Multimed Tools Appl (2016) 75:14641–14668
DOI 10.1007/s11042-015-3050-2

The protocol design and New approach for SCADA security enhancement during sensors broadcasting system

Aamir Shahzad[1] · Malrey Lee[1] · Changhoon Lee[2] · Naixue Xiong[3] · Suntae Kim[4] · Young-Keun Lee[5] · Kangmin Kim[6] · Seon-mi Woo[7] · Gisung Jeong[8]

Received: 25 February 2015 / Revised: 12 June 2015 / Accepted: 26 October 2015 /
Published online: 20 November 2015

Abstract Several security mechanisms have been investigated and deployed that provide protection for real time platforms. Each security mechanism is contributed to enhance the SCADA system security, but at the same time, the mechanism is limited and depended on the other protocols for the purposes of message security, and its delivery. Few researches are

✉ Malrey Lee
mrlee@jbnu.ac.kr

✉ Kangmin Kim
activase@jbnu.ac.kr

Aamir Shahzad
mail2aamirshahzad@gmail.com

Naixue Xiong
xiongnaixue@gmail.com

Suntae Kim
stkim@jbnu.ac.kr

Young-Keun Lee
trueyklee@naver.com

Seon-mi Woo
smwoo@jbnu.ac.kr

Gisung Jeong
jgskor@wku.ac.kr

[1] M561-756, Center for Advanced Image and Information Technology,School of Electronics & Information Engineering, Chon Buk National University, 664-14, 1Ga, Deokjin-Dong, Jeonju, Chon Buk, South Korea

[2] Department of Computer Science and Engineering, Seoul National University of Science and Technology (SeoulTech), Seoul, Republic of Korea

[3] School of Computer Science, Colorado Technical University, Colorado, USA

[4] M561-756, Department of Software Engineering, Chon Buk National University, 664-14, 1Ga, Deokjin-Dong, Jeonju, Chon Buk, South Korea

conducted on security for SCADA broadcasting system, but these are limited to end-to-end designs and developments. The security developments for multicasting and broadcasting systems are much complicated, time consumed and/or overloaded with the cryptography mechanisms. After conducting the detail survey, a simulation environment for SCADA water pumping system is designed in-which number of nodes is configured and well known cryptography algorithms are selected, and deployed as an inclusive development for SCADA/DNP3 broadcasting system. The inclusive security development is considered with the best performance, and with predominant weakness in mind, which are present in SCADA/DNP3 broadcasting system. However, overall communication is initiated, monitored and controlled at main controller side with the user defined human machine interface (HMI).

Keywords Supervisory control and data acquisition · SCADA Broadcasting system · Distributed network protocol · Cryptography dynamic buffer · Cryptography mechanisms

1 Introduction

In SCADA (Supervisory control and data acquisition) system, numbers of field devices are remotely connected with main controller. At controller center, human machine interface (HMI) is designed with high definition resolution which visualizes the overall communication and performs data acquisition and operations of SCADA system. HMI is a major source for end users or operators to manage and control the field devices from control center. Usually, data acquisition and control information are visualized graphically which made convenient for SCADA operator [7, 25, 39, 42]. Meaning that, each end user can views the system processing and information in the style of mimic figures and using simulation designs [25, 39].

In simulation designs [7, 25, 28, 39], SCADA system is entirely presented and operators graphically in HMI. For example, in water pumping system the water levels are monitor using sensors. In-case water level is high or low in water tank, the sensors define the status and this status would be simultaneously visualized on HMI, at controller center. If number of sensors or other field devices are configured in SCADA system and generate the input/output points thus, all information or each device points are presented separately in integrated HMI.

In SCADA system, the nodes (or field devices) are connected with main controller and usually, the communication is also initiated and controlled from main controller. This type of communication is also designated as unbalanced systems, where main controller is authorized to initiates the communication. While in distributed network protocol (DNP3), which is designed as balanced systems [7, 42] provides advance characteristics and flexibilities, in which each and every node within SCADA hierarchical network is authorized to initializes the

[5] M561-756, Department of Orthopedic Surgery, Chonbuk National University Hospital, Jeonju, Republic of Korea

[6] M570-752, Division of Biotechnology, College of Environmental & Bioresource Sciences, Chonbuk National University, Iksan 570-752, Republic of Korea

[7] M561-756,JINI Co. Ltd., B-102, Technobill, 109 banryong-load, Deokjin Gu, JeonJu si, Jeollabuk-do, Republic of Korea

[8] M561-756 , Department of Fire Service Administration, WonKwang University, Iksan, Republic of Korea

communication as primary or main controller with others as secondary or sub-controllers. In balanced systems, all configured nodes in the SCADA network are treated equally, and each node has right to initialize the communication. The terms unbalanced and balanced systems have created confusion at the data link layer of DNP3 protocol. At data link layer, it is difficult to identify which station is primary or secondary, when each node has equaled rights in balance systems. However, primary and secondary stations are distinguished at application layer of DNP3. Data link layer provides reliable communication between the recipients in SCADA system. At the data link layer, up to 65536 addresses are provided and the range FFF0-FFFF is designated for broadcasting communication [7, 42]. The message broadcasting is the important aspect of SCADA/ DNP3 system and usually, addresses are assigned logically to each participated node during SCADA broadcasting (transmission). In broadcasting, security is one of the major challenge because the initial design of DNP3 protocol is without any security concerns therefore, security mechanism is required that should able to provide protection [8, 9, 21, 24, 44].

A new trusted solution is implemented in SCADA system through TCP/IP protocols without accounts of performance impact [24, 28]. The research reviews the underlying security threads and specifies the recommendations for SCADA system such as intranet uses, security system requirements, security system implementation which includes active, passive, half-Active, and tunnel modes and secure system applications. Few challenges are specified against SCADA broadcasting security issues such as user authentication, access control, information interception or modification, and internet security [8, 9, 17, 21] and potential cyber-attacks are also analyzed in SCADA broadcasting transmission, such as unauthorized access, modifications, denial of service, eavesdropping, information leakage, masquerade, sniffers, repudiation, data alter, and others [10, 18, 26, 40]. As the result, security, access availability and reliability are the most importance performance factors acquired within SCADA system and must be achieved, while connected with open networks [24, 28]. Therefore, the proposed study reviews the importance of message broadcasting during unbalanced systems connectivity, over internet and subsequently, deployed an inclusive cryptography solution to overcome the security issues that are commonly founded in SCADA system [9, 21, 28, 32, 44].

Contribution This study follows the concept of unbalance system of DNP3 protocol, the main controller is authorized to initiates the transmission, and transmits message to connected nodes in SCADA system. Using the phenomena of unbalance system, a new security development has made for SCADA/DNP3 broadcasting system.

2 Problem statement

The broadcasting communication is a main concerned of SCADA (Supervisory control and data acquisition) systems. In many situations, main controller transmits a message to each and every sub-controller which is connected in SCADA system (s). The message will be any type such as initiates communication, command, and alert alarm and file information [7]. In broadcasting transmission, security is a big issue that has been faced by SCADA systems such as electric stations, water pumping systems, wastes water system and oil and gas industries [7, 28, 33, 42]. Several security solutions have been employed to secure the SCADA systems, but they have several dependences, and limited to unicasting communication [28, 34, 41]. A cryptography based security solutions are accounted as best approaches for SCADA security enhancements, without any dependency from other protocols [12, 13, 22, 28].

At other side, the security developments via cryptography techniques are real in the case of SCADA broadcasting communication because these types of implementations are usually ponderous in the terms of key generation, and distribution with complex mathematical workflow [28, 38]. So, an optimized cryptography based security solution is required that significantly secure the SCADA broadcasting communication, without performance impact. In proposed implementation; an inclusive security mechanism via cryptography is selected with broadcasting aspects in mind and also significantly enhanced the SCADA security during message broadcasting.

Research objectives The main objective of this study is threefold:

i. A DNP3 stack is designed and security is deployed before broadcasting to open networks.
ii. A new cryptography dynamic buffer (CDB) is employed, which keeps and tracks the information of protocol bytes, and security development.
iii. SCADA/DNP3 testbed is designed to perform the experiments and to compute the security results for evaluation. The formal proofs are employed that validate the proposed security development.

3 Simulation model and design

In simulation, SCADA water pumping system is designed; information is collected through sensors, and delivered to sub-controllers. In system, water is collected from main storage and distributed to local storage through motor. A level sensor is directly attached with local storage which checks the water level inside the local storage (or tank) and pressure sensor is used to check water pressures (incoming/outgoing) inside the pipes. If the water level is according to set points of main controller then, operation either heating or cooling is performs. In-case, water is low inside local storage then; sub-controller deactivates the pumping operations and activates the water pump to fill the local storage according to set points.

Main controller set the points or transmits the bytes to, and remote units (RUs) set the instructions according to set points [33]. Typically, remote units (RUs) are connected with, and sensors are configured to send information that is relative with pumping system [28, 44]. In this study, word ‘simulation’ is used for water pumping system which takes the information from, and sensor controls with remote unit (RU). While each RU receives, and transmits the information from/to main controller. The word ‘testbed’ is used which shows the detail configuration and connectivity between main controller and remote units (RUs). Figure 1 shows the water pumping system with sensors connectivity.

The overall interaction between main controller and sub-controllers are specified by DNP3 protocol which employed non-proprietary protocol such as UDP (user datagram protocol) to communicate over internet [31]. In transmission, DNP3 protocol is act as upper layer or standing above then UDP [31, 34]. Meaning that, DNP3 frames are encapsulated in UDP packets and then, transmitted over internet. Upon receiving, sub-controllers response back to main controllers via TCP/IP protocols usually, the transmission is one-to-one or unicasting transmission [28]. At man controller side, human machine interface (HMI) keep and monitor

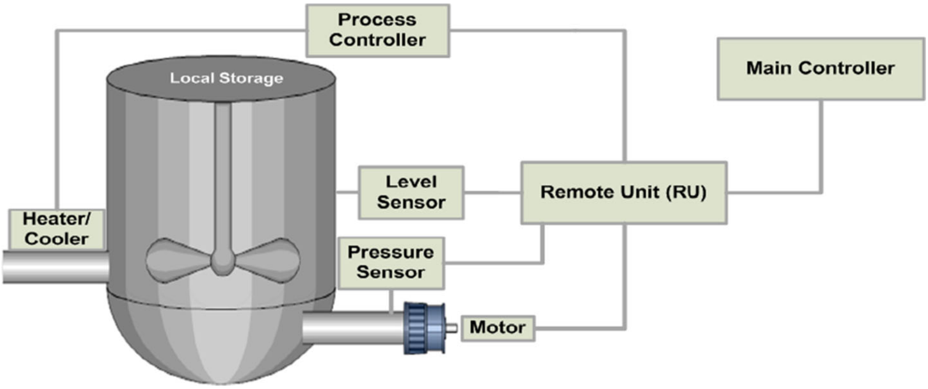


Fig. 1 Simulation design of water pumping system

the information of each sub-controller, and collected information would be simultaneously stored in historian or in database [28, 31]. In study, the historian is deployed in MySQL and DNP3 model is designed and deployed in C#.

