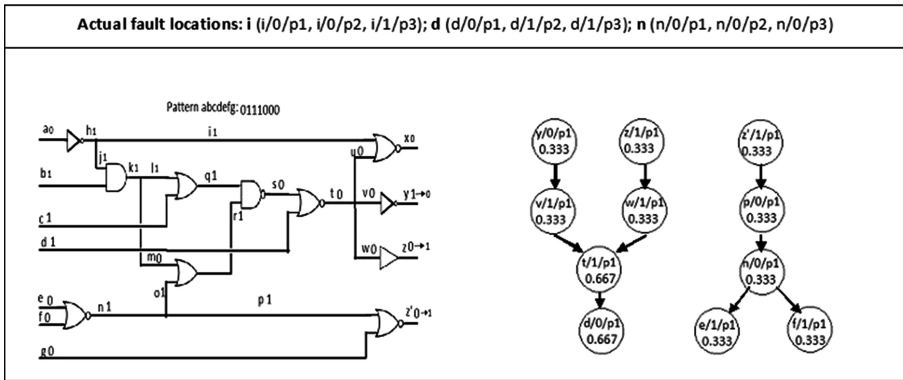


Fig. 3. Circuit and FEG for P1: 0111000

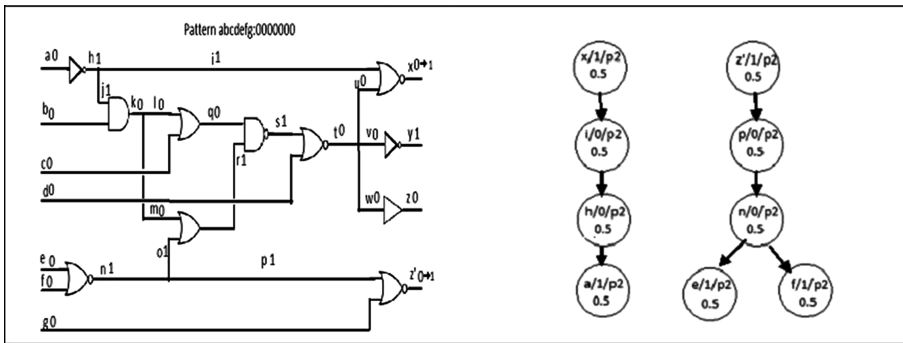


Fig. 4. Circuit and FEG for P2: 0000000

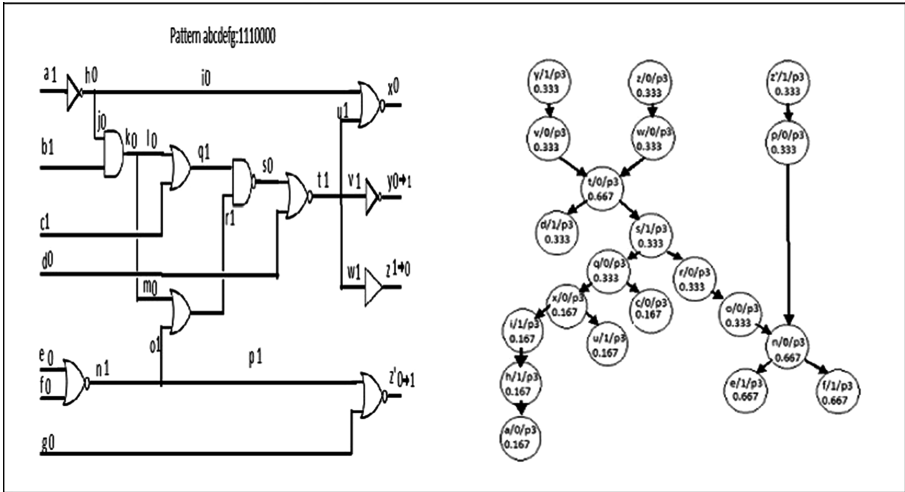


Fig. 5. Circuit and FEG for P3: 1110000

Table 1. Summation of scores in iteration 1

Fault element	Test pattern 1	Test pattern 2	Test pattern 3	Sum
a	-	0.5	0.222	0.722
c	-	-	0.167	0.167
d	0.667	-	0.333	1
e	0.333	0.5	0.667	1.5
f	0.333	0.5	0.667	1.5
h	-	0.5	0.167	0.667
i	-	0.5	0.167	0.667
n	0.333	0.5	0.667	1.5
o	-	-	0.167	0.167
p	0.333	0.5	0.333	1.167
r	-	-	0.333	0.333
s	-	-	0.333	0.333
t	0.667	-	0.667	1.334
u	-	-	0.167	0.167
v	0.333	-	0.333	0.666
w	0.333	-	0.333	0.666
x	-	0.5	0.167	0.666
y	0.333	-	0.333	0.666
z	0.333	-	0.333	0.666
z'	0.333	0.5	0.333	1.166

2.6 Identification of Solutions

Once the FEGs for the required test patterns are constructed and scored, the scores of all the fault elements are summed correspondingly. The fault element with the highest summed score is the candidate location for carrying out the first iteration. In continuation with the example given in Figs. 2, 3 and 4, results are shown in Table 1. Therefore, the candidate locations are ‘e’, ‘f’ and ‘n’, the highest summed score being 1.5. Since the fault elements e/1, f/1 and n/0 has the same effect on the faulty outputs, the three candidate locations can be pruned simultaneously.

For the FEG in Fig. 2, when the fault elements e/1/p1, f/1/p1 and n/0/p1 are pruned together, the nodes that are affected by these fault elements are also pruned. Any change in n/0/p1 will affect p/0/p1 and in turn affect z/1/p1, therefore, p/0/p1 and z/1/p1 are removed from the FEG. This process is done for all three FEGs as shown in Figs. 6 and 7.

The candidate locations are again determined by finding the fault elements with the maximum score sum, as shown in Table 2.

Iteration 1:

The summed score 1 is the maximum and therefore, ‘a’, ‘d’, ‘h’, ‘i’, ‘t’ and ‘x’ are the candidate locations.

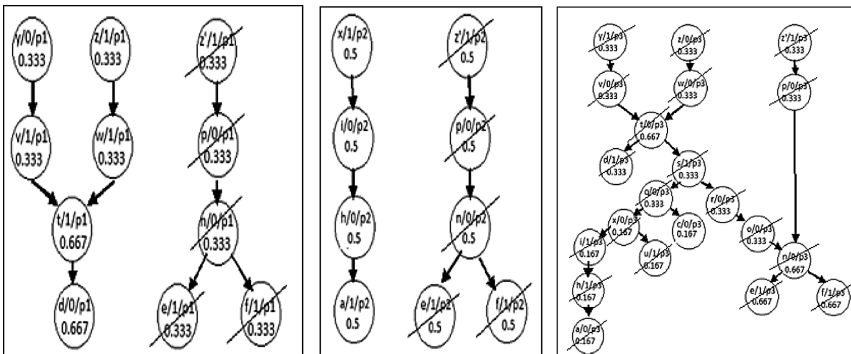


Fig. 6. Pruned FEG of p1, pruned FEG of p2 and pruned FEG of p3

Since the fault elements ‘a’, ‘h’, ‘i’ and ‘x’ have different effects on the faulty outputs from that of fault elements ‘d’ and ‘t’. Therefore, these two groups of fault elements must be pruned in parallel.

After pruning the candidate locations, the results are shown in Fig. 8(1) and (2).

Iteration 2:

Candidate locations in solution 1: The highest scored fault elements when ‘x’, ‘i’, ‘h’, ‘a’ are pruned: ‘d’, ‘t’

Candidate locations in solution 2: The highest scored fault elements when ‘d’ and ‘t’ are pruned: ‘x’, ‘i’, ‘h’, ‘a’

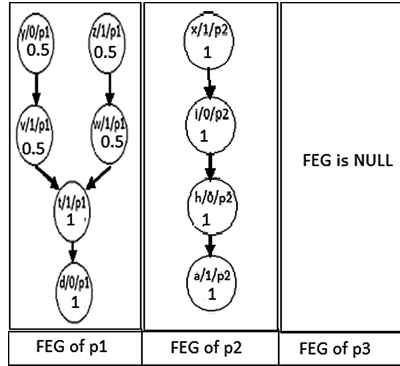


Fig. 7. FEG after pruning and rescoring

Table 2. Summation of scores in iteration 2

Fault Elements	Test Pattern 1	Test Pattern 2	Sum	Fault Elements	Test Pattern 1	Test Pattern 2	Sum
a	-	1	1	v	0.5	-	0.5
d	1	-	1	w	0.5	-	0.5
h	-	1	1	x	-	1	1
i	-	1	1	y	0.5	-	0.5
t	1	-	1	z	0.5	-	0.5

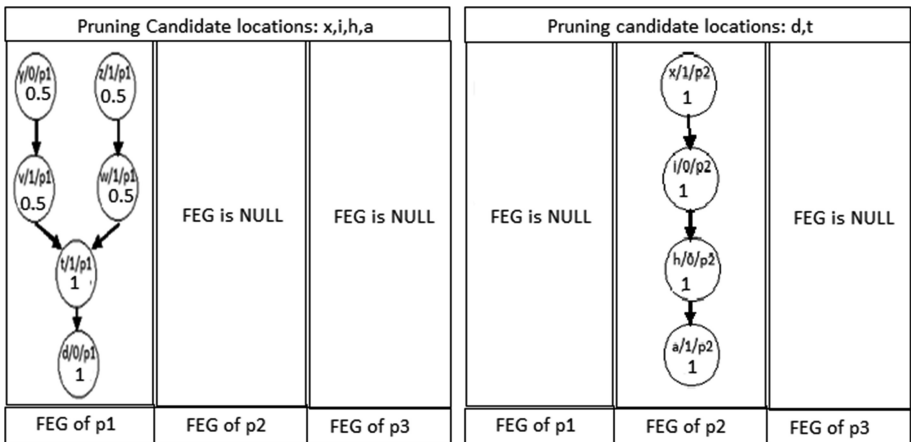


Fig. 8. (1) FEG after pruning x, i, h, a. (2) FEG after pruning d, t

After pruning these candidate locations in parallel, the results are shown in Fig. 9 (1) and (2).

Pruning Candidate locations: d,t			Pruning candidate locations: x,i,h,a		
FEG is NULL	FEG is NULL	FEG is NULL	FEG is NULL	FEG is NULL	FEG is NULL
FEG of p1	FEG of p2	FEG of p3	FEG of p1	FEG of p2	FEG of p3

Fig. 9. (1) FEG after pruning fault elements d and t. (2) FEG after pruning the fault elements x, i, h and a

After three iterations, it is found that the FEG becomes NULL, as shown in Table 3

Table 3. Probable candidate locations

Iteration	Candidate locations in solution 1	Candidate locations in solution 2
1 st	n, f, e	–
2 nd	a, i, h, x	d, t
3 rd	d, t	a, i, x, h

2.7 Ranking of Candidate Locations

The solutions found in Sect. 2.6 are ranked using two metrics:

Metric 1: The number of solutions that contain a candidate location.

Table 4. Ranking

Ranking	Candidate location	Iteration	Appears in
Rank 1	a, i, h, x, d, t	2 nd and 3 rd	Two solutions
Rank 2	n, f, e	1 st	One solution

Metric 2: The order of selection for a candidate location.

From the probability perspective metric 1 is given more priority than the latter during ranking.

According to the data in Table 4, there might be at least one actual fault location among the fault elements ‘a’, ‘i’, ‘h’, ‘x’, ‘d’, ‘t’ of Rank 1 and that there might be another actual fault location among the fault elements ‘n’, ‘f’, ‘e’ of Rank 2.

3 Experimental Results

The experiments are conducted on a Windows 8.1 operating system with four 2.50-GHz CPUs. Ten circuits, five each of ISCAS’85 and ISCAS’89 benchmark circuits were used for testing the fault coverage of the proposed algorithm for test patterns which use minimum switching power are shown in Table 5 (Represents the data for triple stuck-at faults).

In the experiments here, fault injection was divided into two categories, namely, single stuck-at fault and multiple stuck-at faults. Multiple stuck-at faults were subdivided into double and triple stuck-at faults. The test patterns for all the benchmark circuits were found by a self-coded ATPG. After the test patterns were found, Synopsis primetime tool was used with an input netlist which was inclusive of single or multiple stuck-at faults. Test benches were then written for finding the switching power of different groups of test patterns, out of which the group with the least power was chosen for FEG diagnosis.

After choosing the test patterns with the minimum power value, the next phase of identification and diagnosis of the faults was done. Tabulation of the probability of fault coverage is shown in Tables 6a and 6b. For each circuit on test; single, double and triple stuck-at faults were injected and their respective fault coverage is shown. A comparison on the change of accuracy of fault detection was observed when the ‘L’ value (as mentioned in the Sect. 2.1) was changed.

This is due to the fact that the different nodes in the circuit must stabilize to the desirable input test patterns after the injection of faults, which requires ‘L’ number of iterations. The value of ‘L’ may vary for different circuits and a common accurate value cannot be stated.

A mixture of both stuck-at 0 and stuck-at 1 were injected which would be homogeneously varying from primary output to primary input. For each type of fault category mentioned above, ten diagnosis cases were run and the fault coverage was found. For example, the total number of outcomes for triple stuck-at fault would be 30, since there are 10 diagnosis cases and each case consists of 3 faults.

The probability of the fault coverage is given by Eq. 1

$$p(f) = \sum_1^{10} n(\text{Successful diagnosis})/n(\text{Total no of diagnosis cases}) \quad (1)$$

Table 5. Switching power analysis (Tn denotes the input pattern for n)

Benchmark circuit	Test patterns	Total switching power
C499	T0-T1-T2	1.248*10 ⁻¹¹ W
	T1-T2-T3	3.045*10 ⁻⁹ W
	T2-T3-T4	6.09*10 ⁻⁹ W
	T3-T4-T7	5.962*10 ⁻⁹ W
	T4-T7-T9	7.837*10 ⁻⁹ W
C880	T0-T1-T2	5.688*10 ⁻¹² W
	T2-T3-T8	7.286*10 ⁻¹¹ W
	T3-T8-T9	3.643*10 ⁻¹¹ W
	T6-T11-T15	4.448*10 ⁻¹¹ W
	T3-T7-T9	1.457*10 ⁻¹¹ W
C1355	T0-T1-T2	1.996*10 ⁻¹¹ W
	T1-T2-T4	1.585*10 ⁻¹¹ W
	T0-T5-T9	2.93*10 ⁻⁹ W
	T2-T3-T7	4.798*10 ⁻⁹ W
	T2-T4-T7	2.985*10 ⁻⁹ W
C1908	T0-T1-T2	2.089*10 ⁻⁹ W
	T2-T3-T4	7.879*10 ⁻⁹ W
	T3-T4-T6	9.093*10 ⁻⁹ W
	T2-T4-T9	1.351*10 ⁻⁸ W
	T3-T4-T9	1.4304*10 ⁻⁸ W
C2670	T0-T1-T2	1.831*10 ⁻⁹ W
	T0-T2-T7	6.49*10 ⁻¹⁰ W
	T2-T6-T11	6.5*10 ⁻¹⁰ W
	T7-T8-T10	6.74*10 ⁻¹⁰ W
	T5-T9-T10	4.4*10 ⁻⁹ W
S444	T0-T1-T5	2.789*10 ⁻⁹ W
	T0-T7-T5	5.578*10 ⁻⁹ W
	T0-T3-T2	5.096*10 ⁻¹⁰ W
	T3-T7-T2	3*10 ⁻¹⁰ W
	T0-T2-T7	1.02*10 ⁻⁹ W
S526	T0-T5-T7	1.1673*10 ⁻⁸ W
	T1-T2-T3	1.5*10 ⁻⁹ W
	T1-T5-T0	1.063*10 ⁻⁸ W
	T1-T3-T5	5.57*10 ⁻⁹ W
	T1-T2-T5	6.096*10 ⁻⁹ W
S1234	T1-T2-T3	5.112*10 ⁻⁹ W
	T4-T2-T5	5.77*10 ⁻⁹ W
	T3-T6-T7	4.85*10 ⁻⁹ W
	T3-T6-T5	4.582*10 ⁻⁹ W
	T1-T0-T3	5*10 ⁻⁹ W
S1423	T1-T0-T2	4.7*10 ⁻¹⁰ W
	T1-T6-T3	2.32*10 ⁻⁹ W
	T2-T6-T3	1.78*10 ⁻⁹ W
	T7-T5-T3	1.71*10 ⁻⁹ W
	T1-T2-T3	7.01*10 ⁻¹⁰ W
S1488	T1-T0-T2	1.16*10 ⁻⁹ W
	T1-T3-T2	1.3*10 ⁻⁹ W
	T3-T5-T7	1.753*10 ⁻⁹ W
	T7-T10-T12	3.768*10 ⁻⁹ W
	T8-T10-T14	9.425*10 ⁻⁷ W

