

Junji Shikata (Ed.)

LNCS 10681

Information Theoretic Security

10th International Conference, ICITS 2017
Hong Kong, China, November 29 – December 2, 2017
Proceedings



Springer

Commenced Publication in 1973

Founding and Former Series Editors:

Gerhard Goos, Juris Hartmanis, and Jan van Leeuwen

Editorial Board

David Hutchison

Lancaster University, Lancaster, UK

Takeo Kanade

Carnegie Mellon University, Pittsburgh, PA, USA

Josef Kittler

University of Surrey, Guildford, UK

Jon M. Kleinberg

Cornell University, Ithaca, NY, USA

Friedemann Mattern

ETH Zurich, Zurich, Switzerland

John C. Mitchell

Stanford University, Stanford, CA, USA

Moni Naor

Weizmann Institute of Science, Rehovot, Israel

C. Pandu Rangan

Indian Institute of Technology, Madras, India

Bernhard Steffen

TU Dortmund University, Dortmund, Germany

Demetri Terzopoulos

University of California, Los Angeles, CA, USA

Doug Tygar

University of California, Berkeley, CA, USA

Gerhard Weikum


Max Planck Institute for Informatics, Saarbrücken, Germany

More information about this series at <http://www.springer.com/series/7410>

Junji Shikata (Ed.)

Information Theoretic Security

10th International Conference, ICITS 2017
Hong Kong, China, November 29 – December 2, 2017
Proceedings

Editor
Junji Shikata 
Yokohama National University
Yokohama
Japan

ISSN 0302-9743 ISSN 1611-3349 (electronic)
Lecture Notes in Computer Science
ISBN 978-3-319-72088-3 ISBN 978-3-319-72089-0 (eBook)
<https://doi.org/10.1007/978-3-319-72089-0>

Library of Congress Control Number: 2017959633

LNCS Sublibrary: SL4 – Security and Cryptology

© Springer International Publishing AG 2017

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. The publisher remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Printed on acid-free paper

This Springer imprint is published by Springer Nature
The registered company is Springer International Publishing AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

Preface

ICITS 2017, the 10th International Conference on Information Theoretic Security, was held in Hong Kong, China, during November 29 – December 2, 2017. The conference took place on the campus of The Chinese University of Hong Kong. ICITS 2017 was held in cooperation with the International Association for Cryptologic Research (IACR), and supported by IEEE Information Theory Society Hong Kong Chapter. The General Chair of the conference was Kenneth Shum.

ICITS is the successor conference to the 2005 IEEE Information Theory Workshop on Theory and Practice in Information Theoretic Security, held on Awaji Island, Japan. It is now an international conference which deals with all aspects of information-theoretic security and brings together researchers from various areas including cryptography, information theory, and quantum computing. Information-theoretic security is the cryptographic security that does not depend on computational assumptions, and it is achieved by utilizing techniques or methods from various fields such as information theory, discrete mathematics, and quantum physics.

ICITS 2017 had two tracks, a conference track and a workshop track, as did the previous ICITS. This two-track format was started with ICITS 2012, and it has the advantage of bringing together researchers from various areas with different publication cultures. The proceedings contain the accepted papers for the conference track. The accepted works for the workshop track were presented at the conference but do not appear in this volume. The list of the contributions in the workshop track is given before the Table of Contents.

The Program Committee received a total of 42 submissions, of which 12 were accepted for the conference track and 7 for the workshop track. All submitted papers were reviewed by at least 3 members of the Program Committee, who sometimes were assisted by external reviewers.

In addition to the 19 contributed presentations, there were 6 invited talks:

- “Randomness Extraction in the Quantum World” by Kai-Min Chung, Academia Sinica, Taiwan
- “Sufficiently Myopic Adversaries Are Blind” by Sidharth Jaggi, The Chinese University of Hong Kong, Hong Kong
- “Quantum Wiretap Channel Coding and Information Spectrum Methods” by Tomohiro Ogawa, The University of Electro-Communications, Japan
- “A Unified Paradigm of Organized Complexity and Semantic Information Theory” by Tatsuaki Okamoto, NTT, Japan
- “Physical Assumptions for Long-Term Secure Communication” by Rei Safavi-Naini, University of Calgary, Canada
- “Secret Sharing Schemes: Some New Approaches and Problems” by Huaxiong Wang, Nanyang Technological University, Singapore

I would like to thank all the people who have contributed to the success of ICITS 2017. First of all, I would like to thank all the authors who submitted their papers to ICITS 2017. I would also like to thank all members of the Program Committee, who completed the reviews in a timely and professional manner. It was a great honor for me to work together with them. Moreover, I would like to thank the Steering Committee of ICITS, in particular Yvo Desmedt and Rei Safavi-Naini, for their kind support from the initial stage of the conference. I am grateful to the Program Chairs of previous conferences for their advice and assistance, in particular Anderson Nascimento, Stefan Wolf, and Anja Lehmann. I would especially like to thank the General Chair, Kenneth Shum, for organizing and managing the wonderful conference ICITS 2017, and the Treasurer, Chee Wei, for financial management, Wei Kang for taking charge of publicity, and Hoover Yin for designing the website of the conference. I would also like to thank the CANS 2017 program co-chairs, Sran Ćapkun and Sherman S. M. Chow, and the CANS 2017 General Chair, Kehuan Zhang, for the collaboration, because ICITS and CANS were co-located in the campus of The Chinese University of Hong Kong from November 29 to December 2, 2017. Finally, I would like to thank Alfred Hofmann, Elke Werner, and Anna Kramer and other LNCS staff at Springer for their help in publishing the proceedings. Our sponsor was the Institute of Network Coding, The Chinese University of Hong Kong.

October 2017

Junji Shikata

ICITS 2017

The 10th International Conference on Information Theoretic Security

Hong Kong, China, November 29 – December 2, 2017

In cooperation with the International Association for Cryptologic Research (IACR)
Supported by IEEE Information Theory Society Hong Kong Chapter

General Chair

Kenneth Shum The Chinese University of Hong Kong, Hong Kong

Program Chair

Junji Shikata Yokohama National University, Japan

Program Committee

Divesh Aggarwal	National University of Singapore, Singapore
Paulo Barreto	University of Washington, Tacoma, USA
Mario Berta	Imperial College London, UK
Matthieu Bloch	Georgia Institute of Technology, USA
Ignacio Cascudo	Aalborg University, Denmark
Paolo D'Arco	University of Salerno, Italy
Frédéric Dupuis	CNRS, LORIA, Université de Lorraine, France
Benjamin Fuller	University of Connecticut, USA
Peter Gazi	IOHK Research, Hong Kong
Goichiro Hanaoka	AIST, Japan
Masahito Hayashi	Nagoya University, Japan
Mitsugu Iwamoto	The University of Electro-Communications, Japan
Takeshi Koshihara	Waseda University, Japan
Yuan Luo	Shanghai Jiao Tong University, China
Hemanta Maji	Purdue University, USA
Keith Martin	University of London, Royal Holloway, UK
Kirill Morozov	The University of Tokyo, Japan
Anderson Nascimento	University of Washington, USA
Frédérique Oggier	Nanyang Technological University, Singapore
Carles Padró	Universitat Politècnica de Catalunya, Spain
Vinod M. Prabhakaran	Tata Institute of Fundamental Research, India
Rei Safavi-Naini	University of Calgary, Canada
Rafael Schaefer	Technische Universität Berlin, Germany
Vincent Tan	National University of Singapore, Singapore

Stefano Tessaro	University of California, Santa Barbara, USA
Huaxiong Wang	Nanyang Technological University, Singapore
Shun Watanabe	Tokyo University of Agriculture and Technology, Japan

ICITS Steering Committee

Carlo Blundo	University of Salerno, Italy
Yvo Desmedt (Chair)	University College London, UK and University of Texas at Dallas, USA
Yuval Ishai	Technion, Israel
Kaoru Kurosawa	Ibaraki University, Japan
Ueli Maurer	ETH Zurich, Switzerland
C. Pandu Rangan	Indian Institute of Technology, Madras, India
Rei Safavi-Naini	University of Calgary, Canada
Junji Shikata	Yokohama National University, Japan
Stefan Wolf	Università della Svizzera italiana, Switzerland
Moti Yung	Snapchat and Columbia University, USA
Yuliang Zheng	University of Alabama at Birmingham, USA

External Reviewers

Carsten Baum	Fuyuki Kitagawa	David Sutter
Christopher Chubb	Fuchun Lin	Mingyuan Wang
Romar Dela Cruz	Tomoyuki Morimae	Yohei Watanabe
Deepesh Data	Varun Narayanan	Sophia Yakoubov
Rafael Dowsley	Ali Poostindouz	Kenji Yasunaga
Serge Fehr	Manoj Prabhakaran	Lei Yu
Christoph Hirche	Varun Raj	Yun Zhang
Andreas Hülsing	Kaushik Seshadreesan	Lin Zhou
Shaoquan Jiang	Setareh Sharifian	Sufang Zhou
Chethan Kamath	Kazumasa Shinagawa	
Akinori Kawachi	Noah Stephens-Davidowitz	

Sponsor

Institute of Network Coding, The Chinese University of Hong Kong

Workshop Track Presentations

The following papers were accepted to the workshop track of ICITS 2017. They were presented at the conference but do not appear as papers in these proceedings.

1. On Secure Asymmetric Multilevel Diversity Coding Systems
Congduan Li, Xuan Guang, Chee Wei Tan, and Raymond W. Yeung
2. Secure Wireless Communication under Spatial and Local Gaussian Noise Assumptions
Masahito Hayashi
3. Secrecy and Robustness for Active Attack in Secure Network Coding and its Application to Network Quantum Key Distribution
Masahito Hayashi, Masaki Owari, Go Kato, and Ning Cai
4. Information-theoretic Physical Layer Security for Satellite Channels
Angeles Vazquez-Castro and Masahito Hayashi
5. Compressed Secret Key Agreement
Chung Chan
6. Computing on Quantum Shared Secrets
Yingkai Ouyang, Si-Hui Tan, Liming Zhao, and Joseph Fitzsimons
7. Worst-Case Guessing Secrecy Is Meaningful in Secret Sharing Schemes
Mitsugu Iwamoto

Contents

Linear-Time Non-Malleable Codes in the Bit-Wise Independent Tampering Model	1
<i>Ronald Cramer, Ivan Damgård, Nico Döttling, Irene Giacomelli, and Chaoping Xing</i>	
Disproving the Conjectures from “On the Complexity of Scrypt and Proofs of Space in the Parallel Random Oracle Model”	26
<i>Daniel Malinowski and Karol Żebrowski</i>	
Broadcast Encryption with Guessing Secrecy	39
<i>Yohei Watanabe</i>	
Contrast Optimal XOR Based Visual Cryptographic Schemes.	58
<i>Sabyasachi Dutta and Avishek Adhikari</i>	
Verifiably Multiplicative Secret Sharing	73
<i>Maki Yoshida and Satoshi Obana</i>	
Round and Communication Efficient Unconditionally-Secure MPC with $t < n/3$ in Partially Synchronous Network	83
<i>Ashish Choudhury, Arpita Patra, and Divya Ravi</i>	
Catching MPC Cheaters: Identification and Openability	110
<i>Robert Cunningham, Benjamin Fuller, and Sophia Yakoubov</i>	
Secure Grouping Protocol Using a Deck of Cards	135
<i>Yuji Hashimoto, Kazumasa Shinagawa, Koji Nuida, Masaki Inamura, and Goichiro Hanaoka</i>	
Four Cards Are Sufficient for a Card-Based Three-Input Voting Protocol Utilizing Private Permutations.	153
<i>Takeshi Nakai, Satoshi Shirouchi, Mitsugu Iwamoto, and Kazuo Ohta</i>	
Single-Shot Secure Quantum Network Coding for General Multiple Unicast Network with Free Public Communication	166
<i>Go Kato, Masaki Owari, and Masahito Hayashi</i>	

Secure Network Coding for Multiple Unicast: On the Case
of Single Source 188
Gaurav Kumar Agarwal, Martina Cardone, and Christina Fragouli

Rényi Resolvability and Its Applications to the Wiretap Channel 208
Lei Yu and Vincent Y. F. Tan

Author Index 235

Linear-Time Non-Malleable Codes in the Bit-Wise Independent Tampering Model

Ronald Cramer^{1,2}, Ivan Damgård³, Nico Döttling⁴, Irene Giacomelli⁵(✉),
and Chaoping Xing⁶

¹ CWI, Amsterdam, Netherlands

² Leiden University, Leiden, Netherlands

³ Aarhus University, Aarhus, Denmark

⁴ Friedrich-Alexander-University Erlangen-Nürnberg, Erlangen-Nurnberg, Germany

⁵ University of Wisconsin-Madison, Madison, USA

igiacomelli@wisc.edu

⁶ Nanyang Technological University, Singapore, Singapore

Abstract. Non-malleable codes were introduced by Dziembowski et al. (ICS 2010) as coding schemes that protect a message against tampering attacks. Roughly speaking, a code is non-malleable if decoding an adversarially tampered encoding of a message m produces the original message m or a value m' (possibly \perp) completely unrelated to m . It is known that non-malleability is possible only for restricted classes of tampering functions. Since their introduction, a long line of works has established feasibility results of non-malleable codes against different families of tampering functions. However, for many interesting families the challenge of finding “good” non-malleable codes remains open. In particular, we would like to have *explicit constructions* of non-malleable codes with *high-rate* and efficient encoding/decoding algorithms (*i.e.* low computational complexity). In this work we present two explicit constructions: the first one is a natural generalization of the work of Dziembowski et al. and gives rise to the first constant-rate non-malleable code with *linear-time* complexity (in a model including bit-wise independent tampering). The second construction is inspired by the recent works about non-malleable codes of Agrawal et al. (TCC 2015) and of Cheraghchi and Guruswami (TCC 2014) and improves our previous result in the bit-wise independent tampering model: it builds the first non-malleable codes with *linear-time* complexity and *optimal-rate* (*i.e.* rate $1 - o(1)$).

Keywords: Non-malleable codes · Linear-time
Bit-wise independent tampering · Secret-sharing

1 Introduction

Non-malleable codes are a relaxation of error-correcting and error-detecting codes that have useful applications in cryptography. For example, they can be used to protect keys that are stored in non-robust devices against tampering

attacks. Recently, they also found application to computational cryptography (*e.g.* construction of non-malleable commitments [7,36] and domain extension for public-key encryption schemes [20,21]). Roughly speaking, a coding scheme (Enc, Dec) is non-malleable with respect to the tampering function f if decoding $f(\text{Enc}(\mathbf{m}))$ produces the original message \mathbf{m} or a value \mathbf{m}' (possibly \perp) completely unrelated to \mathbf{m} . Moreover, the probability of which one of these two events happens is also independent of \mathbf{m} . As an illustration of the notion, consider a key that is stored in a device. The adversary is able to tamper with the key and gets to see the effect of using the device with the tampered key inside. If the key was coded with a non-malleable code and is decoded before use, this attack becomes useless, as the key actually used after tampering is either unchanged or is unrelated to the original key.

Since a tampering function can always try to decode, modify the message, and encode again, it is clear that non-malleable codes are impossible without restrictions on the tampering function. We therefore restrict the adversary to using functions from a specific class \mathcal{F} . In this case, we say that we have a non-malleable code with respect to the family \mathcal{F} . For example, if the encoding is made by n symbols from a finite field \mathbb{F} , then we can restrict the tampering function to be a function with n independent components (f_1, \dots, f_n) (symbol-wise independent tampering, or bit-wise independent tampering if $\mathbb{F} = \{0, 1\}$). Other important features of the coding scheme are the rate and the computational complexity¹.

Non-malleable codes were introduced in 2010 by Dziembowski et al. [30]. Previously, Cramer et al. [24] introduced the notion of “Algebraic Manipulation Detection” (AMD) codes. Such codes guarantee error-detection with respect to the family of additive tampering functions. Since 2010, a line of works has established increasingly stronger results concerning the feasibility of non-malleable codes against different families of tampering functions. However, for many interesting families the challenge of finding “good” non-malleable codes remains open. In particular, we would like to have *explicit* constructions of non-malleable codes with *high rate* and efficient encoding/decoding algorithm (*i.e. low computational complexity*).

This paper follows this research direction studying the following natural question: can we achieve the optimal properties of linear-time complexity and rate approaching 1 simultaneously (via an explicit construction)? This is not known, even for the restricted case of bit-wise independent tampering, and even if we only ask for linear-time complexity².

Many of the known constructions of non-malleable codes (see for example [7,8,15,17,30]) use *linear secret-sharing schemes* (LSSS) as one of the main

¹ The rate of the coding scheme (Enc, Dec) is the quotient of the length of the message \mathbf{m} over the length of its encoding $\text{Enc}(\mathbf{m})$. The computational complexity of the scheme is maximum of the computational complexities of the two algorithm Enc and Dec in function of the length of \mathbf{m} .

² Determining which cryptographic primitives can be instantiated in linear-time is an interesting and challenging program started by Ishai et al. in [37].

building blocks. This holds also for the constructions presented in this paper. Roughly speaking, a secret-sharing scheme is a randomised algorithm that encodes a message \mathbf{m} as a longer vector \mathbf{s} such that \mathbf{m} can be computed from large enough sets of entries in \mathbf{s} , while smaller sets give no information about \mathbf{m} . LSSS with extra properties (uniformity and distance) are used already by Dziembowski et al. in [30] where they introduce and motivate the formal notion of non-malleable codes and also construct the first family of non-malleable codes in the bit-wise independent tampering model. The computational complexity of the code is quadratic in the size of the input length. Secondly, via the probabilistic method they show that for any family \mathcal{F} of tampering functions such that $|\mathcal{F}| \leq 2^{2^{\alpha n}}$ for some constant $\alpha < 1$ (n is the length of the encoding) there exist constant-rate non-malleable codes with respect to \mathcal{F} . In this case, the description of the code is of exponential size, thus the encoding and decoding algorithms are inefficient. More recently, Cheraghchi and Guruswami [14, 16] prove that for this kind of families the optimal rate is $1 - \alpha$; they construct non-malleable codes approaching this rate. Again, the construction is non-explicit and gives rise to inefficient codes. For families of single exponential size, *i.e.* $|\mathcal{F}| \leq 2^{p(n)}$ for some polynomial p , efficient (*i.e.* polynomial time) non-malleable codes were constructed in [33]. This construction is also randomized, *i.e.* the construction succeeds with overwhelming probability in providing non-malleable codes achieving optimal rate $1 - o(1)$. On the other hand, in [15] an explicit (deterministic) construction of non-malleable codes with rate arbitrarily close to 1 in the bit-wise independent tampering model is given. The construction is based on the concatenation of a linear error-correcting secret-sharing scheme of rate close to 1 and a constant-size non-malleable code. This construction is instantiated using Reed-Solomon codes and has thus computational complexity at least $O(n \text{ polylog}(n))$ (super-linear).

In [38], Jafargholi and Wichs introduce *tamper-detection codes* (TD) and use them together with leakage-resilient codes [27] to construct non-malleable codes that achieve optimal rate when $|\mathcal{F}| \leq 2^{2^{\alpha n}}$ and efficient encoding and decoding when $|\mathcal{F}| \leq 2^{p(n)}$.

Our Contribution. In this paper, we study the above question and achieve positive results. In the first part of our work, we push forward the idea of using linear secret sharing, and show that when the family of tampering functions has a clear structure (as in the symbol-wise independent tampering model), then simple constructions based on LSSS can achieve good results: we get constant-rate non-malleable codes with optimal computational complexity $O(k)$, where k is the length of the input message. To obtain this, we also use known results about linear-time encodable error-correcting codes and linear-time computable universal hash functions [28, 37].

Building on the first result, we then achieve both linear-time complexity and optimal rate, that is rate $1 - o(1)$, for non-malleable codes in the bit-wise independent tampering model. It is instructive to observe that optimal-rate non-malleable codes with superlinear time complexity were constructed in [8, 15],

and that these codes are based on secret sharing schemes with (relatively) large privacy and reconstruction thresholds. The problem we face is that there are no constructions of linear secret sharing schemes with linear-time complexity for the required parameter range³. We therefore propose a novel construction which is based on slightly weaker primitives which can be instantiated for the rate $1 - o(1)$ and linear-time complexity regime.

Overview of our Constructions. As mentioned, we present two deterministic constructions for linear-time non-malleable codes: Construction 1 can be seen as a generalization of the original construction of [30] and gives rise to the first linear-time non-malleable codes with constant rate in the symbol-wise independent tampering model. More generally, we prove that given a family of TD codes with any computational complexity and rate, it is possible to explicitly construct a family of non-malleable codes with constant rate and linear-time complexity. The other ingredients of this first construction are constant-rate AMD codes and constant-rate LSSS with good privacy (but where one needs almost all shares to reconstruct). We present linear-time instantiations of both these primitives using the results of [28]. Construction 1 encodes a message \mathbf{m} with three sequential steps: first \mathbf{m} is encoded with an AMD code, then the result is shared by a LSSS with privacy and finally each share is encoded by a tamper-detection code (see Fig. 1).

$$\begin{array}{ccccccc}
 \mathbb{F}^k & \xrightarrow{\text{AMD}} & \mathbb{F}^{\Theta(k)} & \xrightarrow{\text{LSSS}} & (\mathbb{F}^\ell)^m & \xrightarrow{\text{component-wise TD}} & (\mathbb{F}^{\ell'})^m \\
 \mathbf{m} & \longmapsto & \mathbf{m}' & \longmapsto & \mathbf{s} & & \\
 & & & & \parallel & & \\
 & & & & (\mathbf{s}_1, \dots, \mathbf{s}_m) & \longmapsto & (\mathbf{c}_1, \dots, \mathbf{c}_m)
 \end{array}$$

Fig. 1. The encoding algorithm of Construction 1 ($m = \Theta(k)$ and ℓ constant).

In particular, in Construction 1 if the tamper-detection code is secure against the family of tampering functions \mathcal{F} with constant error, then the resulting code is non-malleable with respect to the family \mathcal{F}^+ of functions of the form (f_1, \dots, f_m) where each f_i is a function from \mathcal{F} , a constant function or the identity and it has error negligible in the length of the input. Hence, depending on how one instantiates the components of the construction, one can handle more general tampering models than bit-wise⁴. A key point for the efficiency is that the shares produced by the LSSS used are of constant size. This implies that applying the tamper-detection code to all the shares results only in a constant overhead for the computational complexity.

³ A Monte-Carlo construction by Cramer et al. [22] can be instantiated for a parameter range where the rate of the secret sharing scheme is bounded away from 1 by a constant, but not for rate approaching 1.

⁴ The concrete instantiation we give in Corollary 3 leads to bit-wise independent tampering.

With Construction 2, we achieve linear-time non-malleable codes with optimal rate approaching 1, still with an explicit (deterministic) construction. The most efficient constructions of optimal rate non-malleable codes in the bit-wise independent tampering model are from [8, 15]. Both these constructions require a secret sharing scheme with good privacy and non-trivial reconstruction threshold. Together with the rate close to 1 constraint, these are challenging features to achieve in linear-time. In our construction, we also use a secret-sharing scheme with rate close to 1, but we do not require any reconstruction property for this scheme. Instead, we combine the sharing scheme with two other tailored primitives, each implementable in linear-time, and a short constant-rate non-malleable code. The modular design of our construction makes the security proof much simpler and more intuitive than previous constructions: each primitive takes care of a specific property needed to prove non-malleability. The encoding is done in the following way: first the input message is shared with a sharing scheme that has rate $1 - o(1)$ and t -uniformity (that is, if \mathbf{s} is the share vector of \mathbf{m} , then each set of t components of \mathbf{s} are distributed uniformly on \mathbb{F}^t). Then we use the two tailored primitives: first, a keyed almost universal function is used to compute the first hash of \mathbf{s} , $h_{\mathbf{k}}(\mathbf{s})$. Second, we compute short deterministic hash $\text{Comp}(\mathbf{s})$, using a new primitive that we call a *compressor*. This compressed value $\text{Comp}(\mathbf{s})$ comes with the guaranty of having high entropy. The two hash values and the key for the almost universal hash function can be thought of as an “authentication tag” of \mathbf{m} . The final encoding is given by the share vector \mathbf{s} and a non-malleable encoding of this tag, this encoding does not have to be high-rate nor linear-time (see Fig. 2).

$$\begin{array}{ccccccc}
 \mathbb{F}^k & \xrightarrow{\text{sharing}} & \mathbb{F}^{k+o(k)} & \xrightarrow{\text{hashing}} & \mathbb{F}^{k+o(k)} \times \mathbb{F}^{o(k)} \times \mathbb{F}^{o(k)} & \xrightarrow{\text{short NM}} & \mathbb{F}^{k+o(k)} \times \mathbb{F}^{o(k)} \\
 \mathbf{m} & \longmapsto & \mathbf{s} & \longmapsto & (\mathbf{s}, h_{\mathbf{k}}(\mathbf{s}), \text{Comp}(\mathbf{s})) & & \\
 & & & & \parallel & & \\
 & & & & (\mathbf{s}, \mathbf{h}, \mathbf{c}) & \longmapsto & (\mathbf{s}, \text{NM}(\mathbf{k}, \mathbf{h}, \mathbf{c}))
 \end{array}$$

Fig. 2. The encoding algorithm of Construction 2.

More related work. Bit-wise independent tampering functions act on each bit of the encoding independently. In the more general, C -split state model the encoding is partitioned into C blocks (C is a constant) and each block can be tampered arbitrarily but independently of the others blocks (*e.g.* [18]). For $C = 10$, an efficient and explicit construction of constant rate non-malleable codes was given in [13]. Several results can be found in the literature when $C = 2$ (split-state model) [3–5, 15, 29, 32, 40, 41]. In [41] the non-malleability property is guaranteed only against computationally bounded adversaries, while the scheme proposed by [29] is secure in the information-theoretic setting, but it can encode only 1-bit messages. The first explicit construction of non-malleable codes with information-theoretic security and message space larger than $\{0, 1\}$ in the split-state model was proposed in [4] and have rate polynomially small (k -bit strings

are encoded into codewords of length $\approx k^7$). This result was recently improved in [2], where the codeword length is decreased to $O(k^5)$. In 2015, Aggarwal et al. [3] constructed the first explicit non-malleable codes in the split state model achieving constant-rate. Rate approaching to 1 is achieved in [1, 39] in the computational setting.

In [7, 8], Agrawal et al. construct explicit and non-malleable codes which are simultaneously resilient against bit-wise independent tampering and permutations. [7] gets optimal rate, but has super-linear computational complexity. In [9] constant-rate and explicit non-malleable codes with respect to the family of functions $f : \{0, 1\}^n \rightarrow \{0, 1\}^n$ such that any output bit depends only on n^δ input bits ($0 \leq \delta < 1$ constant). Finally, notice that many variants of non-malleable codes have been introduced in the literature: *e.g.* continuous non-malleable codes [6, 12, 32, 38], leakage-resilient non-malleable codes [5, 31, 41], block-wise non-malleable codes [10, 35] and local non-malleable codes [11, 25, 26].

Structure of the paper: In Sect. 2, we fix the notation and give the basic definitions we need further on in the paper. In Sect. 3 first we give linear-time construction for AMD codes and LSSS with privacy, then we present Construction 1 in general and finally, we instantiate it for the binary case (bit-wise independent tampering model). Section 4 is also divided in two parts: in the first one we define and instantiate the primitives that are necessary for Construction 2; the latter is described in the second part of the section together with its instantiation in the bit-wise independent tampering model.