3.1 DNP3 modeling and security development

In this section, we have explained the basic terminologies if DNP3 protocol message structure, basic bytes flow in stack, security development and formal proof, with corresponding examples.

DNP3 is a layered based open protocol, and based on enhanced performance architecture (EPA) model which has three layers in its stack such as application layer, data link layer and physical layer. DNP3 protocol adds additional layer called pseudo-transport layer which performs the limited functionalities of transport and network layers of OSI (open system interconnection) model [7, 42].

Figure 2 shows the DNP3 model design, in-which protocol bytes are constructed and security is deployed, and tested as additional layer at each layer of DNP3 protocol. Physical layer is not accounted because; DNP3 frames are directly encapsulated in UDP protocol which provides the services to make connectivity with remote devices through internet.

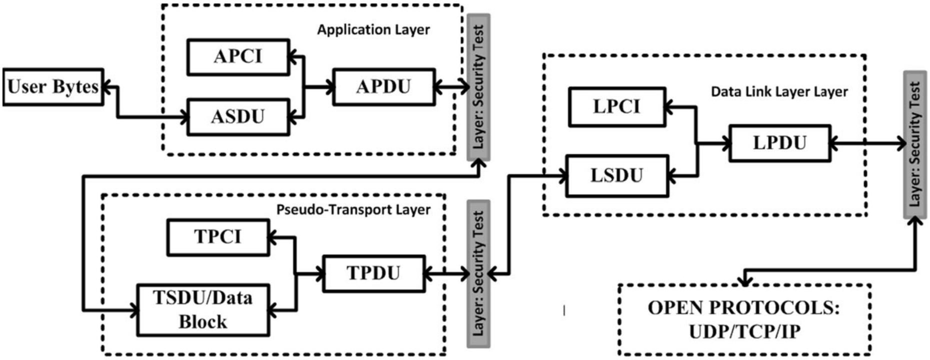


Fig. 2 DNP3 stack model and design

Random user bytes are received from human machine interface (HMI) (or user application layer) to application layer stack, these bytes are managed into fixed sized blocks called application service data units (ASDUs). Header bytes are added with ASDU/ASDUs which made the application protocol data unit (APDU). However, size of ASDU blocks are not fixed but each APDU size limited to 1990/1992 bytes, in-cases of sends/response message. Each ASDU block contains main two subfields including object header and data objects, while object header field contains 3—11 bytes information, and is further divided into three subfields including object (O), Qualifier (Q) and Range (R). More detail related with object header is depicted in Table 1 and the acronyms are not exactly defined by original documentation of DNP3 protocol [7, 28].

Two types of messages are defined at application layer such as, sending message form main controller and response message from sub-controller (s), The messages are distinct by header field or application protocol control information (APCI), sending message header contains two bytes length fields called application control AC) and function code (FC), while two additional bytes length is employed as internal indication (IIN) in response header or/and in response message [7, 28].

In APCI, number of function codes is employed to specify the meaning of message in the cases of send message and response message. The application layer function codes, and related fields information of sending/response message are as follows [7]:

In Table 2, the security bytes such as IE, EE, 1A and EE, are sampled specified bytes which indicate the security development of application layer message. In other words, these bytes represent the security codes which ensure that security has been deployed and should be tested at other side (or at receiver side). The bytes shown as 'XX' are bytes which are not placed in command (or message). The shaded area shows the padding bytes which ensure that bytes are constructed with security development within boundary of application layer stack.

The application layer or APDU bytes are treated as user bytes in pseudo-transport layer. We can also say that, APDU bytes are assembled as transport service data unit (TSDU) bytes in pseudo-transport layer and further subdivide into fixed size data block, and each block size is limited up to 249 bytes. One byte length of header field or transport protocol control information (TPCI) is added with each data block and transport protocol data unit (TPDU) is formed. At received side, the TPCI is stripped off from each data block and TSDU bytes are reformed which should be further employed at application layer. The keyword 'pseudo' is placed with transport layer which performs the limited functionalities of transport layer and network layer of OSI seven layer model.

Each TPDU is limited up to 250 bytes in length and further assemble as link service data unit (LSDU) in data link layer of DNP3 protocol. Link header field or link protocol control

Table 1 Object header fields

Object Header	Occupied Bytes	Description
Object (O)	2 Bytes	This is two bytes length field, and further subdivides into subfield such as object group (OG) and object variation (OV). The OG defines the data type, while OV defines the distinct variation.
Qualifier (Q) Range (R)	1 Byte 0—8 Bytes	These fields are follow the object field and employed in both sending and response message. Usually, these fields are employed to define the associated data points of object group (OG) and object variation (OV). The range field contains two numbers such as start (St) and stop (Sp) which define the data object values.

Table 2 Function codes, fields and security bytes

Functions	AC	FC	Object field		Q	Range		Security bytes				Padding
0x0000 (hex)			OG	OV		St	Sp					
Read Function	C3	01	1E	02	00	04	07	1E	EE	1A	EE	
Write Function	C3	02	32	01	07	01		1E	EE	1A	EE	
Select Function	C3	03	OC	01	17	01		1E	EE	1A	EE	
Operate Function	C3	04	OC	01	17	01		1E	EE	1A	EE	
Direct Operate Function	C3	05	OC	01	17	01		1E	EE	1A	EE	
Direct Operate Function, no Acknowledgement	C3	06	OC	01	17	01		1E	EE	1A	EE	
Freeze Function	C3	07	14	00	06	XX		1E	EE	1A	EE	
Freeze Function, no Acknowledgement	C3	08	14	00	06	XX		1E	EE	1A	EE	
Freeze and Clear Function	C3	09	14	00	06	XX		1E	EE	1A	EE	
Freeze and Clear Function, no Acknowledgement	C3	0A	14	00	06	XX		1E	EE	1A	EE	
Freeze Function –AT-Time	C3	0B	32	02	07	01		1E	EE	1A	EE	
Freeze –Time-AT Function, no Acknowledgement	C3	0C	32	02	07	01		1E	EE	1A	EE	
Cold_Restart Function	C3	0D	34	01	07	01		1E	EE	1A	EE	
Warm_Restart Function	C3	0E	34	01	07	01		1E	EE	1A	EE	
Initialize_ Data Function	C3	0 F	32	02	07	01		1E	EE	1A	EE	
Initialize_Application Function	C3	10	5A	01	5B	01		1E	EE	1A	EE	
Start Application Function	C3	11	5A	01	5B	01		1E	EE	1A	EE	
Stop Application Function	C3	12	5A	01	5B	01		1E	EE	1A	EE	
Save Configuration Function	C3	13	5A	01	5B	01		1E	EE	1A	EE	
Enable_Unsolicited Function	C3	14	32	01	07	01		1E	EE	1A	EE	
Disable_Unsolicited Function	C3	15	3C	02	06	XX		1E	EE	1A	EE	
Assign Class Function	C3	16	3C	02	06	XX		1E	EE	1A	EE	
Delay Measure Function	C3	17	XX	XX	XX	XX		1E	EE	1A	EE	
Record_Current_Time Function	C3	18	XX	XX	XX	XX		1E	EE	1A	EE	
Open File Function	C3	19	46	03	5B	01		1E	EE	1A	EE	
Close File Function	C3	1A	46	03	5B	01		1E	EE	1A	EE	
Delete File Function	C3	1B	46	03	5B	01		1E	EE	1A	EE	
Get File Information Function	C3	1C	46	03	5B	01		1E	EE	1A	EE	
Authentication Function	C3	1D	46	02	5B	01		1E	EE	1A	EE	
Abort File Function	C3	1E	46	03	5B	01		1E	EE	1A	EE	
Activate Configuration Function	C3	1 F	XX	XX	XX	XX		1E	EE	1A	EE	
Function Code:0x20 (32)---0x80 (128): Reserved												
Function Code: 0x81 (129): Unsolicited Response								1E	EE	1A	EE	

information (LPCI) contains 10 bytes and added with LSDU, which makes the link protocol data unit (LPDU) or frame. In link layer, each frame size is limited up to 260 bytes, plus 32 bytes of cyclic redundancy check (CRC). However, in this research the CRC bytes are considered as optional bytes or would be unitized in cryptography dynamic buffer (CDB), upon needs [28, 40].

Data link layer is used to setup the logical connection between sender and receiver and keeps reliable message (or data) communication over physical channel. In LPCI, 2 bytes are allocated to source and destination addresses, which define 65536 distinct addresses and the addresses range FFF0–FFFF is designated for broadcasting communication. The terms balance and unbalance systems are also specified at data link layer. In balance system, each node can initials the communication either main controller or sub-controller (s). In this research, unbalance system is employed therefore; only main controller is authorized to initiates communication and sub-controllers should response accordingly. In LPCI, number of function codes is defined that perform initialization operation, and also test the logical connectivity between main controller and sub-controller (s) [7]. Some link layer function codes and corresponding descriptions are depicted in Table 3.

Algorithm: bytes flow within DNP3 Stack: Logical 'n' bytes are being broadcasted from main controller to remote units (RUs). The bytes are constructed and manipulated in SCADA/DNP3 protocol (stack) and then, encapsulated in UDP protocol in internet protocol suite which defines the ways to transmit DNP3 protocol frames to the destination address.