Table 6a. Probability of fault coverage with $L < 500$

Sr. no.	Circuits	1 fault	2 faults	3 faults
1	C499	0.6	0.55	0.5
2	C880	0.6	0.65	0.5
3	C1355	0.6	0.5	0.5
4	C1908	0.7	0.55	0.5
5	C2670	0.6	0.65	0.56
6	S444	0.7	0.5	0.5
7	S526	0.6	0.5	0.5
8	S1234	0.6	0.5	0.53
9	S1423	0.7	0.5	0.53
10	S1488	0.6	0.55	0.5

Table 6b. Probability of fault coverage with $L > 500$

Sr. no.	Circuits	1 fault	2 faults	3 faults
1	C499	0.8	0.7	0.767
2	C880	0.9	0.75	0.767
3	C1355	0.9	0.75	0.7
4	C1908	0.9	0.8	0.876
5	C2670	0.9	0.8	0.876
6	S444	0.8	0.8	0.8
7	S526	0.8	0.8	0.7
8	S1234	0.8	0.75	0.7
9	S1423	0.8	0.7	0.7
10	S1488	0.7	0.8	0.67

4 Future Scope

The proposed method of fault diagnosis can be improved by choosing the test patterns that can detect more faults and has less switching power. They can be chosen in such a way that there is a trade-off between power dissipated and accuracy achieved. The method can also be extended to larger circuits with over 5000 nodes.

5 Conclusion

In this paper, a method for diagnosis of multiple stuck-at faults with reduced switching power is proposed. Here, only those test patterns that have minimum switching power are used. In order to find the initial set of test patterns for each circuit, the outputs of faulty value and fault-free value simulators are compared. Then, different combinations of test patterns are taken and their switching powers were found. This is then arranged in ascending order of power dissipation. From this consolidated data, the first combination of test patterns was chosen which was applied in the method for fault

diagnosis. For multiple fault diagnosis, a Fault Element Graph (FEG) algorithm was used which is discussed in detail in the previous sections. The obtained results proved the efficiency of the proposed method over multiple fault diagnosis based on restricted single sensitized path.

References

1. Jha, N., Gupta, S.: Testing of Digital Systems. Cambridge University Press, Cambridge (2003)
2. Aitken, R.C.: Modeling the unmodelable: algorithmic fault diagnosis. *IEEE Des. Test Comput.* **14**(3), 98–103 (1997)
3. Huisman, L.M.: Diagnosing arbitrary defects in logic designs using single location at a time (SLAT). *IEEE Trans. Comput. Aided Des. Integr. Circ. Syst.* **23**(1), 91–101 (2004)
4. Lin, Y.C., Lu, F., Cheng, K.-T.: Multiple-fault diagnosis based on adaptive diagnostic test pattern generation. *IEEE Trans. Comput. Aided Des. Integr. Circ. Syst.* **26**(5), 932–942 (2007)
5. Lavo, D., Hartanto, I., Larrabee, T.: Multiplets, models, and the search for meaning: improving per-test fault diagnosis. In: Proceedings of International Test Conference, pp. 250–259, January 2002
6. Wang, Z., Tsai, K.-H., Kun-Han, T., Marek-Sadowska, M., Rajske, J.: An efficient and effective methodology on the multiple fault diagnosis. In: Proceedings of International Test Conference, pp. 329–338, October 2003
7. Wang, Z., Marek-Sadovvska, M., Kun-Han, T., Rajske, J.: Analysis and methodology for multiple-fault diagnosis. *IEEE Trans. Comput. Aided Des. Integr. Circ. Syst.* **25**(3), 558–575 (2006)
8. Yu, X., Blanton, R.D.: Multiple defect diagnosis using no assumptions on failing pattern characteristics. In: Proceedings of 45th ACM/IEEE Design Automation Conference, pp. 361–366, June 2008
9. Yu, X., Blanton, R.D.: Diagnosis of integrated circuits with multiple defects of arbitrary characteristics. *IEEE Trans. Comput. Aided Des. Integr. Circ. Syst.* **29**(6), 977–987 (2010)
10. Veneris, A., Liu, J., Amiri, M., Abadir, M.S.: Incremental diagnosis and correction of multiple faults and errors. In: Proceedings of Design Automation Test European Conference and Exhibition, pp. 716–721, March 2002
11. Smith, A., Veneris, A., Viglas, A.: Design diagnosis using Boolean satisfiability. In: Proceedings Asian South Pacific Design Automation Conference, pp. 218–223, January 2004
12. Takahashi, H., Boateng, K.O., Saluja, K.K., Takamatsu, Y.: On diagnosing multiple stuck-at faults using multiple and single fault simulation in combinational circuits. *IEEE Trans. Comput. Aided Des. Integr. Circ. Syst.* **21**(3), 362–368 (2002)
13. Hu, Y., Li, X., Cheng, W., Huang, Y., Tang, H.: Diagnose failures caused by multiple locations at a time. *IEEE Trans. VLSI* **22**(4), 824–837 (2014)
14. Mohan, N., Anita, J.P.: A zero suppressed binary decision diagram based test set relaxation for single and multiple stuck-at faults. *Int. J. Math. Model. Numer. Optim.* **7**(1), 83–96 (2016)
15. Sinduja, V., Raghav, S., Anita, J.P.: Efficient don't-care filling method to achieve reduction in test power. In: Proceedings of International Test Conference, Advances in Computing, Communications and Informatics, pp. 478–482, August 2015

Intrusion Detection Using Improved Decision Tree Algorithm with Binary and Quad Split

Shubha Puthran¹(✉) and Ketan Shah²

¹ CE and IT Department, Mukesh Patel School of Technology Management and Engineering, Vile Parle, 400056 Mumbai, India

`shubha.puthran@nmims.edu`

² IT Department, Mukesh Patel School of Technology Management and Engineering, Vile Parle, 400056 Mumbai, India

`ketanshah@nmims.edu`

Abstract. Security is a big issue for all servers including defence and government organisations. The Intrusion detection system (IDS) is one that scans server's incoming data activities and attempts to detect the attacks. Data mining based IDS have shown good detection rates for normal and DoS attacks, but do not perform well on Probe, U2R and R2L attacks.

The paper highlights the poor performance of existing ID3 algorithm for Probe, R2L and U2R attacks. The paper also proposes improved decision tree algorithm using binary split (IDTBS) and improved decision tree algorithm using quad split (IDTQS) for improving the detection rate of Probe, U2R and R2L attacks. In this research, KDD99 dataset is used for the experimentation. The True Positive Rate (TPR) accuracy of both the algorithms are compared with the existing ID3 decision tree algorithm. IDTQS algorithm outperforms with the True Positive Rates (TPR) accuracy for Probe, R2L and U2R attacks with values of 99.23 %, 95.57 % and 56.31 % respectively.

Keywords: KDD 1999 · Decision tree · Quad split · Binary split · Intrusion detection

1 Introduction

Identifying the illegal use of information on the organisation's server is defined as intrusion detection. Usual techniques for network security includes user authentication, intrusion prevention systems like firewalls. An intrusion detection system is a device or software application that monitors a systems for malicious activity or policy violations. Any detected activity or violation is typically reported to an administrator. Harmless and harmful users are the types of users accessing data from servers in an organisation. The information on the open hub is available to both types of users. Harmful users can get access to an organizations internal systems. Eighty percent of the intruders are internal users, as these users knows the systems very well than the external users [2].

1.1 Intrusion Detection System (IDS) and Its Need

Intrusion detection system is concerned with monitoring and analysing events occurring in the network information system. IDS can discover the malicious activity [18]. IDS is a safety system that monitors the information systems and the network. It analyzes the network information for possible attacks from outside and from inside the organization [12].

1.2 Challenges in Intrusion Detection [4, 8]

1. Accuracy and speedy detection is the major challenge. The more accurate the detector, the higher the computational overhead. The IDS is reactive in most cases rather than proactive. Proactive Intrusion detection is a challenge.
2. Large load of data is another challenge.
3. Large number of false positives and false negatives.
4. Incremental updation of attacks in the database.

1.3 Need for Datamining in IDS

Data mining can classify and cluster the large amount of data. It can correlate the data patterns to identify the alarms and update the database. To study and identify unknown attack patterns. Data mining helps to establish patterns for normal behaviour and considers the activity that lies outside this pattern as attacks. The data mining is also helpful in data reduction without the loss of useful information. The challenges mentioned in the previous section can be resolved using datamining techniques [8].

1.4 Authors Contributions

1. Generated Binary split tree and Quad split tree. The rules generated are applied on the testing data.
2. 22 types of attacks are seen in KDD99 dataset. Preprocessing of these 22 types of attacks are done. After preprocessing, the four main categories of attacks are found.
3. In order to get the continuous values, preprocessing of attributes like attack type, protocol type and flag are done. This preprocessed methods and DecisionTree (DT) splits are exclusively suitable for KDD 99 dataset.
4. Training and Test data is preprocessed from the KDD-99 dataset. Cross Validation is performed. Average of crossvalidated results are compared with the existing ID3 algorithm.

In this research the Sect. 2 describes the Related work in data mining approaches for intrusion detection and Decision Tree algorithms. Section 3 discusses about Research Approach, Sects. 4 and 5 discusses about the IDTBs, IDTQS approach and the results. Conclusions and the future work is discussed in Sect. 6.

2 Related Work

Redundant features of the data leads to false alarm rates. By reducing the feature dimensions of the dataset, volume can be reduced. By parallelising the data processing one can achieve the speedy detection [1].

Feature selection is very important for reducing the volume of data. Only those features which are relevant for Intrusion Detection is selected. This selection has been performed by implementing information gain. [2].

Mazid et al. [3], discussed the improvement to C4.5, which is rule based algorithm. Their objective was to reduce the dataset. They have used entropy of information theory to select the efficient attributes. Then used the correlation coefficient measure to find the correlation among selected attributes.

Memar et al. discussed the Intrusion detection using Data Mining Techniques [4]. The classification tree and SVM methods were used to detect intrusions in the network. The detection of attacks and false alarm rates are better in C4.5 than in SVM.

Adetunmbi et al. discussed the analysis of KDD99 dataset and the selection of relevant features. KDD 99 data is the raw TCP dump of nine weeks data on a local area Network. [5].

Authors discussed the bagging algorithm for IDS. Bagging is a kind of voting algorithm. This changes the distribution of the training data and runs the classifier. The trained data will be combined to test with the testing data. The false positive and true positive rates are better in this proposed algorithm. But C4.5 algorithm outperforms [6].

Trained model has sufficient storage environment and large data pattern. The outlier data is one which lies outside the specified threshold [7,17]. One can use IDS's for recognising issues with security policies, identifying threats and individuals from violating security policies. IDS's have become a need to the information system of every firm [2].

DT constructs tree like structure to model the training data. DT has a solution for many issues namely medical diagnosis, banking sectors and societal issues. These trees study the patterns by considering the attributes threshold and creates simple rules to differentiate between the various types of data. These rules can be applied for the classification of the testing data. Our literature survey [8,13–15] shows that, DT algorithm is most suitable for Intrusion Detection system.

Some of the surveyed algorithms and their limitations are listed as shown in Table 1. In the paper [19], the accuracy is calculated with respect to all the attacks. This experimentation used only discrete valued attributes. In the paper [20], Accuracy is used as the performance metric. Accuracy metric is not suitable if the dataset does not have uniformly distributed training data set. In [22], the proposed pruning technique is concentrating on the removal of primary key data. The KDD'99 dataset does not contain unique valued attribute, which can be considered as the primary key.

Table 1. Surveyed techniques and limitations

Technique name	Technique/Algorithm used	Drawbacks and limitations
Decision tree [19]	C4.5 with pruning	Accuracy in general
Empirical tech. [20]	10 DM algo's including DT	Accuracy metric
Combination of DM technique [21]	Random forest with random tree	Large false alarm rates
Reoptimization of DT [22]	ID3 and C4.5	Pruning

3 Research Approach

The steps followed to classify the attacks. 1. Database Selection 2. Preprocessing of Database 3. Selection of Training Data and testing data 4. Applying the improved DT algorithm (Binary split/quad split) to cross validated data 5. Applying the rules on the test data 6. Classification of attacks (Fig. 1).

3.1 KDD Database

The KDD99 is a gathering of TCP dump of nine weeks [9]. The KDD 99 is freely available for researchers. This dataset can be applied as an effective benchmark to help researchers. This consists of 42 attributes as shown in the Table 3. This constitutes normal recordset as well as 21 different attacks. The simulated attacks fall in one of the following four categories [10, 11]. The Table 2 shows the types of attacks and attack class.

3.2 Preprocessing

In this experiment, KDD99 dataset is preprocessed. Categorised the attack classes into four types. The numerical values to the attack types are as shown in

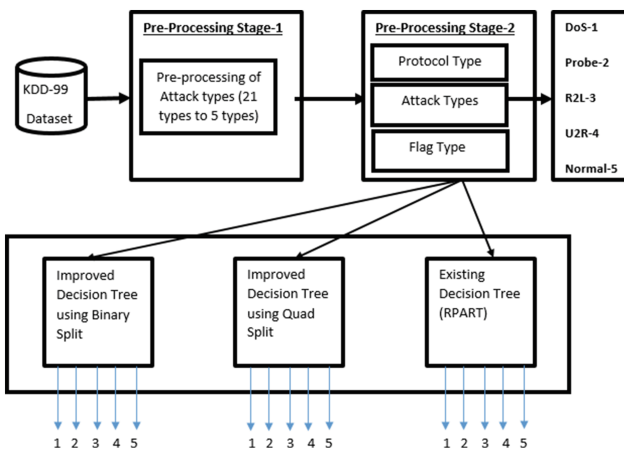


Fig. 1. Research approach

Table 2. KDD dataset attack types

Types of attacks	Attack class
DoS	back, lnd, neptun, pod, smurf, teardrp
U2R	bufferoverflow, perl, load module, rootkt
R2L	ftpwrte, guesspasswd, imp, multhop
	phf,spy, warezclint, warzmaster
Prbe	ipweep, nmap, portweep, satn

Table 3. KDD dataset attributes

Duration	Protocoltype	Srvserrorate
service	srcbytes	srvdiff host rate
dstbytes	flg	dsthost srvcount
land	wrongfragment	dst hostdiff srvrate
urgnt	hot	srvrerror rate
numfailed logins	loggedin	dsthost countl
lnumcompromised	rootshel	dsthost samesrv rate
lsuattempted	lnumroot	dsthost same src port rate
lnumfile creations	lnumshells	dsthost srvdiff host rate
lnumaccess files	lnumoutbound	dsthost srvserror rate
is hostlogin	is guestlogin	dst hostsrv rerrorate
count	serrorate	dst hosterror rate
rerrorate	same srvrate	dsthost rerrorate
diffsrvrate	srvcount	label (class)

Table 4. Attacktype

Attack type	Numerical value
DoS	1
Probe	2
R2L	3
U2R	4
Normal	5

Table 4. The protocoltype attribute is also preprocessed to the numerical values as shown in Table 5. The numerical conversion of Flag attribute is shown in Table 6. The four attack types are explained here below.