2 Preliminaries

For an integer n , we write $[n] = \{1, 2, \dots, n\}$ and, given $A \subseteq [n]$, $|A|$ denotes the cardinality of A , while A^c indicates the complement set of A , i.e. $A^c = [n] \setminus A$. With the notation (z_1, \dots, z_n) we indicate an element of the n -times cartesian product of \mathbb{F}^ℓ , where \mathbb{F} is a finite field of cardinality q and ℓ is a positive integer. Given $\mathbf{z} = (z_1, \dots, z_n) \in (\mathbb{F}^\ell)^n$ and a subset $A \subseteq [n]$, we will use \mathbf{z}_A to denote the vector $(z_i)_{i \in A} \in (\mathbb{F}^\ell)^{|A|}$. Given two vectors $\mathbf{z} = (z_1, \dots, z_n), \mathbf{v} = (v_1, \dots, v_n) \in (\mathbb{F}^\ell)^n$, the *generalized Hamming Distance* between \mathbf{z} and \mathbf{v} is defined by $d_{\text{Ham}}^\ell(\mathbf{z}, \mathbf{v}) = |\{i \in [n] \mid z_i \neq v_i\}|$. If Alg is an algorithm (randomized or not) that takes as input a value from \mathbb{F}^n , then the computational complexity of Alg is the number of field elementary operations that Alg executes to compute the output. We indicate with id the identity function. We say that a function ε is *negligible* in n ($\varepsilon(n) = \text{negl}(n)$) if for every polynomial p there exists a constant c such that $\varepsilon(n) < \frac{1}{p(n)}$ when $n > c$. For a random variable X , the notation $v \leftarrow X$ denotes that v is sampled randomly according to X . For a set S , $v \leftarrow S$ denotes that v is sampled uniformly at random from S . Given two random variables X and Y with finite range S , the *statistical distance* between X and Y is defined as $\text{SD}(X, Y) = \frac{1}{2} \sum_{i \in S} |\Pr[X = i] - \Pr[Y = i]|$. Let $X = (X_1, \dots, X_n)$ be a random variable with range S^n and t be a positive integer less or equal to n . We say that X is *t-wise independent* if for any $A = \{i_1, \dots, i_t\}$ subset

of $[n]$ of cardinality t and for any vector $\mathbf{b} = (\mathbf{b}_1, \dots, \mathbf{b}_t) \in S^t$, it holds that $\Pr[X_A = \mathbf{b}] = \prod_{j=1}^t \Pr[X_{i_j} = \mathbf{b}_j]$. We say that X is *t-wise uniform* on S^n if for any $A \subseteq [n]$ of cardinality t , X_A has the uniform distribution on S^t . If $t = n$ we simply say that X is an uniform random variable on S^n .

2.1 Tamper-Detection and Non-Malleability

Let \mathbb{F} be a finite field and n, ℓ, k be positive integers. An ℓ -folded n -code over \mathbb{F} is a non-empty subset \mathcal{C} of $(\mathbb{F}^\ell)^n$; we will refer to n as the length of the code. Given a set $A \subseteq [n]$, with the notation \mathcal{C}_A we indicate the set $\{\mathbf{c}_A \mid \mathbf{c} \in \mathcal{C}\}$. If $\psi : \mathcal{C} \rightarrow \mathbb{F}^k$ is a regular function, the pair (\mathcal{C}, ψ) is called ℓ -folded (n, k) -coding scheme. The rate of a scheme is the ratio $k/\ell n$. If $\mathbb{F} = \{0, 1\}$, the scheme is called binary. When $\ell = 1$, we simply call it (n, k) -coding scheme. If \mathcal{C} is a vector space over \mathbb{F} , then the code is called *linear*. The dimension of a linear code is its dimension as vector space over \mathbb{F} . Moreover, if the map ψ is an \mathbb{F} -linear map, also the scheme (\mathcal{C}, ψ) is called linear.

Remark 1. Given an ℓ -folded (n, k) -coding scheme (\mathcal{C}, ψ) , any randomized algorithm $\text{Enc} : \mathbb{F}^k \rightarrow \mathcal{C}$ that on input $\mathbf{m} \in \mathbb{F}^k$ outputs $\mathbf{c} \in \psi^{-1}(\{\mathbf{m}\})$ selected uniformly at random is called *encoding algorithm*. On the other side, *decoding algorithm* is the name used for the deterministic algorithm $\text{Dec} : (\mathbb{F}^\ell)^n \rightarrow \mathbb{F}^k \cup \{\perp\}$ that maps \mathbf{c} to $\mathbf{m} = \psi(\mathbf{c}) \in \mathbb{F}^k$ if $\mathbf{c} \in \mathcal{C}$ and to \perp otherwise. For convenience⁵, in the following we will always identify a coding scheme (\mathcal{C}, ψ) with the pair (Enc, Dec) .

While keeping \mathbb{F} fixed, we will assume throughout that $n = n(k)$. The computational complexity (as a function of k) of a coding scheme is the maximum taken over the computational complexities of Enc and Dec , respectively. We say that a coding scheme is *linear-time* if both Enc and Dec have complexity $O(k)$.

Let (Enc, Dec) be an ℓ -folded (n, k) -coding scheme over \mathbb{F} . Given an encoding $\mathbf{c} \leftarrow \text{Enc}(\mathbf{m})$ for the message $\mathbf{m} \in \mathbb{F}^k$, tampering with \mathbf{c} can be represented by considering a function $f : (\mathbb{F}^\ell)^n \rightarrow (\mathbb{F}^\ell)^n$ that modifies the encoding \mathbf{c} in $\tilde{\mathbf{c}} = f(\mathbf{c})$. The output of $\text{Dec}(\tilde{\mathbf{c}})$ now depends on the original message \mathbf{m} and also on the tampering function f . To represent this, we consider the following random variable Real_f^m .

$$\text{Real}_f^m = \begin{cases} \text{sample } \mathbf{c} \leftarrow \text{Enc}(\mathbf{m}); \\ \text{compute } \tilde{\mathbf{c}} = f(\mathbf{c}); \\ \text{output } \tilde{\mathbf{m}} = \text{Dec}(\tilde{\mathbf{c}}); \end{cases}$$

A simple but strong property that we can ask for is that the coding scheme is able to detect with overwhelming probability the tampering caused by all the functions f from a specific family \mathcal{F} .

⁵ The two definitions are equivalent. Given the pair (Enc, Dec) such that for any \mathbf{m} it holds $\Pr[\text{Dec}(\text{Enc}(\mathbf{m})) = \mathbf{m}] = 1$, define \mathcal{C} as the image of Enc in $(\mathbb{F}^\ell)^n$ and ψ as the map Dec restricted to \mathcal{C} .

Definition 1 (TD Code, [38]). Given a family \mathcal{F} of functions over $(\mathbb{F}^\ell)^n$, an ℓ -folded (n, k) -tamper detection code with respect to \mathcal{F} and with error ϵ is an (n, k) -coding scheme such that $\Pr[\text{Real}_f^m \neq \perp] \leq \epsilon$, $\forall \mathbf{m} \in \mathbb{F}^k$ and $\forall f \in \mathcal{F}$.

For example, any error-correcting code from coding theory with minimal distance d is a TD code with respect to the family \mathcal{F}_{dist} of functions that modify less than d components in the input vector (i.e. $d_{\text{Ham}}^\ell(f(\mathbf{x}), \mathbf{x}) < d$). The name *algebraic manipulation detection (AMD) code*, introduced by [24], is used for TD codes with respect to the family \mathcal{F}_{amd} of additive tampering functions. That is, functions of the form $f_e(\mathbf{x}) = \mathbf{x} + \mathbf{e}$ where the vector \mathbf{e} is a non-zero constant vector independent of \mathbf{x} .

Unfortunately, tampering detection can not be achieved for many natural families. For example, consider the family \mathcal{F}_{const} of all constant functions $f_c(\mathbf{x}) = \mathbf{c}$ for $\mathbf{c} \in (\mathbb{F}^\ell)^n$; if \mathbf{c} is a valid encoding, then $\Pr[\text{Real}_{f_c}^m \neq \perp] = 1$ for all $\mathbf{m} \in \mathbb{F}^k$. In order to be able to consider larger families of tampering functions, the definition of tampering detection needs to be relaxed. Instead of asking that the tampering is detected, we can ask that the result of the tampering action is independent of the original message. This property, called non-malleability is weaker than tampering-detection, nevertheless it offers enough protection against tampering attacks: an adversary that actively modifies encoded data can not control the practical effect of his action on the encoded message.

Definition 2 (NM Code, [30]). An ℓ -folded (n, k) -coding scheme (Enc, Dec) is said to be non-malleable with respect to a family \mathcal{F} with error ϵ if the following holds for any $f \in \mathcal{F}$. There exists a random variable D_f on $\mathbb{F}^k \cup \{\perp, \text{same}\}$ such that, given

$$\text{Ideal}_f^m = \begin{cases} \text{sample } \mathbf{m}^* \leftarrow D_f; \\ \text{if } \mathbf{m}^* = \text{same} & \text{then } \mathbf{m}' = \mathbf{m}; \\ & \text{otherwise } \mathbf{m}' = \mathbf{m}^*; \\ \text{output } \mathbf{m}'; \end{cases}$$

then $\text{SD}(\text{Real}_f^m, \text{Ideal}_f^m) \leq \epsilon$ for any $\mathbf{m} \in \mathbb{F}^k$.

In the rest of the paper we will mainly consider the family of *symbol-wise independent tampering* functions. That is, if the encoding has the form $\mathbf{c} = (\mathbf{c}_1, \dots, \mathbf{c}_n) \in (\mathbb{F}^\ell)^n$, then each component \mathbf{c}_i can be modified arbitrarily but independently of the values of the others components. We will use the following notation: $\mathcal{F}_{\ell, n}^q = \{f = (f_1, \dots, f_n) \mid f_i : \mathbb{F}^\ell \rightarrow \mathbb{F}^\ell\}$ and $f(\mathbf{c}) = (f_1(\mathbf{c}_1), \dots, f_n(\mathbf{c}_n))$. Let q be the cardinality of the field \mathbb{F} , note that if $q = 2$ and $\ell = 1$, $\mathcal{F}_{1, n}^2$ is the family considered in the *bit-wise independent tampering* model.

2.2 Secret-Sharing

Suppose that (Enc, Dec) is an ℓ -folded (n, k) -coding scheme over \mathbb{F} . Let t, r be positive integers.

Definition 3. (Enc, Dec) has t -privacy if the following holds for each set $A \subset [n]$ of \mathbb{F}^ℓ -coordinates with $|A| = t$. For each $\mathbf{m}, \mathbf{m}' \in \mathbb{F}^k$, the distributions of $(\text{Enc}(\mathbf{m}))_A$ and $(\text{Enc}(\mathbf{m}'))_A$ on $(\mathbb{F}^\ell)^t$ are identical. The scheme has t -uniformity if these distributions are the uniform ones on $(\mathbb{F}^\ell)^t$. (Enc, Dec) has r -reconstruction if the following holds for each set $A \subset [n]$ of \mathbb{F}^ℓ -coordinates with $|A| = r$. If $\mathbf{c}, \mathbf{c}' \in \mathcal{C}$ satisfy $\mathbf{c}_A = \mathbf{c}'_A$, then $\text{Dec}(\mathbf{c}) = \text{Dec}(\mathbf{c}')$.

Note that any scheme has n -reconstruction. Moreover, if the coding scheme has r -reconstruction and t -privacy, then $t < r$.

Remark 2. Given an ℓ -folded linear (n, k) -coding scheme, it is easy to prove that t -privacy and t -uniformity are equivalent to the following conditions, respectively.

- (t -privacy) for each set $A \subseteq [n]$ of \mathbb{F}^ℓ -coordinates with $|A| = t$, the map that maps \mathbf{c} in \mathcal{C} to the pair $(\text{Dec}(\mathbf{c}), \mathbf{c}_A)$ is surjective;
- (t -uniformity) the same condition as before holds and moreover $\mathcal{C}_A = (\mathbb{F}^\ell)^t$.

Definition 4 (LSSS). An ℓ -folded (n, t, r, k) -secret-sharing scheme over \mathbb{F} (with uniformity) is an ℓ -folded (n, k) -coding scheme over \mathbb{F} with t -privacy (t -uniformity) and r -reconstruction. If the coding scheme is linear then we call it linear secret-sharing scheme (LSSS).

Notice that in the existing literature, the algorithms Enc and Dec of a secret-sharing scheme are often indicated with the notation Sh (*sharing algorithm*) and Rec (*reconstruction algorithm*), respectively. Moreover, if $\mathbf{c} \leftarrow \text{Sh}(\mathbf{m})$, then \mathbf{c} is called share vector. Later on in the paper we will use this notation.

In this work, we will use secret-sharing schemes with different parameters and properties as building blocks for constructing efficient NM codes. In particular, for Construction 1 we are interested in the following aspect: what happens if the reconstruction algorithm of a t -private LSSS is applied to a share vector where at most t components have been tampered arbitrarily but independently from the others. The answer is stated in the next lemma (proof in [23]).

Lemma 1. Let (Sh, Rec) be a t -private ℓ -folded (n, k) -LSSS. Fix a set $A \subseteq [n]$ of \mathbb{F}^ℓ -coordinates with $|A| \leq t$ and an (eventually randomized) function $g : (\mathbb{F}^\ell)^n \rightarrow (\mathbb{F}^\ell)^n$ with the following properties. For any $\mathbf{s} \in (\mathbb{F}^\ell)^n$, $(g(\mathbf{s}))_{A^c} = \mathbf{s}_{A^c}$ and $(g(\mathbf{s}))_A$ depends only on the entries of \mathbf{s}_A . Then, there exists a random variable Δ_g on $(\mathbb{F}^\ell)^n \cup \{\perp\}$ such that for any $\mathbf{m} \in \mathbb{F}^k$, $\text{Rec}(g(\text{Sh}(\mathbf{m})))$ has the same distribution of $\mathbf{m} + \Delta_g$ (with the convention that $\mathbf{m} + \perp = \perp$).

3 Constant-Rate and Linear-Time NM Codes

In this section, we describe our first main result: Construction 1 (Fig. 4) combines an AMD code, a LSSS and a TD code with constant error in order to construct a constant-rate NM code (with negligible error) whose computational complexity is controlled by the complexity of the two first schemes used (the AMD code and the LSSS).

3.1 Building Blocks for Construction 1

Before describing Construction 1, we build linear-time and constant-rate AMD codes and LSSSs.

We recall that a coding scheme (Enc, Dec) (with alphabet \mathbb{F}) is an (n, k) -AMD code⁶ with error ϵ if $\forall \mathbf{m} \in \mathbb{F}^k$ and any non-zero $\mathbf{e} \in \mathbb{F}^n$, it holds that $\Pr[\text{Dec}(\text{Enc}(\mathbf{m}) + \mathbf{e}) \neq \perp] \leq \epsilon$. This special family of TD codes are of particular interest because, despite their simple definition, they can be used as basic tools of generic constructions for coding scheme that achieve security against tampering family larger than \mathcal{F}_{amd} (see for example [30] and our Construction 1). Clearly, the parameters (*i.e.* the rate) and the efficiency of the final schemes depend on the ones of the AMD codes used. In particular, in order to prove our result about constant-rate and linear-time NM codes (Theorem 2), we need to build constant-rate and linear-time AMD codes. Our construction, presented in the following Corollary 1, is based on the family of linear uniform functions from [28].

Lemma 2 (Linear Uniform Family, Theorem 4 in [28]). *For any positive integer c there exists a positive constant b ($b \geq c$) such that for any large enough k there is family of functions $\{g_{\mathbf{k}} : \mathbb{F}^k \rightarrow \mathbb{F}^{ck}\}_{\mathbf{k}}$ with $\mathbf{k} \in \mathbb{F}^{bk}$, such that the following holds:*

1. $g_{\mathbf{k}}$ has computational complexity $O(k)$;
2. $g_{\mathbf{k}}$ is \mathbb{F} -linear and $g_{\mathbf{k}_1 + \mathbf{k}_2} = g_{\mathbf{k}_1} + g_{\mathbf{k}_2}$;
3. for any $\mathbf{y} \in \mathbb{F}^{ck}$ and $\mathbf{x} \in \mathbb{F}^k$ with $\mathbf{x} \neq \mathbf{0}$, if \mathbf{k} is chosen uniformly at random from \mathbb{F}^{bk} then $\Pr[g_{\mathbf{k}}(\mathbf{x}) = \mathbf{y}] = \frac{1}{q^{ck}}$.

Corollary 1 (Linear-Time and Constant-Rate AMD code). *For any large enough integer k , there exists a linear-time (k', k) -AMD code with error q^{-k} and $k' = \Theta(k)$.*

Proof (Sketch). Given k , let \mathcal{G} be the family from Lemma 2 with $c = 1$. For the sake of simplicity we assume that $b = 1$ and we define:

$\text{Enc}_{amd}(\mathbf{m}) = (\mathbf{m}, \mathbf{k}, \mathbf{r}, g_{\mathbf{k}}(\mathbf{m}), g_{\mathbf{k}}(\mathbf{r}), g_{\mathbf{r}}(\mathbf{k}))$, where $\mathbf{k}, \mathbf{r} \in \mathbb{F}^k$ are chosen uniformly at random and

$$\text{Dec}_{amd}(\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3, \mathbf{v}_4, \mathbf{v}_5, \mathbf{v}_6) = \begin{cases} \mathbf{v}_1 & \text{if } g_{\mathbf{v}_2}(\mathbf{v}_1) = \mathbf{v}_4, g_{\mathbf{v}_2}(\mathbf{v}_3) = \mathbf{v}_5, g_{\mathbf{v}_3}(\mathbf{v}_2) = \mathbf{v}_6 \\ \perp & \text{otherwise} \end{cases}$$

It is easy to verify that $(\text{Enc}_{amd}, \text{Dec}_{amd})$ is a $(6k, k)$ -AMD code with error $\frac{1}{q^k}$ and computational complexity $O(k)$. The details of this proof together with its generalization to the case $b > 1$ can be found in [23]. \square

⁶ For Construction 1 we need a “strong” AMD code (as in [30]), while AMD codes were introduced in [24] by a slightly different (weaker) notion ($\forall \mathbf{m}$ and $\forall \mathbf{e}$, $\Pr[\text{Dec}(\text{Enc}(\mathbf{m}) + \mathbf{e}) \notin \{\perp, \mathbf{m}\}] \leq \epsilon$).

For Construction 1, we are interested in linear-time (m, t, m, k) -LSSS with large privacy (i.e. $t > m/2$) and constant-rate. Recently [22], the first linear-time constant-rate LSSS was shown, using a construction based on a combination of suitable linear codes and universal hash functions. More concretely, while being linear over a fixed finite field and supporting an unbounded number of players (or shares) m , there are constants $\epsilon_s, \epsilon_t, \epsilon_r$ with $0 < \epsilon_s, \epsilon_t, \epsilon_r < 1$ and an integer ℓ (the share size) such that the length k of the secret satisfies $k \geq \epsilon_s \ell m$, the privacy parameter t satisfies $t \geq \epsilon_t m$ and the reconstruction parameter r satisfies $r \leq \epsilon_r m$. Moreover, both the sharing and the reconstruction algorithm have complexity linear in m . Although here we also need constant-rate linear-time sharing scheme, we do not use the result from for Construction 1 and instead we construct our constant-rate linear-time sharing scheme for two reasons. First, the construction in [22] is a Monte-Carlo construction, while in this work we are interested only in explicit (deterministic) constructions. Second, later on (Sect. 4) we will require constant-rate sharing scheme with t -uniformity (instead of only t -privacy). Our schemes from Corollary 2 have this extra property that is not satisfied by the schemes presented in [22].

We construct the required LSSS using linear codes. Let \mathcal{D} be an ℓ -folded linear m -code of dimension k over the finite field \mathbb{F} . The minimum distance of \mathcal{D} is defined as $d = \min\{d_{\text{Ham}}^\ell(\mathbf{c}, \mathbf{c}') \mid \mathbf{c}, \mathbf{c}' \in \mathcal{D}, \mathbf{c} \neq \mathbf{c}'\}$. If \mathbf{G} is a $k \times m$ matrix over \mathbb{F}^ℓ , we say that \mathbf{G} is a generator matrix for the code \mathcal{D} if $\mathcal{D} = \{\mathbf{m} \cdot \mathbf{G} \mid \mathbf{m} \in \mathbb{F}^k\}$. We say the \mathcal{D} is a *linear-time encodable code* if the map $\mathbf{m} \rightarrow \mathbf{m} \cdot \mathbf{G}$ can be computed by an algorithm that has computational complexity $O(k)$.

The following Lemma generalizes and rephrases Theorem 2 in [19] asserting that LSSS with t -uniformity can be obtained from linear codes with distance $t + 1$.

Lemma 3. *Let \mathbf{G} be the generator matrix of an ℓ -folded linear code of length m , dimension k and minimum distance d . Assume that $\mathbf{G} = (\mathbf{I}_k, \mathbf{M})$ where \mathbf{I}_k is the $k \times k$ identity matrix (systematic form of the code)⁷. Then the scheme define in Fig. 3 is an ℓ -folded $(m, d - 1, m, k)$ -LSSS with uniformity. If the code is linear-time encodable, then the LSSS obtained has linear-time complexity.*

Proof. According to Remark 2, showing that the map $\psi_A : \mathbf{c} \rightarrow (\mathbf{c} \cdot \mathbf{G}^\top, \mathbf{c}_A)$ is surjective over $\mathbb{F}^k \times (\mathbb{F}^\ell)^{d-1}$ for any $A \subseteq [m]$ of size $d - 1$ is enough to prove that $(\text{Sh}_1, \text{Rec}_1)$ (see Fig. 3) has $d - 1$ uniformity. Clearly \mathbf{G} (and then \mathbf{G}^\top) has rank k (over \mathbb{F}) and the map $\mathbf{c} \rightarrow \mathbf{c} \cdot \mathbf{G}^\top$ is surjective. Moreover since \mathbf{G} generates a code of distance d , we can remove any $d - 1$ columns of \mathbf{G} (i.e. $d - 1$ rows from \mathbf{G}^\top) and the punctured matrix still has rank k (as any two distinct codewords differ in at least d coordinates). This means that for any \mathbf{m} we can solve in \mathbf{x} the linear system $\mathbf{x} \cdot \mathbf{G}^\top = \mathbf{m}$ even when $d - 1$ components of \mathbf{x} are fixed. This trivially implies that also the map ψ_A is surjective and concludes the proof of the uniformity property. Finally, it follows directly from Tellegen's principle

⁷ With $(\mathbf{I}_k, \mathbf{M})$ we indicate that we append the columns of \mathbf{M} to the ones of the identity matrix \mathbf{I}_k .

Input: $\mathbf{m} \in \mathbb{F}^k$ Sh ₁ (\mathbf{m}): Sample $\mathbf{x}' \leftarrow (\mathbb{F}^\ell)^{m-k}$ Compute $\mathbf{x}'' = \mathbf{m} - \mathbf{x}' \cdot \mathbf{M}^\top$ Output $\mathbf{x} = (\mathbf{x}'', \mathbf{x}')$	Input: $\mathbf{c} \in (\mathbb{F}^\ell)^m$ Rec ₁ (\mathbf{c}): Compute $\mathbf{m} = \mathbf{c} \cdot \mathbf{G}^\top$ Output \mathbf{m}
--	---

Fig. 3. Linear-time and constant-rate LSSS

(see Appendix A.1) that if the underlying code is linear-time encodable, then both the algorithms Sh₁ and Rec₁ are linear-time. \square

Instantiating Lemma 3 with ad-hoc linear-time encodable codes (derived by the linear-time encodable codes of [28]) provides us with LSSS with the required properties.

Lemma 4 (Linear-Time Codes, Theorem 2 in [28]). *For any real number $\delta \in (0, 1)$ and large enough integer k , there exist a real number $\rho \in (0, 1)$, a positive integer ℓ and a linear code over \mathbb{F} such that the following hold. The code is ℓ -folded; if m is the length of the code and d is its minimum distance, then $\frac{k}{\ell} < m \leq \frac{k}{\ell\rho}$ and $d \geq \delta m$. Furthermore, the code is linear-time encodable.*

Corollary 2 (Linear-Time and Constant-Rate LSSS). *For any real number $\delta \in (0, 1)$ there exists a positive integer ℓ such that for any large enough k there exists an (m, k) -coding scheme over \mathbb{F} with the following properties. The scheme is an ℓ -folded linear-time LSSS with δm -uniformity and $m = \Theta(k)$.*

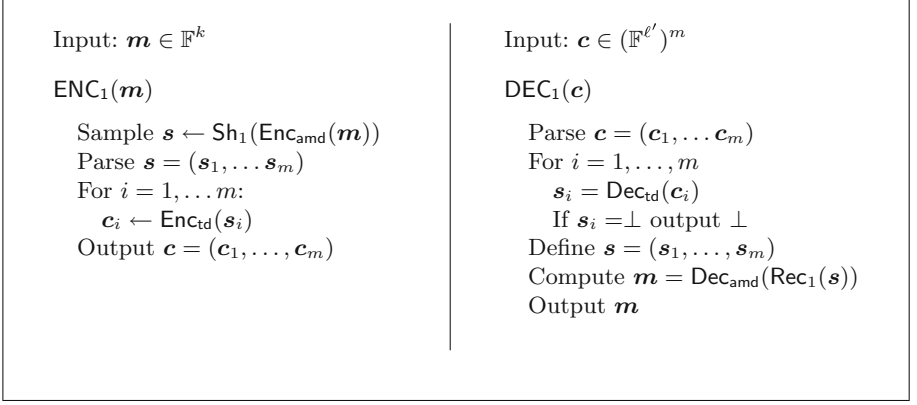
Proof. Given δ and k , let \mathbf{M} be the generator matrix of the code of Lemma 4, then the matrix $\mathbf{G} = (\mathbf{I}_k, \mathbf{M})$ defines a ℓ -folded linear code of dimension k , length $m+k$ and distance at least $\delta m+1$. The Corollary follows from Lemma 3. \square

3.2 Construction 1

Finally, we are ready to give the details of Construction 1 and its security proof. All the schemes in the following are defined over the finite field \mathbb{F} and are 1-folded if it is not explicitly stated otherwise. Consider the following building blocks:

- Let $(\text{Enc}_{\text{amd}}, \text{Dec}_{\text{amd}})$ be a (k', k) -AMD code with error ϵ ;
- Let $(\text{Sh}_1, \text{Rec}_1)$ be an ℓ -folded (m, t, m, k') -LSSS with privacy;
- Finally let $(\text{Enc}_{\text{td}}, \text{Dec}_{\text{td}})$ be an (ℓ', ℓ) -TD codes with respect to the family \mathcal{F} and with error α .

The new coding scheme $(\text{ENC}_1, \text{DEC}_1)$ is defined in Fig. 4. We indicate with \mathcal{F}^+ the set of tampering functions $f : (\mathbb{F}^{\ell'})^m \rightarrow (\mathbb{F}^{\ell'})^m$ in $\mathcal{F}_{\ell', m}^q$ such each f_i is a function from $\mathcal{F} \cup \mathcal{F}_{\text{const}} \cup \{\text{id}\}$. That is, each block \mathbf{c}_i of the encoding is modified by the adversary using a function $f_i : \mathbb{F}^{\ell'} \rightarrow \mathbb{F}^{\ell'}$, which can be any function from $\mathcal{F} \cup \mathcal{F}_{\text{const}} \cup \{\text{id}\}$ provided that it doesn't depend on the others blocks of the encoding.