1: Compute application layer message as $\mathbf{X}^f(D,H)$, number of 'X' bytes are received from user application layer and function 'f' is performed to compute the data bytes and header bytes.

$$f_1 \rightarrow \mathrm{Comp}(\mathrm{D},\mathrm{H}),\ \ f_1 \in f$$

Here, the compute bytes or $f_1 \rightarrow$Comp(D) are application services data unit (ASDU) bytes and application protocol control information (APCI) bytes as $f_1 \rightarrow$Comp(H) are

Table 3 Link layer function codes [7]

Function codes from main controller		
Codes	Frame type	Function
0	SEND-CONFIRM expected	Reset of remote link
1	SEND-CONFIRM expected	Reset of user process
2	SEND-CONFIRM expected	Test function for link
3	SEND-CONFIRM expected	User data
4	SEND-NO REPLY expected	Unconfirmed user data
9	REQUEST-RESPOND expected	Request link status
Noted: Codes 5–8, 10–15 are not employed		
Function codes from sub-controller		
Codes	Frame Type	Function
0	CONFIRM	ACK-positive acknowledgement
1	CONFIRM	NACK-Message not accepted, link busy
11	RESPOND	Status link
14		Link service not functioning
15		Link service not used or implemented
Note: Codes 2–10, 12–13 are not employed		

added to form the application protocol data unit (APDU). Each APDU size is limited to 1992 bytes.

$$= \sum_{k=0}^{n} [\mathrm{Comp(D)}, \mathrm{Comp(H)}]^n$$

Fragment=APDU=APCI+ ASDU

Request: APCI or header=1 byte (AC)+1 byte (FC)=2 bytes

Reponses: APCI=1 byte (AC)+1 byte (FC)+2 bytes (IIN)=4 bytes

Application Control (AC):

[FIR, FIN, CON, SEQ Number] // Application Control Field

[FIR, FIN, CON, MU/RU Timeout] // Application Flow Control

Function Code (FC): Code='0' is used for confirmation and other detail codes are depicted in Table 2.

Here are some flow sequences which show the general transaction of message (or bytes) followed by format as [FIR, FIN, CON, SEQ Number] or [First APDU, Final APDU, Confirmation, sequence]. These flows are optional, and employed to understand the basic message transactions of DNP protocol.

Flow 1: Single APDU request from MU to RU without Confirmation:

Master application control=MU_AC [1, 1, 0, 9] // Request send to RU without Confirmation.

If RU send response to MU request .Then,

Master application control=MU_AC [1, 1, 0, 9] // Confirmation from MU

Flow 2: Single APDU request from MU to RU with Confirmation:

Master application control=MU_AC [1, 1, 1, 9] // Request send to RU with Confirmation

If RU sends response to MU request. Than Master application control=MU_AC [1, 1, 0, 9] // Confirmation from MU

Flow 3: Single APDU request from MU to RU and multiply APDUs response from RU to MU, without Confirmation:

Master application control=MU_AC [1, 1, 0, 5] // Request send to RU without Confirmation

RU send response to MU requested APDU 1

Master application control=MU_AC [1, 1, 0, 5] // Confirmation from MU

RU send response to MU requested APDU 2

Master application control=MU_AC [1, 1, 0, 6] // Confirmation from MU

Flow 4: Single APDU request from MU to RU and multiply APDUs response from RU to MU, with Confirmation:

Master application control=MU_AC [1, 1, 1, 5] // Request send to RU with Confirmation

RU sends confirmation to MU.

RU send response to MU requested APDU 1

Master application control=MU_AC [1, 1, 0, 5] // Confirmation from MU

RU send response to MU requested APDU 2

Master application control=MU_AC [1, 1, 0, 6] // Confirmation from MU

After construction of application protocol data unit (APDU) bytes with/without confirmation bit from application layer, control is shifted to lower layers of SCADA/DNP3 protocol.
2: Compute Pseudo-Transport message as $\boldsymbol{Q^f(D,B,H)}$, 'Q' bytes of application layer are assembled and function '*f*' is performed to compute the data bytes (D), data blocks (B) and header bytes (H).

APDU bytes are received from application layer and assembled as TSDU bytes (or data bytes). We can also say, the APDU bytes are directed mapped with TSDU bytes.

$$f_2 \to (\mathrm{Q} \to \mathrm{D})$$

The TSDU bytes are distributed in number of data blocks (B) and each block contains 249 bytes.

D~B (b_0,b_1,b_2,…..,b_n and $f_2 \to$Comp(B,H), $f_2 \in f$

Here, the compute bytes or $f_2 \to$Comp(B) are data blocks and transport protocol control information (TPCI) bytes as $f_2 \to$Comp(H) are added to form the transport protocol data unit (TPDU). Each TPDU size is limited to 250 bytes.

$$== \sum_{k=0}^{i} [\mathrm{Comp(B)}, \mathrm{Comp(H)}]^i$$

Compute Data Link layer message as $\boldsymbol{J^f(D,H)}$, maximum of '*J*' bytes are received and assembled as user bytes from pseudo-transport layer and function '*f*' is performed to compute the data bytes (D) and header bytes (H).

$$f_3 \to \mathrm{Comp(D, H)}, \quad f_3 \in f$$

Here, the compute bytes or $f_3 \to$Comp(D) are link services data unit (LSDU) bytes and link protocol control information (LPCI) bytes as $f_3 \to$Comp(H) are added to form the link protocol data unit (LPDU). Each LPDU size is limited to 260 bytes, plus optional CRC bytes.

$$= \sum_{k=0}^{l} [\mathrm{Comp(D)}, \mathrm{Comp(H)}]^l$$

Link Protocol Data Unit (LPDU)=LPCI+ LSDU=292 bytes

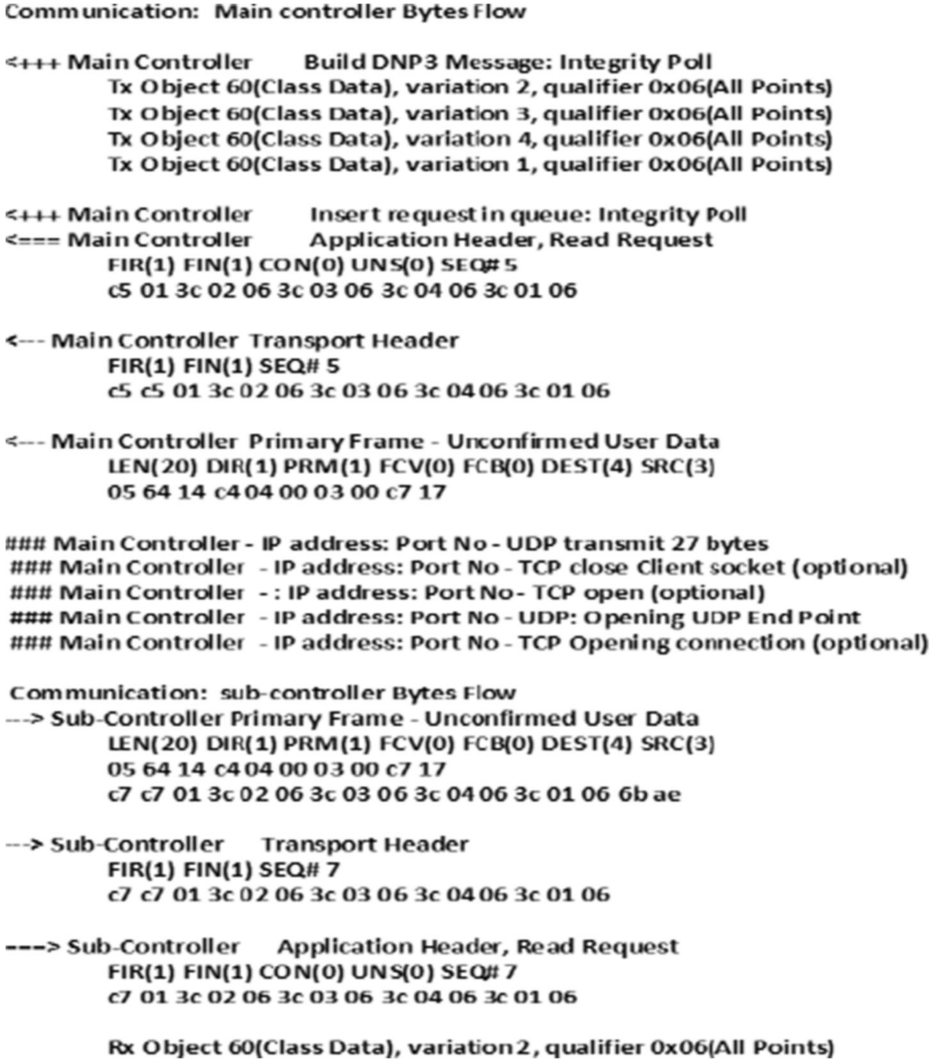
LPCI (Link protocol control information) or header=2 byte (Start)+1 byte (Length)+1 byte (Control)+2 byte (Destination Address)+2 byte (Source Address)+2 byte (CRC, Optional)=10 bytes

LSDU (Link Service Data Unit) or Data bytes=250 byte

Subsequently, data link layer frames are encapsulated into internet protocol suite in-placed of original physical layer of DNP3 protocol. The functions or f_0, f_1,f_2, $f_3 \in f$ and f_0 is employed for special purposes such as test bytes status and perform bytes padding.

More precisely, we have conducted some tests to check the DNP3 protocol stack bytes flow during request and response messages through protocol test harness [43]. More detail is visualized in following screen shots or in Figs. 3 and 4.