1. Denial of Service Attack (DoS): Memory is made too busy by the attacker. In this case, the legitimate users cannot access the machine.
2. Probing Attack: Weaknesses of the machine is studied by the attacker.

Table 5. ProtocolType attribute

Protocoltype	Numerical value
tcp	1
icmp	2
udp	3

Table 6. Flag attribute

Flag	Num. value	Flag	Num. value
OTH	1	REJ	2
RSTO	3	RSTOS0	4
RSTR	5	S0	6
S1	7	S2	8
S3	9	SF	10
SH	11		

3. Remote to Local Attack (R2L): The attacker has the ability to send packets to a machine over a network but he does not have an account on that machine. He exploits some vulnerability to gain local access as a user of that machine.
4. User to Root Attack (U2R): Attacker accesses the system like normal user and tries to exploit some vulnerability to gain root access to the system.

4 Improved Decision Tree Algo with Binary Split (IDTBS)

One of the problem faced with the ID3 algorithm is the number of splits taken depends on the number of categories of an attribute. But it is not suitable for intrusion detection dataset (KDD-99) in this experiment. Some attributes may have ten’s of categories. The proposed algorithm works on the basis of DT algorithms. In the improved algorithm, the following improvements are done. One attribute is considered at each split based on the highest information gain ratio and only Binary split is considered at each level.

4.1 Experimental Setup

The experimentation and analysis of the results are done using the R programming setup. The R package, recursive partitioning (RPART) is used for experimenting ID3. The training set used in this experimentation is the random subset of KDD99. The original KDD 99 dataset had lot of redundant records. Therefore, to overcome this deficiency all the redundant records from KDD99 dataset is removed. The unique dataset used for the experimentation contained 141700

Table 7. Binary split for KDD99

Type	TPR	FPR
DoS	0.9982	0.0028
Probe	0.3017	0
R2L	0.0689	0.00006
U2R	0	0.00066
Normal	0.9952	0.019

records. Separate testing set and training set is pre-processed for our experimentation.

4.2 IDTBS Algorithm

- 1: Finding the entropy
- 2: Finding the information gainratio
- 3: Highest gainratio attribute as Attbest1 is selected as rootnode
- 4: Find the rootnode's binary split

Loop Process:

- 5: **for** *splitobject = 1 to* all splitobject **do**
- 6: Find the highest information gain ratio attribute
- 7: Select the splitting attributes
- 8: Find the median as threshold
- 9: Find the binary split subset
- 10: **end for**
- 11: Classify leaf objects as classes

4.3 Results of IDTBS

In ID3, R2L's TPR accuracy is 0%, whereas IDTBS improvised the same with 6%. The IDTBS does not perform well for Probe and DoS when compared with ID3 algorithm. In order to improve the detection rate of all the attacks IDTQS with quad split (IDTQS) is proposed. The Table 7 shows the results of IDTBS.

5 Improved Decision Tree Algo with Quad Split (IDTQS)

The objective of this algorithm is generating the crisp DT with quad splits for numeric attributes. In quad trees the attributes rarely reappears in any path from root to leaf. Thus the ranking of the attributes based on their contribution can be analysed. Ranking could be done by tracking the classification of an instance by starting at the root to leaf. By quad split the number of levels of split and the complexity of the DT is reduced.

5.1 Experimental Setup

When only information gain is applied to the KDD99 dataset, it selected the attribute which had distinct values. The information gain ratio gives us the relevant attributes to split. In this algorithm the first two highest information gain ratioed attributes are considered as the splitting attributes. Since the two attributes are considered at every node the tree splits into four branches. Following rules are considered for splitting the attributes.

$$\text{Attribute1} \leq m1 \cap \text{attribute2} \leq m2 \quad (1)$$

$$\text{Attribute1} \leq m1 \cap \text{attribute2} > m2 \quad (2)$$

$$\text{Attribute1} > m1 \cap \text{attribute2} > m2 \quad (3)$$

$$\text{Attribute1} > m1 \cap \text{attribute2} \leq m2 \quad (4)$$

where *attribute1* and *attribute2* are the highest information gain ratioed attributes at each level. *m1* and *m2* are the median of *attribute1* and *attribute2* respectively. At each level, the highest information gain ratio (IGR) is calculated. The first and the second highest IGR values are considered as *attribute1* and *attribute2*.

$$\text{InformationGainRatio}(A) = \text{InformationGain}(A) / \text{splitinfo}(A) \quad (5)$$

Gain Ratio is a modification of the information gain that reduces its bias in the selection of root node in classification [19]. The gain ratio divides the gain by the evaluated split information. The information gain ratio concept is used from already existing algorithm C4.5. Improved DT algorithm helps us to explore the structure of the data. The tree is built by the following process. The best attributes for splitting is obtained by finding the highest information gain ratio. These attribute splits the data into four subgroups. The same process is repeated for each subgroup. The process continues until no improvement can be made.

5.2 IDTQS Algorithm

- 1: Finding the entropy
- 2: Finding the information gainratio
- 3: First two highest gainratio Attribute1, Attribute2 is selected as rootnode
- 4: Find the rootnode quad split
- Loop Process:*
- 5: **for** *splitobject = 1* to all splitobject **do**
- 6: Find the highest information gain ratio attributes
- 7: Select the splitting attributes
- 8: Find the median as threshold
- 9: Find the quad split subset
- 10: **end for**
- 11: Classify leaf objects as classes

Table 8. Cross validation

TrainData	TestData
30 %	50000,100000,141700
60 %	50000,100000, 141700
100 %	50000,100000,141700

5.3 Results of IDTQS and Comparison with IDTBS and ID3 (RPART)

Separate testing set and training set is pre-processed for our experimentation. Cross validation is performed as shown in the Table 8 and summarised the results in the Table 12. The results of Existing Decision Tree, Improved DT with quad split is shown in Table 13 (Fig. 2). The accuracy in IDS is based on true positive rates and False positive rates. The TPR and FPR is calculated using the following equations [9].

$$TruePositiveRate(TPR) = TP / (TP + FN) \tag{6}$$

$$FalsePositiveRate(FPR) = FP / (FP + TN) \tag{7}$$

The confusion matrix in Tables 9, 10 and 11 are experimented using IDTQS. The confusion matrix for 100 % training data with 50000, 100000 and 141000 testing data respectively is shown below. The confusion matrix has pruned data. In the Confusion Matrix Table 9, DoS attacks were not found in the testing data. The 50000 records of testing data and 100 % training data is considered in this partition. The confusion matrix Table 10 shows the 100 % training data with 100000 records of testing data. The Table 11 shows Confusion Matrix for 100 % training data and 141000 records of testing data. The experiments were performed for 30 % and 60 % training data. The average of crossvalidation results are shown in the Table 12 [16].

Testing data has 141000 records and training data contains 3500 records. From IDTBS and IDTQS algorithms it is analysed that, only few attributes from the KDD 99 dataset is useful during the binary and quad tree construction. Only

Table 9. Confusion matrix (pruned data) for 100 percent training data and 50000 records of testing data using IDTQS

Type	DoS	Probe	R2L	U2R	Normal
DoS	0	0	0	0	0
Probe	0	1816	5	5	4
R2L	0	7	626	11	11
U2R	0	1	0	28	11
Normal	0	2551	2653	382	28949

Table 10. Confusion matrix (pruned data) for 100 percent training data and 100000 records of testing data using IDTQS

Type	DoS	Probe	R2L	U2R	Normal
DoS	8843	45	0	0	98
Probe	0	1816	5	5	4
R2L	0	7	626	11	11
U2R	0	1	0	28	11
Normal	0	3011	4243	620	61960

Table 11. Confusion matrix (pruned data) for 100 percent training data and 141000 records of testing data using IDTQS

Type	DoS	Probe	R2L	U2R	Normal
DoS	47606	894	0	56	919
Probe	0	1816	5	5	4
R2L	0	7	626	11	11
U2R	0	1	0	28	11
Normal	0	3011	4243	620	61960

Table 12. KDD-99 dataset tested with IDTQS

Type	TPR	FPR
DoS	0.9316	0.0030
Probe	0.9923	0.0288
R2L	0.9557	0.0334
U2R	0.5631	0.0073
Normal	0.9624	0.0540

Table 13. Comparison of TPR accuracy with proposed algorithms and ID3 (RPART)

Type	IDTBS	IDTQS	RPART
DoS	0.9982	0.9316	0.9982
Probe	0.3017	0.9923	0.3534
R2L	0.0689	0.9557	0.0000
U2R	0	0.5631	0.0000
Normal	0.9952	0.9624	0.9979

18 attributes are used during the attack detection process. These attributes are listed in the Table 14. By using this knowledge one can consider only these 18 attributes in further experimentation. This will reduce the size of the dataset, in turn increases the speed of the detection process.

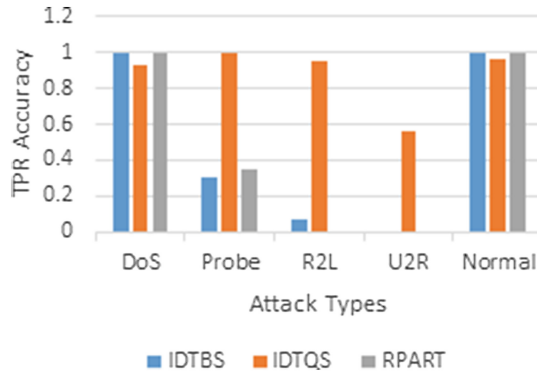


Fig. 2. TPR accuracy comparison

Table 14. KDD dataset's 18 most relevant attributes for ID

Duration	Protocoltype	Logged in
diffsrvrate	srcbytes	dst host diff srv
dst bytes	svrcount	dst hosterror
flag	dst hostsrv count	dst hostsamesrv
diff host srvrate	lnumcompromised	count
dsthostsrc port	dsthostcount	samesrv

6 Conclusions and Future Work

This paper proposes two techniques to improve Probe, U2R and R2L attacks. ID3 algorithm showed poor performance in the detection of Probe, U2R and R2L. IDTBS results shows that, Probe and R2L detection performs with 30.00 % and 6.00 % respectively. IDTQS algorithm outperforms with the True Positive Rates (TPR) accuracy for Probe, R2L and U2R attacks with 99.23 %, 95.57 % and 56.31 % respectively.

False positive resulted majorly for U2R. To improve the false alarm rates the training data quality and the feature selection must be improved. The anomaly detection could be used to upgrade the attack (signature) patterns. The combination of Misuse and Anomaly detection (Classification via clustering) could be applied to improve the detection. Through this experiment, it is found that only 18 out of 42 attributes were used for the tree formation. In the future work, only these 18 attributes could be used to reduce the computing complexity.

References

1. Chen, T., Zhang, X., Kim, S.: Efficient classification using parallel scalable compressed model and its application, pp. 5972–5983. Elsevier, China (2014)

2. David, J., Borghetti, J., Angela, A.: Survey of Distance and Similarity Measures Used Within NW Intrusion Anomaly Detection, pp. 70–91. IEEE, USA (2015)
3. Mazid, M.M., Ali, S., Tickle, K.: Improved C4.5 algorithm for rule based classification. In: Recent Advances in AI knowledge Engineering and Data Bases, pp. 296–301. ACM (2010)
4. Ektefa, M., Memar, S., Serdang: Intrusion detection using data mining techniques. In: CAMP, pp. 200–203. IEEE (2010)
5. Adetunmbi, A., Adeola, S., Abosedo, O.: Analysis of KDD 99 intrusion detection dataset for selection of relevance features. In: WCECS, pp. 162–168 (2010)
6. Gaikwad, D.P., Thool, R.: Intrusion detection system using bagging with partial decision treebase classifier. In: ICAC3, pp. 92–98. Elsevier (2015)
7. Jabez, J., Muthukumar, B.: Intrusion detection system (IDS): anomaly detection using outlier detection approach. In: ICC, pp. 338–346. Elsevier (2015)
8. Wua, S.Y., Yen, E.: Data mining-based intrusion detectors. *Expert Syst. Appl.* **36**(3), 5605–5612 (2009). Elsevier
9. Amudha, P., Rauf, H.A.: Performance Analysis of Data Mining Approaches in Intrusion Detection, India, pp. 1–6. IEEE (2012)
10. Bagheri, E., WeiLu, Ghorbani, A.A.: A Detailed Analysis of the KDD CUP 99 Data Set. IEEE (2009)
11. Tavallaee, M., Bagheri, E., Lu, W., Ghorbani, A.A.: A detailed analysis of the KDD CUP 99 data set. In: Proceedings of the 2009 IEEE Symposium on Computational Intelligence in Security and Defense Applications (2009)
12. SANS Institute Authors: Intrusion Detection Systems: Definition, Need and Challenges, SANS Institute Reading Room (2001)
13. Subrata, S.P.N., Kumar, B.I.: A comparative study of bagging, boosting and C4.5. *Asian J. Inf. Techn.* **9**, 300–306 (2010)
14. Kotsiantis, S.B.: Decision trees: a recent overview. *Artif. Intell. Rev.* **39**, 261–283 (2011). Springer Science and business media
15. Simone, A., Ludwig, F.: Analyzing Gene Expression Data: Fuzzy Decision Tree Algorithm Applied to the Classification of Cancer Data, pp. 1–8. IEEE (2015)
16. Wikipedia, free encyclopedia, cross validation statistics, timestamp: 14: 00 hrs, 22 March 2016
17. Dunham, M.H., Sridhar, S.: Data Mining Introductory and Advanced Topics. Prentice Hall, Saddle River (2015)
18. Bjerkestrand, T., Tsaptsinos, D., Pflugel, E.: Evaluation of feature selection and reduction algorithms for network IDS data. In: Cyber Situational Awareness (CyberSA), London, pp. 1–2 (2015)
19. Neha, G., Dharmaraj, R.: Implementation of network intrusion detection system using variant of decision tree algorithm. In: 2015 International Conference on Nascent Technologies in the Engineering Field (ICNTE-2015), India, pp. 1–5 (2015)
20. Aggarwal, P., Sharma, S.: An empirical comparison of classifiers to analyze intrusion detection. In: Advanced Computing Communication Technologies (ACCT) Fifth International Conference on IEEE, India, pp. 6–12 (2015)
21. Elekar, K.S.: Combination of data mining techniques for intrusion detection system. In: 2015 International Conference on Communication and Control (IC4). IEEE (2015)
22. Thakur, D., Markandaiah, N., Sharan Raj, D.: Re optimization of ID3 and C4. 5 decision tree. In: 2010 International Conference on IEEE Computer and Communication Technology (ICCCT), pp. 448–450 (2010)

MalJs: Lexical, Structural and Behavioral Analysis of Malicious JavaScripts Using Ensemble Classifier

Surendran K^(✉), Prabakaran Poornachandran, Aravind Ashok Nair,
Srinath N, Ysudhir Kumar, and Hrudya P

Amrita Center for Cyber Security, Amrita Vishwa Vidyapeetham,
Amrita University, Kollam, India
{Surendrank, Praba, Aravindashok, Srinathn, Ysudhirk,
hrudyap}@am.amrita.edu

Abstract. Over the past few years javascript has grown up and revolutionized the web by allowing user defined scripts to run inside a web browser. The application of javascript ranges from providing beautiful visualization to performing complex data analytics and modeling machine learning algorithms. However javascript are also widely being used as a channel to execute malicious activities by means of redirection, drive-by-download, vulnerability exploitation and many more in the client side. In this paper we analyze the lexical, structural and behavior characteristics of javascript code to identify malicious javascript in the wild. Performance evaluation results show that our approach achieves better accuracy, with very small false positive and false negative ratios.