Fig. 4. Construction 1

Theorem 1. *If $t > \frac{m}{2}$, then $(\text{ENC}_1, \text{DEC}_1)$ defined in Fig. 4 is an ℓ' -folded (m, k) -NM code with respect to the family \mathcal{F}^+ with error less than or equal to $\max\{\epsilon, \alpha^{2t-m}\}$. Moreover, if ρ is the rate of $(\text{Enc}_{\text{amd}}, \text{Dec}_{\text{amd}})$ and ρ' is the rate of the sharing scheme, then the rate $k/m\ell'$ of the new scheme is $\rho\rho'\frac{\ell}{\ell'}$.*

Proof. The correctness of the scheme $(\text{ENC}_1, \text{DEC}_1)$ (i.e. $\Pr[\text{DEC}_1(\text{ENC}_1(\mathbf{m})) = \mathbf{m}] = 1$ for any $\mathbf{m} \in \mathbb{F}^k$) and the statement about the rate are easy to verify and follow directly from the construction (Fig. 4). Fix $f = (f_1, f_2, \dots, f_m) \in \mathcal{F}^+$, to prove the non-malleability property, we have to define D_f as in Definition 2 and bound the error $\text{SD}(\text{Real}_f^m, \text{Ideal}_f^m)$ for any $\mathbf{m} \in \mathbb{F}^k$. Let $\mathbf{c} = (\mathbf{c}_1, \dots, \mathbf{c}_m) = \text{ENC}_1(\mathbf{m})$ and $\mathbf{s} = (\mathbf{s}_1, \dots, \mathbf{s}_m) = \text{Sh}_1(\text{Enc}_{\text{amd}}(\mathbf{m}))$. Notice that a valid encoding in the new scheme is a vector $\mathbf{c} = (\mathbf{c}_1, \dots, \mathbf{c}_m)$ of m blocks each of which is an encoding done by the constant-size tamper-detection code $(\text{Enc}_{\text{td}}, \text{Dec}_{\text{td}})$. Each block is independently tampered by the function $f_i : \mathbb{F}^{\ell'} \rightarrow \mathbb{F}^{\ell'}$ and since $(\text{Enc}_{\text{td}}, \text{Dec}_{\text{td}})$ is an TD code, for any block such that $f_i \in \mathcal{F}$ we know that the outputs of $\text{Dec}_{\text{td}}(f_i(\mathbf{c}_i))$ is \perp with probability greater or equal to $1 - \alpha$. Using this and the t -privacy property, in the following we will show that we can have enough information on the output of $\text{DEC}_1(f(\text{ENC}_1(\mathbf{m})))$ only looking at how many blocks have been tampered by functions not in \mathcal{F} . More precisely, define the following sets: $I \subseteq [m]$ is the set of indices i such that f_i is the identity function, $C \subseteq [m]$ is the set of indices i such that f_i is a constant function on $\mathbb{F}^{\ell'}$ and $J = [m] \setminus (I \cup C) = (I \cup C)^c$. Consider now the following cases:

- (1) Suppose that many blocks are tampered using constant functions (i.e. $|C| \geq m - t$). Then, the t -privacy implies that the distribution of the blocks not touched by a constant function is the same for any input message \mathbf{m} , while all the other blocks are fixed to known constants. Hence, we define D_f as
 - sample \mathbf{d} accordingly to the distribution of $\text{ENC}_1(\mathbf{0})$ and output the result of $\text{DEC}_1(f(\mathbf{d}))$.

Because of the t -privacy, $\text{DEC}_1(f(\mathbf{d}))$ has the same distribution of $\text{DEC}_1(f(\mathbf{c}))$ and thus we have that $\text{SD}(\text{Real}_f^m, \text{Ideal}_f^m) = 0$.

- (2) Otherwise we can assume that few blocks are tampered by constant functions (*i.e.* $|J| + |I| > t$) and we consider two sub-cases.
- (2.a) Suppose that few blocks are tampered (*i.e.* $|I| \geq m - t$) and look at what happens during the execution of DEC_1 on input $f(\mathbf{c})$. If there exists $i \in I^c$ such that $\text{Dec}_{\text{td}}(f_i(\mathbf{c}_i)) = \perp$, then the entire decoding outputs \perp . Otherwise, we have the situation described by Lemma 1 with⁸ $g = \text{Dec}_{\text{td}} \circ f \circ \text{Enc}_{\text{td}}$. Indeed, in the decoding phase the algorithm Rec_1 is applied to a share vector $\tilde{\mathbf{s}}$ where at most t components have been modified respect to the original share vector \mathbf{s} . It follows by Lemma 1 that $\text{Rec}_1(\tilde{\mathbf{s}})$ has the same distribution as $\text{Enc}_{\text{amd}}(\mathbf{m}) + \Delta_g$. Moreover, by definition of AMD code, if $\Delta_g = \mathbf{0}$, then $\text{DEC}_1(f(\mathbf{c}))$ outputs the original message \mathbf{m} , else it outputs \perp with probability greater than or equal to $1 - \epsilon$. Thus, in this case we define D_f by the following steps:
- sample $\mathbf{r} = (\mathbf{r}_1, \dots, \mathbf{r}_m)$ accordingly to the distribution of $\text{Sh}_1(\mathbf{0})$. If there exists $i \in I^c$ such that $\text{Dec}_{\text{td}}(f_i(\text{Enc}_{\text{td}}(\mathbf{r}_i))) = \perp$, then output \perp . Otherwise continue with the next step;
 - sample \mathbf{e} accordingly to the distribution of Δ_g . If $\mathbf{e} = \mathbf{0}$, D_f outputs *same*; otherwise it outputs \perp .

Because of the t -privacy, the probability that there exists $i \in I^c$ such that $\text{Dec}_{\text{td}}(f_i(\mathbf{c}_i)) = \perp$ is equal to the probability that there exists $i \in I^c$ such that $\text{Dec}_{\text{td}}(f_i(\text{Enc}_{\text{td}}(\mathbf{r}_i))) = \perp$. Moreover, Lemma 1 implies that $\text{SD}(\text{Real}_f^m, \text{Ideal}_f^m) = \Pr[\text{Dec}_{\text{amd}}(\text{Enc}_{\text{amd}}(\mathbf{m}) + \Delta_g) \neq \perp]$ and we know the latter to be less than or equal to ϵ .

- (2.b) Else we can use the assumption on t and m and say that more than $2t - m$ blocks are tampered by functions in \mathcal{F} . That is, $|J| > t - m + t = 2t - m > 0$. Independently for all these blocks, the tamper-detection code outputs a message different from \perp with probability less than or equal to α . Thus, $\text{DEC}_1(f(\mathbf{c})) = \perp$ with probability less than or equal to α^{2t-m} . Therefore, in this last case we define D_f to output \perp and we have that $\text{SD}(\text{Real}_f^m, \text{Ideal}_f^m) = \Pr[\text{Real}_f^m \neq \perp] \leq \Pr[\text{Dec}_{\text{td}}(f_i(\mathbf{c}_i)) \neq \perp \forall i \in J] \leq \alpha^{2t-m}$. \square

We are now ready to state the first of the results about linear-time NM codes that we present in this paper:

Theorem 2 (Linear-Time and Constant-Rate NM codes). *If for infinitely many integers b , there exists an (b', b) -TD code with respect of a family \mathcal{F} and with constant error α , then there exist a positive integer ℓ' such that the following holds. For any large enough integer k there exists an ℓ' -folded (m, k) -NM code $(\text{ENC}_1, \text{DEC}_1)$ with respect of the family \mathcal{F}^+ and $m = \Theta(k)$. Furthermore, the NM code has error negligible in k and linear-time computational complexity.*

⁸ Abuse of notation, with $g = \text{Dec}_{\text{td}} \circ f \circ \text{Enc}_{\text{td}}$ we mean the randomized function $g : (\mathbb{F}^\ell)^m \rightarrow (\mathbb{F}^\ell)^m$ such that $(g(\mathbf{v}))_i = \text{Dec}_{\text{td}}(f_i(\text{Enc}_{\text{td}}(\mathbf{v}_i)))$ for all $i \in [m]$.

Proof. Instantiate Construction 1 with the AMD code given by Corollary 1 and with the LSSS given by Corollary 2. More details in [23].

In [15] an infinite family of TD code with respect the family \mathcal{F} of bit-wise independent tampering functions that are neither the identity nor constant functions is given. Each code in the family has an error less than or equal to $2/3$.

Lemma 5 (Lemma 3.5 in [15]).⁹ *For any $\beta \in (0, 1)$ and any large enough ℓ' (i.e. $\ell' \geq \ell'(\beta) = O(\log^2(1/\beta)/\beta)$), there exists a binary (ℓ, ℓ') -TD code respect to the family $\mathcal{F} = \mathcal{F}_{1,n}^2 \setminus (\mathcal{F}_{const} \cup \{id\})$ with error $2/3$ and with $\ell \geq (1 - \beta)\ell'$.*

The previous lemma together with Theorem 2 implies the following result in bit-wise independent tampering model.

Corollary 3 (Binary Case for Construction 1). *For any large enough integer k , there exists a linear-time binary (N, k) -NM code with respect of the family $\mathcal{F}_{1,N}^2$ and with error negligible in k . Furthermore $N = \Theta(k)$.*

4 Optimal-Rate and Linear-Time NM Codes

In this section, we will construct a linear-time non-malleable code with rate approaching 1 (Construction 2).

4.1 Building Blocks for Construction 2

Before showing our second main result (Construction 2), we present the required building blocks.

In order to achieve linear-time and optimal-rate NM codes, we will employ linear-time (n, t, n, k) -secret-sharing schemes again, however we will need stronger assumptions regarding the rate and the privacy property of the used scheme. Namely, besides linear-time complexity, we require that the rate is not merely constant but that it approaching 1, i.e., length of a full share-vector divided by the length of the secret tends to 1 when the n tends to infinity. By general bounds on secret sharing, this implies that the privacy parameter t is sublinear in the number of players n and that reconstruction is essentially by the full player set only. But that is still fine for our purposes here (as long as privacy is nonconstant). Moreover, we note that we do not require linearity of the scheme either. Besides, we require that any t shares are uniformly and independently distributed over the share-space (t -uniformity). Below we show how to construct the schemes required here by combining results on t -wise independence generators and constant-rate secret sharing. A t -wise independence generator is a deterministic algorithm that expands a short random seed in a longer t -uniform vector. More precisely:

⁹ The construction presented in [15] is randomised, but since in our Construction 1 the parameter ℓ is constant (respect to k) we can exhaustively search for the proper TD code.

Definition 5 (*t*-wise Independence Generator, [34]). Let k, k' and t be positive integers. A function $\text{Gen} : \mathbb{F}^{k'} \rightarrow \mathbb{F}^k$ is a *t*-wise independence generator if the following holds. For each uniform random variable X over $\mathbb{F}^{k'}$ (called the seed), $\text{Gen}(X)$ is *t*-wise uniform over \mathbb{F}^k .

In [23] (Lemma 12) we provide an independence generator with seed-length and independence sub-linear in the output length. Moreover the proposed independence generator has computational complexity linear in the seed-length. Lemma 6 shows how to use the *t*-wise independence generator to build a linear-time secret-sharing scheme with *t'*-uniformity, $t' = \Theta(t)$ and rate $1 - o(1)$. The high-level idea (Fig. 5) is simple, to share a secret $\mathbf{m} \in \mathbb{F}^k$ we do the following. First, we mask \mathbf{m} using $\text{Gen}(\mathbf{s})$ where \mathbf{s} is a uniformly random seed for Gen . Then, we share the seed \mathbf{s} with a constant-rate sharing scheme (for example, the scheme from Corollary 2). The final share vector is defined by the concatenation of $\mathbf{m} + \text{Gen}(\mathbf{s})$ and the share vector of \mathbf{s} .

<p>Sh₂(\mathbf{m}): Sample $\mathbf{s} \leftarrow \mathbb{F}^{k'}$ Compute $\mathbf{c}_1 = \mathbf{m} + \text{Gen}(\mathbf{s})$ Compute $\mathbf{c}_2 \leftarrow \text{Sh}_1(\mathbf{s})$ Output $\mathbf{c} = (\mathbf{c}_1, \mathbf{c}_2)$</p>	<p>Rec₂(\mathbf{c}): Parse $\mathbf{c} = (\mathbf{c}_1, \mathbf{c}_2)$ Compute $\mathbf{s} = \text{Rec}_1(\mathbf{c}_2)$ If $\mathbf{s} = \perp$, then output \perp Otherwise output $\mathbf{c}_1 - \text{Gen}(\mathbf{s})$</p>
---	--

Fig. 5. Linear-time and optimal-rate LSSS

Lemma 6 (Linear-Time and Optimal-Rate LSSS). For any real number $\epsilon \in (0, 1)$ and any large enough k , there exists a linear-time (n, t, n, k) -LSSS with uniformity such that $t = \Omega(k^{1-\epsilon})$ and $n = k + o(k)$.

Proof. Given $\epsilon \in (0, 1)$ and k large enough, there exists a *t*-wise independence generator $\text{Gen} : \mathbb{F}^{k'} \rightarrow \mathbb{F}^k$ with $t = \Omega(k^{1-\epsilon})$ and $k' = \Theta(k^{1-\delta})$ ($\delta \leq \epsilon$, see Lemma 12 in [23]). Let $(\text{Sh}_1, \text{Rec}_1)$ be the (m, t', m, k') -LSSS from Corollary 2. Notice¹⁰ that $m = \Theta(k')$ and that the scheme is *t'*-uniform with $t' = \Theta(k')$. Consider the scheme in Fig. 5 and define $s = \min\{t, t'\}$. It is easy to verify that $(\text{Sh}_2, \text{Rec}_2)$ is a linear-time (n, s, n, k) -LSSS with uniformity. Moreover, $s = \Omega(k^{1-\epsilon})$ and $n = k + m = n + O(k^{1-\delta})$. \square

We introduce a novel primitive, a *compressor*. Suppose we are given a vector whose coordinates are *t*-wise independent random variables. A compressor is a deterministic function that, when applying it to the given vector, results in a shorter vector with nontrivial entropy¹¹, assuming that the original vector contains at least *t* coordinates with nontrivial entropy¹².

¹⁰ The family of LSSSs from Corollary 2 is ℓ folded, where ℓ is a constant respect to k' . Thus, the scheme $(\text{Sh}_1, \text{Rec}_1)$ can be “unfolded” and still it remains a constant-rate scheme.

¹¹ The *min-entropy* of a random variable X is $H_\infty(X) = -\log_2(\max_b \Pr[X = b])$.

¹² Since we require compressors to be deterministic, generic methods for privacy amplification do not apply here.

Definition 6 (compressor). Let t, n, n' be positive integers and r a positive real number. A function $\text{Comp} : \mathbb{F}^n \rightarrow \mathbb{F}^{n'}$ is a (t, r) -compressor if the following holds. Suppose that $X = (X_1, \dots, X_n)$ is a t -wise independent random variable on \mathbb{F}^n such that there is a set $A \subseteq [n]$ of cardinality t and a real number $c > 0$ for which $H_\infty(X_i) \geq c$ for all $i \in A$. Then $H_\infty(\text{Comp}(X)) \geq rc$.

This primitive is used in the security proof of Construction 2 to handle the case of a component-wise tampering function that has many non-constant components. More precisely, we will use the following fact:

Lemma 7. Let $f = (f_1, \dots, f_n) \in \mathcal{F}_{1,n}^q$ be a function such that least t of the functions $f_i : \mathbb{F} \rightarrow \mathbb{F}$ are non-constant. If $\text{Comp} : \mathbb{F}^n \rightarrow \mathbb{F}^{n'}$ is a (t, r) -compressor and X is a t -wise uniform random variable on \mathbb{F}^n , then for any vector $\mathbf{b} \in \mathbb{F}^{n'}$, $\Pr[\text{Comp}(f(X)) = \mathbf{b}] \leq \left(\frac{q-1}{q}\right)^r$.

Proof. By the conditions on f , there is a set $A \subseteq [n]$ of cardinality t such that, for each $i \in A$ it holds that $H_\infty(f_i(X_i)) \geq \log_2(q/(q-1))$. Since X is t -wise independent, it follows by definition of compressor that $H_\infty(\text{Comp}(f(X))) \geq r \log_2(q/(q-1))$. \square

We now show a simple construction of Comp suitable for our purposes later on.

Lemma 8 (Linear-Time Compressor). For any real number $\epsilon \in (0, 1)$ and for any large enough positive integer n there exists an (r^2, r) -compressor $\text{Comp} : \mathbb{F}^n \rightarrow \mathbb{F}^{n'}$ with $r^2 = \Omega(n^{1-\epsilon})$ and $n' = o(n)$. Moreover Comp has computational complexity $O(n)$.

Proof. Given ϵ , for any $n \geq 1$ define $r = \lceil n^{(1-\epsilon)/2} \rceil$ and $n' = \lfloor n/r \rfloor$. Notice that $n'r \leq n$ and, if n large enough, $r^2 \leq n$. Consider the function $\text{Comp} : \mathbb{F}^n \rightarrow \mathbb{F}^{n'}$, $(\mathbf{x}_1, \dots, \mathbf{x}_n) \mapsto (\mathbf{y}_1, \dots, \mathbf{y}_{n'})$ defined by $\mathbf{y}_i = \sum_{j=1}^r \mathbf{x}_{(i-1)r+j}$ for $i = 1, \dots, n'$. Thus, a vector in the domain is viewed as comprising n' consecutive blocks of r coordinates and, for $i = 1, \dots, n'$, the sum taken over the coordinates in the i -th block gives the i -th coordinate in the image of the vector under Comp . We now verify that Comp is a (r^2, r) -compressor. Suppose $X = (X_1, \dots, X_n)$ be a r^2 -wise independent random variable on \mathbb{F}^n and suppose $A \subseteq [n]$ with $|A| = r^2$ satisfies $H_\infty(X_i) \geq c > 0$ for each $i \in A$. Define $(Y_1, \dots, Y_{n'}) = \text{Comp}(X)$. By the pigeonhole principle, there exists a $B \subseteq [n']$ with $|B| = r$ such that each Y_i with $i \in B$ is sum of at least one X_i with $i \in A$. This, together with r^2 -independence of X , implies that the corresponding random variable $Y_B = (Y_i)_{i \in B}$ has the properties that $H_\infty(Y_i) \geq c$ for each $i \in B$ and that the Y_i 's are independent. In conclusion, $H_\infty(\text{Comp}(X)) \geq H_\infty(Y_B) \geq rc$. By inspection, the computational complexity of Comp is $O(n)$. \square

Our Construction 2 that we present later on in Sect. 4.2 depends in particular on *universal hash functions*.

Definition 7 (Almost Universal Family). *Let $\mu \in (0, 1)$ be a real number and let n, m be positive integers. Suppose \mathcal{H} is a family of functions $h_{\mathbf{k}} : \mathbb{F}^n \rightarrow \mathbb{F}^m$, one for each $\mathbf{k} \in \mathbb{F}^a$. Then \mathcal{H} is μ -almost universal if the following holds. For any pair of distinct $\mathbf{x}, \mathbf{x}' \in \mathbb{F}^n$, if \mathbf{k} is chosen uniformly at random from \mathbb{F}^a then $\Pr[h_{\mathbf{k}}(\mathbf{x}) = h_{\mathbf{k}}(\mathbf{x}')] \leq \mu$.*

For our purposes, we require that these functions are linear-time computable and have vanishingly small key- and output-lengths. Hence, the linear uniform family of [28] (see Lemma 2) does not apply directly due to its linear key-length. Note that, besides linear-time, the uniform output property of this particular family enables arbitrary output-length. In [23] we show an easy adaptation of the family from [28] suitable for our purposes. It is a μ -almost universal family. But since μ is very small, it is good enough for our purposes.

Lemma 9. *For any real number $\beta \in (0, 1)$ and any positive integer n , there exists a μ -universal family $\mathcal{H} = \{h_{\mathbf{k}} : \mathbb{F}^n \rightarrow \mathbb{F}^m\}_{\mathbf{k} \in \mathbb{F}^a}$ with $a = o(n)$, $m = o(n)$ and $\mu = \Theta(q^{-n^{(1-\beta)}})$. Moreover, each function $h_{\mathbf{k}}$ has complexity $O(n)$.*

4.2 Construction 2

Finally, we are ready to give the details of Construction 2 and its security proof. Consider the following ingredients (all the scheme are over the finite field \mathbb{F}):

- Let $(\text{Sh}_2, \text{Rec}_2)$ an (n, t, n, k) -SSS with uniformity;
 - Let $\text{Comp} : \mathbb{F}^n \rightarrow \mathbb{F}^{n'}$ be a (t, r) -compressor;
 - Let $\mathcal{H} = \{h_{\mathbf{k}} : \mathbb{F}^n \rightarrow \mathbb{F}^m\}$ be a μ -almost universal family with key-space \mathbb{F}^a ;
 - Let (Enc, Dec) be a (b', b) -NM code with respect to a family \mathcal{F} with error ϵ .
- We require that $b = a + m + n'$.

Let $N = n + b'$, the new (N, k) -coding scheme $(\text{ENC}_2, \text{DEC}_2)$ is defined in Fig. 6.

Theorem 3. *The coding scheme $(\text{ENC}_2, \text{DEC}_2)$ is an (N, k) -non-malleable code with respect to the family $\mathcal{F}_{1,n}^q \times \mathcal{F}$ with error less than or equal to*

$$\max \left\{ \left(\frac{q-1}{q} \right)^t + \mu, \left(\frac{q-1}{q} \right)^r \right\} + \epsilon$$

Proof. It is trivial to verify that the scheme $(\text{DEC}_2, \text{ENC}_2)$ is correct, that is $\Pr[\text{DEC}_2(\text{ENC}_2(\mathbf{m})) = \mathbf{m}] = 1$ for all $\mathbf{m} \in \mathbb{F}^k$. In order to prove non-malleability, for each tampering function F we have to show a simulator which only depends on F and whose output distribution is statistically close to the one of $\text{DEC}_2(F(\text{ENC}_2(\mathbf{m})))$ for any given $\mathbf{m} \in \mathbb{F}^k$. More precisely, according to Definition 2 for any $F = (f, g) \in \mathcal{F}_{1,n}^q \times \mathcal{F}$, we have to define a random variable D_F and bound the error $\epsilon' = \text{SD}(\text{Real}_F^{\mathbf{m}}, \text{Ideal}_F^{\mathbf{m}})$ for any $\mathbf{m} \in \mathbb{F}^k$. Given F and $\mathbf{m} \in \mathbb{F}^k$, we write $\text{ENC}_2(\mathbf{m}) = (\mathbf{c}^{(1)}, \mathbf{c}^{(2)})$. Notice that the left part of the encoding, $\mathbf{c}^{(1)}$, is tampered by the function $f \in \mathcal{F}_{1,n}^q$, while the right part, $\mathbf{c}^{(2)}$, by the function g from \mathcal{F} . Since (Enc, Dec) is a NM-code, there exists the

<p>Input: $\mathbf{m} \in \mathbb{F}^k$</p> <p>ENC₂($\mathbf{m}$):</p> <p> Compute $\mathbf{c}^{(1)} \leftarrow \text{Sh}_2(\mathbf{m})$</p> <p> Sample $\mathbf{k} \leftarrow \mathbb{F}^a$</p> <p> Compute $\mathbf{h} = h_{\mathbf{k}}(\mathbf{c}^{(1)})$</p> <p> Compute $\mathbf{c} = \text{Comp}(\mathbf{c}^{(1)})$</p> <p> Compute $\mathbf{c}^{(2)} = \text{Enc}(\mathbf{k}, \mathbf{h}, \mathbf{c})$</p> <p> Output $(\mathbf{c}^{(1)}, \mathbf{c}^{(2)})$</p>	<p>Input: $\mathbf{c} \in \mathbb{F}^N$</p> <p>DEC₂($\mathbf{c}$):</p> <p> Parse $\mathbf{c} = (\mathbf{c}^{(1)}, \mathbf{c}^{(2)}) \in \mathbb{F}^n \times \mathbb{F}^{b'}$</p> <p> Compute $\mathbf{z} = \text{Dec}(\mathbf{c}^{(2)})$</p> <p> If $\mathbf{z} = \perp$ output \perp</p> <p> Otherwise</p> <p> Parse $\mathbf{z} = (\mathbf{k}, \mathbf{h}, \mathbf{c})$</p> <p> If $\mathbf{h} \neq h_{\mathbf{k}}(\mathbf{c}^{(1)})$ output \perp</p> <p> If $\mathbf{c} \neq \text{Comp}(\mathbf{c}^{(1)})$ output \perp</p> <p> Output $\mathbf{m} = \text{Rec}_2(\mathbf{c}^{(1)})$</p>
---	---

Fig. 6. Construction 2

random variable D_g such that $\text{SD}(\text{Real}_g^z, \text{Ideal}_g^z) \leq \epsilon$ for all $\mathbf{z} \in \mathbb{F}^b$. That is, we can simulate the output of decoding the right part, $\text{Dec}(g(\mathbf{c}^{(2)}))$, using the random variable Ideal_g^z . Specifically, we define the random variable Hyb_F^m as detailed in Fig. 7. Notice that by construction the output of Hyb_F^m depends on $\mathbf{c}^{(1)}$ (the output of $\text{Sh}_2(\mathbf{m})$) and on the output of Ideal_g^z , and the output of Real_F^m depends on $\mathbf{c}^{(1)}$ and the output of Real_g^z in the same way. Thus, we have that $\text{SD}(\text{Real}_F^m, \text{Hyb}_F^m) \leq \text{SD}(\text{Real}_g^z, \text{Ideal}_g^z)$. Given this, defining the random variable D_F in such a way that we can bound $\epsilon'' = \text{SD}(\text{Hyb}_F^m, \text{Ideal}_F^m)$ will conclude the proof. Indeed, we have $\epsilon' \leq \epsilon + \epsilon''$. To define D_F , first sample \mathbf{z}^* randomly according to D_g . The results of the sampling can be classified in three cases: \perp , *same* or some vector $(\mathbf{k}^*, \mathbf{h}^*, \mathbf{c}^*)$. Then, we proceed in the definition of D_F in a different way for each one of the three aforementioned cases. In each case, we will bound the error ϵ'' . In the following, we will write $f = (f_1, \dots, f_n) \in \mathcal{F}_{1,n}^q$. Remember that the value of \mathbf{z}^* determines the output \mathbf{z}' of Ideal_g^z .

- (1) Assume that $\mathbf{z}^* = \perp$, then $\mathbf{z}' = \perp$. We know that $\Pr[\text{Hyb}_F^m = \perp \mid D_g = \perp] = 1$, thus we define D_F to output \perp and we get that $\epsilon'' = 0$.
- (2) If $\mathbf{z}^* = \textit{same}$, then $\mathbf{z}' = (\mathbf{k}, h_{\mathbf{k}}(\mathbf{c}^{(1)}), \text{Comp}(\mathbf{c}^{(1)}))$. Define $I \subseteq [n]$ the set of indices i such that f_i is the identity function on \mathbb{F} . Consider the following two situations.
 - First, assume that many f_i are the identity function (*i.e.* $|I| \geq n-t$). Then the difference $f(\mathbf{c}^{(1)}) - \mathbf{c}^{(1)}$ depends only on the vector $(\mathbf{c}^{(1)})_{I^c}$ whose entries are independent of \mathbf{m} (because of the t -uniformity property). In particular, both the event $f(\mathbf{c}^{(1)}) = \mathbf{c}^{(1)}$ and its complement occur with the same probability for any message \mathbf{m} . If $f(\mathbf{c}^{(1)}) = \mathbf{c}^{(1)}$, then Hyb_F^m obviously outputs the original message \mathbf{m} . Otherwise, we have $f(\mathbf{c}^{(1)}) \neq \mathbf{c}^{(1)}$ and the check done via the hash function $h_{\mathbf{k}}$ fails with probability at

least $1 - \mu$. If the check fails, Hyb_F^m outputs \perp . Given this, we define D_F in the following way:

- sample $\mathbf{r}_i \leftarrow \mathbb{F}$ for all $i \in I^c$; if $f_i(\mathbf{r}_i) = \mathbf{r}_i$ for all $i \in I^c$ then outputs *same*, otherwise output \perp .