In below sections, security is implemented within DNP3 protocol stack, a new inclusive scenario is designed to visualize and representation the logical bytes of DNP3 stack with



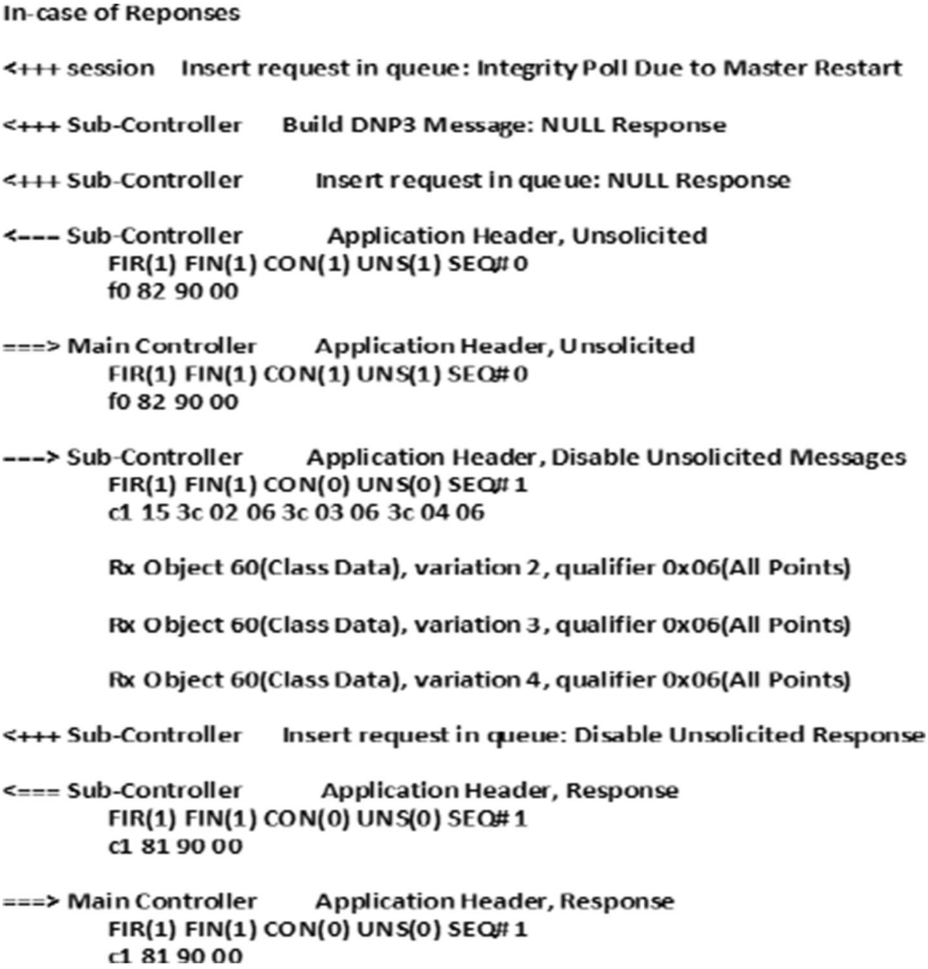
```
Communication:  Main controller Bytes Flow

<+++ Main Controller       Build DNP3 Message: Integrity Poll
         Tx Object 60(Class Data), variation 2, qualifier 0x06(All Points)
         Tx Object 60(Class Data), variation 3, qualifier 0x06(All Points)
         Tx Object 60(Class Data), variation 4, qualifier 0x06(All Points)
         Tx Object 60(Class Data), variation 1, qualifier 0x06(All Points)

<+++ Main Controller        Insert request in queue: Integrity Poll
<=== Main Controller        Application Header, Read Request
         FIR(1) FIN(1) CON(0) UNS(0) SEQ# 5
         c5 01 3c 02 06 3c 03 06 3c 04 06 3c 01 06

<--- Main Controller  Transport Header
         FIR(1) FIN(1) SEQ# 5
         c5 c5 01 3c 02 06 3c 03 06 3c 04 06 3c 01 06

<--- Main Controller  Primary Frame - Unconfirmed User Data
         LEN(20) DIR(1) PRM(1) FCV(0) FCB(0) DEST(4) SRC(3)
         05 64 14 c4 04 00 03 00 c7 17

### Main Controller - IP address: Port No - UDP transmit 27 bytes
 ### Main Controller  - IP address: Port No - TCP close Client socket (optional)
 ### Main Controller  - : IP address: Port No - TCP open (optional)
 ### Main Controller  - IP address: Port No - UDP: Opening UDP End Point
 ### Main Controller  - IP address: Port No - TCP Opening connection (optional)

 Communication:  sub-controller Bytes Flow
---> Sub-Controller Primary Frame - Unconfirmed User Data
         LEN(20) DIR(1) PRM(1) FCV(0) FCB(0) DEST(4) SRC(3)
         05 64 14 c4 04 00 03 00 c7 17
         c7 c7 01 3c 02 06 3c 03 06 3c 04 06 3c 01 06 6b ae

---> Sub-Controller     Transport Header
         FIR(1) FIN(1) SEQ# 7
         c7 c7 01 3c 02 06 3c 03 06 3c 04 06 3c 01 06

---> Sub-Controller     Application Header, Read Request
         FIR(1) FIN(1) CON(0) UNS(0) SEQ# 7
         c7 01 3c 02 06 3c 03 06 3c 04 06 3c 01 06

         Rx Object 60(Class Data), variation 2, qualifier 0x06(All Points)
```

Fig. 3 Request bytes flow of DNP3 protocol [43]

security bytes and in last section, cryptography dynamic buffer (CDB) is employed which keeps the information of whole security development.

3.2 New DNP3 stack design and bytes representation

In Fig. 5, a new detail DNP3 logical stack has been designed which shows the bytes flow from upper layer to lower layer and/or vice versa, with security implementation bytes. Logical 48 bytes are shown in application layer buffer, which represent the constructed bytes included user bytes, application layer header bytes and cryptography bytes (security implementation bytes). These bytes are constructed and placed in number of rows as RW0 to RW9 and columns 'CL0' to 'CLn' with corresponding offsets. The value 'n' shows the total number of



```
In-case of Reponses

<+++ session    Insert request in queue: Integrity Poll Due to Master Restart

<+++ Sub-Controller     Build DNP3 Message: NULL Response

<+++ Sub-Controller         Insert request in queue: NULL Response

<--- Sub-Controller          Application Header, Unsolicited
          FIR(1) FIN(1) CON(1) UNS(1) SEQ# 0
          f0 82 90 00

===> Main Controller        Application Header, Unsolicited
          FIR(1) FIN(1) CON(1) UNS(1) SEQ# 0
          f0 82 90 00

---> Sub-Controller        Application Header, Disable Unsolicited Messages
          FIR(1) FIN(1) CON(0) UNS(0) SEQ# 1
          c1 15 3c 02 06 3c 03 06 3c 04 06

          Rx Object 60(Class Data), variation 2, qualifier 0x06(All Points)

          Rx Object 60(Class Data), variation 3, qualifier 0x06(All Points)

          Rx Object 60(Class Data), variation 4, qualifier 0x06(All Points)

<+++ Sub-Controller     Insert request in queue: Disable Unsolicited Response

<=== Sub-Controller         Application Header, Response
          FIR(1) FIN(1) CON(0) UNS(0) SEQ# 1
          c1 81 90 00

===> Main Controller        Application Header, Response
          FIR(1) FIN(1) CON(0) UNS(0) SEQ# 1
          c1 81 90 00
```

Fig. 4 Response bytes flow of DNP3 protocol

bytes within columns but number of rows is fixed up to nine (or 0—9). Usually, in application layer the request or response bytes are placed and distinct by header bytes and communication is flown as unicast. While during broadcasting of bytes from main controller to several receivers, the address range such as FFF0-FFF is added in link layer header fields: source field and destination address field, based on network setup [7, 40]. The cases, which have been occurred during logical communication of application layer from sender to receiver and/or vice versa, plus pseudo-transport layer and data link layer flows are as followed:

The APDU bytes are constructed in application layer with the implementation of security, then bytes are passed to lower layer for further processing [7, 28]. The shaded bytes in whole DNP3 stack are representing the padding bytes and the bytes 'xx,xx,xx,….', are allocated for CDB, while the bytes in row no.9 with corresponding offset 0x0100 are reserved for especial cases or future development. The more detail related with bytes representation is depicted in Table 4.

The proposed study trends to secure the SCADA/DNP3 broadcasting system using cryptography mechanism. In cryptography [28, 38], public key encryption flow is not appropriates

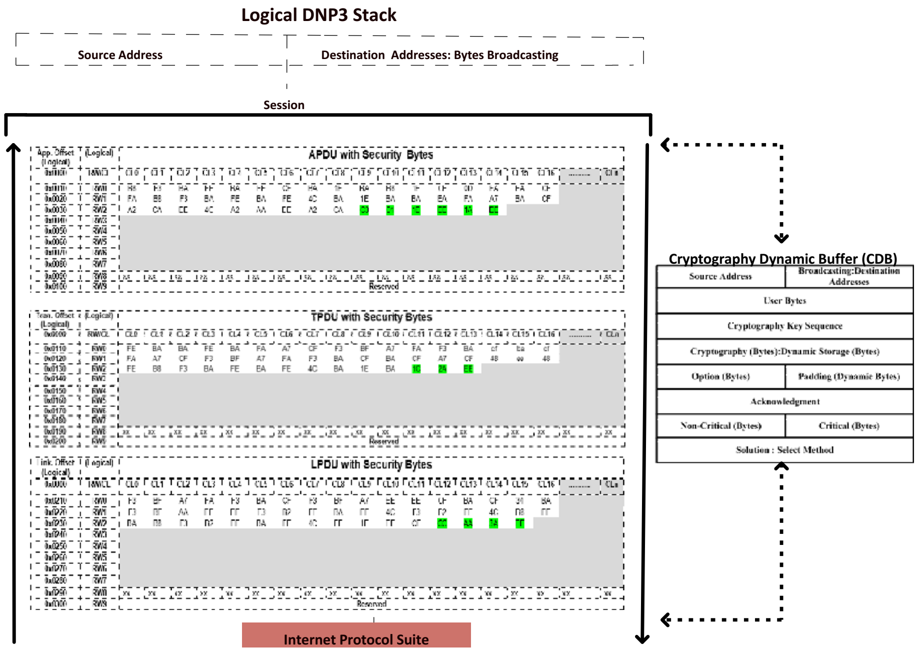


Fig. 5 DNP3 protocol stack with CDB

for broadcasting communication, the main reasons indentified by study are: the number of keys generation, keys distribution and keys utilization [19, 47]. Therefore, this study used