1 Introduction

JavaScript is one of the building blocks of the client-side web applications. In the recent past it has become an open playground for the attackers to spread malware by injecting obfuscated javascripts. Hence Internet users tend to use malware detection softwares to stay away from getting their machines infected. As the malware creators are aware of the malware detection methods, they tend to create intelligent malwares which are difficult to get trapped, hence new detection methods evolves too and vice versa. Even if the malware evolves there are few basics/patterns that remain unchanged. The authors of this paper have developed a framework with the mixture of existing solutions to improve the detection of malicious javascripts in the vast wide Internet. Obfuscated javascripts could be detected by both static and dynamic analysis. The static analysis analyzes the text features and when combined with machine learning it becomes more intelligent enough to recognize the malicious authors obfuscating patterns. However, static analysis has its own drawbacks which could be identified using dynamic analysis. Dynamic analysis performs behavior analysis of the javascript code during run-time.

In this research, the authors have inspected multiple javascript obfuscation and mitigation techniques. The framework developed consists of a collection of good and malicious javascripts for training and modeling purposes. Different types of analysis in sandboxed environment and manual analysis are performed on obfuscated javascripts to extract functional features and automate the same. Following sections include detailed

explanation on Background and literature survey, Attack methodologies, mitigation techniques followed, design, implementation and evaluation of the system.

2 Background and Literature Survey

Static analysis: Static analysis on javascripts like combination of a basic string pattern matching (structural analysis) and calculating N-Grams (lexical analysis) on top of them helps predict the malicious pattern as shown by Choi, et al. [5]. Santos, et al. [2] had shown that using machine learning and data mining together helps better detection of malicious/obfuscated javascript. Also using machine learning algorithms on the presence of shellcode, unescape like functions made Hao, et al. [1] possible to build an intelligent system to detect the obfuscated javascripts. Wang, et al. [3] used SVM to detect malicious javascripts by training the system with known malicious javascripts and its attack patterns. Likarish, et al. [22] shows that static analysis with machine learning algorithms provide better detection rate of obfuscated javascripts. Jim, et al. [27] work shows that how img tags, URL properties of style tags are used for malicious script injection resulting in URL redirection. But these methodologies can't detect the new patterns used by attackers hence machine learning model has to be retrained. Static analysis on javascript may fail to address few attack vectors like buffer overflow or memory based attacks by the malicious javascripts. Dynamic analysis is employed to overcome the drawbacks present in the detection mechanisms of static analysis.

Dynamic analysis: Dynamic analysis on javascripts is needed to examine the runtime attack vectors. Seifert, et al. [23] shows that running obfuscated javascripts in sandboxed environment helps in classifying behavior of malicious webpages. Behavior of eval() functions or document.write() functions can be analyzed through dynamic analysis as they can reveal suspicious behavior as discussed by Kim, et al. [11]. The drive by download attacks is mostly triggered through these functions as an entry point to redirect users to the attacker's server. Egele, et al. [4] had shown that Drive-by download attacks can be detected and prevented by employing dynamic analysis. This could be done by checking for strings loaded with shell code during runtime in Javascript engine where attacks are memory based. Phung, et al. [6] had showed how document.write() function and script tags are used by malicious Javascripts to inject malicious codes dynamically. Dynamic analysis can be carried out in a sandboxed environment where file system and registry changes can be monitored to harness the malicious/suspicious patterns as discussed by Provos, et al. [14]. Christodorescu, et al. [16] has shown how to build semantic aware malware detection using behavior analysis on the binaries. Cova, et al. [17] and Xu, [25] shows how static and dynamic analysis features can be used efficiently to detect obfuscated javascript.

The lexical features of the URLs queried by malicious javascript in sandboxed environment have also been used for analysis. Maliciousness of URL is estimated based on the URL or IP based reputation system. Lexical analysis on URLs can estimate the maliciousness of URLs faster than reputation based system as discussed by Darling, et al. [7]. Canali, et al. [26] work shows that using URL components like domain name, subdomain name, IP address plays important role in ranking the maliciousness of a URL.

3 Attack Methodologies

Attackers writing malicious javascript tend to use the following methods to do malicious activities. These methodologies are widely used by attackers to spread across web swiftly. The attack methods can be classified as follows:

3.1 Malicious URL

Malicious URLs can be injected into benign webpages where third party plugins or javascript used can become backdoor for attackers to inject malicious javascript codes or iFrames. Figure 1 shows an iFrame holding readable malicious URL which is not viewable by users as its having too small height and width properties. So this made attackers to easily participate in wide range of attacks reaping lots of victims. Where victim’s machines can be used to carry out Denial Of Service attacks or personal information can be stolen by attacker in stealth way (like phishing attacks). Li, et al. [21] work shows that how drive-by download, scam, phishing attacks are carried out by URL redirection. An URL within iFrame need not be invisible to perform malicious/suspicious activity as Provos, et al’s [14] work shows that iFrames were used to generate artificial traffic and earn money. Also the same work shows that how efficiently iFrames were used to carry out drive-by download attacks. Zarras, et al. [24] shows that most of the URLs have suspicious redirection resulting one to be malicious webpage.

```
<iframe src="http://www.maliciouswebsite.com/
malicious_ad.js " width=1 height=1 style="visibility
:hidden;position:absolute"></iframe>
```

Fig. 1. Bad iFrame with readable URL.

```
eval(unescape( '%64%6f%63%75%6d%65%6e%74%2e%77%72%69%74%65%28%27%3c%
61%20%68%72%65%66%3d%22%6d%61%69%6c%74%6f%3a%69%64%62%40%62%61%63%6b
%62%65%61%74%6d%65%64%69%61%2e%63%6f%6d%3f%73%75%62%6a%65%63%74%3d%
69%44%6f%77%6e%6c%6f%61%64%42%6c%6f%67%25%32%30%41%64%76%65%72%74%
69%73%69%6e%67%25%32%30%49%6e%66%6f%25%32%30%52%65%71%75%65%73%74%
22%3e%41%64%76%65%72%74%69%73%65%3c%2f%61%3e%27%29%3b' ))|
```

Fig. 2. eval() and unescape()

As shown in CodeSnippet1 below which decodes to location. Replace ([‘http://maliciousurl.com’](http://maliciousurl.com)) static analysis may fail to capture the URL as its generated during runtime. This shows that attackers use of dynamic code injection into users web page for better infection. Also the static URLs may be shortened URLs which in turn will redirect users to the attacker’s website. The recent incident of code injection and URL redirection attack on ebay [15] shows the effective use of code injection and URL redirection on online shopping sites.

```
<SCRIPT language=JavaScript>  
function otqzyu(nemz)juyu="lo";sdfwe78="catio";  
kjj="n.r";vj20=2;uyty="eplac";iuiuh8889="e";vbb25="('";  
awq27="";sftftttft=4;fghdh="'ht";ji87gkol="tp:/";  
polkiuu="/mal";jbj89="icio";jhbhi87="us";hgdxgf="ur";  
jkhuift="l.c";jygyhg="om'";dh4=eval(fghdh+ji87gkol+  
polkiuu+jbj89+jhbhi87+hgdxgf+jkhuift+jygyhg);je15="'");  
if(vj20+sftftttft==6)eval(juyu+sdfwe78+kjj+uyty+  
iuiuh8889+vbb25+awq27+dh4+je15);  
otqzyu();//  
</SCRIPT>
```

3.2 Malicious Use of JavaScript Code

Shell codes and eval(). Injecting shell codes or making use of vulnerability in the third party libraries are effective ways of gaining control over the user’s machine and spreading malware. These injected shell codes are crafted for targeted application/libraries vulnerability. Mostly buffer overflow attacks are carried out to gain root access by attacker.

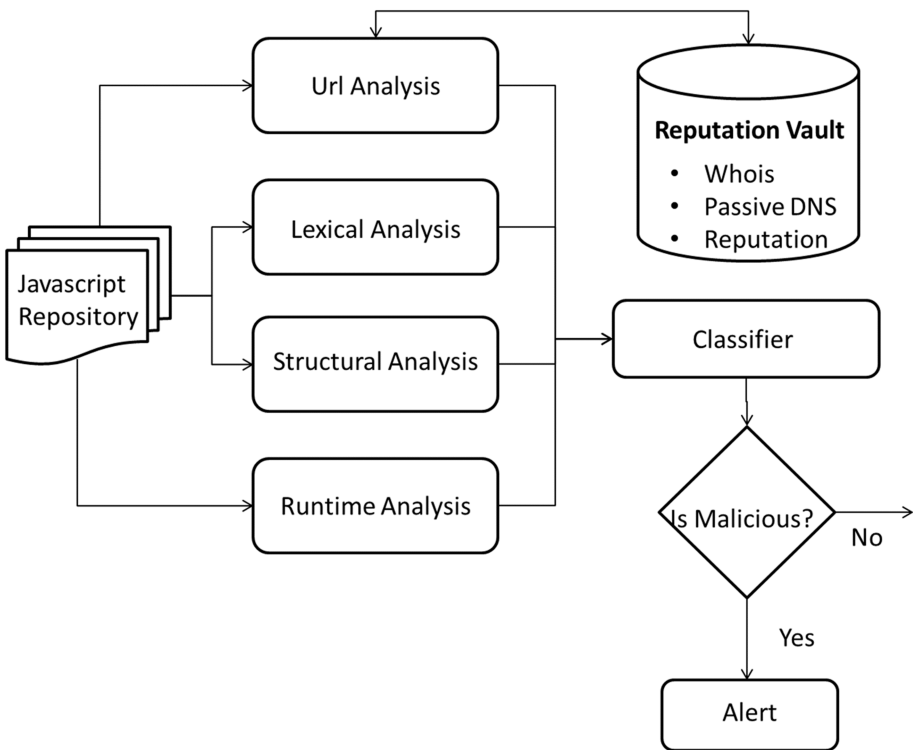


Fig. 3. System flow diagram

This work [8] shows that attackers make use of `unescape()` and `eval()` functions extensively to carry out the attacks using obfuscated codes. `eval()` functions can be used to generate URL in a complex way with `unescape()` function. Figure 2 shows that how `eval()` and `unescape()` functions are used to embed the obfuscated javascript codes and used to carry out attacks. Attacker can hardcode the URLs or can generate the URLs with series of operations recursively with more randomness in URL to evade during cyber forensics leaving the attacker to stay anonymous. This shows that a user's machine can be infected just by landing users on the attacker's page or because of user's interaction on the attacker's page. Other than shell code and `eval()` functions `document.write()` and `unescape()` as presented in Provos, et al's [14] and Feinstein, et al. [20] works these functions are widely used to inject obfuscated codes in multiple layers to carry out drive-by download attacks. Hallaraker, et al. [18] and Hedin, et al. [19] works shows that how dom elements are manipulated to carry out malicious activity through obfuscated javascript.

4 Methodology

In the proposed MalJs detection framework four types of analyses are performed on the javascript codes to distinguish the benign and the malicious scripts. They include:

4.1 URL Analysis

It has been proven that a malicious web page could be identified by just analyzing its URL [7, 26]. This analysis is performed only on the URL of the javascript code. The objective is to find:

- The maliciousness of the URL
- Historic Reputation of the URL
- Historic Reputation of the IP addresses resolved by the URL

4.2 Lexical Analysis

A lexeme is a string of characters which forms a syntactic unit. Performing lexical analysis gives an idea about the basic building blocks and the semantics of the javascript code. The lexical analysis involves finding:

- N-Gram (Uni, Bi, Tri and Quad grams) frequency of the tokens in the code [9]
- Vowel to consonant ratio
- Digit to letter ratio
- Average length of tokens in the code

4.3 Structural Analysis

This analysis involves inspecting the javascript code and looking for the attack patterns and types mentioned in Sect. 3. The objective in this analysis is to find the presence of:

- Hidden iFrames.
- Count and Percentage of whitespaces.
- Number of *unescape()* function used.
- Number of *eval()* function used.
- Percentage of Shell code.

4.4 Run-Time Analysis

In the run-time analysis, the dynamic properties of the javascript are checked in a controlled safe environment. The javascript code is analyzed to examine the dynamic properties including:

- Registry changes
- Processes and sub-processes spawned
- Network monitoring
- Memory changes
- File Permission changes
- Number of dynamic code executions
- Length of dynamically evaluated code

5 Design and Implementation

5.1 URL Analysis

This module makes use of all the URLs captured during the runtime analysis. Using the network traffic we can harvest all the URLs queried by that javascript. All these URLs are then allowed to be lexically analyzed [13] and by checking their domain and IP reputations with an actively updating reputation database for their maliciousness.

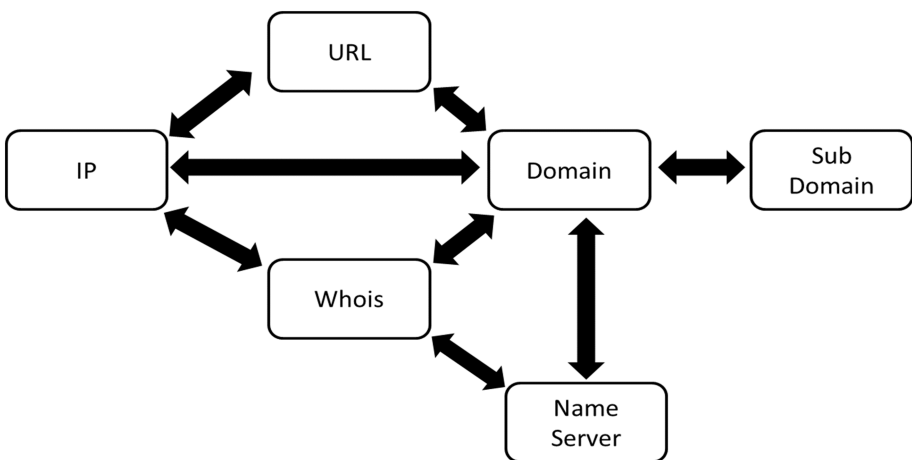


Fig. 4. Graph correlation of reputation vault components

Though reputation based analysis provides historical evidence of a URL, it needs to be updated frequently. Apart from the reputation checkup, the analysis of the URL is also performed to find malicious patterns.

IP and Domain based reputation system is building a passive DNS reputation model as in [30–32] which is stored in a distributed graph database. Mapping the subdomain to the domain they belong to allows us to flag a domain as suspicious based on the number of malicious subdomains found under that particular domain. Figure 4 shows the graph connection of the reputation vault.

5.2 Lexical Analysis

This analysis collects a javascript's n-grams (Uni to Quad grams), and the ratio of vowels, consonants, digits, letters and average length of tokens makes it possible for a machine learning process to extract the semantics used by the attacker and recognize his semantic pattern. All these features are extracted from good and malicious javascript after scanning them. The extracted features are stored to feed them into classifier to build the final model.

5.3 Structural Analysis

i. Hidden iFrames: An iFrame is tagged hidden when its size ranges from too small to negative. These types of iFrames are tagged hidden and hence suspicious. Shown in the example below is an iFrame having too small height and width. These iFrames serves no purpose for the user as it won't be visible for users. Such iFrames are used by attackers to carry out malicious activities.

```
<iframe width="1" height="1" frameborder="0" scrolling="no" marginheight="0" marginwidth="0" src="http://malicious-network.com/getmaliciousfile.php">
</iframe>
```

ii. Unescape() and eval() Functions: Structural analysis also scans for the presence of unescape() and eval() functions. The unescape() and eval() combination are used frequently to inject shell code or malicious javascript, where lots of escape() functions usage tends to generate shellcode. Attacker can make use of this to take control over the victim's machine by exploiting the vulnerabilities found in browser and third party libraries used in guest OS.

iii. Shell Code: As part of structural analysis all the javascript are scanned for presence of shellcode using regular expression. If the percentage of shellcode is more than a particular threshold, then the javascript code becomes more suspicious.