As we have already argued before, the t -uniformity property implies that the event $f_i(\mathbf{r}_i) = \mathbf{r}_i$ for all $i \in I^c$ has the same probability as the event $f(\mathbf{c}^{(1)}) = \mathbf{c}^{(1)}$ and therefore, as a consequence of the check involving the hash function, we can bound the error in the following way:

$$\begin{aligned} \epsilon'' &\leq \Pr[\text{Hyb}_F^m \neq \perp \mid D_g = \textit{same} \text{ and } f(\mathbf{c}^{(1)}) \neq \mathbf{c}^{(1)}] \\ &\leq \Pr[h_{\mathbf{k}}(f(\mathbf{c}^{(1)})) = h_{\mathbf{k}}(\mathbf{c}^{(1)}) \mid f(\mathbf{c}^{(1)}) \neq \mathbf{c}^{(1)}] \leq \mu \end{aligned}$$

- In the second case, assume that many f_i are not the identity function (*i.e.* $|I| < n - t$). Then, there exists a set $A \subseteq I^c$ of size t , and it follows again from the uniformity property that the events $f_i(\mathbf{c}_i^{(1)}) \neq \mathbf{c}_i^{(1)}$ with $i \in A$ are independent and each of them occurs with probability at least $1/q$. Therefore, very likely and independently of \mathbf{m} , $f(\mathbf{c}^{(1)}) \neq \mathbf{c}^{(1)}$ and Hyb_F^m outputs \perp because of the check done using the hash function $h_{\mathbf{k}}$. For this reason, in this case we define D_F to always output \perp and we can bound the error as follows.

$$\begin{aligned} \epsilon'' &\leq \Pr[\text{Hyb}_F^m \neq \perp \mid D_g = \textit{same}] \leq \Pr[h_{\mathbf{k}}(f(\mathbf{c}^{(1)})) = h_{\mathbf{k}}(\mathbf{c}^{(1)})] \\ &\leq \Pr[f(\mathbf{c}^{(1)}) = \mathbf{c}^{(1)}] + \mu \leq \left(\frac{q-1}{q}\right)^t + \mu \end{aligned}$$

- (3) If $\mathbf{z}^* = (\mathbf{k}^*, \mathbf{h}^*, \mathbf{c}^*)$, then we have that $\mathbf{z}' = \mathbf{z}^*$. Let $C \subseteq [n]$ be the set of all indices i such that f_i is a constant function on \mathbb{F} . Consider the following two situations.

- If many f_i are constant functions (*i.e.* $|C| \geq n - t$), then the value of vector $f(\mathbf{c}^{(1)})$ is independent of \mathbf{m} . Indeed, the t -uniformity makes the value of $(f(\mathbf{c}^{(1)}))_{C^c}$ independent of \mathbf{m} , while $(f(\mathbf{c}^{(1)}))_C$ is fixed equal to a constant defined only by f . It follows that, if we define D_F in this way:
 - sample $\mathbf{r} \leftarrow \mathbb{F}^n$, if $\mathbf{h}^* \neq h_{\mathbf{k}^*}(f(\mathbf{r}))$ or $\mathbf{c}^* \neq \text{Comp}(f(\mathbf{r}))$ output \perp ; otherwise output $\text{Rec}_2(f(\mathbf{r}))$.

then we have that $\epsilon'' = 0$.

- Otherwise more than t components f_i are not constant functions (*i.e.* $|C| < n - t$) and it follows from Lemma 7 that $\text{Comp}(f(\text{Sh}_2(\mathbf{m})))$ is a random variable with min-entropy at least $r \log_2(q/(q-1))$. Moreover, $\text{Comp}(f(\text{Sh}_2(\mathbf{m})))$ is independent of the random variable D_g . Therefore, in this case the probability that the check done using the compressor is satisfied is less than or equal to $\left(\frac{q-1}{q}\right)^r$. Remember that if the check is not satisfied then, Hyb_F^m outputs abort. Thus, we can define D_F to output always \perp and we get an error bounded by:

$$\begin{aligned} \epsilon'' &\leq \Pr[\text{Hyb}_F^m \neq \perp \mid D_g = (\mathbf{k}^*, \mathbf{h}^*, \mathbf{c}^*)] \leq \Pr[\text{Comp}(f(\mathbf{c}^{(1)})) = \mathbf{c}^*] \\ &\leq \left(\frac{q-1}{q}\right)^r \quad \square \end{aligned}$$

Real_F^m : <ul style="list-style-type: none"> Compute $\mathbf{c}^{(1)} \leftarrow \text{Sh}_2(\mathbf{m})$ Sample $\mathbf{k} \leftarrow \mathbb{F}^\alpha$ Compute $\mathbf{z} = (\mathbf{k}, h_{\mathbf{k}}(\mathbf{c}^{(1)})), \text{Comp}(\mathbf{c}^{(1)})$ Compute $\mathbf{z}' \leftarrow \text{Real}_g^z$ If $\mathbf{z}' = \perp$ output \perp Otherwise <ul style="list-style-type: none"> Parse $\mathbf{z}' = (\mathbf{k}', \mathbf{h}', \mathbf{c}')$ If $\mathbf{h}' \neq h_{\mathbf{k}'}(f(\mathbf{c}^{(1)}))$ output \perp If $\mathbf{c}' \neq \text{Comp}(f(\mathbf{c}^{(1)}))$ output \perp Output $\mathbf{m} = \text{Rec}_2(f(\mathbf{c}^{(1)}))$ 	Hyb_F^m : <ul style="list-style-type: none"> Compute $\mathbf{c}^{(1)} \leftarrow \text{Sh}_2(\mathbf{m})$ Sample $\mathbf{k} \leftarrow \mathbb{F}^\alpha$ Compute $\mathbf{z} = (\mathbf{k}, h_{\mathbf{k}}(\mathbf{c}^{(1)})), \text{Comp}(\mathbf{c}^{(1)})$ Compute $\mathbf{z}' \leftarrow \text{Ideal}_g^z$ If $\mathbf{z}' = \perp$ output \perp Otherwise <ul style="list-style-type: none"> Parse $\mathbf{z}' = (\mathbf{k}', \mathbf{h}', \mathbf{c}')$ If $\mathbf{h}' \neq h_{\mathbf{k}'}(f(\mathbf{c}^{(1)}))$ output \perp If $\mathbf{c}' \neq \text{Comp}(f(\mathbf{c}^{(1)}))$ output \perp Output $\mathbf{m} = \text{Rec}_2(f(\mathbf{c}^{(1)}))$
---	---

Fig. 7. On the right, the definition of the random variable Hyb_F^m for an input message $\mathbf{m} \in \mathbb{F}^k$ and a tampering function $F = (f, g) \in \mathcal{F}_{1,n}^q \times \mathcal{F}$. On the left, for a quick reference, the random variable Real_F^m (defined in Sect. 2) for the scheme $(\text{ENC}_2, \text{DEC}_2)$.

We are now ready to state the main result about linear-time NM codes that we present in this paper:

Theorem 4 (Linear-Time and Optimal-Rate NM codes). *Suppose that there exists real number $\alpha \in (0, 2)$ such that for any positive integer b there exists a (b', b) -NM-code (Enc, Dec) with respect of a family \mathcal{F} , with error $\epsilon(b) = \text{negl}(b)$ (the error is a negligible function of the message length) and $b' = O(b^\alpha)$, then the following holds. For any positive integer k large enough, there exists an (N, k) -NM code $(\text{ENC}_2, \text{DEC}_2)$ with respect of the family $\mathcal{F}_{1,n}^q \times \mathcal{F}$ and with error negligible in k . Furthermore $N = k + o(k)$ and, if the computational complexity of (Enc, Dec) is sub-quadratic in b , then $(\text{ENC}_2, \text{DEC}_2)$ is linear-time.*

Proof. Instantiate Construction 2 with the LSSS from Lemma 6, the compressor from Lemma 8 and the universal family from Lemma 9. More details in [23].

Corollary 4 (Binary Case for Construction 2). *For any large enough k , there exists linear-time binary (N, k) -NM code with respect of the family $\mathcal{F}_{1,N}^2$ and with error negligible in k . Furthermore, $N = k + o(k)$.*

Acknowledgements. Ivan Damgård and Irene Giacomelli acknowledge support from the Danish National Research Foundation and The National Science Foundation of China (under the grant 61361136003) for the Sino-Danish Center for the Theory of Interactive Computation and from the Center for Research in Foundations of Electronic Markets (CFEM), supported by the Danish Strategic Research Council. Ivan Damgård acknowledges support from the Advanced ERC grant MPCPRO. Ronald Cramer acknowledges the support from ERC Advanced Grant ALGSTRONGCRYPTO.

A Appendix

A.1 Tellegen's Principle

We will briefly discuss a technique know as Tellegen's principle. Assume that we are given a linear algorithm \mathbb{T} computing the function $f(\mathbf{x}) = \mathbf{x} \cdot \mathbf{A}$, where \mathbf{A} is a

$m \times n$ matrix over some ring R and \mathbf{x} is a vector from R^n . Then we can transform T into an algorithm T' computing the function $f'(\mathbf{y}) = \mathbf{y} \cdot \mathbf{A}^\top$, where $\mathbf{y} \in R^m$ and \mathbf{A}^\top is the transpose of the matrix \mathbf{A} , which has the same computational complexity as T . We will discuss this transformation for arithmetic circuits. We can decompose a circuit into a sequence of elementary instructions ϕ_i , where each ϕ_i is a linear transformation on all the wires. We can thus write the matrix \mathbf{A} as $\mathbf{A} = \phi_n \cdot \phi_{n-1} \cdots \phi_2 \cdot \phi_1$. Transposing \mathbf{A} immediately yields $\mathbf{A}^\top = \phi_1^\top \cdot \phi_2^\top \cdots \phi_{n-1}^\top \cdot \phi_n^\top$. Thus, we only have to consider the effect of transposition to the elementary instructions ϕ_i .

- Instruction ϕ_i multiplies a wire \mathbf{x} with a constant $\alpha \in R$ and writes the output in the same register. In this case $\phi_i^\top = \phi_i$, as the transformation matrix ϕ_i is diagonal and thus symmetric.
- Instruction ϕ_i adds wire \mathbf{y} to wire \mathbf{x} . In this case ϕ_i^\top adds wire \mathbf{x} to wire \mathbf{y} .

These two instructions are sufficient to implement any linear transformation. For instance, to clear an (auxiliary) register, simply multiply it by 0. We summarize this in the following Lemma.

Lemma 10 (Tellegen’s Principle [42]). *Let $\mathsf{T}(\mathbf{x})$ be a linear arithmetic circuit or linear RAM algorithm computing the function $\mathbf{x} \cdot \mathbf{A}$. Then there exists a linear arithmetic circuit $\mathsf{T}'(\mathbf{y})$ that computes the function $\mathbf{y} \cdot \mathbf{A}^\top$ and has the same computational complexity as T .*

References

1. Aggarwal, D., Agrawal, S., Gupta, D., Maji, H.K., Pandey, O., Prabhakaran, M.: Optimal computational split-state non-malleable codes. In: Kushilevitz, E., Malkin, T. (eds.) TCC 2016. LNCS, vol. 9563, pp. 393–417. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-49099-0_15
2. Aggarwal, D., Briët, J.: Revisiting the sanders-bogolyubov-ruza theorem in \mathbb{F}_p^n and its application to non-malleable codes. In: IEEE International Symposium on Information Theory, ISIT 2016, Barcelona, Spain, 10–15 July 2016, pp. 1322–1326 (2016)
3. Aggarwal, D., Dodis, Y., Kazana, T., Obremski, M.: Non-malleable reductions and applications. In: Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing, STOC 2015, Portland, OR, USA, 14–17 June 2015, pp. 459–468 (2015)
4. Aggarwal, D., Dodis, Y., Lovett, S.: Non-malleable codes from additive combinatorics. In: Proceedings of the 46th Annual ACM Symposium on Theory of Computing, STOC 2014, pp. 774–783. ACM, New York (2014)
5. Aggarwal, D., Dziembowski, S., Kazana, T., Obremski, M.: Leakage-resilient non-malleable codes. In: Dodis, Y., Nielsen, J.B. (eds.) TCC 2015. LNCS, vol. 9014, pp. 398–426. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-662-46494-6_17
6. Aggarwal, D., Kazana, T., Obremski, M.: Inception makes non-malleable codes stronger. IACR Cryptology ePrint Arch. **2015**, 1013 (2015)

7. Agrawal, S., Gupta, D., Maji, H.K., Pandey, O., Prabhakaran, M.: Explicit non-malleable codes against bit-wise tampering and permutations. In: Gennaro, R., Robshaw, M. (eds.) CRYPTO 2015. LNCS, vol. 9215, pp. 538–557. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-662-47989-6_26
8. Agrawal, S., Gupta, D., Maji, H.K., Pandey, O., Prabhakaran, M.: A rate-optimizing compiler for non-malleable codes against bit-wise tampering and permutations. In: Dodis, Y., Nielsen, J.B. (eds.) TCC 2015. LNCS, vol. 9014, pp. 375–397. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-662-46494-6_16
9. Ball, M., Dachman-Soled, D., Kulkarni, M., Malkin, T.: Non-malleable codes for bounded depth, bounded fan-in circuits. In: Fischlin, M., Coron, J.-S. (eds.) EUROCRYPT 2016. LNCS, vol. 9666, pp. 881–908. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-49896-5_31
10. Chandran, N., Goyal, V., Mukherjee, P., Pandey, O., Upadhyay, J.: Block-wise non-malleable codes. In: 43rd International Colloquium on Automata, Languages, and Programming, ICALP 2016, 11–15 July 2016, Rome, Italy, pp. 31:1–31:14 (2016)
11. Chandran, N., Kanukurthi, B., Raghuraman, S.: Information-theoretic local non-malleable codes and their applications. In: Kushilevitz, E., Malkin, T. (eds.) TCC 2016. LNCS, vol. 9563, pp. 367–392. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-49099-0_14
12. Chattopadhyay, E., Goyal, V., Li, X.: Non-malleable extractors and codes, with their many tampered extensions. In: Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2016, Cambridge, MA, USA, 18–21 June 2016, pp. 285–298 (2016)
13. Chattopadhyay, E., Zuckerman, D.: Non-malleable codes against constant split-state tampering. In: 55th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2014, Philadelphia, PA, USA, 18–21 October 2014, pp. 306–315 (2014)
14. Cheraghchi, M., Guruswami, V.: Capacity of non-malleable codes. In: Proceedings of the 5th Conference on Innovations in Theoretical Computer Science, ITCS 2014, pp. 155–168. ACM, New York (2014)
15. Cheraghchi, M., Guruswami, V.: Non-malleable coding against bit-wise and split-state tampering. In: Lindell, Y. (ed.) TCC 2014. LNCS, vol. 8349, pp. 440–464. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-642-54242-8_19
16. Cheraghchi, M., Guruswami, V.: Capacity of non-malleable codes. *IEEE Trans. Inf. Theor.* **62**(3), 1097–1118 (2016)
17. Cheraghchi, M., Guruswami, V.: Non-malleable coding against bit-wise and split-state tampering. *J. Cryptology* **30**(1), 191–241 (2017)
18. Choi, S.G., Kiayias, A., Malkin, T.: BiTR: built-in tamper resilience. In: Lee, D.H., Wang, X. (eds.) ASIACRYPT 2011. LNCS, vol. 7073, pp. 740–758. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-25385-0_40
19. Chor, B., Goldreich, O., Hasted, J., Freidmann, J., Rudich, S., Smolensky, R.: The bit extraction problem or t-resilient functions. In: 26th Annual Symposium on Foundations of Computer Science, 1985, pp. 396–407. IEEE (1985)
20. Coretti, S., Dodis, Y., Tackmann, B., Venturi, D.: Non-malleable encryption: simpler, shorter, stronger. In: Kushilevitz, E., Malkin, T. (eds.) TCC 2016. LNCS, vol. 9562, pp. 306–335. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-49096-9_13
21. Coretti, S., Maurer, U., Tackmann, B., Venturi, D.: From single-bit to multi-bit public-key encryption via non-malleable codes. In: Dodis, Y., Nielsen, J.B. (eds.) TCC 2015. LNCS, vol. 9014, pp. 532–560. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-662-46494-6_22

22. Cramer, R., Damgård, I.B., Döttling, N., Fehr, S., Spini, G.: Linear secret sharing schemes from error correcting codes and universal hash functions. In: Oswald, E., Fischlin, M. (eds.) EUROCRYPT 2015. LNCS, vol. 9057, pp. 313–336. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-662-46803-6_11
23. Cramer, R., Damgård, I., Döttling, N., Giacomelli, I., Xing, C.: Linear-time non-malleable codes in the bit-wise independent tampering model. IACR Cryptology ePrint Archive 2016/397
24. Cramer, R., Dodis, Y., Fehr, S., Padró, C., Wichs, D.: Detection of algebraic manipulation with applications to robust secret sharing and fuzzy extractors. In: Smart, N. (ed.) EUROCRYPT 2008. LNCS, vol. 4965, pp. 471–488. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-78967-3_27
25. Dachman-Soled, D., Kulkarni, M., Shahverdi, A.: Tight upper and lower bounds for leakage-resilient, locally decodable and updatable non-malleable codes. In: Fehr, S. (ed.) PKC 2017. LNCS, vol. 10174, pp. 310–332. Springer, Heidelberg (2017). https://doi.org/10.1007/978-3-662-54365-8_13
26. Dachman-Soled, D., Liu, F.-H., Shi, E., Zhou, H.-S.: Locally decodable and updatable non-malleable codes and their applications. In: Dodis, Y., Nielsen, J.B. (eds.) TCC 2015. LNCS, vol. 9014, pp. 427–450. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-662-46494-6_18
27. Davì, F., Dziembowski, S., Venturi, D.: Leakage-resilient storage. In: Garay, J.A., De Prisco, R. (eds.) SCN 2010. LNCS, vol. 6280, pp. 121–137. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-15317-4_9
28. Druk, E., Ishai, Y.: Linear-time encodable codes meeting the gilbert-varshamov bound and their cryptographic applications. In: Innovations in Theoretical Computer Science, ITCS 2014, Princeton, NJ, USA, 12–14 January 2014, pp. 169–182 (2014)
29. Dziembowski, S., Kazana, T., Obremski, M.: Non-malleable codes from two-source extractors. In: Canetti, R., Garay, J.A. (eds.) CRYPTO 2013. LNCS, vol. 8043, pp. 239–257. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-40084-1_14
30. Dziembowski, S., Pietrzak, K., Wichs, D.: Non-malleable codes. In: Innovations in Computer Science - ICS 2010, Tsinghua University, Beijing, China, 5–7 January 2010, Proceedings, pp. 434–452 (2010)
31. Faonio, A., Nielsen, J.B.: Non-malleable codes with split-state refresh. In: Public-Key Cryptography - PKC 2017–20th IACR International Conference on Practice and Theory in Public-Key Cryptography, Amsterdam, The Netherlands, 28–31 March 2017, Proceedings, Part I, pp. 279–309 (2017)
32. Faust, S., Mukherjee, P., Nielsen, J.B., Venturi, D.: Continuous non-malleable codes. In: Lindell, Y. (ed.) TCC 2014. LNCS, vol. 8349, pp. 465–488. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-642-54242-8_20
33. Faust, S., Mukherjee, P., Venturi, D., Wichs, D.: Efficient non-malleable codes and key-derivation for poly-size tampering circuits. In: Nguyen, P.Q., Oswald, E. (eds.) EUROCRYPT 2014. LNCS, vol. 8441, pp. 111–128. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-642-55220-5_7
34. Goldreich, O.: Modern Cryptography, Probabilistic Proofs and Pseudorandomness, Algorithms and Combinatorics, vol. 17. Springer, Heidelberg (1998)
35. Goyal, V., Khurana, D., Sahai, A.: Breaking the three round barrier for non-malleable commitments. In: IEEE 57th Annual Symposium on Foundations of Computer Science, FOCS 2016, 9–11 October 2016, Hyatt Regency, New Brunswick, New Jersey, USA, pp. 21–30 (2016)

36. Goyal, V., Pandey, O., Richelson, S.: Textbook non-malleable commitments. In: Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2016, Cambridge, MA, USA, 18–21 June 2016, pp. 1128–1141 (2016)
37. Ishai, Y., Kushilevitz, E., Ostrovsky, R., Sahai, A.: Cryptography with constant computational overhead. In: STOC, pp. 433–442 (2008)
38. Jafarholi, Z., Wichs, D.: Tamper detection and continuous non-malleable codes. In: Dodis, Y., Nielsen, J.B. (eds.) TCC 2015. LNCS, vol. 9014, pp. 451–480. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-662-46494-6_19
39. Kiayias, A., Liu, F., Tselekounis, Y.: Practical non-malleable codes from 1-more extractable hash functions. In: Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, Vienna, Austria, 24–28 October 2016, pp. 1317–1328 (2016)
40. Li, X.: Improved non-malleable extractors, non-malleable codes and independent source extractors. In: Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2017, Montreal, QC, Canada, 19–23 June 2017, pp. 1144–1156 (2017)
41. Liu, F.-H., Lysyanskaya, A.: Tamper and leakage resilience in the split-state model. In: Safavi-Naini, R., Canetti, R. (eds.) CRYPTO 2012. LNCS, vol. 7417, pp. 517–532. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-32009-5_30
42. Tellegen, B.D.H.: A general network theorem, with applications. Philips Res. Rep. **7**, 259–269 (1952)

Disproving the Conjectures from “On the Complexity of Scrypt and Proofs of Space in the Parallel Random Oracle Model”

Daniel Malinowski and Karol Żebrowski^(✉)

University of Warsaw, Warsaw, Poland
daniel.malinowski@crypto.edu.pl, k.zebrowski@mimuw.edu.pl

Abstract. In the paper “On the Complexity of Scrypt and Proofs of Space in the Parallel Random Oracle Model” (Eurocrypt 2016) Joël Alwen et al. focused on proving a lower bound of the complexity of a general problem that underlies both proofs of space protocols [Dziembowski et al. CRYPTO 2015] as well as data-dependent memory-hard functions like `scrypt` — a key-derivation function that is used e.g. as proofs of work in cryptocurrencies like Litecoin.

In that paper the authors introduced a sequence γ_n and conjectured that this sequence is upper bounded by a constant. Alwen et al. proved (among other results) that the Cumulative Memory Complexity of the hash function `scrypt` is lower bounded by $\Omega(n^2/(\gamma_n \cdot \log^2(n)))$. If the sequence γ_n is indeed bounded by a constant then this lower bound can be simplified to $\Omega(n^2/\log^2(n))$.

In this paper we first show that $\gamma_n > c\sqrt{\log(n)}$ and then we strengthen our result and prove that $\gamma_n \geq \frac{\sqrt{n}}{\text{poly}(\log(n))}$.

Alwen et al. introduced also a weaker conjecture, that is also sufficient for their results — they introduced another sequence Γ_n and conjectured that it is upper bounded by a constant. We show that this conjecture is also false, namely: $\Gamma_n \geq c\sqrt{\log(n)}$.

1 Introduction

The purpose of *proofs of work* is to provide a puzzle that requires a worker to dedicate a significant amount of resources to solve it, while still remaining feasible. Originally, this technique was developed to fight spam emails — if the sender had to dedicate some nontrivial amount of resources to send a single message then sending millions of spam emails would be unprofitable. However, proofs of work gained a lot of attention only recently — they are used in cryptocurrencies to solve the problem of double spending of coins.

Originally, the resource used in proofs of work was a *time* spent on the computations, and consequently the focus was on *time complexity* of the worker. In the

This work was supported by the Polish National Science Centre grant 2014/13/B/ST6/03540.

view of recent hardware advances, e.g. tailored ASIC devices, *memory-hardness* appears to be a much better requirement, as memory cost is not reduced by such devices. A candidate memory-hard function `script`, introduced by Percival in [9], aims to require the evaluator to either dedicate significant amount of space for the computation or highly increase the time spent on the evaluation. A similar space-time trade-off is imposed on the worker in *proofs of space* — a concept introduced by Dziembowski et al. in [5]. In proofs of space the worker can either dedicate a specified amount of the memory to generate proofs very efficiently, or save the space and pay increased time cost every time he generates the proof.

Alwen et al. in [1] focus on proving a lower bound of the complexity of a general problem that underlies both proofs of space protocols as well as the `script` function. To prove their main results, the authors of [1] introduced two combinatorial conjectures (either of them is sufficient for their results) and assumed that they are true.

In this paper we disprove both conjectures from [1]. To give a reader intuitions and a good understanding of the definitions required for stating the conjectures we give an introduction to [1] in Sect. 1.1. We remind the parallel Random Oracle Model, the labeling and pebbling games and how to calculate the Cumulative Memory Complexity of algorithms.

1.1 Introduction to [1]

Alwen et al. in [1] investigate lower bounds on the time and memory complexity of an adversary algorithm \mathcal{A} whose goal is to compute labels of nodes in a directed acyclic graph. In this game (we describe it in more details in Sect. 1.1) the label of a node is a hash h of node's index and the labels of its parents¹. The hash function is modeled as a random oracle, so in order to compute the label, \mathcal{A} has to keep the labels of parent nodes in the memory.

Specific instances of this problem underlie *proofs of space* protocols constructed by Dziembowski et al. in [5]. Proofs of space is an alternative concept to proofs of work, in which a prover must dedicate a significant amount of his disc space as opposed to his computing power. Proofs of space are more environmentally friendly than proofs of work, because storage does not require energy. They can be used to create e.g. greener cryptocurrencies [8].

Another application of the problem considered in [1] is an examination of a memory-hard hash function² `script` introduced by Percival in [9]. The honest evaluation of the `script` function invokes underlying hash function h (modeled as a random oracle) n times, and requires storing n labels (where n is a parameter of `script`). As Percival stated, the expectation was that even for the adversary that parallelizes the computation it holds that $S(n) \cdot T(n) \geq n^{2-\epsilon}$, where $S(n)$ and $T(n)$ denote space and time invested, respectively. However, no rigorous proof of that fact was given. Another shortcoming of Percival's analysis was measuring

¹ Parent of a node v is any node w s.t. an edge (w, v) exists in the graph.

² Memory-hard hash functions require large storage during evaluation. They are used as password hashing functions and in proofs of work in cryptocurrencies.

memory complexity in terms of *maximum* memory used during computation. This does not take into account that the adversary could potentially *amortize* memory usage across *multiple* invocations of `script` function for multiple different inputs. To address this issue Alwen et al. consider a *cumulative memory complexity* proposed in [3]. We briefly recall this notion in Sect. 1.1.

Cumulative Memory Complexity in Parallel Random Oracle Model.

Alwen and Serbinenko in [3] developed a new complexity metric better suited for capturing an amortized memory hardness of a given function. The intuition behind their model is that the adversary can use specialized hardware to evaluate many instances of the function in parallel. In such a situation only the amortized cost per single evaluation is important.

The authors of [3] consider an adversary whose goal is to compute a function \mathcal{H}^h (i.e. some function \mathcal{H} that depends on the oracle h) with underlying hash function h modeled as a random oracle. The computation proceeds in steps and ends when the adversary computes \mathcal{H}^h . In each step the adversary gets the previous state σ_{i-1} (the state σ_0 is set to the given initial state σ_{init}), makes unbounded local computations and produces the next state σ_i . Additionally, once per step the adversary can send a *polynomial* (therefore *parallel* in the model name) set of queries to the random oracle and get back the hash values.