Table 4 DNP3 stack bytes representation

Logical Stack (Bytes)	DNP3 (Stack) :Bytes Representation
$OS_{0010}\cdot R_{0000}\cdot C_{0000}\cdot B_{0000}$___$OS_{0010}\cdot R_{0000}\cdot C_{0016}\cdot B_{0016}$	Application Layer User Bytes
$OS_{0020}\cdot R_{0001}\cdot C_{0000}\cdot B_{0000}$___$OS_{0020}\cdot R_{0001}\cdot C_{0016}\cdot B_{0016}$	Application Layer User Bytes
$OS_{0030}\cdot R_{0002}\cdot C_{0000}\cdot B_{0000}$___$OS_{0030}\cdot R_{0002}\cdot C_{0008}\cdot B_{0008}$	Application Layer User Bytes
$OS_{0030}\cdot R_{0002}\cdot C_{0009}\cdot B_{0009}$___$OS_{0030}\cdot R_{0002}\cdot C_{0010}\cdot B_{0010}$	Application Layer Header Bytes
$OS_{0030}\cdot R_{0002}\cdot C_{0011}\cdot B_{0011}$___$OS_{0030}\cdot R_{0002}\cdot C_{0014}\cdot B_{0014}$	Security Bytes
$OS_{0110}\cdot R_{0000}\cdot C_{0000}\cdot B_{0000}$___$OS_{0110}\cdot R_{0000}\cdot C_{0016}\cdot B_{0016}$	Pseudo-transport Layer User Bytes
$OS_{0120}\cdot R_{0001}\cdot C_{0000}\cdot B_{0000}$___$OS_{0120}\cdot R_{0001}\cdot C_{0016}\cdot B_{0016}$	Pseudo-transport User Bytes
$OS_{0130}\cdot R_{0002}\cdot C_{0000}\cdot B_{0000}$___$OS_{0130}\cdot R_{0002}\cdot C_{0010}\cdot B_{0010}$	Pseudo-transport User Bytes
$OS_{0130}\cdot R_{0002}\cdot C_{0011}\cdot B_{0011}$	Pseudo-transport Header Bytes
$OS_{0130}\cdot R_{0002}\cdot C_{0012}\cdot B_{0012}$___$OS_{0130}\cdot R_{0002}\cdot C_{0013}\cdot B_{0013}$	Security Bytes
$OS_{0210}\cdot R_{0000}\cdot C_{0000}\cdot B_{0000}$___$OS_{0210}\cdot R_{0000}\cdot C_{0016}\cdot B_{0016}$	Data Link Layer User Bytes
$OS_{0220}\cdot R_{0001}\cdot C_{0000}\cdot B_{0000}$___$OS_{0220}\cdot R_{0001}\cdot C_{0016}\cdot B_{0016}$	Data Link Layer User Bytes
$OS_{0230}\cdot R_{0002}\cdot C_{0000}\cdot B_{0000}$___$OS_{0230}\cdot R_{0002}\cdot C_{0011}\cdot B_{0011}$	Data Link Layer User Bytes
$OS_{0230}\cdot R_{0002}\cdot C_{0012}\cdot B_{0012}$___$OS_{0230}\cdot R_{0002}\cdot C_{0013}\cdot B_{0013}$	Data Link Layer Header Bytes
$OS_{0230}\cdot R_{0002}\cdot C_{0014}\cdot B_{0014}$___$OS_{0230}\cdot R_{0002}\cdot C_{0015}\cdot B_{0015}$	Security Bytes

'OS' and 'B' represent the number of offsets and bytes utilized within stack,and 'R' and 'C' represent the number of rows and columns corresponding to offsets symmetric and hashing algorithms to enhance the security of SCADA/DNP3 system during bytes broadcasting from main controller to remote units (RUs).

3.3 Security implementation

Two security developments are made to observe the level of security during SCADA/DNP3 broadcasting transmission, and designated as S-bed^1 and S-bed^2, In S-bed^1; 3-way hashing is computed in-which SHA-2 algorithm is deployed at each layer means that, SHA-2 is deployed at application layer, pseudo-transport layer and data link layer of DNP3 and symmetric encryption using AES algorithm is deployed at application layer. In security development or S-bed^2; 3-way hashing function is deployed at each layer which is application layer, pseudo-transport layer and data link layer of DNP3 and symmetric encryption is deployed at application layer and data link layer. More detail related with security development is illustrated in Fig. 6, while in Table 5 cryptography algorithms are selected at each layer in DNP3 stack and shaded area shows the absence of security implementation (or symmetric encryption).

Significance The security developments such as S-bed^1 and S-bed^2 are tested in SCADA/DNP3 testbed setup (or network setup) illustrated in Fig. 8. Subsequently in Fig. 10, the range of RUs is increased up to sixteen with additional router 'R4' and switch 'S3', and also afterward, to check the level of security during traffic increased.

Proof (Security Development and message broadcasting) The BR=(1, 2, 3,…., n-1, n) is a set of recipients R, which have received the broadcasting message 'M' from main controller 'C'. The number of recipients in set 'BR' is static and 'n' is generated by controller 'C'. Such that,

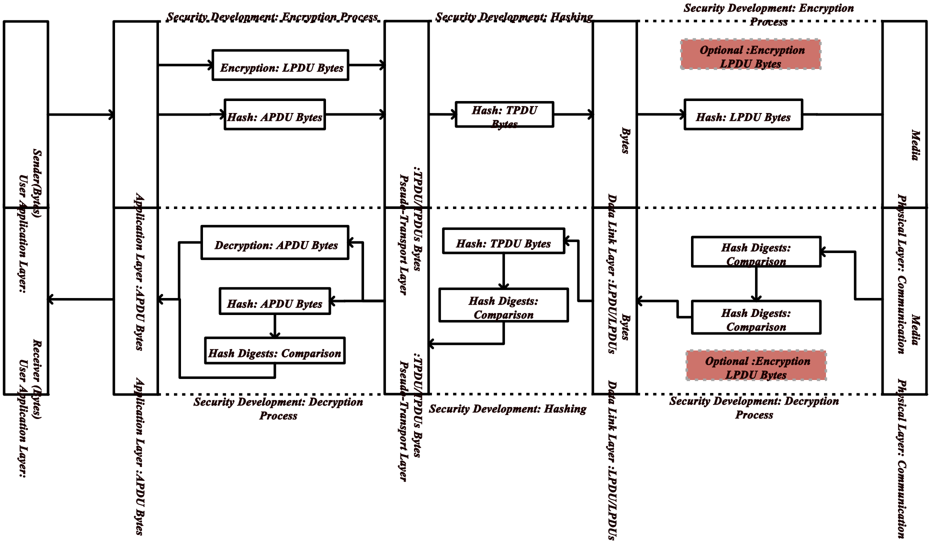

Security Development: Encryption Process
Security Development: Hashing
Security Development: Encryption Process
Encryption: LPDU Bytes
Optional :Encryption LPDU Bytes
Hash: APDU Bytes
Hash: TPDU Bytes
Hash: LPDU Bytes
Sender(Bytes) User Application Layer:
Application Layer :APDU Bytes
Pseudo-Transport Layer :TPDU/TPDUs Bytes
Data Link Layer :LPDU/LPDUs Bytes
Physical Layer: Communication Media
Decryption: APDU Bytes
Hash: APDU Bytes
Hash Digests: Comparison
Hash: TPDU Bytes
Hash Digests: Comparison
Hash Digests: Comparison
Hash Digests: Comparison
Optional :Encryption LPDU Bytes
Receiver (Bytes) User Application Layer:
Application Layer :APDU Bytes
Pseudo-Transport Layer :TPDU/TPDUs Bytes
Data Link Layer :LPDU/LPDUs Bytes
Physical Layer: Communication Media
Security Development: Decryption Process
Security Development: Hashing
Security Development: Decryption Process

Fig. 6 Security developments

Table 5 Security selection in DNP3 stack

Testbed	3-way hashing	Symmetric Encryption
S-bed[1]	Application Layer: APDU Bytes Pseudo-transport Layer: TPDU Bytes Data Link Layer: LPDU Bytes	Application Layer: APDU Bytes
S-bed[2]	Application Layer: APDU Bytes Pseudo-transport Layer: TPDU Bytes Data Link Layer: LPDU Bytes	Application Layer: APDU Bytes Data Link Layer: LPDU Bytes

$X^f \wedge Q^f \wedge J^f \subseteq M^f$, which have constructed in DNP3 stack through function 'f'.

$$\underset{E,H}{\Leftrightarrow} \ \exists : \forall \mu^{AL}(f : \mathrm{X}) \wedge \exists : \forall \alpha^{TL}(f : \mathrm{Q}) \wedge \exists : \forall \beta^{DL}(f : J) \subseteq \mathrm{M}^f \wedge (\mu, \alpha, \beta) \in f$$

The functions $\mu, \alpha, and\ \beta$ are security computing functions which perform the encryption (E) and hashing (H). ε

Bytes Broadcast:

$$\forall \mu^{AL}_{(E,H)} : f\left\{\sum_i^k \mu^{AL}(f : \mathrm{X})\right\} \Rightarrow \forall \alpha^{TL}_{(H)} : f\left\{\sum\nolimits_i^k \alpha^{TL}(f : Q)\right\} \Rightarrow \forall \beta^{DL}_{(H)}$$
$$: f\left\{\sum_i^k \beta^{DL}(f : J)\right\} \in \mathrm{M}^f \Big|\Big| CDB : \mathrm{B}R\big(\mathrm{R}_{(i\ldots\ldots,\ n-1,\ n)}\big) \tag{1}$$

Bytes Received: Direct Function

$$\forall \beta^{DL}_{(H)} : f\left\{\sum_i^k \beta^{DL}(f : J)\right\} \Rightarrow \forall \alpha^{TL}_{(H)} : f\left\{\sum\nolimits_i^k \alpha^{TL}(f : Q)\right\} \Rightarrow \forall \mu^{AL}_{(E,H)}$$
$$: f\left\{\sum\nolimits_i^k \mu^{AL}(f : X)\right\} \in \mathrm{M}^f \Big|\Big| CDB \tag{2}$$

Bytes Received: In-Direct Function. Eq. (2) $\Longrightarrow$

$$\mathrm{R} : \ f_n : j^{DL}_{f \to f_n}\{f_r : j_{HB}(f : j^{DL}_{UB}, f : j^{DL}_{CRC})\} \Rightarrow \mathrm{R} : f_n : j^{DL}_{f \to f_n}(j^{DL}_{UB}) \in \mathrm{M}^f \Big|\Big| CDB \tag{2.1}$$

$$\mathrm{R} : \ f_n : Q^{TL}_{f \to f_n}\{f_r : \mathrm{Q}_{HB}(f : Q^{TL}_{UB})\} \Rightarrow \mathrm{R} : f_n : Q^{TL}_{f \to f_n}(Q^{TL}_{UB}) \in \mathrm{M}^f \Big|\Big| CDB \tag{2.2}$$

$$\mathrm{R} : \ f_n : \mu^{AL}_{f \to f_n}\{f_r : \mu_{HB}(f : \mu^{AL}_{UB})\} \Rightarrow \mathrm{R} : \ f_n : \mu^{AL}_{f \to f_n}(\mu^{AL}_{UB}) \in \mathrm{M}^f \Big|\Big| CDB \tag{2.3}$$

Splitting of header bytes ‘HB’ from user bytes’UB’ within each layer of DNP stack and subsequently, bytes are utilized at upper layer. Where ‘i’ represents the total number of bytes generated within stack with limit ‘k’ and ‘CDB’ shows the number of bytes utilized during security deployment. If the optional confirmation bit is set in message broadcast then, each recipient reply followed by one-to-one or unicast communication.