5.4 Run-Time Analysis

Run-Time analysis as shown in the below figure checks the behavior of each javascript file by visiting the webpage the javascript is used in a sandboxed environment like cuckoo. Cuckoo can keep track of network traffic generated; all types of file operations, memory changes, registry changes when the javascript was loaded. Cuckoo uses OS snapshots to do a clean run or makes sure to avoid infected lineage runs. This analysis draws boundary for the suspicious behavior of javascript during suspicious redirects, local file changes.

All these analysis is done on both good and bad javascript from predefined repository of javascript for training purposes. All the features are extracted for any given javascript and sent to classifier to carry out classification and detects the javascript is malicious or not.

Classification Algorithm: Random Forest, is an average of many deep decision trees which are trained on the different parts of same training set. Also this is an ensemble based supervised classifier. The advantage of using Random Forest algorithm is that, it predicts the output by creating multiple decision trees keeping different features as the root nodes. The advantage of having multiple trees with different root nodes is that it removes/reduces the over-fitting of the data which is very common with the individual decision trees. Also since this classifier predicts the output based on the decision provided by multiple trees, the accuracy is generally good and gives equivalent results when compared with non-linear SVMs which are both time and computationally expensive.

The Random Forest classifier model was trained with a training set of 6240 unique instances of good and malicious javascript. Both the classes had equal amount of data i.e. 3120 instances each to avoid any bias towards any class. A Random Forest model was developed consisting of 10 trees, each constructed while considering 4 random features. One of the advantages of Random Forest classifier is that it doesn't need any cross validation or a distinct test case set for evaluating estimating the efficiency of the model. It has an internal estimation procedure known as out-of-bag (OOB) error estimation which provides unbiased estimate from the given training dataset. The OOB error estimate of the trained model is 0.1228.

6 Evaluation of the System

Our crawled dataset contains more than 6,200 JavaScript and Html files which includes both good and bad datasets. We used Alexa's top 100 websites [28] and listings from malwaredomainlist [29] for mining our dataset. Table 1 compares and shows the different classifiers accuracy result set using the above mentioned features in Sect. 4. Our experimental setup shows that random forest classifier provides better accuracy in discriminating benign and malicious javascripts. Table 2 provides the detailed class-wise accuracy achieved by the random forest classifier (Table 3).

Table 1. Classifier performance comparison

Classifier	TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area	Accuracy
RandomForest	0.889	0.111	0.889	0.889	0.889	0.939	88.8782 %
BFTree	0.888	0.112	0.892	0.888	0.888	0.904	88.8301 %
J48graft	0.887	0.113	0.891	0.887	0.887	0.916	88.734 %
J48	0.887	0.113	0.89	0.887	0.887	0.916	88.6859 %
ADTree	0.875	0.125	0.888	0.875	0.874	0.912	87.5321 %
DecisionStump	0.871	0.129	0.895	0.871	0.869	0.865	87.0673 %
LibSVM	0.866	0.134	0.875	0.866	0.865	0.866	86.5545 %
NaiveBayes	0.863	0.137	0.879	0.863	0.862	0.878	86.3301 %

Table 2. Detailed accuracy result by class for random forest

Class	TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area
Malicious	0.907	0.129	0.875	0.907	0.891	0.939
Benign	0.871	0.093	0.904	0.871	0.887	0.939
Weighted Avg.	0.889	0.111	0.889	0.889	0.889	0.939

Table 3. Confusion matrix

	Malicious	Benign
Malicious	2830	290
Benign	404	2716

7 Conclusion

Malicious javascript’s existence have an adverse effect on the Internet users by making users vulnerable to drive-by-downloads, ransomware attacks, URL redirection, phishing attacks, making all victims participate in Denial Of Service attack, disguise as a legitimate seller on online shopping sites etc. As the malwares are crafted specially to exploit a specific vulnerability from which attacker can get lots of victims, proper measures has to be taken to keep the vulnerable packages updated. Also by embedding the shellcode injection detection mechanisms in the environment which is exposed to such attacks can prevent lots of victims. In this research, we have built an ensemble based supervised classification model to detect the presence of malicious javascripts by analyzing its lexical, structural and behavioral patterns. To improvise the efficiency of the model we also employ a reputation vault having historical evidence of the URL, domain and IP address. The proposed system can detect malicious javascripts present in the internet with a low false positive rate of 0.1.

References

1. Hao, Y., et al.: JavaScript malicious codes analysis based on naive bayes classification. In: 2014 Ninth International Conference on P2P, Parallel, Grid, Cloud and Internet Computing (3PGCIC). IEEE (2014)
2. Santos, I., et al.: N-grams-based file signatures for malware detection. *ICEIS* **2**(9), 317–320 (2009)
3. Wang, W-H, et al.: A static malicious javascript detection using SVM. In: Proceedings of the International Conference on Computer Science and Electronics Engineering, vol. 40 (2013)
4. Egele, M., Wurzinger, P., Kruegel, C., Kirda, E.: Defending browsers against drive-by downloads: mitigating heap-spraying code injection attacks. In: Flegel, U., Bruschi, D. (eds.) DIMVA 2009. LNCS, vol. 5587, pp. 88–106. Springer, Heidelberg (2009)
5. Choi, Y., Kim, T., Choi, S., Lee, C.: Automatic detection for javascript obfuscation attacks in web pages through string pattern analysis. In: Lee, Y.-h., Kim, T.-h., Fang, W.-c., Ślęzak, D. (eds.) FGIT 2009. LNCS, vol. 5899, pp. 160–172. Springer, Heidelberg (2009)
6. Phung, P.H., et al.: Between worlds: securing mixed javascript/actionscript multi-party web content. In: *IEEE Transactions on Dependable and Secure Computing*, 12 April 2015, pp. 443–457 (2015)
7. Darling, M., et al.: A lexical approach for classifying malicious URLs. In: 2015 International Conference on High Performance Computing and Simulation (HPCS). IEEE (2015)
8. Fraiwan, M., et al.: Analysis and identification of malicious javascript code. *Inf. Secur. J. A Global Perspect.* 1–11 (2012)
9. Fang, Z., et al.: A half-dynamic classification method on obfuscated malicious JavaScript detection. *Int. J. Secur. Appl.* **9**(6), 251–262 (2015)
10. Provos, N., Mavrommatis, P., Rajab, M.A., Monroe, F.: All your iframes point to us (2008)
11. Kim, B.-I., Im, C.-T., Jung, H.-C.: Suspicious malicious web site detection with strength analysis of a javascript obfuscation. *Int. J. Adv. Sci. Technol.* **26**, 19–32 (2011)
12. Choi, J., et al.: Efficient malicious code detection using n-gram analysis and SVM. In: 2011 14th International Conference on Network-Based Information Systems (NBiS). IEEE, 2011
13. Chong, C., Liu, D., Lee, W.: Malicious URL detection
14. Provos, N., et al.: The ghost in the browser: analysis of web-based malware. In: *HotBots 2007*, p. 4 (2007)
15. Mutton, P.: EBay scripting flaws being actively exploited by fraudsters. *Netcraft*, 18 February 2016. Web, 4 June 2016. <http://news.netcraft.com/archives/2016/02/18/ebay-scripting-flaws-being-actively-exploited-by-fraudsters.html>
16. Christodorescu, M., et al.: Semantics-aware malware detection. In: 2005 IEEE Symposium on IEEE Security and Privacy (2005)
17. Cova, M., Kruegel, C., Vigna, G.: Detection and analysis of drive-by-download attacks and malicious JavaScript code. In: Proceedings of the 19th international conference on World wide web. ACM (2010)
18. Hallaraker, O., Vigna, G.: Detecting malicious javascript code in mozilla. In: Proceedings. 10th IEEE International Conference on Engineering of Complex Computer Systems, ICECCS 2005. IEEE (2005)
19. Hedin, D., Sabelfeld, A.: Information-flow security for a core of JavaScript. In: 2012 IEEE 25th Computer Security Foundations Symposium (CSF). IEEE (2012)
20. Feinstein, B., Peck, D., SecureWorks, Inc.: Caffeine monkey: Automated collection, detection and analysis of malicious javascript. *Black Hat USA 2007* (2007)

21. Li, Z., et al.: Knowing your enemy: understanding and detecting malicious web advertising. In: Proceedings of the 2012 ACM Conference on Computer and Communications Security. ACM (2012)
22. Likarish, P., Jung, E., Jo, I.: Obfuscated malicious javascript detection using classification techniques. In: MALWARE (2009)
23. Seifert, C., Welch, I., Komisarczuk, P.: Identification of malicious web pages with static heuristics. In: Australasian Telecommunication Networks and Applications Conference. ATNAC 2008. IEEE (2008)
24. Zarras, A., et al.: The dark alleys of madison avenue: understanding malicious advertisements. In: Proceedings of the 2014 Conference on Internet Measurement Conference. ACM (2014)
25. Xu, W., Zhang, F., Zhu, S.: JStill: mostly static detection of obfuscated malicious JavaScript code. In: Proceedings of the Third ACM Conference on Data and Application Security and Privacy. ACM (2013)
26. Canali, D., et al.: Prophiler: a fast filter for the large-scale detection of malicious web pages. In: Proceedings of the 20th International Conference on World Wide Web. ACM (2011)
27. Jim, T., Swamy, N., Hicks, M.: Defeating script injection attacks with browser-enforced embedded policies. In: Proceedings of the 16th International Conference on World Wide Web. ACM (2007)
28. Top Sites. Alexa Top 500 Global Sites. Web, 4 June 2016. <http://www.alexa.com/topsites>
29. Malware Domain List. Web, 4 June 2016. <https://www.malwaredomainlist.com/mdl.php>
30. Zdrnja, B., Brownlee, N., Wessels, D.: Passive monitoring of DNS anomalies. In: Hämmerli, B.M., Sommer, R. (eds.) DIMVA 2007. LNCS, vol. 4579, pp. 129–139. Springer, Heidelberg (2007)
31. Bilge, L., et al. EXPOSURE: finding malicious domains using passive DNS analysis. In: NDSS (2011)
32. Antonakakis, M., et al.: Building a dynamic reputation system for DNS. In: USENIX Security Symposium (2010)

SocialBot: Behavioral Analysis and Detection

Madhuri Dewangan^(✉) and Rishabh Kaushal

Department of Information Technology, Indira Gandhi Delhi Technical University
for Women, New Delhi, India
madhuri.dewangan.15@gmail.com, rishabhkaushal@igdtuw.ac.in

Abstract. Bots refer to automated software that have the capability to execute commands on receiving instructions from BotMaster. *SocialBots* are the bots present in Online Social Network (OSN) which mimic the activities of the real users. They have the capability to automatically perform the basic functionalities offered by the OSN platforms. These socialbots have widespread usage in political campaigning and product marketing, but *SocialBots* can also be used for the purpose of swaying voters, mounting political attacks, manipulating public opinion, etc. Apart from these, *SocialBots* possess various security risks, one of which is befriending an OSN user thereby gaining access to personal details such as birthday, email id, phone number, address, etc. Detection of these *SocialBots* is therefore an important problem to be solved in order to maintain the reputation of OSN. Our work concentrates on behavioral analysis of these SocialBots in the OSN and identifying features to be used to develop a model for detection of these Socialbots using machine learning. The model, thus developed, is further used as a background process to create a web-based tool for detection of SocialBots. In our work, we created a SocialBot to perform behavioral analysis. This SocialBot got a good response from the real users and was able to grab 100+ real followers along with some real interactions in form of retweet, mention and direct messages.

Keywords: Online Social Network · Social engineering · SocialBots · Feature extraction · Machine learning

1 Introduction

Online Social Networks (OSNs) provide virtual connectivity, thereby helping users to keep in touch with person who are miles apart and aim to improve social togetherness. Due to their widespread use, people from various section of society, particularly, politicians and businessman are using it increasingly to reach to their voters and consumers, respectively. As a result, popular OSN sites such as Facebook, Twitter etc. are being used for the purpose of campaigning and marketing. *SocialBots* have emerged as potent tool for social engineering attacks which are used to entrap the victim into revealing their sensitive information which can be used for performing masquerading and helping in identity theft. Hill et al. [13] reports that “Facebook thinks 83 million of its user are fake”

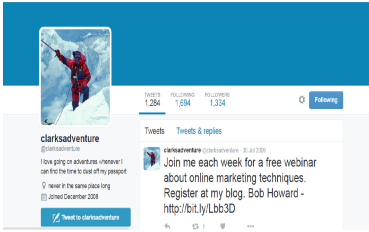


Fig. 1. SocialBot created by Greg Marra [16]

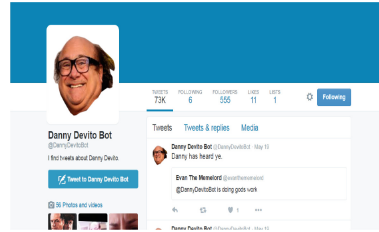


Fig. 2. SocialBot identified during observation.

and that “Upto 29.9 % of Barack Obama’s 17.82 million [Twitter] followers and 21.9 % of Mitt Romney’s 814000 followers may be fake” [13]. Thus it is evident that the phenomena of presence of SocialBots in OSNs is widespread and some of them are found to behave with malicious intent. Figures 1 and 2 are couple of examples of SocialBots on Twitter. From the accounts like these it can be very observed that SocialBot accounts are very actively involved in these OSNs. The focus of this work is to perform behavioral analysis of SocialBots and detect them in order to prevent such fraud and manipulation from happening in OSNs. Our approach to address the problem is to study the issue from a user’s perspective as shall be explained later.

2 Related Work

Since the time the phenomena of SocialBots have emerged, a lot of effort has been made towards analysis and detection of the SocialBots. There has been recent interest towards detection of SocialBots due to its inherent disadvantage of persuading a mob over a topic. Different work done so far are summarized in this section.

“A social bot is piece of software that is designed to have a presence on the Internet especially on social media and is engineered to achieve some purpose. Social bots can be used to create a smear campaign, to promote legislation. Social bots can make a website, a hashtag, a person or user become popular” [9]. Boshmaf et al. [1] has explained in 2011 how the SocialBots can infiltrate Facebook on large scale using its Graph API and interact with the real users. They had collected the 250 GB of data of real user in response to the large scale infiltration. From this successful infiltration they made various conclusions namely (1) Facebook can be infiltrated with a success rate of upto 80 %, (2) Depending upon user’s privacy setting, successful privacy breach can occur and (3) Facebook Immune System was not effective enough in detecting such large scale infiltration. Wagner et al. [2] modelled susceptibility of real users towards SocialBots in OSN to determine who are the users who are most likely to accept the network of a SocialBots. Using features based on user’s network and user’s behavior, they were able to predict the susceptibility of the user for connecting to SocialBots.