The *cumulative memory complexity* (CMC) of a single evaluation of \mathcal{H}^h is measured as $\sum_i |\sigma_i|$. CMC in parallel ROM model of \mathcal{H}^h , denoted $\text{cmc}^{\text{PROM}}(\mathcal{H}^h)$, is defined as minimal (over all the adversaries) expected CMC of the adversary computing \mathcal{H}^h .

Labeling Games. Alwen et al. in [1] proved that the hardness of `script`-like functions, as well as the security of proofs of space, rely on difficulty of the following game, called `computeLabel`.

The game is played on a single source and a single sink directed acyclic graph (DAG) $G = (V, E)$ with subset of challenge nodes $C \subseteq V$ and is parametrized with a hash function h (modeled as a random oracle). Each graph node, with index i , is associated with a label l_i defined recursively as a hash of index i and labels of parents of i , namely $l_i = h(i, l_{p_1}, \dots, l_{p_d})$, where $p_1 < \dots < p_d$ are indices of all the parents of i . The game proceeds in n rounds, where n is a parameter of the game. At each round r a challenge c_r is drawn uniformly at random from C . The player's goal is to compute the label associated with the challenge node c_r , before moving to the next round and learning the next challenge c_{r+1} . As before, we assume the pROM model, i.e. the player can make multiple parallel random oracle calls at each step of his computation. The game ends when the last challenge is answered.

We define a CMC complexity of the `computeLabel` game as the expected value of CMC of the best adversary playing the game. The second result of Alwen et al., described in Sect. 1.1, applies to the CMC complexity of the `computeLabel` game played on a simple path graph, which underlies the `script` function.

Pebbling Games and “entangled” Pebbling Games. A standard pebbling game, similarly to the `computeLabel`, is played on a single source and a single sink DAG $G = (V, E)$. At each step the player can put or remove a pebble from a node of G according to the following two rules: a new pebble can be placed on any node v for which all parents of v have pebbles on them (in particular, a pebble can always be placed on the source), and pebbles can always be removed. The game ends when a pebble is placed on the sink of G . In the parallel pebbling game the player at each step places at the same time as many pebbles as he wants (as long as he follows the rules) and then he removes any number of pebbles of his choice.

The *cumulative complexity* (CC) of the strategy for the (parallel) pebbling game is defined as $\sum_i |S_i|$ where S_i is a set of pebbled nodes at the end of i -th step. The CC of a graph G is defined as CC of the best pebbling strategy for G .

For a graph $G = (V, E)$ one could consider a pebbling analogue of the `computeLabel` game, called `pebble` in [1]. At each round a challenge c_r is sampled uniformly at random from $C \subseteq V$, and the goal of the player is to pebble the challenge node (following the same rules as in the parallel pebbling game), before advancing to the next round and learning the challenge c_{r+1} .

It is easy to see that any pebbling strategy in the `pebble` game can be adapted as a strategy in the `computeLabel` game for a *restricted* adversary who stores in memory only the labels, i.e. random oracle outputs. One could consider a slightly strengthened model in which the adversary can store specific functions of the labels (but not yet *arbitrary* ones). For example, consider an adversary playing the `computeLabel` game who stores in memory the XOR of labels $x := l_i \oplus l_j$. Later, e.g. in the next round, he could compute l_i and use it together with x to recover l_j (or the other way around). This way he could potentially improve the complexity in terms of CMC.

A pebbling abstraction of such an adversary is an adversary playing the *entangled pebbling game*, a new class of randomized pebbling game introduced in [1]. In this game, for a set $\mathcal{V} \subseteq V$ and some integer $0 \leq t < |\mathcal{V}|$ a player who has individual pebbles on all the nodes in \mathcal{V} is allowed to place an *entangled pebble* $\langle \mathcal{V} \rangle_t$ on \mathcal{V} that weights $|\mathcal{V}| - t$. The meaning of such an entangled pebble is that when the player has both $\langle \mathcal{V} \rangle_t$ and individual pebbles on any t nodes from \mathcal{V} then he can at once put individual pebbles on all the nodes in \mathcal{V} . The pebbles used to “disentangle” \mathcal{V} might be a result of disentangling another entangled pebble. Note that the entangled pebble is a generalization of a normal pebble where $\langle v \rangle_0$ corresponds to the individual pebble on vertex v . The initial pebbling in this game consists of a number of entangled pebbles.

Alwen et al. show a clever trick using polynomial interpolation which allows to translate an entangled pebble $\langle \mathcal{V} \rangle_t$ to an encoding of length $w \cdot (|\mathcal{V}| - t)$ such that given any t labels of nodes from \mathcal{V} it is possible to recover the remaining ones (here w is the length of a single label). An adversary allowed to use only such encodings in the `computeLabel` game is called an *entangled adversary*.

It is not obvious if relaxing the restriction by allowing the adversary to use entangled pebbles improves his cumulative complexity of the `pebble` game.

However, Alwen et al. show an example of a graph for which entangled pebbling is strictly more powerful (see [1] Appendix A).

Alwen et al. Conjectures and Their Implications. The authors of [1] define a sequence γ_n (we define it in Definition 7, Sect. 2) and prove that:

1. For any DAG $G = (V, E)$ with $|V| = n$, with high probability over the choice of the random hash function $h: \{0, 1\}^* \rightarrow \{0, 1\}^w$, the pROM *time* complexity to play `computeLabel` on G , for any number of challenges, starting with any initial state of size $k \cdot w$ is roughly at least the time complexity needed to play pebble on G with the same number of challenges and starting with an initial pebbling of size roughly $\gamma_n \cdot k$.
2. The pROM CMC of `computeLabel` for L_n (a simple graph underlying `script`, that consists of a single path from a source to a sink) is $\Omega\left(\frac{n^2}{\gamma_n \cdot \log^2(n)}\right)$.

Alwen et al. conjecture that the sequence γ_n is upper bounded by a constant (see Conjecture 13 in [1] or Conjecture 1, Sect. 2). They use this conjecture to boost their results proved for a restricted class of adversaries, so called entangled adversaries (see Sect. 1.1), to hold for *arbitrary* adversaries in pROM.

Under this conjecture, the first result would solve the main open problem from the work of Dziembowski et al. [5] on proofs of space. The second one would imply a near-quadratic lower bound on CMC of evaluating `script` for arbitrary pROM adversaries.

The authors of [1] also prove the same results using a different sequence Γ_n instead of γ_n . It is easy to show that for each n it holds $\Gamma_n \leq \gamma_n$. Therefore the results would hold assuming only a weaker conjecture — that the sequence Γ_n is upper bounded by a constant (see Conjecture 16 in [1] or Conjecture 2, Sect. 2). However, the authors of [1] concentrate on the stronger conjecture, because the sequence γ_n is more convenient to work with.

1.2 Our Results

As stated before, in this paper we disprove the Conjectures 13 and 16 from [1] (Conjectures 1 and 2 Sect. 2). To do it, we first show how to construct a *transcript* (see Sect. 2) from a graph and we prove that the properties of such graph-derived transcripts are connected to the clique number and the (fractional) chromatic number of that graph.

We disprove Conjecture 1 first using the Mycielski construction [7] (from that we get $\gamma_n \geq \sqrt{\log(n)}/2$) and then we strengthen our result using random graphs (we get $\gamma_n \geq \sqrt{n}/\text{poly}(\log(n))$). We disprove Conjecture 2 using Kneser graphs [6] (we get $\Gamma_n \geq c\sqrt{\log(n)}$).

1.3 Related Work

In recent work [2] Alwen et al. proved that CMC complexity in pROM model of `script` is $\Omega(n^2w)$, where w and n are the output length and number of invocations

of the underlying hash function, respectively. That bound clearly improves the result $\Omega\left(\frac{n^2}{\gamma_n \cdot \log^2(n)}\right)$ from [1] and is tight because CMC complexity of script computed as prescribed is $O(n^2 w)$.

However, the techniques used in [2] do not use the notion of entangled pebbling games and are strictly tailored for script function. Thus the approach from [1] might be of more general use.

2 Preliminaries

In this section we recall the conjectures from [1] and definitions that are necessary to state them. We also give the intuitions behind the notions introduced by Alwen et al. They are, however, not necessary to understand our results from Sect. 3.

Definition 1. For $n \in \mathbb{N}$ an n -transcript T is a set of implications of the form $\tau_j = (i_1, i_2, \dots, i_k \rightarrow i_0)$ for $k, i_0, i_1, \dots, i_k \in [n] = \{1, 2, \dots, n\}$.

The idea behind the notion of a transcript is as follows. Consider an adversary \mathcal{A} playing the `computeLabel` game on a graph G having some fixed initial state σ_{init} . Here \mathcal{A} is unrestricted, which in particular means that σ_{init} can contain any information, not only labels of the vertices. We fix the random oracle h as well. We include the implication $\tau_j = (i_1, i_2, \dots, i_k \rightarrow i_0)$ into the transcript describing $\mathcal{A}^h(\sigma_{\text{init}})$ if for some sequence of challenges c_1, \dots, c_m at some round in the game:

- the labels l_{i_1}, \dots, l_{i_k} are all the labels that appeared as inputs or outputs of the oracle so far,
- the label l_{i_0} did not appear as an input or an output of the oracle before, and
- \mathcal{A} makes a query to the random oracle using l_{i_0} as one of the inputs.

Intuitively, this means that we are able to “extract” the label l_{i_0} (without querying the oracle h for it) from σ_{init} and the labels l_{i_1}, \dots, l_{i_k} , by invoking $\mathcal{A}^h(\sigma_{\text{init}})$.

For example, consider a DAG G from Fig. 1. Suppose \mathcal{A} so far queried only for the label of the vertex 1 i.e. $l_1 = h(1)$ and for the label of the vertex 3 i.e. $l_3 = h(3, l_1)$. At this round he makes a query for a label of the vertex 5 i.e. $l_5 = h(5, l_2, l_3, l_4)$. \mathcal{A} had to extract l_2 and l_4 from σ_{init} and l_1, l_3 , so in this case we would include into the transcript the implications $(1, 3 \rightarrow 2)$ and $(1, 3 \rightarrow 4)$.

Definition 2. A set $U \subseteq [n]$ satisfies an n -transcript T , if for some $s \leq n$ there exists a sequence U_0, \dots, U_s , s.t.:

- $U_0 = U$,
- For each $j = 1, 2, \dots, s$ there exists $\tau_j = (i_1, i_2, \dots, i_k \rightarrow i_0) \in T$ s.t. $U_j = U_{j-1} \cup \{i_0\}$ and $i_1, i_2, \dots, i_k \in U_{j-1}$,
- $U_s = [n]$.

Definition 3. For an n -transcript T we define $ex(T) = n - \min_U |U|$ where the minimum is taken over all sets $U \subseteq [n]$ that satisfy T .

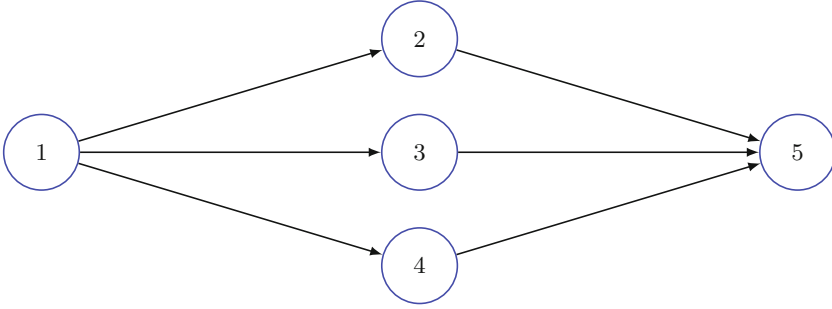


Fig. 1. An example graph used to illustrate definitions from Sect. 2.

Let the transcript T describe an adversary \mathcal{A} playing the `computeLabel` game on a graph G with an initial state σ_{init} (as explained before). Intuitively, $ex(T)$ is the maximal number of labels that can be extracted from σ_{init} and some label set $\{l_i | i \in U\}$ (where U corresponds to the set from Definition 3) by invoking $\mathcal{A}^h(\sigma_{\text{init}})$ on several different challenge sequences, in an optimal way.

Definition 4. *The entangled set $S = \langle q_1, q_2, \dots, q_t \rangle_m$ for $0 \leq m \leq t - 1$ is an object that given m different numbers from $\{q_1, q_2, \dots, q_t\}$ returns all the numbers $\{q_1, q_2, \dots, q_t\}$. The weight of S is defined as $w(S) = t - m$. The weight of the family $F = \{S_1, \dots, S_r\}$ of entangled sets is the sum of weights of the entangled sets $w(F) = w(S_1) + \dots + w(S_r)$. We will write S^* to denote the (real) set $\{q_1, q_2, \dots, q_t\}$.*

Definition 5. *We say that the n -transcript T is covered by the family of entangled sets $F = \{S_1, \dots, S_r\}$ if for every implication $\tau = (i_1, i_2, \dots, i_k \rightarrow i_0) \in T$ there exists a sequence of sets V_0, \dots, V_s s.t.*

- $V_0 = \{i_1, i_2, \dots, i_k\}$,
- For each $j = 1, 2, \dots, s$ there exists an entangled set $S_j = \langle q_1, q_2, \dots, q_t \rangle_m \in F$ s.t. $V_j = V_{j-1} \cup \{q_1, q_2, \dots, q_t\}$ and $|\{q_1, q_2, \dots, q_t\} \cap V_{j-1}| \geq m$,
- $i_0 \in V_s$.

Definition 6. *The weight $w(T)$ of the n -transcript T is the smallest weight $w(F)$ of a family F of entangled sets that covers T .*

The intuition behind the notion of the transcript weight $w(T)$ is as follows. Let T describe an unrestricted adversary \mathcal{A} playing the `computeLabel` game with an initial state σ_{init} on a graph G . Then there exists an entangled pebbling adversary (see Sect. 1.1) \mathcal{A}' playing the `pebble` game on the same graph G with an initial pebbling state σ'_{init} of weight $w(T)$ who is able to mimic the adversary \mathcal{A} in the following sense: whenever \mathcal{A} makes a query to compute some label $l_i = h(i, l_{p_1}, \dots, l_{p_t})$, \mathcal{A}' puts a (normal) pebble on the vertex i . Note that the

initial state of \mathcal{A}' may contain entangled pebbles which cannot be translated to standard pebbles.

For example, consider a DAG G from Fig. 1. Suppose σ_{init} consists of l_2 , a xor of the first half of l_1 with the first half of l_4 and a xor of the second half of l_3 with the second part of l_4 . Then \mathcal{A} can extract l_2 just from σ_{init} and l_4 from $\sigma_{\text{init}}, l_1, l_3$. So $T = \{(\rightarrow 2), (1, 3 \rightarrow 4)\}$. In this case the initial pebbling state σ'_{init} could be equal to $\{\langle 2 \rangle_0, \langle 1, 3, 4 \rangle_2\}$. Then \mathcal{A}' can use $\langle 2 \rangle_0$ when \mathcal{A} extracts l_2 from σ_{init} and use $\langle 1, 3, 4 \rangle_2$ when \mathcal{A} extracts l_4 from $\sigma_{\text{init}}, l_1, l_3$.

Definition 7. We define a sequence γ_n as:

$$\gamma_n = \max_{n\text{-transcript } T} \frac{w(T)}{ex(T)}$$

Conjecture 1 (Conjecture 13 from [1]). There exists a constant C s.t. for all natural n we have $\gamma_n < C$.

Definition 8. Let l_i for $i = 1, 2, \dots, n$ be independent random labels chosen uniformly from $\{0, 1\}^w$. We say that the state $\sigma \in \{0, 1\}^*$, that might depend on those labels, satisfies the transcript T if for every implication $(i_1, i_2, \dots, i_k \rightarrow i_0) \in T$ the label l_{i_0} is a function of a tuple $(l_{i_1}, \dots, l_{i_k}, \sigma)$. Let σ_w denote the shortest state that satisfies T . Then $shannon(T) = \inf_w \frac{|\sigma_w|}{w}$.

The value $shannon(T)$ is the length of the shortest state σ , divided by a label length w , that for any implication $(i_1, i_2, \dots, i_k \rightarrow i_0) \in T$ allows to extract l_{i_0} given labels l_{i_1}, \dots, l_{i_k} .

For example, if $T = \{(1 \rightarrow 2), (2 \rightarrow 1)\}$ then a state $\sigma = l_1 \oplus l_2$ allows to recover l_2 given l_1 , and to recover l_1 given l_2 . So in this case $shannon(T) \leq 1$.

Definition 9. We define a sequence Γ_n as:

$$\Gamma_n = \max_{n\text{-transcript } T} \frac{w(T)}{shannon(T)}$$

Conjecture 2 (Conjecture 16 from [1]). There exists a constant C s.t. for all natural n we have $\Gamma_n < C$.

It is easy to prove that for each transcript T we have $ex(T) \leq shannon(T) \leq w(T)$, so always $\gamma_n \geq \Gamma_n$. To see the first inequality let $U = \{i_1, \dots, i_k\} \subseteq [n]$ be the smallest set that satisfies T and σ_w be a state that satisfies T . By Definition 2 we can expand the set U to the whole set $[n]$ using implications from T and by Definition 8 for each implication $\tau = (j_1, j_2, \dots, j_m \rightarrow j_0) \in T$ we can extract the label l_{j_0} from σ_w and the labels $l_{j_1}, l_{j_2}, \dots, l_{j_m}$. Therefore (l_1, \dots, l_n) is a function of $(\sigma_w, l_{i_1}, \dots, l_{i_k})$ and

$$\begin{aligned} |\sigma_w| &\geq H(\sigma_w) \geq H(\sigma_w | l_{i_1}, \dots, l_{i_k}) = H(\sigma_w, l_{i_1}, \dots, l_{i_k}) - H(l_{i_1}, \dots, l_{i_k}) \geq \\ &\geq H(l_1, \dots, l_n) - H(l_{i_1}, \dots, l_{i_k}) = (n - k) \cdot w = ex(T) \cdot w. \end{aligned}$$

This ends the proof of the first inequality. The second inequality follows from the fact that the family of entangled sets F covering T can be thought of as a special case of a state σ , because on the trick used in [1] to translate an entangled pebble to the encoding of length proportional to the pebble weight (see Sect. 1.1).

Therefore the Conjecture 2 is weaker than the Conjecture 1. As stated before, both Conjecture 1 and Conjecture 2 are sufficient for the main result of [1].

3 Our Results

We disprove Conjecture 1 in Sect. 3.1 and Conjecture 2 in Sect. 3.2.

3.1 Disproving Conjecture 1

Let G denote an undirected simple³ graph with vertex set equal to $[n]$. We call such a graph an n -graph.

Definition 10. *Let G be an n -graph. By $T(G)$ we denote an n -transcript $T(G) = \{\tau_1, \dots, \tau_n\}$ where $\tau_i = (i_1, i_2, \dots, i_k \rightarrow i)$ and i_1, i_2, \dots, i_k are all the vertices in $[n] \setminus \{i\}$ that are not adjacent to the vertex i .*

Lemma 1. *Let G be an n -graph. Then $ex(T(G)) = \omega(G)$ where $\omega(G)$ is the clique number of G i.e. the size of the largest clique in G .*

Proof. First we show that $ex(T(G)) \geq \omega(G)$. Let V be the largest clique in G and $U = [n] \setminus V$. Then $|U| = n - \omega(G)$ and U satisfies $T(G)$. That is because we can add elements of V to U , in any order, using implications from $T(G)$. Formally, let $i_0 \in [n] \setminus U = V$. Then $\tau_{i_0} = (i_1, i_2, \dots, i_k \rightarrow i_0) \in T(G)$ where i_1, i_2, \dots, i_k are the vertices not adjacent to i_0 . But V is a clique, so all the vertices i_1, i_2, \dots, i_k are contained in U , so we can use τ_{i_0} and by that add i_0 to U . We can do the same for all $i \in [n] \setminus U = V$ and at the end we get the whole set $[n]$. Therefore the set U satisfies $T(G)$, so $ex(T(G)) \geq n - |U| = \omega(G)$.

Now we show that $ex(T(G)) \leq \omega(G)$. Let U be the smallest set that satisfies $T(G)$. Assume by contradiction, that $|U| < n - \omega(G)$. Then $V = [n] \setminus U$ is not a clique (as $|V| > \omega(G)$) — there exist $i_0 \neq j_0 \in V$ that are not adjacent. As U satisfies $T(G)$, we have to add both i_0 and j_0 to U . But the only implication in $T(G)$ with i_0 on the right side has j_0 on the left side, and the only implication in $T(G)$ with j_0 on the right side has i_0 on the left side. In other words i_0 depends on j_0 and j_0 depends on i_0 . Therefore we cannot add either of i_0, j_0 to U because the other element would have to be added first. Consequently U does not satisfy $T(G)$. We have a contradiction — U cannot be smaller than $n - \omega(G)$.

Lemma 2. *Let G be an n -graph. Then $\sqrt{\chi(G)} \leq w(T(G)) \leq \chi(G)$ where $\chi(G)$ is the chromatic number of G i.e. the smallest number of colors that has to be used to properly color the vertices of G .*

³ A simple graph is a graph containing no graph loops or multiple edges.

Proof. First we prove the second inequality. Let $\mathcal{C}: [n] \rightarrow [\chi(G)]$ be the proper coloring of G . Let $\mathcal{C}^{-1}(i) = \{q_1^i, q_2^i, \dots, q_{p_i}^i\}$, $S_i = \langle q_1^i, q_2^i, \dots, q_{p_i}^i \rangle_{p_i-1}$ and $F = \{S_1, S_2, \dots, S_{\chi(G)}\}$. Then $w(S_i) = 1$, $w(F) = \chi(G)$ and $T(G)$ is covered by F . The proof of this claim is easy. Let $\tau_{i_0} = (i_1, i_2, \dots, i_k \rightarrow i_0) \in T(G)$. Then the vertices i_1, i_2, \dots, i_k are all the vertices in G that are not adjacent to i_0 . Let $j = \mathcal{C}(i_0)$ be the color of the vertex i_0 . All the other vertices with the color j are not adjacent to i_0 , which means that $S_j^* \cap \{i_1, i_2, \dots, i_k\} = S_j^* \setminus \{i_0\}$. So we can use the entangled set S_j to get a vertex i_0 .

Now we prove the first inequality. Assume that $T(G)$ is covered by the family of entangled sets $F = \{S_1, S_2, \dots, S_r\}$. It is enough to show a proper coloring of vertices of G using $w(F)^2$ colors. First we show a coloring using r colors in which every vertex has less than $w(F)$ same color neighbors. Then, using a greedy algorithm, we can change it to a proper coloring using $r \cdot w(F) \leq w(F)^2$ colors

Let $i_0 \in [n]$ be any vertex in G and $\tau_{i_0} = (i_1, i_2, \dots, i_k \rightarrow i_0) \in T(G)$. Let V_0, V_1, \dots, V_s be any shortest sequence of sets as in Definition 5. Let $S_{j_i} = \langle q_1^{j_i}, q_2^{j_i}, \dots, q_{t_{j_i}}^{j_i} \rangle_{m_{j_i}}$ be the entangled set opened at the step number $i = 1, 2, \dots, s$, i.e. $V_i = V_{i-1} \cup S_{j_i}^*$ and $|V_{i-1} \cap S_{j_i}^*| \geq m_{j_i}$. We assign the color number j_s to the vertex i_0 . Obviously $j_s \in [r]$ and the coloring is unambiguous as there is exactly one implication in $T(G)$ with the element i_0 on the right side. Additionally, as the sequence V_0, V_1, \dots, V_s is the shortest, we know that $i_0 \in S_{j_s}^*$ and all the indices j_i are different.

Now we show that for any vertex $i_0 \in [n]$ there are less than $w(F)$ other vertices that are adjacent to i_0 and have the same color j_s . Let $N(i_0) \subseteq [n] \setminus \{i_0\}$ denotes the set of neighbors of the vertex i_0 . We know that all the vertices with the color j_s are contained in the set $S_{j_s}^*$. So it is enough to show, that $|S_{j_s}^* \cap N(i_0)| < w(F)$.

We know that V_0 is exactly the set $[n] \setminus N(i_0) \setminus \{i_0\}$. So $S_{j_s}^* \cap N(i_0) \subseteq V_s \setminus V_0 \setminus \{i_0\}$. On the other hand we have $|V_i \setminus V_{i-1}| \leq w(S_{j_i}) = t_{j_i} - m_{j_i}$ as we add at most t_{j_i} elements but only if there were already m_{j_i} elements present. We now have:

$$|S_{j_s}^* \cap N(i_0)| < |V_s \setminus V_0| = \sum_{i=1}^s |V_i \setminus V_{i-1}| \leq \sum_{i=1}^s w(S_{j_i}) \leq \sum_{i=1}^r w(S_i) = w(F).$$

Now we change the coloring into a proper coloring using colors from the set $[r] \times [w(F)]$. We use a greedy algorithm. For each vertex i_0 , with color j_s , we assign it the color (j_s, k) where k is any number from $[w(F)]$ s.t. the color (j_s, k) was not assigned earlier to any neighbor of i_0 . We can always find such k because there are less than $w(F)$ neighbors of i_0 which previously had the color j_s .

We have constructed a proper coloring of vertices of G using at most $w(F)^2$ colors, so $w(F)^2 \geq \chi(G)$.

To prove that γ_n is unbounded we use graphs that have big chromatic number but small clique number. We first give an example of explicit graphs using Mycielski construction [7] that satisfy these conditions. In Sect. 3.1 we use random graphs to get a stronger result.

The Mycielski construction generates graph $\mu(G)$ from a given graph G . The construction has the following properties:

- $|V(\mu(G))| = 2 \cdot |V(G)| + 1$,
- $\chi(\mu(G)) = \chi(G) + 1$,
- $\omega(\mu(G)) = \max(2, \omega(G))$.

The proof of these properties can be found in [7].

Corollary 1. *Using the Mycielski construction [7], starting from M_2 equal to a single edge, we can create a graph M_k that has $n = 3 \cdot 2^{k-2} - 1 < 2^k$ vertices, $\omega(M_k) = 2$ and $\chi(M_k) = k$. That means that $ex(T(M_k)) = 2$ and $w(T(M_k)) \geq \sqrt{\chi(M_k)} = \sqrt{k}$. So $\gamma_n \geq \frac{w(T(M_k))}{ex(T(M_k))} = \frac{\sqrt{k}}{2} > \frac{\sqrt{\log(n)}}{2}$ is unbounded therefore the Conjecture 1 is false.*

Stronger Result. In this section we show that $\gamma_n \geq \frac{\sqrt{n}}{\text{poly}(\log(n))}$.

Let $G(n, p)$ denote a random n -graph s.t. each edge is present with probability p .

Lemma 3. *We have:*

- $\mathbb{P}(\omega(G(n, 1/2)) \geq M) \leq \binom{n}{M} \cdot 2^{-\binom{M}{2}}$,
- $\lim_{n \rightarrow \infty} \mathbb{P}(\omega(G(n, 1/2)) \geq \log(n)^2) = 0$.

Proof. The first part of the lemma follows from the union bound — there are $\binom{n}{M}$ candidate sets to be a clique of size M , each of them is a clique with probability $2^{-\binom{M}{2}}$.

The second part of the lemma follows from the first part:

$$\begin{aligned} \mathbb{P}(\omega(G(n, 1/2)) \geq \log(n)^2) &\leq \binom{n}{\log(n)^2} \cdot 2^{-\binom{\log(n)^2}{2}} \leq n^{\log(n)^2} \cdot 2^{-\log(n)^4/4} = \\ &= 2^{\log(n)^3 - \log(n)^4/4}, \end{aligned}$$

So $\lim_{n \rightarrow \infty} \mathbb{P}(\omega(G(n, 1/2)) \geq \log(n)^2) = 0$.