3.4 Cryptography dynamic buffer

In Fig. 4, the cryptography dynamic buffer (CDB) contains number of fields including source address, destination broadcasting addresses, cryptography key sequence, cryptography (bytes): dynamic storage (bytes), option (bytes), padding (dynamic bytes), acknowledgment, non-critical (bytes), critical (bytes), and solution (select method) [40]. CDB is employed to keep the tracks of security in proper sequence and store and monitor the overall information of stack and security development. The CDB is based on 56 bytes from application layer stack and remaining 1992 bytes are constructed as application protocol data unit (APDU) bytes which would further utilize in lower layer (s) of DNP3. More detail related with CDB fields is depicted in Table 6.

Table 6 CDB fields and description

No.	Cryptography Dynamic Buffer (CDB) Fields	Occupied Bytes	Description
1	Source Address	1 Bytes	Additional source address is added which would be helpful during bytes decryption process.
2	User Bytes	8 Bytes	Keeps the information of DNP3 Stack bytes such as APDU, TPDU and LPDU.
3	Cryptography Key Sequence	1 Byte	Cryptography keys are employed and counted with unique sequence numbers.
4	Cryptography: Dynamic Storage	18—56 Bytes	Padding bytes are added and bytes are allocated to each field upon needs.
5	Padding	Dynamic Bytes	Stack bytes are constructed and remaining bytes are padded.
6	Optional	1 Byte	Verify the message contents before transmitting to open networks.
7	Critical	1 Byte	Define the abnormal transmission status.
8	Non-Critical	1 Byte	Define the normal transmission status.
9	Solution: Select Method	1 Byte	Define and keeps the information of the security method being employed such as S-bed^1 and S-bed^2.
10	Acknowledgment	1 Byte	To Acknowledge, bytes confirmations and also user defined exceptions.
11	Destination Broadcasting Addresses	2 Bytes	Addresses are added which would be helpful during bytes decryption process.

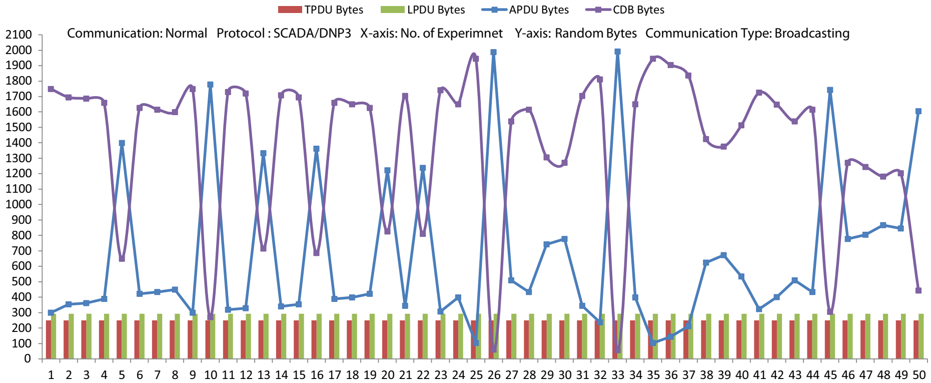


Fig. 7 CDB bytes allocation and utilization

The performance measurements in Fig. 7 shows that CDB space is sufficient during security development and relevant information storage, even in-case of maximum bytes have been received from application layer.

4 Testbed configuration and setup

In SCADA tesbed in Fig. 8, eight remote units (RUs) are employed which are indirectly connected with the physical environment through the direct access of sensors in water pumping system. Four remote units (RUs) are located in station 1 and remaining RUs are located in station 2, and are monitored and controlled from main controller (side). Main controller is superior in SCADA network, which is designed and configured to control, and to send supervisory commands to designated remote units (RUs) [28, 31]. In station 1, remote units (RUs), such as RU1, RU2, RU3, and RU4, are connected with switch 'S1'; while in station 2, remote units (RUs), such as RU5, RU6, RU7, and RU8, are connected with switch 'S2'. Main controller initiates the transmission and broadcast the message (or command) to each station via router 'R1', which is configured and further connected and/or accessed with the designated routers 2 and 3. This study follows a static structure for SCADA/

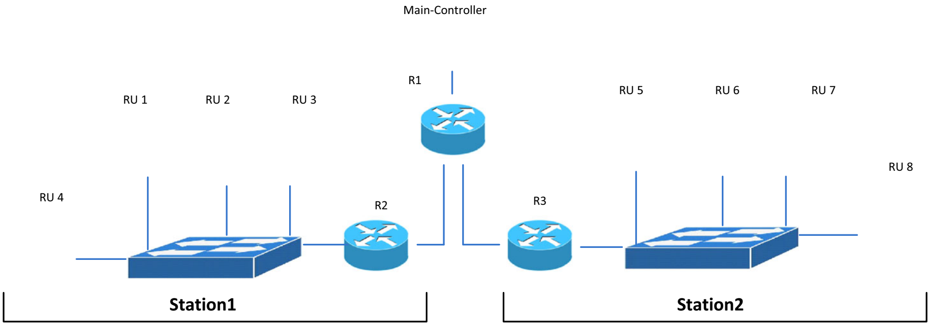


Fig. 8 SCADA/DNP3 testbed configuration and setup

Fig. 9 SCADA lab views

DNP3 broadcasting system, in which, remote units (RUs) are defined and known in advance, and accounted at main controller side. As consequence to restrict the dynamic entry of RU, the man-in-the-middle attack could not successful in broadcasting communication [3, 14, 29]. Figure 9 shows the SCADA lab views where the measurements are conducted.

In Fig. 10, the remote units (RUs) are increased up to sixteen. The additional eight remote units (RUs), such as RU9, RU10, RU11, RU12, RU13, RU14, RU15, and RU16, are located in station 3 via switch 'S3' and router 'R4', and are statically configured and/or updated at main controller side or in routing table of Router 'R1'.

Two security developments have been made for SCADA broadcasting communication followed by the scenario of Figs. 8 and 10. Thus, the shorter name is assigned for each security development as S-bed[1] and S-bed[2].

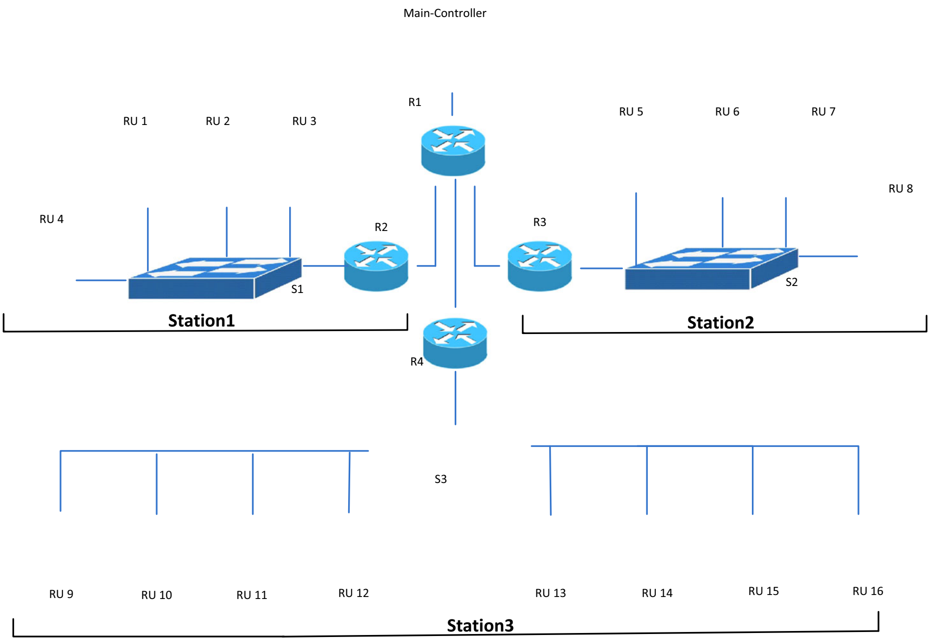


Fig. 10 SCADA/DNP3 testbed nodes increased

5 Performance measurement and analysis

In Table 7, the attacks including, authentication attacks: guessing shared key, brute force, and password guessing; integrity attacks: frame injection, data replay, and data deletion; and confidentiality attacks: eavesdropping, key cracking, and man-in-the-middle, are launched by employing of attack tools such as, cracking tools, sniffer, dsniff, winsniffer, and password dictionary for authentication attacks; airpwn, file2air, dinject/reinject, capture and injection tools, jamming and injection tools for integrity attacks; and ethereal, ettercap, kismet, aircrack, airsnort, dsniff, and ettercap for confidentiality attacks, which interrupt/change the normal flow of SCADA broadcasting transmission (or warm the SCADA broadcasting traffic) [12, 28, 31]. The attack tools are employed and designated as potential attackers for SCADA/DNP3 broadcasting system, and to change the normal sequence of communication in between the main controller and the remote units (RUs). The performances that were observed during abnormal scenarios prove the validation (process) of security implementation and also assess the security that attained corresponding to impact, and attack impact percentages were measured on the basis of attack detection percentages in broadcasting system. For example, some attacks were detected but their influences (or impacts) were minimal or equivalent to zero thus, these attacks are not accounted in total of attacks detection percentages.

In performance Fig. 11, the attack detection is 15 percentages, while attack impact on system is 7 percentages and security attained is approximately 93 percentages, which validate the proposed security implementation. At the other side in performance Fig. 12, the attack detection and impact percentages have decreased to 8 percentages and 3 percentages, and the security is increased up to 97 percentages in SCADA/DNP3 broadcasting system. The total number of authentication and confidentiality attacks exceed due to the absence of symmetric encryption at data link layer, which created performance difference between S-bed^1 and S-bed^2 measurements.

In performance Fig. 13, the network traffic has been increased and successful experiments are performed 100 times to measure the security level during abnormal communication. As results analysis, the level of security is decreased as increasing of network nodes within SCADA/DNP3 testbed. As consequence, this study also shows the significant security enhancement while comparing with existing works [1, 4–6, 11, 23, 27, 30, 32].