The conclusion drawn from the study was that the susceptible user tends to use the OSN for conversation purposes [2], use more social words and show more affection than non susceptible user. Analysis of the activities of the SocialBots leads to the identification of the issues and challenges which arise with the detection of these SocialBots. Boshmaf et al. [3] had categorised these challenges in three categories namely web automation, identity binding and user security. Some of the challenges described in paper were strong CAPTCHA as it poses a problem to the real user, large scale crawling, online offline identity binding, usable security and privacy controls. Nowadays, it is observed that SocialBots, instead of targeting the entire domain of user, concentrate on the special group of users such as those belonging to any organization, this was explained by Elisher et al. [6] who used the concept of SocialBot to explain how the SocialBots can be used for the purpose of mining information from users belonging to a targeted organization. Here the authors instead of targeting entire domain of OSN platform, wisely select the organization and infiltrate users of that organization. Their results are of a greater concern because the employees targeted belonged to technology companies who are assumed to have technical awareness and yet they got entrapped. They had stated that any OSN platform can be subjected to large scale infiltration, users privacy settings can result in privacy breaches and the large scale infiltration. In demonstrating such an infiltration, the authors created 102 socialbots and earned 8,750 friend requests of which 3,055 requests were accepted. This infiltration was 80% successful within the run of 8 weeks. Results of their work were that possibilities of request getting accepted increases with mutual friends and considerably large number of friends also effect the acceptance rate. They concluded that CAPTCHA is not efficient mechanism to help them to take informed decision while accepting the new connection. According to them the social-technical solution are required to mitigate the problem of SocialBot Attack. Ferrara et al. [4] described the *feature based SocialBot detection* to differentiate the real human user from that of computer algorithm i.e. *Socialbots* as SocialBots have improved with the time and have became more sophisticated, as they perfectly imitate the real human user of OSN. The authors had identified several features to compare the human users and SocialBots, their activity and responses with respect to these feature to distinguish the SocialBots. They provided an implementation to their algorithm for detection of Socialbots as well.

3 Proposed Approach

In our research work we have proposed a method to detect whether, the given screen name (it refers to user name in Twitter) is a SocialBot or not. From our investigations, we came to know that each user profile on the OSN platform has some information within the OSN framework which can be used to detect SocialBots. In case of Twitter, this information comprises of screen name, url(s), hash-tags, mentions, followers, following, username, created at, source, list, favourite, retweet count, tweet id and several other information about user who tweets

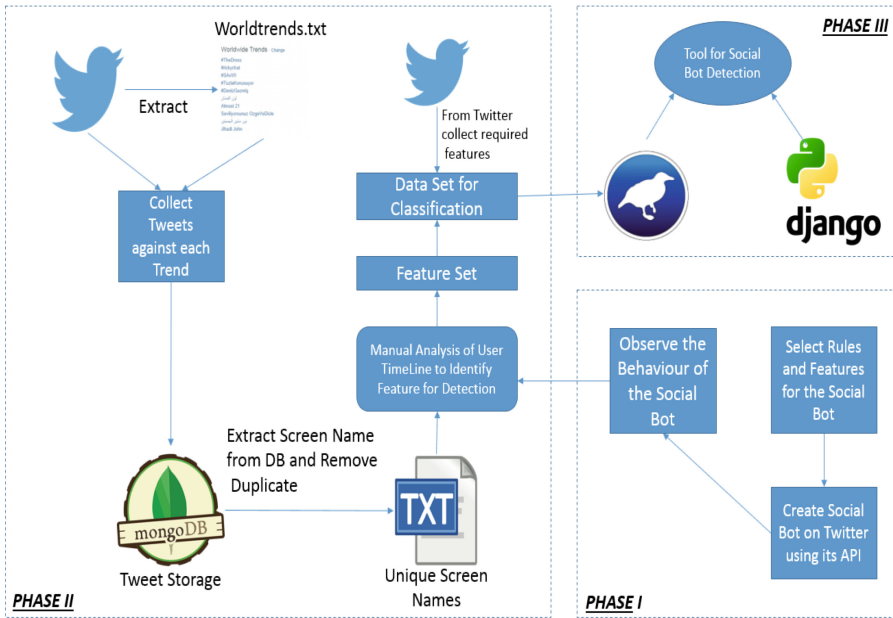


Fig. 3. The flow of work for classification of SocialBot.

and regarding the tweet timing and many more. For the purpose of detection, the workflow described in Fig. 3 was used. Here we first created a SocialBot in *phase I* in order to observe the behaviour of the SocialBot. Then in *phase II*, we had identified the features for detection of SocialBot and created dataset for detection. Finally in *phase III*, we classified the computed dataset and created a tool to automate the process. Its different phases and stages are described in detail in subsequent subsections.

3.1 Phases of Proposed Approach

The research work was organised in three phases as depicted in the Fig. 3:

Phase I: Creation of SocialBot, in which we had created SocialBot for twitter to observe its activity and behavior in Twitter. Also it helped to entrap some of the SocialBots for the detection purpose. The SocialBot that we created was able gather 100+ followers and managed to interact with other users like real user.

The activities of the Socialbot is clearly visible from the Figs. 4 and 5. With the help of Twitter API, we were very easily able to create a SocialBot. The SocialBot created was regularly tweeting, retweeting and favoriting to attract more followers. This experiment helped us to identify different features of the Social Bots and their activities.



Fig. 4. SocialBot created for observing behavior of SocialBot.

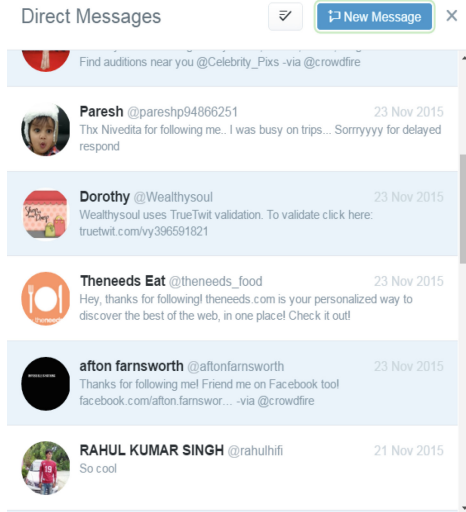


Fig. 5. Interaction received from user by Socialbot create by us.

Phase II: Analysis and Detection. Next phase after observing the behaviour was to identify the feature set for detection of SocialBots. This phase is most crucial for detection of SocialBots. The following procedure is used for identifying the feature set.

1. **Tweets Collection:** To create dataset for analysis and detection we needed tweets. Tweets were obtained using Twitter API [15] specifically using the wrapper library called Tweepy [18]. To collect tweets we followed approach as below:
 - Captured top world trends using automated python scripts. The world trends dated 27-02-2016 were captured.
 - For each world trend captured, we collected tweets using python scripts and stored them in a database.
2. **Preprocessing:** From database we extracted *unique* users because the same user may have tweeted in different world trends.
3. **Feature Identification:** To identify the difference between SocialBots and users, we manually inspected near about 200 timelines, to get an understanding in regard to differences in the presentation of bot and real user visually. These 200 accounts included 100 user profiles and 100 known automated profiles of brands and some of them were bots given in list [14], which we had searched. One of the most prominent feature observed was that *bots mostly tweets for each world trend* to make itself visible to larger audience, for interaction.
4. **Dataset Creation:** After identifying the features which helped us to distinguish among legitimate user and socialbot from manual analysis mentioned

Table 1. Dataset details

Analysis	Number of screen_names
Total tweets collected	10432
Total unique screen name extracted	3392
Total users profile already blocked	208
Total users having privacy setting enabled	159
Total instances to create dataset	3025
Number of socialbots	467
Number of legitimate User	2557

in above step, we created dataset using automated methods. Data computed at our end contains *3,025 instances out of which 2,557 are legitimate users and 467 are SocialBots*. Table 1 summarises the dataset that had been created by us.

Phase III: System Development. The final phase is system development. The instances of dataset were classified against five machine learning algorithms namely *Random Forest, ADT tree, Naive Bayes, LAD tree and j48* and their results were compared. Out of 5 models, the model developed by Random Forest algorithm was found to be most suitable for detection of SocialBots. This model was then used for creation of the system which was developed using Django Framework [19]. For any Screen name the user enter, the system gives the result as bot or user depending upon its features collected.

3.2 Assumptions

For creation of SocialBot in Twitter and attracting real users, SocialBot ought to have the following characteristics:-

- Attractive Display Picture to grab visitors attention.
- Proper Information about the SocialBot as a real user.
- Operation in normal usage time of the real user.
- Prompt reaction to activities from other user such as *Retweet, Direct Message, Mention*.

To mark the instances as bot or real user, manual analysis done for feature identification took following into consideration:

- Bots basically work on the programmed scripts and do not show the dynamic behaviour hence for every given feature they would not show any strikingly increasing or decreasing trend.
- They can either be retweeting based on certain keyword or user or posting urls if they frame their own tweet.

Table 2. Feature set

Features	Its relevance with SocialBot
Average mention	Not more than 2 mentions per tweet
Average hashtag	More than 2 hashtags per tweet
URL count	More than 30% of all tweets in timeline contain Urls
Retweet count	Either all tweets are retweet or none is retweet but not both
Original tweet count	Either all tweets are fresh tweets or all is retweet but not both
InterTweet delay	Frequent posting
Number of sources	0 or 1 as they post via APIs

- Normal user posts from various sources like from the Web Client, Twitter for iPhone, Twitter for Android, Twitter for Mac etc., bot usually operated the account from the single source.
- Bot posted on regular interval and predefined time period as they are programmed unlike human who tweet when they want, so their tweet time patterns has randomness and for SocialBot, there is some regularity.
- Bots try to present themselves to larger audience so they uses more hashtags and posts URLS with almost every tweet where as users they mention more user to communicate.

3.3 Features

Identifying features is a crucial step, Table 2 describes features in detail which are further explained in this section.

- **Average Mention:**

Number of mentions a user posts per tweet is termed as **Average Mention**. Legitimate users generally get involved in the conversation with other users so they usually mention other users in every tweet, while in case for the Socialbots, they are designed for the specific purpose so it is not necessary every time they be mentioning other screen names.

- **Average Hashtag:**

Number or hashtags a user posts per tweet is **Average Hashtags**. For the legitimate user they will not post hashtags more than 2 or 3 per tweet where as bots uses greater than 3 hashtags per tweet as it gives them opportunity to mark their presence. Usually Socialbots captures the trending hashtags and posts them to gain the attention of other users.

- **URL Count:**

Total count of URLs posted by the user in his timeline is termed as **URL count**. This is very important feature to look for as user usually does not post URLs frequently but Socialbot users the URLs as means of promotion thus for any account suspected for Socialbot they might be having more than 300 urls posted in their original tweets.

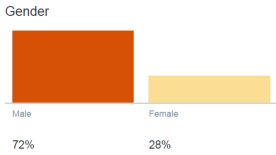


Fig. 6. Distribution of socialbots follower on the basis of gender.

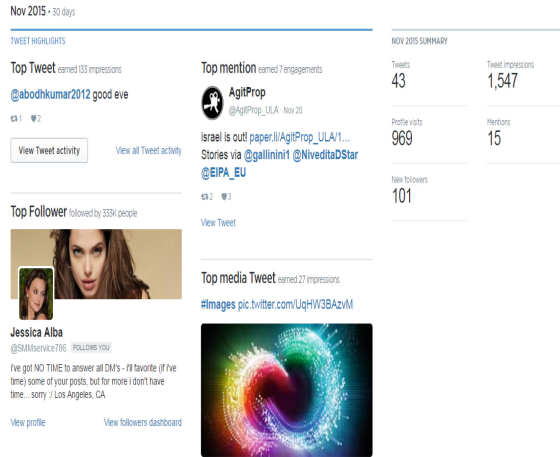


Fig. 7. Summary of SocialBot activity.

– **Number of Sources:**

Total count of platforms from which the users posts their tweets is termed as **Number of Sources**. Now for the legitimate user this field will be very high as users, generally uses different apps, platform and operating system to post the tweet, where as the Socialbot, since are operated from scripts they do not have different source to post from.

– **Retweet Count:**

Total number of retweets posted by the user is termed as **Retweet Count**. For the Socialbot account the Retweet Count will be very high and will be having no Original Tweet and vice versa for the legitimate users, as such Social bots are designed to survive on the retweets.

– **Original Tweet Count:**

Total number of original tweet posted by the user is termed as **Original Tweet Count**. For the Socialbot account the original tweet will be very high and will be having no Retweet Count and vice versa for the legitimate users. These are those bots which will be posting urls and images.

– **Intertweet Delay (in hrs):** The average difference between two consecutive posts is termed as **InterTweet Delay**. Socialbot will be having smaller intertweet delay as they have script running behind thus they tweet more often and in regular interval whereas the users will be having larger intertweet delay as compared with the Socialbots.

4 Observations and Results

Results of Phase I clearly depicts that its very easy to create a SocialBot and interact with the real users.

The distribution of followers with respect to gender in Fig.6 depicts that female SocialBot grabs more male followers. Figure 7 helps to identify how the

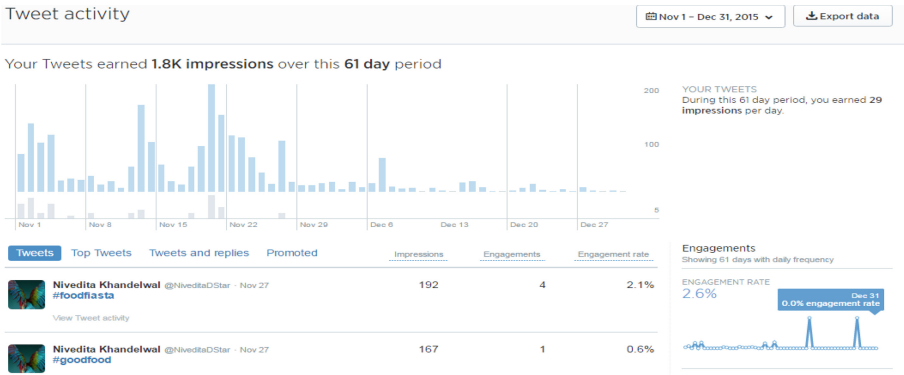


Fig. 8. Activity of SocialBot created by us.

Table 3. Observations during dataset creation

Analysis	Number of screen_names
Total users timeline analyzed	3392
Total user time captured	3025 (out of which 2557 are Legitimate users and 467 are SocialBots)
Total Users profile already blocked	208
Total users having privacy setting enabled	159

socialbot grabs the attention of the follower to interact with it as we can see one of its mention has gained seven engagements¹.

Figure 8 [17] clearly depicts the activity of SocialBot. This shows impression² that Socialbot was able to obtain over that period, thus denoting the reach of tweet made by Socialbot.

The result of the Phase I has helped us to identify the Socialbot in Twitter also it gave the glimpse of the feature which we later used in the Feature Set. During manual analysis of 200 users’ timelines, feature set was created comprising of 100 of legitimate users and 100 of SocialBots. Those were summarized in Table 3. Most Important observation was “Not Authorized” error which was returned by the API when either the account has privacy setting enabled or the account is suspended by Twitter.

Figure 9 gives the comparison of the various classifier’s result. The Red Bar on the graph depicts the performance for Random Forest. From the graph we can see that the FP Rate for both Bot and User is least for Random Forest classifier. Thus **Random Forest Classifier Model** is used for Detection.

¹ Number of times users has interacted with the Tweet. This include all clicks anywhere on the tweet [17].

² When we say “impression”, we mean that a tweet has been delivered to the Twitter stream of a particular account [20].

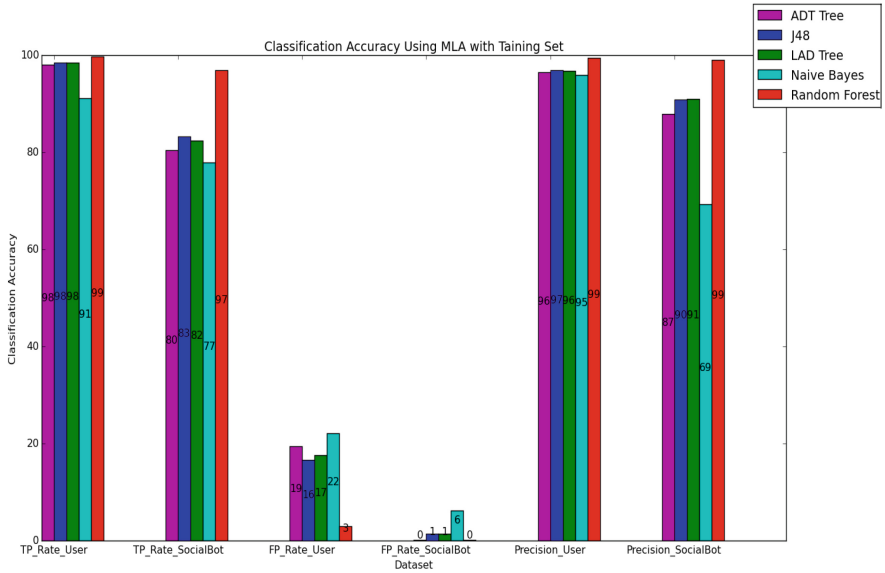


Fig. 9. Comparison of results of the classifier.