Lemma 4. *There exists $d > 0$ s.t. $\lim_{n \rightarrow \infty} \mathbb{P}(\chi(G(n, 1/2)) \leq d \frac{n}{\log n}) = 0$.*

Proof of Lemma 4 can be found in [4].

Theorem 1. *There exists $c > 0$ s.t. $\gamma_n \geq c \frac{\sqrt{n}}{\log(n)^{5/2}}$.*

Proof. From the previous lemmas we know that there exists $d > 0$ s.t. with probability $1 - o(1)$ a random graph $G \leftarrow G(n, 1/2)$ has the following properties:

- $\omega(G) < \log(n)^2$,
- $\chi(G) > d \frac{n}{\log n}$.

$$\text{So } \gamma_n \geq \frac{w(T(G))}{ex(T(G))} \geq \frac{\sqrt{\chi(G)}}{\omega(G)} > \sqrt{d} \frac{\sqrt{n}}{\log(n)^{5/2}} = c \frac{\sqrt{n}}{\log(n)^{5/2}}.$$

3.2 Disproving Conjecture 2

Definition 11. Let G be an n -graph. The b -fold coloring of G is an assignment of sets of size b (i.e. b colors) to each vertex of G s.t. adjacent vertices receive disjoint sets. The $a : b$ coloring is a b -fold coloring using a colors. $\chi_b(G)$ is the smallest number a of colors s.t. $a : b$ coloring exists. Thpfe fractional chromatic number of G is $\chi_F(G) = \inf_b \frac{\chi_b(G)}{b}$.

Remark: For each n -graph G there exists an $a : b$ coloring s.t. $\chi_F(G) = a/b$ (see e.g. [6]).

Lemma 5. Let G be an n -graph. Then $\chi_F(G) \geq \text{shannon}(T(G))$.

Proof. Let $\mathcal{C} : [n] \rightarrow 2^{[a]}$ be a fixed b -fold coloring of G s.t. $\chi_F(G) = a/b$. Let $l_i \in \{0, 1\}^b$ for $i = 1, 2, \dots, n$ be random labels of the vertices of G and let $l_i[r]$ denote the r -th bit of l_i . We will construct a state σ_b of length a that satisfies $T(G)$.

Let $\mathcal{C}(i) = \{j_1^i, j_2^i, \dots, j_b^i\}$ where $j_1^i < j_2^i < \dots < j_b^i$. We say that the bit $l_i[r]$ has the color j_r^i . Let $\sigma_b[c]$ be the xor of all the bits $l_i[r]$ (where $i \in [n], r \in [b]$) which have the color c , for $c = 1, 2, \dots, a$.

It should be easy to see that σ_b satisfies $T(G)$. That is because for $\tau = (i_1, i_2, \dots, i_k \rightarrow i) \in T(G)$ and $r \in [b]$ we can calculate $l_i[r]$. Let c be the color of the bit $l_i[r]$. Then $l_i[r] = \sigma_b[c] \oplus l_{j_1}[r_1] \oplus \dots \oplus l_{j_s}[r_s]$ where $l_i[r], l_{j_1}[r_1], l_{j_2}[r_2], \dots, l_{j_s}[r_s]$ are all the bits of color c . Vertices i, j_1, j_2, \dots, j_s have common color, therefore i is not adjacent to any of j_1, j_2, \dots, j_s . Thus by the definition of $T(G)$ we know that all the numbers j_1, j_2, \dots, j_s are present on the left side of the implication τ . Now we can read the bits $l_{j_1}[r_1], l_{j_2}[r_2], \dots, l_{j_s}[r_s]$ from the labels $l_{j_1}, l_{j_2}, \dots, l_{j_s}$ and calculate $l_i[r]$. This can be done for any $\tau \in T(G)$ and any $r \in [b]$.

To prove that Γ_n is unbounded, we use graphs that have big chromatic numbers, but small fractional chromatic numbers. An example of graphs with these properties are Kneser graphs [6]. The vertices of the Kneser graph $K_{a:b}$ for $a \geq b$ are all b -element subsets of the set $[a]$. Two vertices are adjacent if their corresponding subsets are disjoint. The Kneser graph $K_{a:b}$ for $a \geq 2b$ has the following properties:

- $|V(K_{a:b})| = \binom{a}{b}$,
- $\chi(K_{a:b}) = a - 2b + 2$,
- $\chi_F(K_{a:b}) = a/b$.

The proofs of these properties can be found in [6].

Corollary 2. Let $K_{a:b}$ be a Kneser graph [6] for $a := 3b$ and $n := \binom{a}{b}$. We have:

$$\Gamma_n \geq \frac{w(T(K_{a:b}))}{\text{shannon}(T(K_{a:b}))} \geq \frac{\sqrt{\chi(K_{a:b})}}{\chi_F(K_{a:b})} = \frac{\sqrt{a - 2b + 2}}{a/b} = \frac{\sqrt{b + 2}}{3} = \Omega(b^{0.5})$$

so Γ_n is unbounded.

In this example $n = \binom{3b}{b} < 2^{3b}$ therefore we have proved that Γ_n is greater than $\Omega(\sqrt{\log(n)})$.

References

1. Alwen, J., Chen, B., Kamath, C., Kolmogorov, V., Pietrzak, K., Tessaro, S.: On the complexity of scrypt and proofs of space in the parallel random oracle model. In: Fischlin, M., Coron, J.-S. (eds.) EUROCRYPT 2016. LNCS, vol. 9666, pp. 358–387. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-49896-5_13
2. Alwen, J., Chen, B., Pietrzak, K., Reyzin, L., Tessaro, S.: Scrypt is maximally memory-hard. In: Coron, J.S., Nielsen, J. (eds.) Advances in Cryptology - EUROCRYPT 2017. EUROCRYPT 2017. LNCS, vol. 10212, pp. 33–62. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-56617-7_2
3. Alwen, J., Serbinenko, V.: High parallel complexity graphs and memory-hard functions. In: STOC (2015)
4. Bollobás, B.: The chromatic number of random graphs (1988)
5. Dziembowski, S., Faust, S., Kolmogorov, V., Pietrzak, K.: Proofs of space. In: Gennaro, R., Robshaw, M. (eds.) CRYPTO 2015. LNCS, vol. 9216, pp. 585–605. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-662-48000-7_29
6. Ullman, D.H., Scheinerman, E.R.: Fractional graph theory: A rational approach to the theory of graphs (2013)
7. Mycielski, J.: Sur le coloriage des graphs. *Colloquium Math.* **3**(2), 161–162 (1955)
8. Park, S., Kwon, A., Alwen, J., Fuchsbauer, G., Gaži, P., Pietrzak, K.: SpaceMint: A Cryptocurrency Based on Proofs of Space. *Cryptology ePrint Archive*, Report 2015/528 (2015). <http://eprint.iacr.org/2015/528>
9. Percival, C.: Stronger key derivation via sequential memory-hard functions (2009)

Broadcast Encryption with Guessing Secrecy

Yohei Watanabe^{1,2}(✉)

¹ Graduate School of Informatics and Engineering,
The University of Electro-Communications, Tokyo, Japan
watanabe@uec.ac.jp

² Information Technology Research Institute, AIST, Tokyo, Japan

Abstract. Perfect secrecy, which is a fundamental security notion introduced by Shannon, guarantees that no information on plaintexts is leaked from corresponding ciphertexts in the information-theoretic sense. Although it captures the strongest security, it is well-known that the secret-key size must be equal or larger than the plaintext-size to achieve perfect secrecy. Furthermore, probability distribution on secret keys must be uniform. Alimomeni and Safavi-Naini (ICITS 2012) proposed a new security notion, called *guessing secrecy*, to relax the above two restrictions, and showed that unlike perfect secrecy, even non-uniform keys can be used for providing guessing secrecy. Iwamoto and Shikata (ISIT 2015) showed secure concrete constructions of a symmetric-key encryption scheme with non-uniform keys in the guessing secrecy framework. In this work, we extend their results to the broadcast encryption setting. We first define guessing secrecy of broadcast encryption, and show relationships among several guessing-secrecy notions and perfect secrecy. We derive lower bounds on secret keys, and show the Fiat-Naor one-bit construction with non-uniform keys is also secure in the sense of guessing secrecy.

Keywords: Broadcast encryption · Guessing secrecy
Information-theoretic security · Non-uniform distribution

1 Introduction

Broadcast encryption (BE), which was introduced by Berkovitz [2] and later formalized by Fiat and Naor [11], enables a sender to control which receivers can decrypt ciphertexts. Over a quarter of a century, BE schemes, which are encryption schemes with such a simple but convenient functionality, have been investigated both in the computational security setting [5, 9, 13, 18, 21] and in the unconditional security setting [2–4, 6, 11, 12, 16, 17, 19, 20]. BE schemes are used in various situations such as copyright protection in the real world. In this paper, we focus on unconditionally secure BE schemes.

Roughly speaking, we can classify unconditionally secure BE schemes into two types depending on its functionality: $(t, \leq \omega)$ -secure BE schemes [3, 4, 16, 17, 19] and $(\leq n, \leq \omega)$ -secure BE schemes [3, 11, 25], where t is the number of

receivers who can decrypt ciphertexts (called *privileged users*), n is the number of all receivers, and ω is the maximum number of colluders. Let S be a sender and $\mathcal{R} := \{R_1, R_2, \dots, R_n\}$ be a receiver set. In $(t, \leq \omega)$ -secure BE schemes, S encrypts a plaintext for some subset (called a *privileged set*) $\mathcal{P} \subset \mathcal{R}$ such that the cardinality of \mathcal{P} is exactly t (i.e. $|\mathcal{P}| = t$). On the other hand, in $(\leq n, \leq \omega)$ -secure BE schemes S can encrypt a plaintext for *any* $\mathcal{P} \subset \mathcal{R}$. Namely, the latter realizes more flexible functionality than the former instead of significantly larger secret-key sizes [3, 11]. It is known that there is a trade-off between the secret-key sizes and the ciphertext sizes in BE schemes [4, 19, 25].

After Shannon’s seminal work [23], several relaxations of perfect secrecy, which is the strongest security notion of confidentiality, have been proposed [1, 10, 15, 22, 26]. Alimomeni and Safavi-Naini [1] introduced a security notion based on success probability of guessing plaintexts, called *guessing secrecy*, as a natural extension of perfect secrecy. They showed a non-uniform key (randomness) is sufficient to construct encryption schemes that satisfy guessing secrecy, whereas perfect secrecy requires uniform randomness to realize secure encryption schemes. Iwamoto and Shikata [14] showed concrete constructions of encryption schemes that meet guessing secrecy. It is important to investigate cryptographic protocols with non-uniform randomness from a practical perspective.

In this paper, we consider $(\leq n, \leq \omega)$ -secure BE with guessing secrecy by extending results on symmetric-key encryption [14]. We formalize guessing secrecy for BE, derive lower bounds on ciphertexts and secret keys, and propose a construction that meets guessing secrecy. More specifically, our contributions are as follows. In Sect. 3, we formalize four types of guessing secrecy for BE depending on conditions on ciphertexts and secret keys of colluders: *average guessing secrecy* (A-GS), *strong average guessing secrecy* (sA-GS), *weak optimal guessing secrecy* (wO-GS), and *optimal guessing secrecy* (O-GS). We derive lower bounds on ciphertexts, and show that the ciphertext size for any privileged set must be larger than the plaintext size. Further, we derive tight lower bounds on sizes of encryption keys and decryption keys when the plaintext size is equal to the ciphertext size. Those lower bounds are derived in Sect. 4. Finally, we analyze the Fiat-Naor construction, which is the perfectly secure, most efficient construction in the sense that secret-key sizes attain lower bounds with equalities, in the guessing secrecy setting in Sect. 5. We show the Fiat-Naor (one-bit) construction meets A-GS and sA-GS and is most efficient if and only if the probability distribution on plaintexts is more biased than that on randomness (used for generating secret keys). In other words, non-uniform randomness is sufficient to construct $(\leq n, \leq \omega)$ -secure BE schemes that satisfy A-GS or sA-GS. We also show that randomness used in a BE scheme must be uniform to meet wO-GS or O-GS (or perfect secrecy).

2 Preliminaries

Notation. For $n \in \mathbb{N}$, let $[n] := \{1, 2, \dots, n\}$. The calligraphy \mathcal{X} indicates a set, and $|\mathcal{X}|$ denotes the cardinality of \mathcal{X} . The roman capital X indicates a random

variable which takes values in \mathcal{X} (e.g., A, B , and C are random variables which take values in \mathcal{A}, \mathcal{B} , and \mathcal{C} , respectively).

2.1 Information Theoretic Tools

P_X denotes a probability distribution over a set \mathcal{X} . Throughout the paper, we assume that all of probability distributions assign non-zero values to elements of the corresponding sets. Namely, for any P_X , it holds $P_X(x) > 0$ for all $x \in \mathcal{X}$.

We briefly describe Shannon entropy. For details, see [7, 8] for the excellent instruction. Let X and Y be random variables which take values in sets \mathcal{X} and \mathcal{Y} , respectively. Shannon entropy $H(X)$ is defined by $H(X) := -\sum_{x \in \mathcal{X}} \Pr(X = x) \log \Pr(X = x)$. The joint entropy $H(X, Y)$ and conditional entropy $H(X|Y)$ of a pair of random variables (X, Y) with a joint probability distribution P_{XY} are defined by $H(X, Y) := -\sum_{x \in \mathcal{X}} \sum_{y \in \mathcal{Y}} \Pr(X = x, Y = y) \log \Pr(X = x, Y = y)$, and $H(X|Y) := \sum_{y \in \mathcal{Y}} \Pr(Y = y) H(X|Y = y)$, respectively.

2.2 Symmetric-Key Encryption

We define information-theoretically secure symmetric-key encryption (SKE). Let \mathcal{M} , \mathcal{C} , and \mathcal{K} be sets of possible plaintexts, ciphertexts, and secret keys, respectively. Let P_M and P_K be probability distributions on plaintexts and secret keys, respectively.

Definition 1 (SKE). *A symmetric-key encryption (SKE) scheme π consists of the following three tuple of algorithm (G, E, D) with three finite spaces, \mathcal{M} , \mathcal{C} , and \mathcal{K} , where D is a deterministic algorithm.*

- $k \leftarrow G(P_K)$: It outputs a common key $k \in \mathcal{K}$ according to P_K .
- $c \leftarrow E(k, m)$: It takes k and a plaintext $m \in \mathcal{M}$ generated according to P_M as input, and outputs a ciphertext $c \in \mathcal{C}$.
- m or $\perp \leftarrow D(k, c)$: It takes k and c as input, and outputs m or \perp , which is a special symbol that indicates decryption failure.

π satisfies the following correctness: For all P_K , all $k \leftarrow G(P_K)$, and all $m \in \mathcal{M}$, it holds that $m \leftarrow D(k, E(k, m))$.

Perfect secrecy (PS) is defined as follows (guessing secrecy for SKE is discussed in Sect. 3).

Definition 2 (Perfect Secrecy for SKE [23]). *An SKE scheme π is said to satisfy PS if it holds $H(M | C) = H(M)$.*

2.3 Broadcast Encryption

Suppose that there are $n + 1$ entities, a sender S and n receivers R_1, R_2, \dots, R_n . Let $\mathcal{R} := \{R_1, R_2, \dots, R_n\}$ be a set of all receivers, and \mathcal{M} be a set of possible plaintexts. For any subset $\mathcal{J} := \{R_{i_1}, R_{i_2}, \dots, R_{i_j}\} \subset \mathcal{R}$, let $\mathcal{C}_{\mathcal{J}}$ be a set of all possible ciphertexts for \mathcal{J} , and let $\mathcal{C} := \bigcup_{\mathcal{J} \subset \mathcal{R}} \mathcal{C}_{\mathcal{J}}$. Let \mathcal{EK} be a set of possible

encryption keys. Let \mathcal{DK}_i be a set of possible decryption keys for R_i , and let $\mathcal{DK} := \bigcup_{i=1}^n \mathcal{DK}_i$.

Model. First, S generates an encryption key $ek \in \mathcal{EK}$ and n decryption keys $(dk_1, dk_2, \dots, dk_n) \in \prod_{i=1}^n \mathcal{DK}_i$ according to probability distribution P_{SK} , where $SK := (EK, DK_1, DK_2, \dots, DK_n)$. S then distributes each decryption key dk_i to each receiver R_i via secure channels, respectively. When encrypting a plaintext $m \in \mathcal{M}$ by using ek , S can choose a non-empty subset \mathcal{P} (called a *privileged set*) of \mathcal{R} so that only receivers in \mathcal{P} can decrypt the resulting ciphertext $c_{\mathcal{P}} \in \mathcal{C}_{\mathcal{P}}$. A receiver R_i can decrypt the ciphertext by using his/her decryption key dk_i if $R_i \in \mathcal{P}$. Otherwise, he/she fails to decrypt it.

Definition 3 (BE). A broadcast encryption (BE) scheme Π consists of the following three-tuple of algorithms (Setup, Enc, Dec) with four finite spaces, $\mathcal{M}, \mathcal{C}, \mathcal{EK},$ and \mathcal{DK} , where Setup is a probabilistic algorithm and Enc and Dec are deterministic algorithms.

- $(ek, dk_1, \dots, dk_n) \leftarrow \text{Setup}(n, P_{SK})$: It outputs an encryption key $ek \in \mathcal{EK}$, n decryption keys $(dk_1, \dots, dk_n) \in \prod_{i=1}^n \mathcal{DK}_i$ according to P_{SK} , where $SK := (EK, DK_1, DK_2, \dots, DK_n)$.
- $c_{\mathcal{P}} \leftarrow \text{Enc}(ek, m, \mathcal{P})$: It takes ek , a plaintext $m \in \mathcal{M}$ generated according to a probability distribution P_M , and a privileged set $\mathcal{P} \subset \mathcal{R}$ as input, and outputs a ciphertext $c_{\mathcal{P}} \in \mathcal{C}_{\mathcal{P}}$.¹
- m or $\perp \leftarrow \text{Dec}(dk_i, c_{\mathcal{P}})$: It takes dk_i of a receiver R_i and the ciphertext $c_{\mathcal{P}}$ for \mathcal{P} as input, and outputs m or \perp .

The following decryption correctness is required for Π : For all $n \in \mathbb{N}$, all P_{SK} , all $(ek, dk_1, \dots, dk_n) \leftarrow \text{Setup}(n, P_{SK})$, all $m \in \mathcal{M}$, all $\mathcal{P} \subset \mathcal{R}$, and all $R_i \in \mathcal{P}$, $m \leftarrow \text{Dec}(dk_i, \text{Enc}(ek, m, \mathcal{P}))$.

Security definition. In BE, we consider PS against at most ω colluders. Namely, an adversary not only observes a ciphertext but also has at most ω decryption keys. PS for BE guarantees that no information is leaked from a ciphertext for any $\mathcal{P} \subset \mathcal{R}$ even if any $\mathcal{W} \subset \mathcal{R} \setminus \mathcal{P}$ such that $|\mathcal{W}| \leq \omega$ is corrupted.

Before formally describing PS for BE, we define several notations, which are specific to BE and will be used for definitions, a construction, and proofs:

- (i) $\mathcal{P}^{(i)} := \{\mathcal{P} \subset \mathcal{R} \mid R_i \in \mathcal{P}\}$,
- (ii) $\mathcal{W}(\omega) := \{\mathcal{W} \subset \mathcal{R} \mid |\mathcal{W}| \leq \omega\}$,
- (iii) $\mathcal{W}^{(i)}(\omega) := \{\mathcal{W} \in \mathcal{W}(\omega) \mid R_i \notin \mathcal{W}\}$,
- (iv) $\mathcal{W}^{(\mathcal{W})} := \bigcup_{R_i \in \mathcal{W}} \mathcal{W}^{(i)}(\omega)$,

¹ For simplicity, we assume that all entities share the information on \mathcal{P} of $c_{\mathcal{P}}$, e.g., by a publicly accessible authenticated bulletin board, and therefore we omit a description of \mathcal{P} from $c_{\mathcal{P}}$.

- (v) $\mathscr{W}(\mathcal{P}, \omega) := \{\mathcal{W} \subset (\mathcal{R} \setminus \mathcal{P}) \mid |\mathcal{W}| \leq \omega\}$,
- (vi) $\widehat{\mathscr{W}}(\mathcal{P}, \omega) := \{\mathcal{W} \subset (\mathcal{R} \setminus \mathcal{P}) \mid |\mathcal{W}| = \min\{\omega, n - |\mathcal{P}|\}\}$,

where (i) is a family of subsets of \mathcal{R} such that each subset contains R_i , (ii) is a family of possible sets of at most ω colluders in the BE scheme, (iii) is a family of possible sets of at most ω colluders such that each set does not contain R_i , (iv) is a union of (iii) according to $\mathcal{W} \subset \mathcal{R}$, (v) is a family of possible sets of at most ω colluders against \mathcal{P} , and (vi) is a family of possible sets that contain the maximum number of colluders against \mathcal{P} .

We are ready to define the notion of PS for BE. For any subset $\mathcal{J} := \{R_{i_1}, R_{i_2}, \dots, R_{i_j}\} \subset \mathcal{R}$, let $\mathcal{DK}_{\mathcal{J}} := (\mathcal{DK}_{i_1}, \mathcal{DK}_{i_2}, \dots, \mathcal{DK}_{i_j})$. Let $DK_{\mathcal{J}}$ be its corresponding random variable (i.e., $DK_{\mathcal{J}} := (DK_{i_1}, DK_{i_2}, \dots, DK_{i_j})$), and $dk_{\mathcal{J}} := (dk_{i_1}, dk_{i_2}, \dots, dk_{i_j})$.

Definition 4 (Perfect Secrecy for BE [11]). *A BE scheme Π is said to be $(\leq n, \leq \omega)$ -PS secure if it holds $H(M \mid C_{\mathcal{P}}, DK_{\mathcal{W}}) = H(M)$ for any $\mathcal{P} \subset \mathcal{R}$ and any $\mathcal{W} \in \mathscr{W}(\mathcal{P}, \omega)$.*

3 Security Definition Based on Guessing Secrety

We consider guessing secrety for BE analogous to traditional perfect secrety. More precisely, we assume at most ω ($< n$) colluders, and define guessing secrety against the colluders. As in previous works (e.g., [3, 11, 25]), we consider one-time secrety, which means that only one plaintext is encrypted and an adversary (i.e., colluders) observes the resulting ciphertext. We also consider two types of guessing secrety, *average guessing secrety* and *optimal guessing secrety*, which are due to [14], in the BE setting. The former, called A-GS, intuitively means that the advantage of adversary's optimal strategy for a random ciphertext, and the latter, called O-GS, intuitively means that the advantage of adversary's optimal strategy for all possible ciphertexts. Note that Alimomeni and Safavi-Naini first considered A-GS in [1], and later Iwamoto and Shikata introduced O-GS in [14]. We here recall definitions of A-GS and O-GS as follows.

Definition 5 (Guessing Secrety for SKE [1, 14]). *Let π be an SKE scheme. Two kinds of adversary's guessing probabilities are defined as follows:*

$$\text{A-GS}(\pi) := \sum_{c \in \mathcal{C}} P_C(c) \max_{m \in \mathcal{M}} P_{M|C}(m \mid c) = \sum_{c \in \mathcal{C}} \max_{m \in \mathcal{M}} P_{MC}(m, c),$$

$$\text{O-GS}(\pi) := \max_{c \in \mathcal{C}} \max_{m \in \mathcal{M}} P_{M|C}(m \mid c).$$

π is said to satisfy A-GS if $\text{A-GS}(\pi) = \max_{m \in \mathcal{M}} P_M(m)$. Further, π is said to satisfy O-GS if $\text{O-GS}(\pi) = \max_{m \in \mathcal{M}} P_M(m)$.

The following relationships among PS, A-GS, and O-GS are known.

Proposition 1 ([14]). *For any SKE scheme π , it holds that $\text{PS}(\pi) \Rightarrow \text{O-GS}(\pi) \Rightarrow \text{A-GS}(\pi)$, where “ $\text{A} \Rightarrow \text{B}$ ” means “ A implies B ”.*

Based on Definitions 4 and 5, we define guessing secrecy for BE. However, it becomes more complicated than the above definition since there are at most ω colluders. Namely, we have to consider the advantage of adversary’s strategy with colluders’ decryption keys. Therefore, we consider the following four types of guessing secrecy notions for any $\mathcal{P} \subset \mathcal{R}$ and any colluders $\mathcal{W} \in \mathscr{W}(\mathcal{P}, \omega)$.

1. A-GS for \mathcal{P} with randomly chosen decryption keys of \mathcal{W} .
2. A-GS for \mathcal{P} with all possible decryption keys of \mathcal{W} .
3. O-GS for \mathcal{P} with randomly chosen decryption keys of \mathcal{W} .
4. O-GS for \mathcal{P} with all possible decryption keys of \mathcal{W} .

We refer to the first and the last notions as average guessing secrecy (A-GS) and optimal guessing secrecy (O-GS) since they are the most “average” notion and the most “optimal” notion, respectively. Moreover, we refer to the second one as strong average guessing secrecy (sA-GS) and to the third one as weak optimal guessing secrecy (wO-GS). We formally define the above notions as follows.

Definition 6 (Guessing Secrecy for BE). *Let Π be a BE scheme. For any $\mathcal{P} \subset \mathcal{R}$ and any $\mathcal{W} \in \mathscr{W}(\mathcal{P}, \omega)$, four kinds of colluders’ guessing probabilities are defined as follows:*

$$\begin{aligned}
\text{A-GS}(\Pi, \mathcal{P}, \mathcal{W}) &:= \sum_{dk_{\mathcal{W}} \in \mathcal{DK}_{\mathcal{W}}} P_{DK_{\mathcal{W}}}(dk_{\mathcal{W}}) \sum_{c_{\mathcal{P}} \in \mathcal{C}_{\mathcal{P}}} P_{C_{\mathcal{P}}|DK_{\mathcal{W}}}(c_{\mathcal{P}} | dk_{\mathcal{W}}) \\
&\quad \cdot \max_{m \in \mathcal{M}} P_{M|C_{\mathcal{P}}DK_{\mathcal{W}}}(m | c_{\mathcal{P}}, dk_{\mathcal{W}}) \\
&= \sum_{dk_{\mathcal{W}} \in \mathcal{DK}_{\mathcal{W}}} \sum_{c_{\mathcal{P}} \in \mathcal{C}_{\mathcal{P}}} \max_{m \in \mathcal{M}} P_{MC_{\mathcal{P}}DK_{\mathcal{W}}}(m, c_{\mathcal{P}}, dk_{\mathcal{W}}), \\
\text{sA-GS}(\Pi, \mathcal{P}, \mathcal{W}) &:= \max_{dk_{\mathcal{W}} \in \mathcal{DK}_{\mathcal{W}}} \sum_{c_{\mathcal{P}} \in \mathcal{C}_{\mathcal{P}}} P_{C_{\mathcal{P}}|DK_{\mathcal{W}}}(c_{\mathcal{P}} | dk_{\mathcal{W}}) \\
&\quad \cdot \max_{m \in \mathcal{M}} P_{M|C_{\mathcal{P}}DK_{\mathcal{W}}}(m | c_{\mathcal{P}}, dk_{\mathcal{W}}) \\
&= \max_{dk_{\mathcal{W}} \in \mathcal{DK}_{\mathcal{W}}} \sum_{c_{\mathcal{P}} \in \mathcal{C}_{\mathcal{P}}} \max_{m \in \mathcal{M}} P_{MC_{\mathcal{P}}|DK_{\mathcal{W}}}(m, c_{\mathcal{P}} | dk_{\mathcal{W}}), \\
\text{wO-GS}(\Pi, \mathcal{P}, \mathcal{W}) &:= \sum_{dk_{\mathcal{W}} \in \mathcal{DK}_{\mathcal{W}}} P_{DK_{\mathcal{W}}}(dk_{\mathcal{W}}) \max_{c_{\mathcal{P}} \in \mathcal{C}_{\mathcal{P}}} \max_{m \in \mathcal{M}} P_{M|C_{\mathcal{P}}DK_{\mathcal{W}}}(m | c_{\mathcal{P}}, dk_{\mathcal{W}}) \\
\text{O-GS}(\Pi, \mathcal{P}, \mathcal{W}) &:= \max_{dk_{\mathcal{W}} \in \mathcal{DK}_{\mathcal{W}}} \max_{c_{\mathcal{P}} \in \mathcal{C}_{\mathcal{P}}} \max_{m \in \mathcal{M}} P_{M|C_{\mathcal{P}}DK_{\mathcal{W}}}(m | c_{\mathcal{P}}, dk_{\mathcal{W}}).
\end{aligned}$$

Π is said to be $(\leq n, \leq \omega)$ -X-GS ($X \in \{\text{A}, \text{sA}, \text{wO}, \text{O}\}$) secure (or, it is said to satisfy $(\leq n, \leq \omega)$ -X-GS) if it holds that

$$\text{X-GS}(\Pi) = \max_{m \in \mathcal{M}} P_M(m),$$

where $\text{X-GS}(\Pi) := \max_{\mathcal{P} \in \mathcal{R}, \mathcal{W} \in \mathscr{W}(\mathcal{P}, \omega)} \text{X-GS}(\Pi, \mathcal{P}, \mathcal{W})$.