6 Related study

Four field devices are employed to conduct the experiments. Master station or 'merging unit (MU)' send the message to remote station or substation with the speed of 960 Hz. Upon remote station message receive; the *intelligent electronic device*

Table 7 Performance measurement and evaluation

Results (Performance)	T-bed^1	T-bed^2	T-bed^2 (Nodes Increased)
Attack Detection (%)	15 %	8 %	14 %
Attack Impact (%)	7 %	3 %	8 %
Security (%)	93 %	97 %	92 %

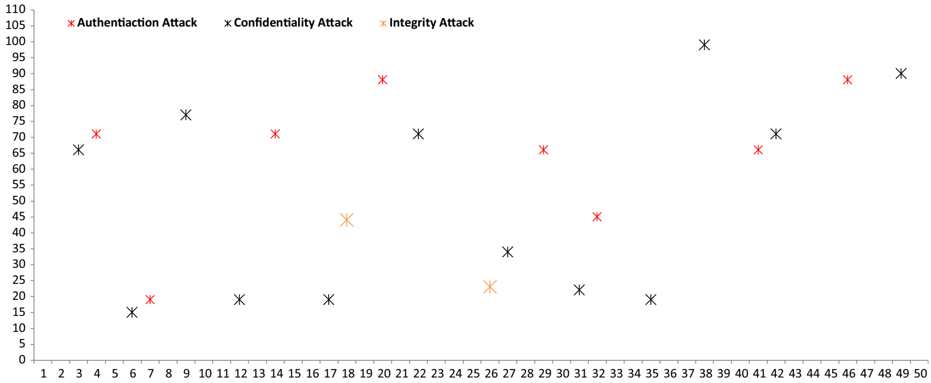


Fig. 11 S-bed^1: attack detection

(IED) collect and sends back response to master station through remote station. If *IED detects any intrusion or anomaly during transmission, then response* 'GOOSE message' *is transmitted back to master station and further processing (of actuator functions) will depend on master station acknowledgment. As the result; message authentication, confidentiality and integrity mechanisms have been deployed between master station and remote station, using of advance encryption standard (AES) 256, HMAC and MD5 as part of cryptography* [2, 22, 35]. *The* IEC 61850 standard is used to simulate the SCADA communication between the field devices and deployed the cryptography solutions for message security. The performance overhead is also calculated that is based on encryption and decryption operations [16, 36, 45].

Traditionally, control systems had been designed without security consideration in mind and emphasize only on physical security. Nowadays, large number of developments relevant with SCADA security is conducted such as using firewalls, DMZs and other Key encryption and distribution solutions, but these developments are based on end-to-end message delivery [20, 28]. Schweitzer Engineering Laboratories, Inc reviews several exiting security developments and then an inclusive security solution is proposed for SCADA electric industry, and cryptography based solutions are suggested as best approaches for SCADA system security. The research paper [13]

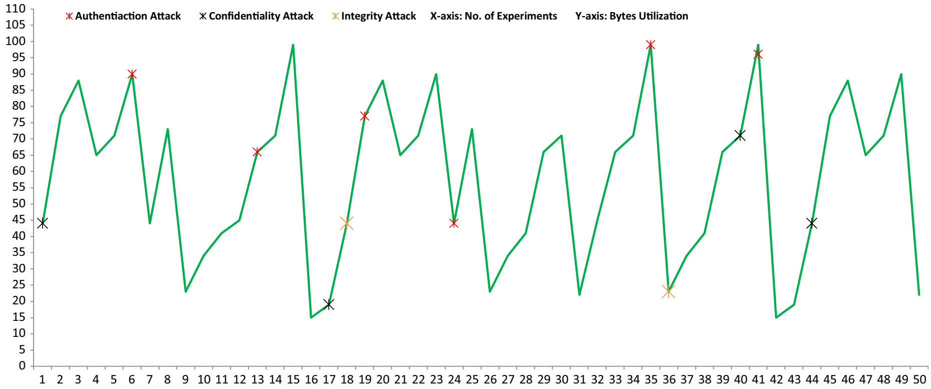


Fig. 12 S-bed^2: attack detection

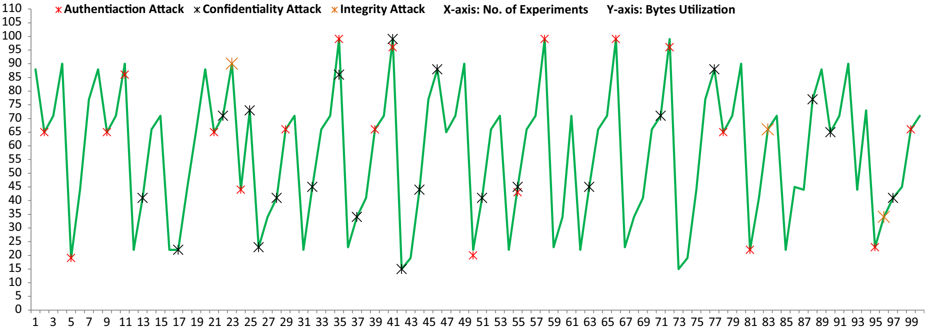


Fig. 13 S-bed^2: attack detection during nodes increased

provides detail review: on cryptography algorithms, their implementation, advantages and disadvantages of cryptography solutions, and the major attacks that creates vulnerabilities in SCADA communication [33, 34, 46].

A message is encrypted at master station and transmitted to all remote stations within SCADA network. Upon receiving, each remote station performs decryption process using asymmetric keys. This approach is infeasible because same message is encrypted many times and all stations have acquired to generate the public and private keys during SCADA broadcasting communication [38]. In another research; master station generates multiple packets and encrypts each packet with each remote station public key, but this solution acquired much time and also impact on performance. The hash function provides integrity mechanism and safety measurements from attacks, such as data reply, data modify, and data delete. Master station generates the hash digest of message that is being transmitted to remote stations and also encrypts the hash digest with private key; this function act as digital signature [28, 34]. Upon receiving at each remote station; each station uses master public key to decrypt the message. This would verify and concludes that the message is secured and unauthorized entity (or attacker) is involved during transmission. The encryption and hash functions are placed at the each end of IEC 61850 protocol message header that are to achieve the security goals, such as authentication and integrity, within SCADA communication [15, 22, 37, 45].

7 Conclusion and future work

Security is a main concern that has been accounted in information technology (IT), many security mechanisms are available but are limited in design and development for secure SCADA broadcasting system, and are also not commonly available for the traditional systems. In this study, an inclusive development was made that secure the SCADA/DNP3 broadcasting system with the best performance evaluation and validation of security, which have been measured in abnormal communication. As consequence, the performance results concluded that the level of security is successfully enhanced for SCADA/DNP3 broadcasting system, while implementing of proposed inclusive security within SCADA/DNP3 protocol before transmitting the protocol bytes to open protocols or networks.

In the future work, the proposed inclusive security solution will be deployed and tested in real environment in which hundreds of SCADA nodes are interconnected with main controller and/or several sub-controllers are interconnected with main-controller, as similar to distributed computing. However, the proposed security development was limited in the terms to contribute for other security parameter, such as non-repudiation. In cryptography, the digital signature algorithm is available and considered as a best approach for achieving of non-repudiation security for sensitive information. Therefore, digital signature algorithm would be deployed to keep the SCADA/DNP3 broadcasting system secure against non-repudiation attacks.

Acknowledgment This research was supported by next generation information computing development program through the national research foundation of Korea, and funded by the ministry of science, ICT & Future Planning (2012M3C4A7033348).

References

1. Arango IM, Izquierdo J, Campbell E, Pérez-García R (2014) Cloud-based decision making in water distribution systems. Procedia Eng 2014(89):488
2. Benrhouma O, Hermassi H, Benrhouma O, Hermassi H, Belghith S (2015) Security analysis and improvement of a partial encryption scheme. Multimed Tools Appl 74(11):3617–3634. doi:10.1007/s11042-013-1790-4
3. Bhattacherjee S, Palash S (2013) Complete tree subset difference broadcast encryption scheme and its analysis. Des Codes Crypt 66(1–3):335–362
4. Chen HC (2013) Secure multicast key protocol for electronic mail systems with providing perfect forward secrecy. Security Comm 2013 Net 6:100–107. doi:10.1002/sec.536
5. Cho EJ, Hong CS, Lee S, Jeon S (2013) A partially distributed intrusion detection system for wireless sensor networks. Sensors 13(12):15863–15879. doi:10.3390/s131215863
6. Chunsheng G, Jixing G (2014) Known-plaintext attack on secure kNN computation on encrypted databases. Security Comm Net 2014(7):2432–2441. doi:10.1002/sec.954
7. Clarke D, Reynders E Wright (2004) Practical Modern SCADA Protocols: DNP3, 60870.5 and Related Systems
8. Coates, G. M.Hopkinson, K.M., Graham, S. R., Kurkowski, S.H (2008) Collaborative, Trust-Based Security Mechanisms for a Regional Utility Intranet. Power Systems, IEEE Transactions 2008, Volume:23, Issue: 3, 10.1109/TPWRS.2008.926456
9. Coates, G. M. Hopkinson, K.M., Graham, S. R., Kurkowski, S.H. (2010) A Trust System Architecture for SCADA Network Security. Power Delivery, IEEE Transactions, Volume: 25, Issue: 1, 10.1109/TPWRD.2009.2034830
10. Davis JE, Okhravi TH, Grier C, Overbye TJ, Nicol D (2006) SCADA cyber security testbed development. Power Symp 2006, NAPS, 38th North American, IEEE. doi:10.1109/NAPS.2006.359615
11. Eghbal Ghazizadeh, Mazdak Zamani, Jamalul-lail Ab Manan, Mojtaba Alizadeh (2014) Trusted Computing Strengthens Cloud Authentication. The Scientific World Journal, vol. 2014, Article ID 260187, 17 pages, doi:10.1155/2014/260187
12. El Mrabet N, Fournier JJA, Louis G, Ronan L (2015) A survey of fault attacks in pairing based cryptography. Cryptogr Commun 7(1):185–205. doi:10.1007/s12095-014-0114-5
13. Fujisaki E, Okamoto T. (1999) Secure integration of asymmetric and symmetric metric encryption schemes. In Advances in Cryptology – CRYPTO'99, LNCS, Vol.1666. Spring-Verlag, 537–554
14. Gentry C, Waters B (2009) Adaptive security in broadcast encryption systems (with short Ciphertexts). advances in cryptology - EUROCRYPT 2009. Lecture Notes in Comput Science 5479:171–188. doi:10.1007/978-3-642-01001-9_10
15. Herbert F (2008) Securing IEC 61850. Power and Energy Society General Meeting - Conversion and Delivery of Electrical Energy in the 21st Century 2008, IEEE, doi:10.1109/PES.2008.4596335
16. Hong S (2010) Experiments for embedded protection device for secure SCADA communication. Power and Energy Eng Conf (APPEEC). doi:10.1109/APPEEC.2010.5448606
17. Jang U, Lim H, Kim H (2014) Privacy-enhancing security protocol in LTE initial attack. Symmetry 6:1011–1025
18. Jeong-Han Yun, Sung-Ho Jeon, Kyoung-Ho Kim, Woo-Nyon Kim (2013) Burst-based Anomaly Detection on the DNP3 Protocol. International Journal of Control and Automation, Vol. 6, No. 2
19. Jongkil Kim, Susilo, W., Man Ho Au, Seberry, J (2015) Adaptively Secure Identity-Based Broadcast Encryption With a Constant-Sized Ciphertext. Information Forensics and Security, IEEE Transactions on, vol.10, no.3, pp.679,693, doi: 10.1109/TIFS.2014.2388156
20. Kang, H. M. Kim. (2007) A Proposal for Key Policy of Symmetric Encryption Application to Cyber Security of KEPCO SCADA Network. FGCN '07 Proceedings of the Future Generation Communication and Networking 2007,Volume 02, doi:10.1109/FGCN.2007.36