5 Conclusion and Future Work

The study and experimentation done for the behavioral analysis of Socialbot has help us to conclude several important points. First of all, it is very easy to create a SocialBot in Twitter using its API and infiltrate among the users in the network. Secondly, people in the social network have casual attitude towards incoming connection request for any user. Most of User accept the request regardless they know the other user or not, this might lead them to face social engineering attack from a malicious user or SocialBot. Thirdly, for an account driven by program it is very easy to capture personal information about the target user if, that user has not enabled the privacy setting. Thus leading to a identity theft attack. Detection Phase of the research uses account based features for detection which makes it faster and reliable mechanism for Detection of the SocialBot. Further in this project we can increase the dataset to further improve the accuracy, embed classifiaction of good socialbot and bad socialbot, extend it to identify what percentage of the followers of a given screen name are socialbot to determine the person’s actual popularity.

References

1. The socialbot network: when bots socialize for fame and money. In: Proceeding ACSAC 2011 Proceedings of the 27th Annual Computer Security Applications Conference, pp. 93–102. ACM, New York (2011)

2. Wagner, C., Mitter, S., Strohmaier, M., Krner, C.: When social bots attack: modeling susceptibility of users in online social networks. In: MSM (2012)
3. Boshmaf, Y., Muslukhov, I., Beznosov, K., Ripeanu, M.: Key challenges in defending against malicious socialbots. In: 5th USENIX Workshop on Large Scale Exploits and Emergent Threats LEET 2012 (2014)
4. Ferrara, E., Varol, O., Davis, C., Menczer, F., Flammini, A.: The Rise of Social Bots
5. Hwang, T., Pearce, I., Nanis, M.: SocialBots: voices from the fronts. Mag. Interact. **19**(2), 38–45 (2012). ACM, New York
6. Elishar, A., Fire, M., Kagan, D., Elovici, Y.: Organizational intrusion: organization mining using socialbots. In: 2012 International Conference on Social Informatics (SocialInformatics) (2012)
7. Yazan, B., Muslukhov, I., Beznosov, K., Ripeanu, M.: Design and analysis of a social botnet. Comput. Netw.: Botnet Act.: Anal. Detect. Shutdown **57**(2), 556–578 (2013)
8. FaceBook API. <https://developers.facebook.com/docs/graph-api>
9. Steps Toward Tracking and Managing Your Digital Footprint. http://articles10972.rssing.com/chan-30099336/all_p3.html
10. Facebook has more than 83 million illegitimate accounts. <http://www.bbc.com/news/technology-19093078>
11. Bot or Not? <http://truthy.indiana.edu/botornot/>
12. Boshmaf, Y., Ripeanu, M., Beznosov, K., Santos-Neto, E.: Thwarting fake OSN accounts by predicting their victims. In: Proceeding AISec 2015 Proceedings of the 8th ACM Workshop on Artificial Intelligence and Security. ACM, New York (2015)
13. Dickerson, J.P., Kagan, V., Subrahmanian, V.S.: Using Sentiment to Detect Bots on Twitter: Are Humans More Opinionated than Bots?
14. List of Twitter bots 2015. <http://qz.com/572763/the-best-twitter-bots-of-2015/>
15. Twitter. www.twitter.com
16. Gepettos Army: Creating International Incidents with Twitter Bots. <http://www.slideshare.net/gregmarra/gepettos-army-creating-international-incidents-with-twitter-bots>
17. Twitter Analytics. <https://analytics.twitter.com/>
18. Tweepy. www.tweepy.org
19. Django. <https://www.djangoproject.com/>
20. Impression. <https://unionmetrics.zendesk.com/hc/en-us/articles/201201636-What-do-you-mean-by-Twitter-reach-exposure-and-impressions->

Vulnebdroid: Automated Vulnerability Score Calculator for Android Applications

Sugandha Gupta^(✉) and Rishabh Kaushal

Department of Information Technology, Indira Gandhi Delhi Technical University
for Women, New Delhi, India

sugandha.gupta515@gmail.com, rishabhkaushal@igdtw.ac.in

Abstract. Nowadays mobile phone users download lots of applications for various purposes like learning, entertainment, businesses, etc. For a naive user, it is very difficult to identify whether the permissions provided to the application at the time of installation are being used properly or not. There are tools available for the detection of android malware but many of them are not open source or give tricky results which are not easily understandable. Various online services like VirusTotal uses the updated anti viruses for computing the malware detection ratio. However, since most of these anti-viruses are based on signature based detection methodology, therefore, it detection can be circumvented by using obfuscation methods. In our work we have implemented VULNEBDROID, an automated light weight obfuscation-tolerant static tool for computing the vulnerability score and assessing the vulnerability level of android applications. To assess the vulnerability, this tool selects the features of the application, like dangerous permissions used; vulnerable functions which can be used in order to misuse the application and can exploit the Application Programming Interface (API) to access the resources. Using this assessment tool, we are able to detect 96 % of malicious application as vulnerable either with high or medium degree of vulnerability.

Keywords: Android application · Vulnerability score · Malware · Obfuscation

1 Introduction

Thousands of new applications are introduced to the various market stores each passing day. With the growth of applications, the malwares are also growing exponentially every year. The hackers are getting more attracted towards the android market because of the open source and availability of the application from the various market stores like Google play. Therefore, security of the applications is the main concern for both the users and the developers.

An application typically behave maliciously only if it has the required permissions and there are vulnerabilities in the code which a malicious user can exploit to gain access to the sensitive private information or can gain the root privileges to control the handheld device as a bot. Therefore, instead of classifying the applications in malware or benign category, we are focusing in finding

the vulnerabilities in the application. Vulnerabilities are introduced mainly in the code because developers do not wisely use the permissions, application components and unintentionally use the vulnerable functions of the Android API. To optimize searching for vulnerabilities, the tool should have no human interaction. The battery and the memory both are valuable resources for a mobile device thus the tool should utilize them efficiently and minimally. Therefore, we introduce Vulnebdroid an automated light weight static tool that analyses the reverse engineered Dalvik code and manifest file and compute the vulnerability score of the application. On the basis of the vulnerability score, the application is evaluated as high, medium or low vulnerable.

Virustotal website [7] is commonly used online service by many researchers to predict the accuracy of their malware detection systems. Virustotal shows the aggregated result of the different anti-viruses available in the market and most of the anti-viruses have signature-based malware detection methodology. These detection techniques can be obfuscated by transforming the malware into its polymorphic (different form with same code) or metamorphic (mutate code) forms. This type of attack is known as transformation attack. DroidChameleon [15] and Pandora [16] have proposed systems to obfuscate malicious application from getting detected. These tools have lots of obfuscation techniques to change the signature, semantic as well as behavior of the application. Our tool Vulnebdroid is tested for obfuscated applications and as shall be shown in the paper later that it is tolerant to transformation attack.

So the main contributions of our work are -

- A vulnerability score metric is proposed which is dependent on the features based on Permission Count, Audit Points and Accessibility Points, expressed as below:
- We developed Vulnebdroid, an automated light weight static tool for calculating the vulnerability score and assess the application based on score. It is evaluated on 1100 malicious applications and it classifies 96 % application as high or medium vulnerable.
- Our tool is obfuscation tolerant because it is statically analysing the code of the application. It gives same vulnerability score while in case of VirusTotal detection, the accuracy is found to be reduced by 23.05 % for the obfuscated applications.

2 Related Work

As the users of mobile applications are increasing so are the threat of privacy leakages. To deal with this many researchers are coming up with the ideas of detecting the malwares. Wang [1] introduced a detection technique through static analysis of manifest file of application in which they uses the permissions request, high priority receivers and older version number as their key features. Zurutura [3] discussed a behavior based detection system focused on crowdsourcing i.e. Crowddroid is installed on users device which monitors and send every interaction to remote server. Partition clustering is used to detect the malicious applications

at server side. Permission use analysis technique is proposed by Zhang [8] in which the application is tracked using the permissions provided to utilize the system resources. A network traffic analysis based malware detection is proposed by Malik et al. [17] in which malicious application is identified on the basis of domain name server queries and data transmitted to the remote servers.

Another method for detecting the malware by the use of labeling the sensitive private information as taints which can be tracked in different ways. William et al. [4] developed an information tracking system which tracks the taint at variable level, method level, message level and file level and whenever the tainted data leaves the system, logs are generated. Artz et al. [5] proposed a static based analysis system, Flowdroid. It analyzes the data on the basis of context, flow, field and object-sensitive. This approach track sensitive tainted information through the application by starting a pre-defined source and then following the data flow until it reaches a given sink giving precise information about the data and its destination of leakage. This modeling takes care of multiple entry points, asynchronously executing components and call backs.

Grace et al. [9] has proposed a proactive scheme to identify zero day malwares. They developed an automated system called RiskRanker that categorize the application in medium, high- level risk if the applications are exploiting the platform level vulnerabilities or showing the behavior that could result in sending private information to remote server or unexpected costing money by the user.

Our work is different from these works in the way that our tool is calculating the vulnerability score which gives the common user a trusted parameter to judge the application in benign or malicious. It also needs no human interaction as it is automatically decompiling the application package file (.APK) and after statically analyzing the code gives the vulnerability assessment of the application. It has negligible memory overhead and resource consumption.

3 Methodology

This section describes our approach to assess the vulnerability of the android applications. Assessment is done by considering the features of the application selected and the data set.

3.1 Data Set

We have used the dataset released by Zhou et al. [2] used by them as part of their research work. The Android Malware Genome Project has collection of over 1200 Android applications covering 49 malware families.

3.2 Feature Selection

Feature selection plays a vital role in assessing the vulnerability of application. To select the features, the main aim is to collect all possible information about

Table 1. Feature selection

Feature	Description
PERMISSION_COUNT	# of Dangerous permissions commonly used By malicious apps
AUDIT_POINTS	# of vulnerable functions through which application can be misused
ACCESSIBLE_POINTS	# of functions which can exploit the resources through API

the application which can indicate the malicious behavior of it. Selected features and their description is shown in Table 1 are explained further to have a better understanding. First feature is the count of the dangerous permissions that can be misused to compromise the private data. According to the survey done by Zhou [10], there are permissions such as READ_SMS, WRITE_SMS, CHANGE_WIFI_STATE etc. which are more requested by the malicious application than the benign application. So, we have considered such 16 permissions as dangerous.

Mobile Security Checklist released by OWASP [6] in 2016 contains the list of vulnerabilities to be checked to develop a secure mobile application. Some of the static check points are to check for invalid SSL certificates, ignoring SSL certificate errors, improper exception handling in code and debug set to true. So, Http_Audit checks for the function like *setHostnameVerifier()* as it will not throw any exception if SSL Certificate is not verified while establishing a connection, which is the breach of security. *Webview_Audit* checks for the functions which have the vulnerability of remote execution of JavaScript. *Register_Receiver* counts all the broadcast receivers registered to be run in main activity. Audit_points feature counts *Http_Audit*, *Webview_audit*, *Register_Receiver*, *Debuggable* and all the application components.

Accessible.Points handles the use of vulnerable functions which exploits the android API to reach the resources available to the application. It counts the input/output files which are accessible from the application; Function *Getname()* which can reveal the internal paths of the application to the malicious user; functions of the intent class like *getserializableextra* and *getparcelable* which have the 0day deserialization vulnerability which can lead to DOS attack and Privilege Escalation attack.

3.3 Proposed Approach

Workflow of vulnebdroid tool is represented by Fig. 1. The main components of our tool are - Vtdroid¹, Parser, Scaler and Score Calculator. Vtdroid tool contains decompiler and apk parser which are used for creating the profile of the application. Profile contains all the features of the application which are

¹ <https://github.com/Xbalienvetdroid>.

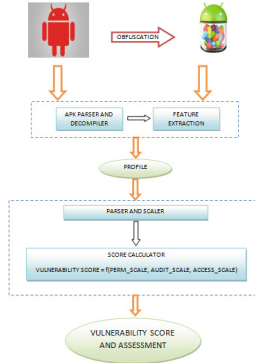


Fig. 1. The VULNEBDROID workflow

useful in indicating the vulnerable applications. Firstly, Decompiler decompiles the application and searches for the vulnerable functions defined in the selected features and the extracted features are recorded in the profile of the application. Then, Apk Parser parses the manifest file of the application and records the dangerous permissions and the application components- Activity, Service, Receiver and Provider in the profile. Parser and Scaler is used for feature selection and scaling the features. Application Profile is sent to parser which analyses the profile and set the value of the features after counting them. Then, scaler scales the features according to their mean values and sends the feature vector to vulnerability score calculator. Vulnerability score calculator calculates the score and gives the assessment and score as output.

Our tool is tested for malicious applications as well as obfuscated applications.

4 Implementation and Observations

The tool is implemented in Linux environment. Coding of the tool is done in python language. Our experiment is divided into 3 phases. Phase I represents the equation generation phase. In phase II, vulnerability assessment of all the malware families is done using the equation generated in phase I and in phase III, firstly obfuscation of various applications is done and then the vulnerability score calculated from Vulnebdroid tool and malware detection ratio of VirusTotal website are compared for the obfuscated applications. In phase I, for generation of equation, profiles of randomly selected 106 applications are created and the features are selected. The profiles are created using Vetdroid which uses an open source tool Androguard [11] for decompiling and manifest file parsing. It is the training phase of our work in which we have used data analytics tool Minitab 17 [12] for observing the contribution of the selected features in different malware families.

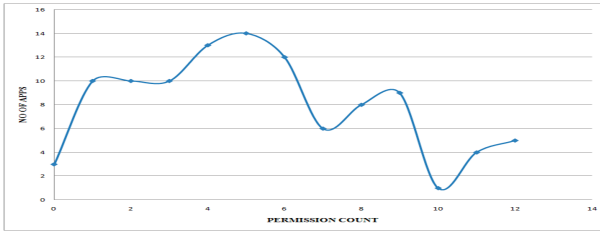


Fig. 2. Permission_count v/s number of apps

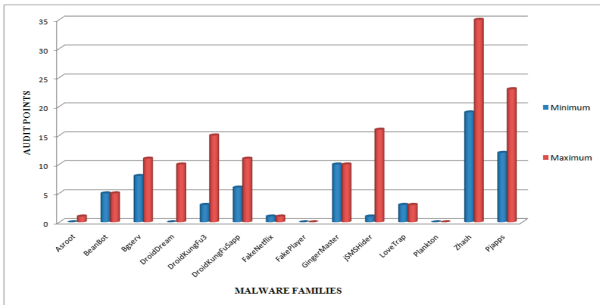


Fig. 3. Audit_points v/s number of apps

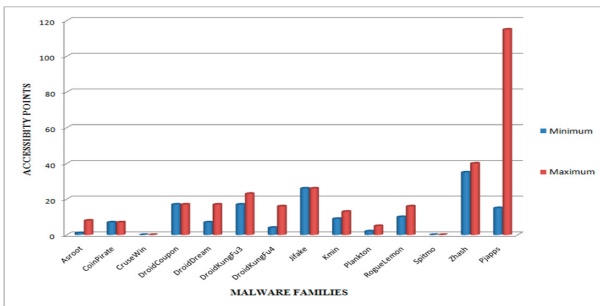


Fig. 4. Accessibility_points v/s number of apps

1. Permission Count.

From the Fig. 2, it is observed that 5 dangerous permissions are requested by maximum number of applications. Permission Count is having the direct relationship with the vulnerability score as more dangerous permissions will be given to application, it will have more privileges to access the different components to behave maliciously.