The following relationships among the above security notions and PS are hold. The proof is omitted since it is straightforward.

Proposition 2. *For any BE scheme Π , it holds that*

$$\text{PS}(\Pi) \Rightarrow \text{O-GS}(\Pi) \Rightarrow \left\{ \begin{array}{l} \text{wO-GS}(\Pi) \\ \text{sA-GS}(\Pi) \end{array} \right\} \Rightarrow \text{A-GS}(\Pi).$$

4 Lower Bounds on Sizes of Ciphertexts and Secret Keys

In this section, we derive lower bounds on sizes of ciphertexts and secret keys required for BE schemes with guessing secrecy. Although the derived lower bounds are similar to those required for perfectly secure BE schemes [3, 16, 24],² the deriving techniques are different from those.

Theorem 1. *Let Π be an $(\leq n, \leq \omega)$ -X-GS ($X \in \{\text{A}, \text{sA}, \text{wO}, \text{O}\}$) secure BE scheme. Then, it holds that for any $\mathcal{P} \subset \mathcal{R}$,*

$$|\mathcal{C}_{\mathcal{P}}| \geq |\mathcal{M}|. \quad (1)$$

Moreover, if $|\mathcal{C}_{\mathcal{P}}| = |\mathcal{M}|$ for any $\mathcal{P} \subset \mathcal{R}$, it then holds that

$$\log |\mathcal{EK}| \geq \sum_{j=0}^{\omega} \binom{n}{j} \log |\mathcal{M}|, \quad (2)$$

$$\log |\mathcal{DK}_i| \geq \sum_{j=0}^{\omega} \binom{n-1}{j} \log |\mathcal{M}| \text{ for every } i \in [n]. \quad (3)$$

Proof. We can easily show Eq. (1) since Enc is an injective mapping for every $ek \in \mathcal{EK}$ and $\mathcal{P} \subset \mathcal{R}$ (i.e., $\text{Enc}(ek, \cdot, \mathcal{P}) : \mathcal{M} \rightarrow \mathcal{C}_{\mathcal{P}}$ is injective).

We next show Eqs. (2) and (3) for an $(\leq n, \leq \omega)$ -A-GS secure BE scheme Π since the bounds for A-GS secure scheme can be applied for any BE scheme satisfying sA-GS, wO-GS, or O-GS.

Proof of Eq. (2). Since the Enc algorithm is deterministic, we can consider a set of encryption keys $\mathcal{EK}_{\mathcal{P}}$ for a privileged set $\mathcal{P} \subset \mathcal{R}$, and \mathcal{EK} as $\prod_{\mathcal{P} \subset \mathcal{R}} \mathcal{EK}_{\mathcal{P}}$. In other words, we consider $ek \in \mathcal{EK}$ as a vector $(ek_{\mathcal{P}})_{\mathcal{P} \subset \mathcal{R}} \in \prod_{\mathcal{P} \subset \mathcal{R}} \mathcal{EK}_{\mathcal{P}}$, and $\text{Enc} : \mathcal{EK} \times \mathcal{M} \times \mathcal{P} \rightarrow \mathcal{C}_{\mathcal{P}}$ as $\mathcal{EK}_{\mathcal{P}} \times \mathcal{M} \rightarrow \mathcal{C}_{\mathcal{P}}$. We then show that the size of encryption keys $ek_{\mathcal{P}} \in \mathcal{EK}_{\mathcal{P}}$ must be larger than the size of ciphertexts $c_{\mathcal{P}} \in \mathcal{C}_{\mathcal{P}}$ (and hence the plaintext size). Finally, we show that for any two sets of colluders $\mathcal{W}, \mathcal{W}' \in \mathcal{W}(\omega)$, two subsets $\mathcal{EK}_{\mathcal{R} \setminus \mathcal{W}}$ and $\mathcal{EK}_{\mathcal{R} \setminus \mathcal{W}'}$ are disjoint. Namely, it indicates that we have to use a different encryption key depending on the set of possible colluders.

First, we show that it holds $|\mathcal{EK}_{\mathcal{P}}| \geq |\mathcal{C}_{\mathcal{P}}|$ for any $\mathcal{P} \subset \mathcal{R}$ in an $(\leq n, \leq \omega)$ -A-GS secure BE scheme Π . We now assume that for some \mathcal{P} , it

² For the lower bounds required for perfectly secure BE schemes, see Appendix A.

holds $|\mathcal{EK}_{\mathcal{P}}| < |\mathcal{C}_{\mathcal{P}}|$. Let m^* be a plaintext satisfying $m^* \in \arg \max_{m \in \mathcal{M}} P_M(m)$.

For m^* , we have ℓ ciphertexts $c_{\mathcal{P}}^{(1)}, c_{\mathcal{P}}^{(2)}, \dots, c_{\mathcal{P}}^{(\ell)}$, where $c_{\mathcal{P}}^{(i)} \leftarrow \text{Enc}(ek, m^*, \mathcal{P})$ for every $ek \in \mathcal{EK}_{\mathcal{P}}$ and $\ell \leq |\mathcal{EK}_{\mathcal{P}}|$ ($\text{Enc}(\cdot, m^*)$ might output the same $c_{\mathcal{P}}^{(j)}$ with some two keys $ek_{\mathcal{P}}, ek'_{\mathcal{P}} \in \mathcal{EK}_{\mathcal{P}}$, and hence $\ell \leq |\mathcal{EK}_{\mathcal{P}}|$). It means that $\text{Enc}(\cdot, m^*) : \mathcal{EK} \rightarrow \{c_{\mathcal{P}}^{(j)}\}_{j=1}^{\ell}$, and we have $\mathcal{C}_{\mathcal{P}} \setminus \{c_{\mathcal{P}}^{(j)}\}_{j=1}^{\ell} \neq \emptyset$ by the assumption. For any $\mathcal{W} \in \mathscr{W}(\mathcal{P}, \omega)$, we have

$$\begin{aligned}
\max_{m \in \mathcal{M}} P_M(m) &= P_M(m^*) \\
&= \sum_{dk_{\mathcal{W}} \in \mathcal{DK}_{\mathcal{W}}} P_{DK_{\mathcal{W}}}(dk_{\mathcal{W}}) P_M(m^*) \\
&= \sum_{dk_{\mathcal{W}} \in \mathcal{DK}_{\mathcal{W}}} P_{DK_{\mathcal{W}}}(dk_{\mathcal{W}}) P_M(m^*) \sum_{i=1}^{\ell} P_{C_{\mathcal{P}} | MDK_{\mathcal{W}}}(c_{\mathcal{P}}^{(i)} | m^*, dk_{\mathcal{W}}) \\
&= \sum_{dk_{\mathcal{W}} \in \mathcal{DK}_{\mathcal{W}}} \sum_{i=1}^{\ell} P_{MC_{\mathcal{P}} DK_{\mathcal{W}}}(m^*, c_{\mathcal{P}}^{(i)}, dk_{\mathcal{W}}) \\
&< \sum_{dk_{\mathcal{W}} \in \mathcal{DK}_{\mathcal{W}}} \sum_{i=1}^{\ell} P_{MC_{\mathcal{P}} DK_{\mathcal{W}}}(m^*, c_{\mathcal{P}}^{(i)}, dk_{\mathcal{W}}) \\
&\quad + \sum_{dk_{\mathcal{W}} \in \mathcal{DK}_{\mathcal{W}}} \sum_{c_{\mathcal{P}} \in \mathcal{C}_{\mathcal{P}} \setminus \{c_i\}_{i=1}^{\ell}} \max_{m \in \mathcal{M}} P_{MC_{\mathcal{P}} DK_{\mathcal{W}}}(m, c_{\mathcal{P}}, dk_{\mathcal{W}}) \\
&\leq \text{A-GS}(II, \mathcal{P}, \mathcal{W}).
\end{aligned}$$

This is a contradiction with A-GS of II , and hence we have $|\mathcal{EK}_{\mathcal{P}}| \geq |\mathcal{C}_{\mathcal{P}}|$ for any $\mathcal{P} \subset \mathcal{R}$.

We next show that for any distinct $\mathcal{W}, \mathcal{W}' \in \mathscr{W}(\omega)$, $\mathcal{P} := \mathcal{R} \setminus \mathcal{W}$, and $\mathcal{P}' := \mathcal{R} \setminus \mathcal{W}'$, $\mathcal{EK}_{\mathcal{P}}$ and $\mathcal{EK}_{\mathcal{P}'}$ are disjoint (i.e., $\mathcal{EK}_{\mathcal{P}} \cap \mathcal{EK}_{\mathcal{P}'} = \emptyset$). We assume that it holds $\mathcal{EK}_{\mathcal{P}} \cap \mathcal{EK}_{\mathcal{P}'} \neq \emptyset$ for some distinct $\mathcal{W}, \mathcal{W}' \in \mathscr{W}(\omega)$, $\mathcal{P} := \mathcal{R} \setminus \mathcal{W}$, and $\mathcal{P}' := \mathcal{R} \setminus \mathcal{W}'$. Then, there exists $\widetilde{ek}_{\mathcal{P} \cap \mathcal{P}'} \in \mathcal{EK}_{\mathcal{P}} \cap \mathcal{EK}_{\mathcal{P}'}$ and some receiver R_{i^*} such that $R_{i^*} \in \mathcal{P} \cap \mathcal{W}'$ but $R_{i^*} \notin \mathcal{P}'$.

We have $\text{Enc}(\widetilde{ek}_{\mathcal{P} \cap \mathcal{P}'}, \cdot) : \mathcal{M} \rightarrow \mathcal{C}_{\mathcal{P}} \cap \mathcal{C}_{\mathcal{P}'}$ since ciphertexts encrypted by $\widetilde{ek}_{\mathcal{P} \cap \mathcal{P}'}$ have to be decrypted by every receiver in \mathcal{P} or \mathcal{P}' . Then, R_{i^*} can decrypt $c_{\mathcal{P}'} \leftarrow \text{Enc}(\widetilde{ek}_{\mathcal{P} \cap \mathcal{P}'}, m^*, \mathcal{P}') \in (\mathcal{C}_{\mathcal{P}} \cap \mathcal{C}_{\mathcal{P}'})$ since $R_{i^*} \in \mathcal{P}$ can decrypt all ciphertexts in $\mathcal{C}_{\mathcal{P}}$. This is a contradiction with A-GS of II , and thus it holds $\mathcal{EK}_{\mathcal{R} \setminus \mathcal{W}} \cap \mathcal{EK}_{\mathcal{R} \setminus \mathcal{W}'} = \emptyset$ for any distinct $\mathcal{W}, \mathcal{W}' \in \mathscr{W}(\omega)$. Since $|\mathcal{EK}_{\mathcal{P}}| \geq |\mathcal{C}_{\mathcal{P}}| = |\mathcal{M}|$ for any $\mathcal{P} \subset \mathcal{R}$ and $|\mathscr{W}(\omega)| = \sum_{j=0}^{\omega} \binom{n}{j}$, we have

$$|\mathcal{EK}| \geq \prod_{\mathcal{W}(\omega)} |\mathcal{EK}_{\mathcal{P}}| \geq \prod_{\mathcal{W}(\omega)} |\mathcal{C}_{\mathcal{P}}| = \prod_{\mathcal{W}(\omega)} |\mathcal{M}| = |\mathcal{M}|^{\sum_{j=0}^{\omega} \binom{n}{j}}.$$

Proof of Eq. (3). We can prove Eq. (3) in a way similar to the proof of Eq. (2). We first give an intuition. Since the Dec algorithm is deterministic, we can

consider a set of R_i 's decryption keys for every privileged set $\mathcal{P} \in \mathcal{P}^{(i)}$, and \mathcal{DK}_i as $\prod_{\mathcal{P} \in \mathcal{P}^{(i)}} \mathcal{DK}_{i,\mathcal{P}}$. In other words, we consider $dk_i \in \mathcal{DK}_i$ as a vector $(dk_{i,\mathcal{P}})_{\mathcal{P} \in \mathcal{P}^{(i)}} \in \prod_{\mathcal{P} \in \mathcal{P}^{(i)}} \mathcal{DK}_{i,\mathcal{P}}$, and $\text{Dec} : \mathcal{DK}_i \times \mathcal{C}_{\mathcal{P}} \rightarrow \mathcal{M}$ as $\mathcal{DK}_{i,\mathcal{P}} \times \mathcal{C}_{\mathcal{P}} \rightarrow \mathcal{M}$ for every $\mathcal{P} \in \mathcal{P}^{(i)}$. We then show that the size of R_i 's decryption keys $dk_{i,\mathcal{P}} \in \mathcal{DK}_{i,\mathcal{P}}$ must be larger than the size of ciphertexts $c_{\mathcal{P}} \in \mathcal{C}_{\mathcal{P}}$ (and hence the plaintext size). Finally, we show that for any two sets of colluders $\mathcal{W}, \mathcal{W}' \in \mathcal{W}^{(i)}(\omega)$, two subsets $\mathcal{DK}_{i,\mathcal{R} \setminus \mathcal{W}}$ and $\mathcal{DK}_{i,\mathcal{R} \setminus \mathcal{W}'}$ are disjoint. Namely, it indicates that each receiver R_i must have a different decryption key depending on the possible set of colluders.

First, we show that for any $\mathcal{P} \in \mathcal{P}^{(i)}$, it holds $|\mathcal{DK}_{i,\mathcal{P}}| \geq |\mathcal{C}_{\mathcal{P}}|$ in an $(\leq n, \leq \omega)$ -A-GS secure BE scheme Π . We now assume it holds for some $\mathcal{P} \in \mathcal{P}^{(i)}$, $|\mathcal{DK}_{i,\mathcal{P}}| < |\mathcal{C}_{\mathcal{P}}|$. For every $dk_{i,\mathcal{P}} \in \mathcal{DK}_{i,\mathcal{P}}$, $\text{Dec}(dk_{i,\mathcal{P}}, \cdot) : \mathcal{C}_{\mathcal{P}} \rightarrow \mathcal{M}$ is bijective from the assumption $|\mathcal{C}_{\mathcal{P}}| = |\mathcal{M}|$. Namely, for any $m \in \mathcal{M}$ and any $dk_{i,\mathcal{P}} \in \mathcal{DK}_{i,\mathcal{P}}$, there is a unique ciphertext $c_{\mathcal{P}} \in \mathcal{C}_{\mathcal{P}}$ such that $\text{Dec}(dk_{i,\mathcal{P}}, c_{\mathcal{P}}) \rightarrow m$. We have the number of ciphertexts of any $m \in \mathcal{M}$ for \mathcal{P} is at most $|\mathcal{DK}_{i,\mathcal{P}}|$ since some two keys $dk_{i,\mathcal{P}}, dk'_{i,\mathcal{P}} \in \mathcal{DK}_{i,\mathcal{P}}$ might share the same ciphertext $c_{\mathcal{P}}$. Let m^* be a plaintext satisfying $m^* \in \arg \max_{m \in \mathcal{M}} P_{\mathcal{M}}(m)$, and we here assume the number of ciphertexts of m^* for \mathcal{P} is $\ell' (\leq |\mathcal{DK}_{i,\mathcal{P}}|)$. They are denoted by $c_{\mathcal{P}}^{(1)}, c_{\mathcal{P}}^{(2)}, \dots, c_{\mathcal{P}}^{(\ell')}$. Therefore, we have

$$\sum_{i=1}^{\ell'} P_{\mathcal{C}_{\mathcal{P}}|\mathcal{M}}(c_{\mathcal{P}}^{(i)} | m^*) = 1. \quad (4)$$

We have $\mathcal{C}_{\mathcal{P}} \setminus \{c_{\mathcal{P}}^{(j)}\}_{j=1}^{\ell'} \neq \emptyset$ by the assumption. For any $\mathcal{W} \in \mathcal{W}(\mathcal{P}, \omega)$, we have

$$\begin{aligned} & \max_{m \in \mathcal{M}} P_{\mathcal{M}}(m) \\ &= P_{\mathcal{M}}(m^*) \\ &= \sum_{dk_{\mathcal{W}} \in \mathcal{DK}_{\mathcal{W}}} P_{\mathcal{DK}_{\mathcal{W}}}(dk_{\mathcal{W}}) P_{\mathcal{M}}(m^*) \\ &= \sum_{dk_{\mathcal{W}} \in \mathcal{DK}_{\mathcal{W}}} P_{\mathcal{DK}_{\mathcal{W}}}(dk_{\mathcal{W}}) P_{\mathcal{M}}(m^*) \sum_{i=1}^{\ell'} P_{\mathcal{C}_{\mathcal{P}}|\mathcal{M}\mathcal{DK}_{\mathcal{W}}}(c_{\mathcal{P}}^{(i)} | m^*, dk_{\mathcal{W}}) \quad (5) \\ &= \sum_{dk_{\mathcal{W}} \in \mathcal{DK}_{\mathcal{W}}} \sum_{i=1}^{\ell'} P_{\mathcal{MC}_{\mathcal{P}}\mathcal{DK}_{\mathcal{W}}}(m^*, c_{\mathcal{P}}^{(i)}, dk_{\mathcal{W}}) \\ &< \sum_{dk_{\mathcal{W}} \in \mathcal{DK}_{\mathcal{W}}} \sum_{i=1}^{\ell'} P_{\mathcal{MC}_{\mathcal{P}}\mathcal{DK}_{\mathcal{W}}}(m^*, c_{\mathcal{P}}^{(i)}, dk_{\mathcal{W}}) \\ &\quad + \sum_{dk_{\mathcal{W}} \in \mathcal{DK}_{\mathcal{W}}} \sum_{c_{\mathcal{P}} \in \mathcal{C}_{\mathcal{P}} \setminus \{c_i\}_{i=1}^{\ell'}} \max_{m \in \mathcal{M}} P_{\mathcal{MC}_{\mathcal{P}}\mathcal{DK}_{\mathcal{W}}}(m, c_{\mathcal{P}}, dk_{\mathcal{W}}) \\ &\leq \text{A-GS}(\Pi, \mathcal{P}, \mathcal{W}). \end{aligned}$$

where Eq. (5) follows from Eq. (4). This is a contradiction with A-GS of Π , and hence we have $|\mathcal{DK}_{i,\mathcal{P}}| \geq |\mathcal{C}_{\mathcal{P}}|$ for any $\mathcal{P} \in \mathcal{P}^{(i)}$.

We next show that for any distinct $\mathcal{W}, \mathcal{W}' \in \mathcal{W}^{(i)}(\omega)$, $\mathcal{P} := \mathcal{R} \setminus \mathcal{W}$, and $\mathcal{P}' := \mathcal{R} \setminus \mathcal{W}'$, $\mathcal{DK}_{i,\mathcal{P}}$ and $\mathcal{DK}_{i,\mathcal{P}'}$ are disjoint (i.e., $\mathcal{DK}_{i,\mathcal{P}} \cap \mathcal{DK}_{i,\mathcal{P}'} = \emptyset$). We assume that it holds $\mathcal{DK}_{i,\mathcal{P}} \cap \mathcal{DK}_{i,\mathcal{P}'} \neq \emptyset$ for some distinct $\mathcal{W}, \mathcal{W}' \in \mathcal{W}^{(i)}(\omega)$, $\mathcal{P} := \mathcal{R} \setminus \mathcal{W}$, and $\mathcal{P}' := \mathcal{R} \setminus \mathcal{W}'$. Then, there exists $\widetilde{dk}_{i,\mathcal{P} \cap \mathcal{P}'} \in \mathcal{DK}_{i,\mathcal{P}} \cap \mathcal{DK}_{i,\mathcal{P}'}$ and some receiver $R_{j^*} \in \mathcal{P} \cap \mathcal{W}'$ (i.e., $R_{j^*} \in \mathcal{P} \setminus \mathcal{P}'$). Note that $R_i \in \mathcal{P}$ and $R_i \in \mathcal{P}'$.

We then have $\text{Dec}(\widetilde{dk}_{i,\mathcal{P} \cap \mathcal{P}'}, \cdot) : \mathcal{C}_{\mathcal{P}} \cap \mathcal{C}_{\mathcal{P}'} \rightarrow \mathcal{M}$ since for any $m \in \mathcal{M}$, R_i can decrypt both $\text{Enc}(\widetilde{ek}, m, \mathcal{P}) \in \mathcal{C}_{\mathcal{P}}$ and $\text{Enc}(\widetilde{ek}, m, \mathcal{P}') \in \mathcal{C}_{\mathcal{P}'}$ by using $\widetilde{dk}_{i,\mathcal{P} \cap \mathcal{P}'}$, where $\widetilde{ek} \in \mathcal{EK}$ is an encryption key generated along with $\widetilde{dk}_{i,\mathcal{P} \cap \mathcal{P}'}$ by Setup. Namely, we have $\text{Enc}(\widetilde{ek}, \cdot, \mathcal{P}) : \mathcal{M} \rightarrow \mathcal{C}_{\mathcal{P}} \cap \mathcal{C}_{\mathcal{P}'}$ and $\text{Enc}(\widetilde{ek}, \cdot, \mathcal{P}') : \mathcal{M} \rightarrow \mathcal{C}_{\mathcal{P}} \cap \mathcal{C}_{\mathcal{P}'}$ since ciphertexts encrypted by \widetilde{ek} have to be decrypted by $\widetilde{dk}_{i,\mathcal{P} \cap \mathcal{P}'}$ due to the decryption correctness. Therefore, $R_{j^*} (\notin \mathcal{P}')$ can decrypt $c_{\mathcal{P}'} \leftarrow \text{Enc}(\widetilde{ek}, m^*, \mathcal{P}') \in (\mathcal{C}_{\mathcal{P}} \cap \mathcal{C}_{\mathcal{P}'})$ since $R_{j^*} \in \mathcal{P}$ can decrypt all ciphertexts in $\mathcal{C}_{\mathcal{P}}$. This is a contradiction with A-GS of Π , and thus it holds $\mathcal{DK}_{i,\mathcal{R} \setminus \mathcal{W}} \cap \mathcal{DK}_{i,\mathcal{R} \setminus \mathcal{W}'} = \emptyset$ for any distinct $\mathcal{W}, \mathcal{W}' \in \mathcal{W}^{(i)}(\omega)$. Since $|\mathcal{DK}_{i,\mathcal{P}}| \geq |\mathcal{C}_{\mathcal{P}}| = |\mathcal{M}|$ for every $\mathcal{P} \in \mathcal{P}^{(i)}$ and $|\mathcal{W}^{(i)}(\omega)| = \sum_{j=0}^{\omega} \binom{n-1}{j}$, we have

$$|\mathcal{DK}_i| \geq \prod_{\mathcal{W}^{(i)}(\omega)} |\mathcal{DK}_{i,\mathcal{P}}| \geq \prod_{\mathcal{W}^{(i)}(\omega)} |\mathcal{C}_{\mathcal{P}}| = \prod_{\mathcal{W}^{(i)}(\omega)} |\mathcal{M}| = |\mathcal{M}|^{\sum_{j=0}^{\omega} \binom{n-1}{j}}. \quad \square$$

In fact, the derived bounds are tight since in the next section, we will see a construction that attains every bound of Theorem 1 with equality. We define what is the most efficient BE scheme in terms of sizes of ciphertexts and secret keys as follows.

Definition 7 (BE with the Shortest Ciphertexts and Keys). *A construction of an $(\leq n, \leq \omega)$ -X-GS ($X \in \{A, sA, wO, O\}$) secure BE scheme is said to achieve the shortest ciphertexts and keys³ if it meets equality in every bound of Eqs. (1), (2), and (3) in Theorem 1.*

5 Construction

In this section, we propose a one-bit construction that meet guessing secrecy with non-uniform keys. Specifically, we analyze (a variant of) the Fiat-Naor construction [11] with non-uniform keys.

The idea of the Fiat-Naor construction is simple: Random elements $r_{\mathcal{W}}$ are chosen for every set of colluders $\mathcal{W} \in \mathcal{W}(\omega)$, and a ciphertext $c_{\mathcal{P}}$ for $\mathcal{P} \subset \mathcal{R}$ is the one-time pad by the sum of all random elements $r_{\mathcal{W}}$ such that $\mathcal{W} \in \mathcal{W}(\mathcal{P}, \omega)$. Any $\mathcal{W} \subset (\mathcal{R} \setminus \mathcal{P})$ does not have at least one random element $r_{\mathcal{W}}$ used for

³ Information-theoretically secure schemes that attain all lower bounds of secret keys with equalities are often said to be *optimal*. However, in this paper we do not use the terminology here to avoid confusion since we already use it for O-GS.

the encryption, and therefore the Fiat-Naor construction meets PS when the probability distribution on the randomness is uniform [11].

In this section, we simplify (or, fine-tune) the Fiat-Naor construction as follows: Random elements $r_{\mathcal{W}}$ for a ciphertext $c_{\mathcal{P}}$ are chosen according to $\mathcal{W} \in \widehat{\mathscr{W}}(\mathcal{P}, \omega)$, instead of $\mathscr{W}(\mathcal{P}, \omega)$. The ciphertext is the one-time pad by the sum of them. In fact, colluder sets that we have to pay attention are \mathcal{W} with the maximum cardinality (ω or $n - |\mathcal{P}|$), since the construction is secure against \mathcal{W} then it implies it is secure against $\mathcal{W}' \subset \mathcal{W}$. Therefore, this modification reduces the random elements used for creating a ciphertext while keeping the construction secure, and hence, it makes the security analysis of the modified construction easy.

We analyze whether or not the modified construction meets X-GS ($X \in \{A, sA, wO, O\}$), and clarify requirements if it meets any of them. As a result, we show that the construction meets sA-GS (and also A-GS) if probability distribution of plaintexts are more biased than that of randomness used in the construction. Furthermore, we show that wO-GS, O-GS, and PS are equivalent under the construction. In other words, uniform distribution of randomness is required for wO-GS (and O-GS).

The modified Fiat-Naor construction with common biased randomness. Suppose that $\mathcal{M} = \mathcal{C} = \{0, 1\}$, and $P_{\mathcal{M}}(0) = q$. We assume a biased random source $R_{\mathcal{W}}$ which takes values in $\{0, 1\}$ such that $P_{R_{\mathcal{W}}}(0) = p$ for any $\mathcal{W} \in \mathscr{W}(\omega)$. Without loss of generality, we assume $1/2 \leq q < 1$ and $1/2 \leq p < 1$.