21. Kang DJ, Joo LJ, Joo KS, Hyuk PJ (2009) Analysis on cyber threats to SCADA systems. Trans Distrib Conf Expos Asia and Pacific 2009, IEEE. doi:10.1109/TD-ASIA.2009.5357008
22. Khalil IM, Khreishah A, Azeem M (2014) Cloud computing security: a survey. Computers 3(1):1–35. doi:10.3390/computers3010001
23. Kim I, Oh D, Yoon MK, Yi K, Ro WW (2013) A distributed signature detection method for detecting intrusions in sensor systems. Sensors 2013 13(4):3998–4016. doi:10.3390/s130403998
24. Kiuchi, M. Serizawa, Yoshizumi (2009) Security technologies, usage and guidelines in SCADA system networks. ICCAS-SICE, IEEE,IAN: 10982993
25. Krajewski J (2014) Situational awareness –the next leap in industrial human machine interface design. Schneider Electric 2014:1–11
26. Lee C-M (2015) Criminal profiling and industrial security. Multimed Tools Appl 74(5):1689–1696. doi:10.1007/s11042-014-2014-
27. Moon D, Im H, Lee JD, Park JH (2014) MLDS: multi-layer defense system for preventing advanced persistent threats. Symmetry 2014 6(4):997–1010. doi:10.3390/sym6040997
28. Musa, A. Shahzad, A.Aborujilah (2013) Secure security model implementation for security services and related attacks base on end-to-end, application layer and data link layer security. Proceeding ICUIMC ’13 Proceedings of the 7th International Conference on Ubiquitous Information Management and Communication 2013, doi:10.1145/2448556.2448588
29. Nikolic I, Wang L, Wu S (2015) The parallel-cut meet-in-the-middle attack. Cryptogr Commun 7(3):331–345. doi:10.1007/s12095-014-0118-1
30. Obaidat MS, Woungang I, Dhurandher SK, Koo V (2014) A cryptography-based protocol against packet dropping and message tampering attacks on mobile ad hoc networks. Security Comm Net 2014(7):376–384. doi:10.1002/sec.731
31. Patel SC, Bhatt GD, Graham JH (2009) Improving the cyber security of SCADA communication networks. Commun ACM 52(7):139–142. doi:10.1145/1538788.1538820
32. Premnath A.P., Ju-Yeon Jo, Yoohwan Kim (2014) Application of NTRU Cryptographic Algorithm for SCADA Security. Information Technology: New Generations (ITNG) 2014, 11th International Conference on, vol., no., pp.341, 346, 7–9, doi: 10.1109/ITNG.2014.38
33. Risley, J. Roberts, P. Ladow. (2003) Electronic Security Of Real-Time Protection and Scada Communications. Schweitzer Engineering Laboratories Inc 2003, https://www.selinc.com/WorkArea/linkit.aspx
34. Robles RJ, Balitanas M, Kim T-h (2011) Security encryption schemes for internet SCADA: comparison of the solutions. Commun Comput Inf Sci 2010 223:19–27. doi:10.1007/978-3-642-23948-9_4
35. Robles RJ, Kim T-h (2010) An encryption scheme for communication internet SCADA components. Commun Comput Inf Sci 2010 74:56–64. doi:10.1007/978-3-642-13346-6_6
36. Robles, R., Kim, T.H. (2012) Encryption Schemes Applied in SCADA Components Communication. 15, (3) pp. 1241–1252. ISSN 1344–8994, Refereed Article, University of Tasmania, http://ecite.utas.edu.au/8309
37. Saxena, O. Pal, S. Saiwan, Z. Saquib (2011) Token Based Key Management Scheme for SCADA Communication. International Journal of Distributed and Parallel Systems (IJDPS) 2011, Vol.2, No.4, http://airccse.org/journal/ijdps/papers/0711dps07.pdf
38. Saxena A, Pal O, Saquib Z (2011) Public Key cryptography based approach for securing SCADA communications. Commun Comput Inf Scie 142(2011):56–62. doi:10.1007/978-3-642-19542-6_10
39. SCADA, http://en.wikipedia.org/wiki/SCADA
40. Shahzad A, Irfan SM (2014) Deployment of New dynamic cryptography buffer for SCADA security enhancement. J Appl Sci. doi:10.3923/jas.2014.2487.24
41. Sommestad, T., Ericsson, G.N., Nordlander, J (2010) SCADA system cyber security — A comparison of standards. Power and Energy Society General Meeting, 2010 IEEE, vol., no., pp.1,8, doi: 10.1109/PES.2010.5590215
42. Stouffer J, Falco K Kent (2006) Guide to Supervisory Control and Data Acquisition (SCADA) and Industrial Control Systems Security. Recommendations of the National Institute of Standards and Technology
43. Test Harness, Triangle MicroWorks, www.trianglemicroworks.com
44. Wang Chunlei; Fang Lan; Dai Yiqi (2010) A Simulation Environment for SCADA Security Analysis and Assessment. Measuring Technology and Mechatronics Automation (ICMTMA) 2010, International Conference on, vol.1, no., pp.342,347, 13–14, doi: 10.1109/ICMTMA.2010.603
45. Yoo H, Shon T (2015) Novel approach for detecting network anomalies for substation automation based on IEC 61850. Multimed Tools Appl 74(1):303–318. doi:10.1007/s11042-014-1870-0
46. Zhang Y, Xiao Y, Ghaboosi K, Zhang J, Deng H (2012) A survey of cyber crimes. Secur Comm Net 5:422–437. doi:10.1002/sec.331
47. Zhu, Wen Tao (2013) Towards secure and communication-efficient broadcast encryption systems. Journal of Network and Computer Applications 36.1 (2013): 178–186.

AAmir Shahzad received “PhD degree in Network Security and Cryptography” from University Kuala Lumpur, Malaysia and is teaching computer science and I.T courses at University Kuala Lumpur. He involves his students in “traditional and real time networks, security and policies”. His professional interests focus on “networks security via cryptography, networks protocols configuration and related security, software engineering, artificial intelligence, multimedia design and applications, and biometric information systems.” Currently, he is appointed as a researcher in Chon Buk National University, Korea.

Malrey Lee is a Professor in the Department of Electronics and information Engineering and member of Research Center for Advanced Image and Information Technology at Chonbuk National University, South Korea. She has over forty publications in various areas of Computer Science, concentrating on Artificial Intelligence, Robotics, Medical Healthcare and Software Engineering.

Dr Changhoon Lee received his Ph.D. degree in Graduate School of Information Management and Security (GSIMS) from Korea University, Korea. In 2008, he was a research professor at the Center for Information Security Technologies in Korea University. In 2009–2011, he was a professor in the School of Computer Engineering in Hanshin University. He is now a professor at the Department of Computer Science and Engineering, Seoul National University of Science and Technology (SeoulTech), Korea. He has been serving not only as chairs, program committee, or organizing committee chair for many international conferences and workshops but also as a (guest) editor for international journals by some publishers. His research interests include information security, cryptography, digital forensics, smart grid security, computer theory etc. He is currently a member of the IEEE, IEEE Computer Society, IEEE Communications, IACR, KIISC, KDFS, KIPS, KITCS, KMMS, KONI, and KIIT societies.

Dr. Naixue Xiong is currently holding a position of professor in School of Computer Science Colorado Technical University, USA.

Dr. Suntae Kim is an assistant Professor of the Department of Software Engineering at Chonbuk National University. He received his BS in Computer Science and Engineering from Chung-Ang University in 2003, and MS and PhD in Computer Science and Engineering from Sogang University in 2007 and 2010. He worked in Software Craft Co. Ltd., as a Senior Consultant for Financial Enterprise Systems during 2002 to 2004. Also, he developed Android-based Smart TV middleware for two years after 2009 at Any point Media Group. His research focuses on software architecture, design patterns, requirements engineering and mining software repository.

Dr.Young-Keun Lee received his Ph.D. in Orthopaedics from University of ChonbukNational University. He used to work at Eulji University as a professor. Currently, he was working at Dason Orthopaedic Clinic Center as a doctor. He is working at chonbuk National University as a professor now. He has over forty publications in various areas of orthopaedic surgery. Also, he is concentrating on artificial intelligence, robotics, medical healthcare and software engineering.

Dr. Kangmin Kim received a Ph.D. from University of Nevada at Reno, and He has been research professor at the Chonbuk National University, He is interested in healthcare, agriculture robot, biomedical and so on.

Dr. Seon-mi Woo received a Ph.D. from Chonbuk National University, She is working at JINI Company, and she is interested in healthcare, multimedia, artificial intelligence and other related disciplines.

Dr. Gisung Jeong received a Ph.D. from Chonbuk National University. He has been a Professor at the WonKwang University. He is interested in healthcare, fire service, and robotics.