2. Audit Points.

From the Fig. 3, 35 audit points are counted as maximum. Audit points are also having the direct relationship with the vulnerability score as the malicious

application requires more audit points to work in background. This feature shows a high value for most of the malware families (eg. Kmin, Gingermaster, BaseBridge, Crusewin) which are connecting to a remote server to steal the sensitive information.

3. Accessibility Points.

Figure 4 represents that maximum number of applications dont show any accessibility point indicating there are lots of application which are not accessing the resources (e.g. files stored) available to the application. Malware families like Pjapps, ADRD etc. having nature of stealing information and transforming the phone as bot have more accessible points than the mean value and the malware families like Crusewin, FakePlayer, GGTracker, Walkinwat, LoveTrap etc. which send the messages to premium rate numbers have accessible points near to 0.

The value of all the features are ranging differently i.e. the permission count will always have the value between 0 to 16 while Accessibility points are ranging from 0 to 115 and can be more. Therefore, to give features equal weightage, scaling is done on the basis of their mean values i.e. the feature having count more than mean value then the scaled value is set to 2 and for lower values it is set to 1 and 0 is assigned in the absence of feature in profile. Hence, the feature vector of scaled values contains only 0s, 1s and 2s. The scaled values are then used as parameters for expressing vulnerability score equation. Hence, we are providing scaled features and VirusTotal detection ratio as Vulnerability score of the selected applications to find the co-efficient of each variable by using the regression modeling. We have constructed a predictive model on the basis of the linear combination as below:

$$V = \sum W_i X_i \quad (1)$$

In the above equation, V stands for the Vulnerability Score, W represents the co-efficient of the i th continuous predictor variable and X is the selected feature variable.

For phase II, a parser and scaler is developed in python to count the selected features from the profile created by Vetdroid and then scales the features according to their mean values. The scaled feature vector is passed to the score calculator to calculate the vulnerability score. The tool is tested on 1100 malicious applications from all the malware families in our data set. In phase III, Obfuscation Technique used is repacking, disassembling and reassembling of the application. It is done with the help of reverse engineering tools like Apktool [13] and Dex2jar [14] tool. Virustotal website has around 56 updated anti-viruses on which the detection ratio of the applications is evaluated. Virustotal score is used to compare with our vulnerability score to show that we have developed an obfuscation tolerant technique.

5 Results

Result 1:

From the Phase I, the vulnerability score equation generated is following:

$$Vulneb_score = 28 * Perm_scale + 14.63 * Audit_scale + 8.20 * Access_scale \quad (2)$$

Equation 2 indicates that the critical permissions are majorly contributing to the vulnerability score while the accessibility points are contributing minimum. Permissions are always the important parameter for detecting the android malware as various researchers have used them to detect the malwares [1, 8, 10].

Result 2:

Figure 5 represents the category-wise distribution of the vulnerability in the MalGenome dataset used for the evaluation. We are able to categorize 96 % of applications as highly or medially vulnerable indicating that by using our tool, we can also detect the malware to an extent.

Result 3:

Vulnerability Assessment of all malware families in the dataset is represented in Table 2. High vulnerability score shows the presence of all the selected features with one of them exploited to its extreme level. Zhash malware family shows the maximum vulnerability score i.e. it contains the maximum vulnerable components defined by us. Malware applications containing dangerous permissions along with some vulnerability in code come in medium vulnerable category. Also the applications which do not have dangerous permission but have other two feature more than mean values comes in this category.

Result 4:

By obfuscating the application, the VirusTotal score is decreased, represented in Fig. 6, with the average of 23.5 % having standard deviation of 12.12 % while the vulnerability score remains exactly the same shown in Fig. 7. The VirusTotal

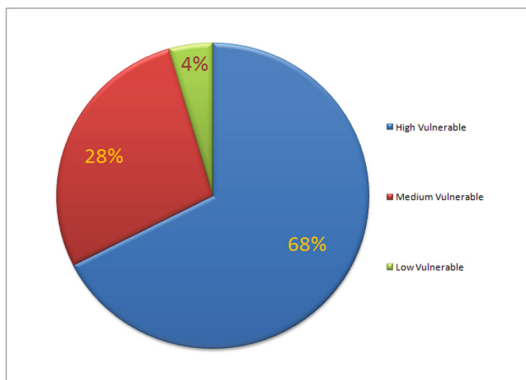


Fig. 5. Categorization of vulnerable applications

Table 2. Vulnerability Assessment of malware families

Malware family	# Samples	mean vulneb_score%	Assessment	Malware family	# Samples	mean vulneb_score%	Assessment
Zhash	11	100	HIGH	DroidDream	15	60.15	MED
Kmin	52	97.84	HIGH	DroidKungFu2	29	59.97	MED
Bgserv	9	94.62	HIGH	RogueSPPush	9	59.18	MED
CoinPirate	1	91.93	HIGH	DroidCoupon	1	58.07	MED
BeanBot	8	89.49	HIGH	Jifake	1	58.07	MED
AnserverBot	186	85.20	HIGH	GPSSMSpy	6	51.11	MED
GGTracker	1	83.87	HIGH	DogWars	1	50	MED
RogueLemon	2	82.20	HIGH	Zsone	12	50	MED
Geinimi	35	81.75	HIGH	CruseWin	2	41.93	MED
ADRD	21	81.29	HIGH	FakeNetflix	1	41.93	MED
DroidKungFu3	276	78.46	HIGH	Gone60	9	41.93	MED
Endofday	1	77.54	HIGH	LoveTrap	1	41.93	MED
GamblerSMS	1	77.54	HIGH	SMSReplicator	1	41.93	MED
NickySpy	2	77.54	HIGH	SndApps	10	41.93	MED
YZHC	21	77.54	HIGH	Tapsnake	2	41.93	MED
BaseBridge	122	75.68	HIGH	Walkinwat	1	41.93	MED
HippoSMS	4	75.52	HIGH	Zitmo	1	41.93	MED
Pjapps	27	75.43	HIGH	JSMShider	16	40.65	MED
GoldDream	29	74.49	HIGH	Plankton	11	39.53	MED
GingerMaster	4	72.46	HIGH	Asroot	8	28.4	LOW
DridKungFuSapp	3	69.77	MED	FakePlayer	6	27.54	LOW
DroidKungFu4	73	66.50	MED	Spitmo	1	27.54	LOW
DroidDreamLight	46	64.24	MED	DroidDelux	1	22.46	LOW
DroidKungFu1	21	60.89	MED				

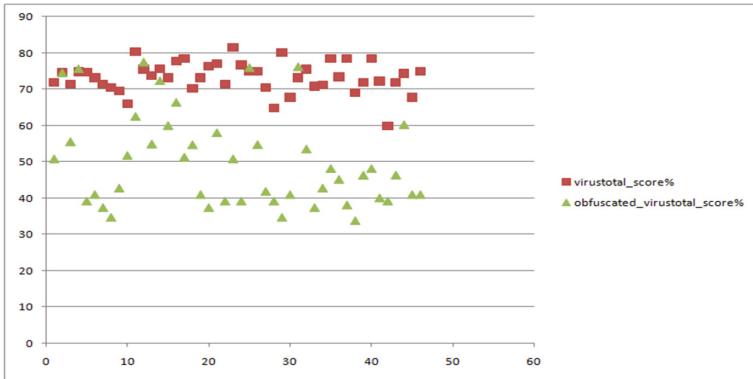


Fig. 6. Comparison between VirusTotal score and obfuscated VirusTotal score

score is changed because while the compilation of source code into Dalvik executable code the name of the variables are changed, thus, changing the signature of the malware where in our tool the application is statically analyzed therefore, it does not show any changes. It is the simplest technique applied for our exper-

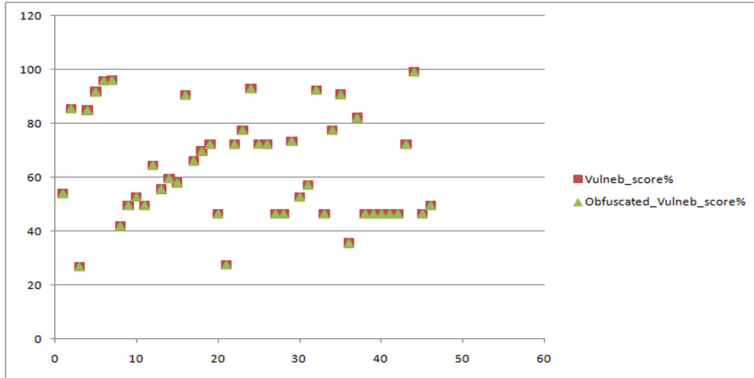


Fig. 7. Comparison between vulnerability score and obfuscated vulnerability score

iment while a complex obfuscation technique can result in high decrement in detection ratio by the anti-viruses.

6 Conclusion and Future Work

20 out of 47 malware families in our dataset shows the high vulnerability level. 96% of variant applications of all malware families are detected as high or medium degree of vulnerability. As they all are malicious applications therefore, the proposed technique should be considered as positive approach for doing vulnerability assessment. We conclude that higher the Vulnerability Score, there will be more chances that application will exploit the permissions provided to it and leak the sensitive information or gain root privileges. It is also concluded that by using our approach obfuscation does not affect the vulnerability score of the application.

References

1. Feldman, S., Stadther, D., Wang, B.: Manilyzer: automated android malware detection through manifest analysis. In: IEEE 11th International Conference on Mobile Ad Hoc and Sensor Systems (2014)
2. Zhou, Y., Jiang, X.: Dissecting android malware: characterization and evolution. In: IEEE Symposium on Security and Privacy (SP) (2012)
3. Burguera, I., Zurutuza, U., Nadjm-Tehrani, S.: Crowdroid: behavior-based malware detection system for android. In: 1st ACM Workshop on Security and Privacy in Smartphones and Mobile Devices (2011)
4. Enck, W., Gilbert, P., McDaniel, P., Chun, B.-G.: TaintDroid: an information-flow tracking system for realtime privacy monitoring on smartphones. ACM, October 2010
5. Arzt, S., Rasthofer, S., Fritz, C., Bodden, E., Bartel, A., Klein, J., Le Traon, Y., Octeau, D., McDaniel, P.: FlowDroid: precise context, flow, field, object-sensitive and lifecycle-aware taint analysis for android apps. ACM (2014)

6. OWASP Mobile Security. https://www.owasp.org/index.php/OWASP_Mobile_Security_Project
7. Virutota. <https://VirusTotal.com/>. Accessed Mar 2016
8. Yuan, Z., Min, Y., Yang, Z., Gu, G., Ning, P., Zang, B.: Permission use analysis for vetting undesirable behaviors in android apps. *IEEE Trans. Inf. Forensics Secur.* **9**(11), 1828–1842 (2014)
9. Grace, M., Zhou, Y., Zhang, Q., Zou, S., Jiang, X.: RiskRanker: scalable and accurate zero-day android malware detection. In: *MobiSys*. ACM (2012)
10. Jiang, X., Zhou, Y.: A survey of android malware. In: Jiang, X., Zhou, Y. (eds.) *Android Malware*. SpringerBreifs in Computer Science, pp. 3–20. Springer, New York (2013)
11. Desnos, A.: Androguard (2011). <https://code.google.com/p/androguard/>
12. Minitab. <https://www.minitab.com/en-us/products/minitab/>, Accessed 17 Mar 2016
13. Apktool. <http://ibotpeaches.github.io/Apktool/install/>
14. Dex2jar. <http://sourceforge.net/projects/dex2jar/files/>
15. Rastogi, V., Chen, Y., Jiang, X.: DroidChameleon: evaluating android anti-malware against transformation attack. In: *ACM ASIA CCS*, May 2013
16. Protsenko, M., Mller, T.: PANDORA applies non-deterministic obfuscation randomly to android. In: *2013 8th International Conference on Malicious and Unwanted Software: “The Americas” (MALWARE)* (2013)
17. Malik, J., Kaushal, R.: CREDROID: android malware detection by network traffic analysis’. In: *1st ACM Workshop on Privacy-Aware Mobile Computing* (2016)

Author Index

- Adhikari, Avishek 74
Agrawal, Himanshi 188
Amandeep, Bagga 295
Anita, J.P. 414
Arora, Amar 251
Ashwini, B. 369
- Bakshi, Aditya 63
Bala, Rajni 305
Basu, Srijita 137
Bhattacharya, Uma 347
- Chacko, Anu Mary 152
Chakraborty, Chandan 283
Chakraborty, Rupak 283
Chandran, Aditya 390
Chandrasekaran, K. 199
Chatterjee, Sumanta 161
- Das, Sukanta 30
Deokamble, Vasant 283
Dewangan, Madhuri 450
Dey, Prakash 74
Dhane, Dhiraj Manohar 283
Dhope, Tanuja 403
- Fairooz, Munavar 152
- Gayathri, N.B. 225
Geetha, G. 87, 295
Gilbert, Kethzi 377
Goel, Rajat 239
Gokkul Nath, T.S. 269, 414
Gosain, Anjana 251
Govil, Mahesh Chandra 239
Gupta, Savita 318
Gupta, Sugandha 461
- Jain, Lokesh 390
Jain, Prakhar 112
Jayakumar, M. 258, 269
John, Asha Liza 18
- K., Surendran 439
Kaur, Mandeep 318
Kaur, Pankaj Deep 173
Kaushal, Rishabh 188, 450, 461
Kaushik, Kulvaibhav 99
Kharat, Reena 52, 336
Khethavath, Praveenkumar 213
Ko, Ryan K.L. 3
Kumar, Sanjeet 124
Kumar, Ysudhir 439
Kumari, Anu 124
Kuruba, Chandrasekhar 377
- Ladhake, Siddharth 328
Lekshmi, Binitaa 414
- Madhu Kumar, S.D. 152
Maity, Maitreya 283
Malkani, Manu 124
Malviya, Ashwini 328
Mazumdar, Chandan 137
Menon, Vijay Krishna 369
Midhila, E.R. 414
Mishra, Anurag 305
Mohamad, Salwa O. 358
Mungle, Tushar 283
Murmu, Bhavya Ishaan 124
- N., Srinath 439
Nair, Aravind Ashok 439
- P., Hrudya 439
Pais, Alwyn R. 42, 161
Pal, Doyel 213
Pareek, Gaurav 377
Pawar, Komal 403
Poornachandran, Prabaharan 439
Puthran, Shubha 427
- Rajpal, Ankit 305
Rangappa, Purushothama Byrapura 377
Rashid, Tarik A. 358
Rawat, Sanjay 390

Rohit, Raghvendra Singh 74
Roy, Sohini 347

Sanyasi Naidu, P. 52, 336
Sarkar, Santanu 74
Saurabh, Sumit 161
Senchury, Gobinda 213
Sengupta, Anirban 137
Sethi, Biswanath 30
Shah, Arvind Kumar 347
Shah, Ketan 427
Shankar, Shashi Kant 63
Sharma, Manmohan 63
Shukla, K.K. 112
Sidhaye, Prabhav 377
Singh, Girdhari 239
Singh, Inderjeet 42

Singh, Parampreet 173
Sobti, Rajeev 87
Soman, K.P. 369
Srinathan, Kannan 390
Sudheesh, P. 258, 269
Swaminathan, Ashwin 414

Thampi, Sabu M. 18
Thangam, V. 199
Tomar, Anurag Singh 63

Varadharajan, Vijayaraghavan 99
Vasudeva Reddy, P. 225
Vikranth, S. 258

Will, Mark A. 3