1. $(ek, dk_1, \dots, dk_n) \leftarrow \text{Setup}(n, P_{SK})$: Output $ek := \{r_{\mathcal{W}}\}_{\mathcal{W} \in \mathscr{W}(\omega)}$ and $dk_i := \{r_{\mathcal{W}}\}_{\mathcal{W} \in \mathscr{W}^{(i)}(\omega)}$ for any $i \in [n]$.
2. $c_{\mathcal{P}} \leftarrow \text{Enc}(ek, m, \mathcal{P})$: Compute and output

$$c_{\mathcal{P}} := m \bigoplus_{\mathcal{W} \in \widehat{\mathscr{W}}(\mathcal{P}, \omega)} r_{\mathcal{W}}.$$

3. m or $\perp \leftarrow \text{Dec}(dk_i, c_{\mathcal{P}})$: Output \perp if $R_i \notin \mathcal{P}$. Otherwise, output

$$m = c_{\mathcal{P}} \bigoplus_{\mathcal{W} \in \widehat{\mathscr{W}}(\mathcal{P}, \omega)} r_{\mathcal{W}}.$$

Theorem 2. *A BE scheme Π given by the above construction is $(\leq n, \leq \omega)$ -sA-GS secure and achieves the shortest ciphertexts and keys if and only if $p \leq q$.*

Proof. Since it is straightforward that Π achieves the shortest ciphertexts and keys, we omit the proof. Without loss of generality, we fix some $\mathcal{P} \subset \mathcal{R}$ such that $|\mathcal{P}| = n - \omega$ and $\mathcal{W} = \mathcal{R} \setminus \mathcal{P}$. This maximizes the amount of \mathcal{W} 's information related to $c_{\mathcal{P}}$, and simplifies the analysis since only one random element $r_{\mathcal{W}}$ is used for computing $c_{\mathcal{P}}$ (i.e., $c_{\mathcal{P}} := m \oplus r_{\mathcal{W}}$). An analysis for the case of $|\mathcal{W}| < \omega$ is more complicated, however it basically follows the following proof. Therefore, we omit it here, and the more detailed analysis will appear in the full version. Let $r_{\mathscr{W}(\omega)} := \{r_{\mathcal{W}}\}_{\mathcal{W} \in \mathscr{W}(\omega)}$. Namely, $r_{\mathscr{W}(\omega)}$ denotes all random elements that

\mathcal{W} has (i.e., $dk_{\mathcal{W}}$), and we denote it as a $|\mathcal{Y}^{\mathcal{W}}|$ -bits string for simplicity. Let $R_{\mathcal{Y}^{\mathcal{W}}}$ be a random variable of $r_{\mathcal{Y}^{\mathcal{W}}}$.

Then, we have

$$\begin{aligned}
& \text{sA-GS}(II, \mathcal{P}, \mathcal{W}) \\
&= \max_{dk_{\mathcal{W}} \in \mathcal{DK}_{\mathcal{W}}} \sum_{c_{\mathcal{P}} \in \mathcal{C}_{\mathcal{P}}} P_{C_{\mathcal{P}}}(c_{\mathcal{P}}) \max_{m \in \mathcal{M}} P_{M|C_{\mathcal{P}}DK_{\mathcal{W}}}(m | c_{\mathcal{P}}, dk_{\mathcal{W}}) \\
&= \max_{r_{\mathcal{Y}^{\mathcal{W}}} \in \{0,1\}^{|\mathcal{Y}^{\mathcal{W}}|}} \left(\underbrace{(P_{C_{\mathcal{P}}}(0) \max_{m \in \mathcal{M}} P_{M|C_{\mathcal{P}}R_{\mathcal{Y}^{\mathcal{W}}}}(m | 0, r_{\mathcal{Y}^{\mathcal{W}}}))}_{(a)} \right. \\
&\quad \left. + \underbrace{P_{C_{\mathcal{P}}}(1) \max_{m \in \mathcal{M}} P_{M|C_{\mathcal{P}}R_{\mathcal{Y}^{\mathcal{W}}}}(m | 1, r_{\mathcal{Y}^{\mathcal{W}}}))}_{(b)} \right).
\end{aligned}$$

As for (a), we have

$$\begin{aligned}
& P_{C_{\mathcal{P}}}(0) \max_{m \in \mathcal{M}} P_{M|CR_{\mathcal{Y}^{\mathcal{W}}}}(m | 0, r_{\mathcal{Y}^{\mathcal{W}}}) \\
&= P_{C_{\mathcal{P}}}(0) \max_{m \in \mathcal{M}} \frac{P_{MC_{\mathcal{P}}R_{\mathcal{Y}^{\mathcal{W}}}}(m, 0, r_{\mathcal{Y}^{\mathcal{W}}})}{P_{C_{\mathcal{P}}R_{\mathcal{Y}^{\mathcal{W}}}}(0, r_{\mathcal{Y}^{\mathcal{W}}})} \\
&= P_{C_{\mathcal{P}}}(0) \max_{m \in \mathcal{M}} \frac{P_{MC_{\mathcal{P}}}(m, 0)P_{R_{\mathcal{Y}^{\mathcal{W}}}}(r_{\mathcal{Y}^{\mathcal{W}}})}{P_{C_{\mathcal{P}}}(0)P_{R_{\mathcal{Y}^{\mathcal{W}}}}(r_{\mathcal{Y}^{\mathcal{W}}})} \quad (6) \\
&= \max_{m \in \mathcal{M}} P_{MC_{\mathcal{P}}}(m, 0),
\end{aligned}$$

where Eq. (6) follows from that $r_{\mathcal{Y}^{\mathcal{W}}}$ is independent of $(m, r_{\mathcal{W}})$. Since $P_{MC_{\mathcal{P}}}(0, 0) = pq$ and $P_{MC_{\mathcal{P}}}(1, 0) = (1-p)(1-q)$, we have

$$\max_{r_{\mathcal{Y}^{\mathcal{W}}} \in \{0,1\}^{|\mathcal{Y}^{\mathcal{W}}|}} P_{C_{\mathcal{P}}}(0) \max_{m \in \mathcal{M}} P_{M|CR_{\mathcal{Y}^{\mathcal{W}}}}(m | 0, r_{\mathcal{Y}^{\mathcal{W}}}) = pq. \quad (7)$$

Similarly, as for (b), we have

$$P_{C_{\mathcal{P}}}(1) \max_{m \in \mathcal{M}} P_{M|CR_{\mathcal{Y}^{\mathcal{W}}}}(m | 1, r_{\mathcal{Y}^{\mathcal{W}}}) = \max_{m \in \mathcal{M}} P_{MC_{\mathcal{P}}}(m, 1).$$

Since $P_{MC_{\mathcal{P}}}(0, 1) = (1-p)q$ and $P_{MC_{\mathcal{P}}}(1, 1) = p(1-q)$, we have

$$P_{C_{\mathcal{P}}}(1) \max_{m \in \mathcal{M}} P_{M|CR_{\mathcal{Y}^{\mathcal{W}}}}(m | 1, r_{\mathcal{Y}^{\mathcal{W}}}) = \max\{(1-p)q, p(1-q)\}. \quad (8)$$

From Eqs. (7) and (8), we have

$$\text{sA-GS}(II, \mathcal{P}, \mathcal{W}) = pq + \max\{(1-p)q, p(1-q)\}.$$

If $p \leq q$, we have

$$\text{sA-GS}(II, \mathcal{P}, \mathcal{W}) = pq + (1-p)q = q = P_M(0) = \max_{m \in \mathcal{M}} P_M(m).$$

Therefore, Π is $(\leq n, \leq \omega)$ -sA-GS secure.

On the other hand, if $p > q$, then we have

$$\text{sA-GS}(\Pi, \mathcal{P}, \mathcal{W}) = pq + p(1 - q) = p > q = \max_{m \in \mathcal{M}} P_M(m).$$

Therefore, Π is not $(\leq n, \leq \omega)$ -sA-GS secure if $p > q$. \square

A toy example. Let $n = 4$ and $\omega = 2$. An encryption key and decryption keys are

$$\begin{aligned} ek &:= \{r_{\mathcal{W}}\}_{\mathcal{W} \in \mathscr{W}(\omega)} \\ &= \{r_{\emptyset}, r_{\{1\}}, r_{\{2\}}, r_{\{3\}}, r_{\{4\}}, r_{\{1,2\}}, r_{\{1,3\}}, r_{\{1,4\}}, r_{\{2,3\}}, r_{\{2,4\}}, r_{\{3,4\}}\}, \\ dk_1 &:= \{r_{\mathcal{W}}\}_{\mathcal{W} \in \mathscr{W}^{(1)}(\omega)} = \{r_{\emptyset}, r_{\{2\}}, r_{\{3\}}, r_{\{4\}}, r_{\{2,3\}}, r_{\{2,4\}}, r_{\{3,4\}}\}, \\ dk_2 &:= \{r_{\mathcal{W}}\}_{\mathcal{W} \in \mathscr{W}^{(2)}(\omega)} = \{r_{\emptyset}, r_{\{1\}}, r_{\{3\}}, r_{\{4\}}, r_{\{1,3\}}, r_{\{1,4\}}, r_{\{3,4\}}\}, \\ dk_3 &:= \{r_{\mathcal{W}}\}_{\mathcal{W} \in \mathscr{W}^{(3)}(\omega)} = \{r_{\emptyset}, r_{\{1\}}, r_{\{2\}}, r_{\{4\}}, r_{\{1,2\}}, r_{\{1,4\}}, r_{\{2,4\}}\}, \\ dk_4 &:= \{r_{\mathcal{W}}\}_{\mathcal{W} \in \mathscr{W}^{(4)}(\omega)} = \{r_{\emptyset}, r_{\{1\}}, r_{\{2\}}, r_{\{3\}}, r_{\{1,2\}}, r_{\{1,3\}}, r_{\{2,3\}}\}, \end{aligned}$$

where we denote R_i as i for simplicity. Suppose $\mathcal{P} = \{R_1, R_2\}$. Since $\widehat{\mathscr{W}}(\mathcal{P}, 2) = \{\{R_3, R_4\}\}$, $c_{\mathcal{P}} := m \oplus r_{\{3,4\}}$.⁴ For $\mathcal{W} = \{R_3, R_4\}$, we have

$$\mathscr{W}^{(\mathcal{W})} = \{\emptyset, \{R_1\}, \{R_2\}, \{R_3\}, \{R_4\}, \{R_1, R_2\}, \{R_1, R_3\}, \{R_1, R_4\}, \{R_2, R_3\}, \{R_2, R_4\}\}.$$

Therefore, we have $\mathscr{W}(\omega) \setminus \mathscr{W}^{(\mathcal{W})} = \{\mathcal{W} = \{R_3, R_4\}\}$. Then, we have

$$\begin{aligned} &\text{sA-GS}(\Pi, \mathcal{P} = \{R_1, R_2\}, \mathcal{W} = \{R_3, R_4\}) \\ &= \max_{dk_{\mathcal{W}} \in \mathcal{DK}_{\mathcal{W}}} \sum_{c_{\mathcal{P}} \in \mathcal{C}_{\mathcal{P}}} P_{C_{\mathcal{P}}}(c_{\mathcal{P}}) \max_{m \in \mathcal{M}} P_{M|C_{\mathcal{P}}DK_{\mathcal{W}}}(m \mid c_{\mathcal{P}}, dk_{\mathcal{W}}) \\ &= \max_{r_{\mathscr{W}^{(\mathcal{W})}} \in \{0,1\}^{|\mathscr{W}^{(\mathcal{W})}|}} \left(\underbrace{(P_{C_{\mathcal{P}}}(0) \max_{m \in \mathcal{M}} P_{M|C_{\mathcal{P}}R_{\mathscr{W}^{(\mathcal{W})}}}(m \mid 0, r_{\mathscr{W}^{(\mathcal{W})}}))}_{(a)} \right. \\ &\quad \left. + \underbrace{P_{C_{\mathcal{P}}}(1) \max_{m \in \mathcal{M}} P_{M|C_{\mathcal{P}}R_{\mathscr{W}^{(\mathcal{W})}}}(m \mid 1, r_{\mathscr{W}^{(\mathcal{W})}})}_{(b)} \right). \end{aligned}$$

As for (a), we have

$$\begin{aligned} &P_{C_{\mathcal{P}}}(m \oplus r_{\{3,4\}} = 0) \max_{m \in \mathcal{M}} P_{M|CR_{\mathscr{W}^{(\mathcal{W})}}}(m \mid m \oplus r_{\{3,4\}} = 0, r_{\mathscr{W}^{(\mathcal{W})}}) \\ &= P_{C_{\mathcal{P}}}(m \oplus r_{\{3,4\}} = 0) \max_{m \in \mathcal{M}} P_{M|C_{\mathcal{P}}}(m \mid m \oplus r_{\{3,4\}} = 0) \quad (9) \\ &= \max_{m \in \mathcal{M}} P_{MC_{\mathcal{P}}}(m, m \oplus r_{\{3,4\}} = 0) \\ &= P_{MR_{\mathcal{W}}}(m = 0, r_{\{3,4\}} = 0) \\ &= pq. \quad (10) \end{aligned}$$

⁴ $c_{\mathcal{P}} := m \oplus r_{\{3,4\}} \oplus r_{\emptyset} \oplus r_3 \oplus r_4$ in the original Fiat-Naor construction.

where Eq. (9) follows from that $\mathcal{W} = \{\mathbf{R}_3, \mathbf{R}_4\} \notin \mathscr{W}^{(\mathcal{W})}$ and m and each $r_{\mathcal{W}}$ are independently chosen.

Similarly, as for (b), we have

$$\begin{aligned} P_{C_{\mathcal{P}}}(m \oplus r_{\{3,4\}} = 1) &= \max_{m \in \mathcal{M}} P_{M|CR_{\mathscr{W}^{(\mathcal{W})}}}(m \mid m \oplus r_{\{3,4\}} = 1, r_{\mathscr{W}^{(\mathcal{W})}}) \\ &= \max_{m \in \mathcal{M}} P_{MR_{\mathcal{W}}}(m, r_{\{3,4\}} = 1) \\ &= \max\{(1-p)q, p(1-q)\}. \end{aligned} \quad (11)$$

From Eqs. (10) and (11), if $p \leq q$ we have

$$\text{sA-GS}(II) = \text{sA-GS}(II, \mathcal{P}, \mathcal{W}) = pq + (1-p)q = q = P_M(0) = \max_{m \in \mathcal{M}} P_M(m).$$

Therefore, II is $(\leq 4, \leq 2)$ -sA-GS secure.

We also show that the Fiat-Naor construction satisfies wO-GS (and O-GS) if and only if it satisfies PS.

Theorem 3. *For a BE scheme II given by the above construction, the following statements are equivalent:*

- (I) *All the probability distributions of the randomness $P_{R_{\mathcal{W}}}$ for $\mathcal{W} \in \mathscr{W}(\omega)$ are uniform (i.e., $p = 1/2$).*
- (II) *II is $(\leq n, \leq \omega)$ -PS secure.*
- (III) *II is $(\leq n, \leq \omega)$ -O-GS secure.*
- (IV) *II is $(\leq n, \leq \omega)$ -wO-GS secure.*

Furthermore, $(\leq n, \leq \omega)$ -X-GS ($X \in \{\mathbf{wO}, \mathbf{O}\}$) secure II given by the construction achieves the shortest ciphertexts and keys.⁵

Proof. Since it is straightforward that II achieves the shortest ciphertexts and keys if it is $(\leq n, \leq \omega)$ -X-GS ($X \in \{\mathbf{wO}, \mathbf{O}\}$) secure, we omit the proof.

(I)→(II). We omit the proof of (I) since it can be easily proved in a way similar to PS of the Fiat-Naor construction.

(II)→(III) and (III)→(IV). It is straightforward from Proposition 2.

(IV)→(I). We prove it by showing that the construction is $(\leq n, \leq \omega)$ -wO-GS secure only if $p = 1/2$.

⁵ It is already known that (the variant of) the Fiat-Naor construction is $(\leq n, \leq \omega)$ -PS secure and meets lower bounds of sizes of ciphertexts and secret keys (Proposition 3 in Appendix A) with equalities [3, 11].

Suppose that Π meets wO-GS. As in Theorem 2, we fix some $\mathcal{P} \subset \mathcal{R}$ such that $|\mathcal{P}| = n - \omega$ and $\mathcal{W} = \mathcal{R} \setminus \mathcal{P}$ without loss of generality. Then, it holds

$$\begin{aligned}
& \text{wO-GS}(\Pi, \mathcal{P}, \mathcal{W}) \\
&= \sum_{dk_{\mathcal{W}} \in \mathcal{DK}_{\mathcal{W}}} P_{DK_{\mathcal{W}}}(dk_{\mathcal{W}}) \max_{c_{\mathcal{P}} \in \mathcal{C}_{\mathcal{P}}} \max_{m \in \mathcal{M}} P_{M|C_{\mathcal{P}}DK_{\mathcal{W}}}(m | c_{\mathcal{P}}, dk_{\mathcal{W}}) \\
&= \sum_{r_{\mathcal{W}(\omega)} \in \{0,1\}^{|\mathcal{W}(\omega)|}} P_{R_{\mathcal{W}(\omega)}}(r_{\mathcal{W}(\omega)}) \max_{c_{\mathcal{P}} \in \mathcal{C}_{\mathcal{P}}} \max_{m \in \mathcal{M}} \frac{P_{MC_{\mathcal{P}}R_{\mathcal{W}(\omega)}}(m, c_{\mathcal{P}}, r_{\mathcal{W}(\omega)})}{P_{C_{\mathcal{P}}R_{\mathcal{W}(\omega)}}(c_{\mathcal{P}}, r_{\mathcal{W}(\omega)})} \\
&= \sum_{r_{\mathcal{W}(\omega)} \in \{0,1\}^{|\mathcal{W}(\omega)|}} P_{R_{\mathcal{W}(\omega)}}(r_{\mathcal{W}(\omega)}) \max_{c_{\mathcal{P}} \in \mathcal{C}_{\mathcal{P}}} \max_{m \in \mathcal{M}} \frac{P_M(m)P_{C_{\mathcal{P}}|M}(c_{\mathcal{P}} | m)}{P_{C_{\mathcal{P}}}(c_{\mathcal{P}})} \quad (12) \\
&= \max_{c_{\mathcal{P}} \in \mathcal{C}_{\mathcal{P}}} \max_{m \in \mathcal{M}} \frac{P_M(m)P_{C_{\mathcal{P}}|M}(c_{\mathcal{P}} | m)}{P_{C_{\mathcal{P}}}(c_{\mathcal{P}})} \quad (13) \\
&= \frac{pq}{pq + (1-p)(1-q)},
\end{aligned}$$

where $r_{\mathcal{W}(\omega)}$ and $R_{\mathcal{W}(\omega)}$ are the same as the proof of Theorem 2, Eq. (12) follows from that $r_{\mathcal{W}(\omega)}$ is independent of $(m, r_{\mathcal{W}})$, and Eq. (13) follows from $\sum P_{R_{\mathcal{W}(\omega)}}(r_{\mathcal{W}(\omega)}) = 1$.

Since Π meets wO-GS by the assumption, we have

$$\text{wO-GS}(\Pi, \mathcal{P}, \mathcal{W}) = \frac{pq}{pq + (1-p)(1-q)} = q = P_M(0) = \max_{m \in \mathcal{M}} P_M(m).$$

We

also have $\text{wO-GS}(\Pi) = \max_{\mathcal{P} \subset \mathcal{R}, \mathcal{W} \in \mathcal{Y}(\mathcal{P}, \omega)} \text{wO-GS}(\Pi, \mathcal{P}, \mathcal{W}) = pq/(pq + (1-p)(1-q))$. Since Π satisfies wO-GS, it holds

$$\text{wO-GS}(\Pi) = \frac{pq}{pq + (1-p)(1-q)} = q.$$

It means it holds $(2p-1)q(1-q) = 0$ for Π with wO-GS security. Since $1/2 \leq q < 1$, Π satisfies wO-GS only if $p = 1/2$. \square

We have considered that randomness $r_{\mathcal{W}}$ is independently chosen according to some biased probability distribution. However, there might be a situation where the probability distribution is changed as sampling elements (e.g., the probability distribution of $r_{\mathcal{W}_i}$ depends on whether $r_{\mathcal{W}_{i-1}} = 0$ or $r_{\mathcal{W}_{i-1}} = 1$). We can also obtain a similar result with various biased randomness. Namely, we can prove the security of (the variant of) the Fiat-Naor construction even if each randomness $r_{\mathcal{W}}$ in the construction is chosen according to different probability distribution. For details, see Appendix B.

6 Concluding Remarks

In this paper, we considered guessing secrecy for BE. We defined four notions of guessing secrecy, and derived lower bounds on ciphertexts and secret keys.

We showed that the Fiat-Naor construction is also secure in the sense of guessing secrecy, and achieves the shortest ciphertexts and keys.

The derived lower bounds of secret keys and the construction are only for BE schemes when the plaintext size is equal to the ciphertext size. Namely, analyzing BE schemes that meet guessing secrecy with more general ciphertext sizes is an open problem. More broadly, there is an interesting open problem in this research topic: Providing complete solution to trade-offs between sizes of ciphertexts and secret keys in BE schemes even in the perfect security setting.

Acknowledgments. We would like to thank the anonymous reviewers for fruitful comments. We would also like to thank Junji Shikata for his feedback. The author is supported by JSPS Research Fellowship for Young Scientists. This work was supported by Grant-in-Aid for JSPS Fellows Grant Number JP16J10532 and JP17H01752.

Appendix

A Lower Bounds for Perfectly Secure BE Schemes

Previous works [3, 16, 24] derived lower bounds on sizes of ciphertexts and secret keys required for perfectly secure BE schemes in various contexts. We here describe the bounds from [24] since it explicitly showed the lower bound on the encryption-key size.

Proposition 3 ([24]). *Let Π be an $(\leq n, \leq \omega)$ -PS secure BE scheme. Then, it holds that for any $\mathcal{P} \subset \mathcal{R}$,*

$$H(C_{\mathcal{P}}) \geq H(M).$$

Moreover, if $H(C_{\mathcal{P}}) = H(M)$ for any $\mathcal{P} \subset \mathcal{R}$, it then holds that

$$H(EK) \geq \sum_{j=0}^{\omega} \binom{n}{j} H(M),$$

$$H(DK_i) \geq \sum_{j=0}^{\omega} \binom{n-1}{j} H(M) \text{ for every } i \in [n].$$

B The Fiat-Naor Construction with Various Biased Randomness

We consider a more complicated situation than the construction in Sect. 5. Suppose that $\mathcal{M} = \mathcal{C} = \{0, 1\}$, and $P_M(0) = q$. We assume biased random sources $R_{\mathcal{W}}$ which take values in $\{0, 1\}$ for any $\mathcal{W} \in \mathscr{W}(\omega)$. We also assume $P_{R_{\mathcal{W}}}(0) = p_{\mathcal{W}}$ for all $\mathcal{W} \in \mathscr{W}(\mathcal{P}, \omega)$.

We assume a biased random source $R_{\mathcal{W}}$ which takes values in $\{0, 1\}$ such that $P_{R_{\mathcal{W}}}(0) = p$ for any $\mathcal{W} \in \mathscr{W}(\omega)$. Without loss of generality, we assume $1/2 \leq q < 1$ and $1/2 \leq p_{\mathcal{W}} < 1$ for any $\mathcal{W} \in \mathscr{W}(\mathcal{P}, \omega)$.

Note that the construction is the same as the previous one (i.e., the modified Fiat-Naor construction). We then have the following theorem.

Theorem 4. *A BE scheme Π given by the modified Fiat-Naor construction is $(\leq n, \leq \omega)$ -A-GS secure and achieves the shortest ciphertexts and keys if and only if $\max\{p_{\mathcal{W}}\}_{\mathcal{W} \in \mathscr{W}(\mathcal{P}, \omega)} \leq q$.*

Proof (Sketch). As in the proof of Theorem 2, we fix some $\mathcal{P} \subset \mathcal{R}$ such that $|\mathcal{P}| = n - \omega$ and $\mathcal{W} = \mathcal{R} \setminus \mathcal{P}$.

Then, we have

$$\begin{aligned}
 & \text{A-GS}(\Pi, \mathcal{P}, \mathcal{W}) \\
 &= \sum_{dk_{\mathcal{W}} \in \mathcal{DK}_{\mathcal{W}}} \sum_{c_{\mathcal{P}} \in \mathcal{C}_{\mathcal{P}}} \max_{m \in \mathcal{M}} P_{MC_{\mathcal{P}}DK_{\mathcal{W}}}(m, c_{\mathcal{P}}, dk_{\mathcal{W}}) \\
 &= \sum_{r_{\mathscr{W}(\mathcal{W})} \in \{0,1\}^{|\mathscr{W}(\mathcal{W})|}} \left(\max_{m \in \mathcal{M}} P_{MC_{\mathcal{P}}R_{\mathscr{W}(\mathcal{W})}}(m, 0, r_{\mathscr{W}(\mathcal{W})}) \right. \\
 & \qquad \qquad \qquad \left. + \max_{m \in \mathcal{M}} P_{MC_{\mathcal{P}}R_{\mathscr{W}(\mathcal{W})}}(m, 1, r_{\mathscr{W}(\mathcal{W})}) \right) \\
 &= \sum_{r_{\mathscr{W}(\mathcal{W})} \in \{0,1\}^{|\mathscr{W}(\mathcal{W})|}} P_{R_{\mathscr{W}(\mathcal{W})}}(r_{\mathscr{W}(\mathcal{W})}) \\
 & \qquad \qquad \qquad \left(\max_{m \in \mathcal{M}} P_{MC_{\mathcal{P}}}(m, 0) + \max_{m \in \mathcal{M}} P_{MC_{\mathcal{P}}}(m, 1) \right) \quad (14) \\
 &= \max_{m \in \mathcal{M}} P_{MC_{\mathcal{P}}}(m, 0) + \max_{m \in \mathcal{M}} P_{MC_{\mathcal{P}}}(m, 1),
 \end{aligned}$$

where $\mathscr{W}(\mathcal{W})$, $r_{\mathscr{W}(\mathcal{W})}$, and $R_{\mathscr{W}(\mathcal{W})}$ are the same as those in Theorem 2, and Eq. (14) follows from $r_{\mathscr{W}(\mathcal{W})}$ is independent of $(m, r_{\mathcal{W}})$. Since it holds

$$\begin{aligned}
 P_{MC_{\mathcal{P}}}(0, 0) &= p_{\mathcal{W}}q, & P_{MC_{\mathcal{P}}}(1, 0) &= p_{\mathcal{W}}(1 - q), \\
 P_{MC_{\mathcal{P}}}(0, 1) &= (1 - p_{\mathcal{W}})q, & P_{MC_{\mathcal{P}}}(1, 1) &= (1 - p_{\mathcal{W}})(1 - q),
 \end{aligned}$$

we have $\text{A-GS}(\Pi, \mathcal{P}, \mathcal{W}) = p_{\mathcal{W}}q + \max\{p_{\mathcal{W}}(1 - q), (1 - p_{\mathcal{W}})q\}$. If $p_{\mathcal{W}} \leq q$, then we have $\text{A-GS}(\Pi, \mathcal{P}, \mathcal{W}) = q = \max_{m \in \mathcal{M}} P_M(m)$. Otherwise, we have $\text{A-GS}(\Pi, \mathcal{P}, \mathcal{W}) = p_{\mathcal{W}} > q = \max_{m \in \mathcal{M}} P_M(m)$.

Therefore, it holds $\text{A-GS}(\Pi) = \max_{m \in \mathcal{M}} P_M(m)$ if $\max\{p_{\mathcal{W}}\}_{\mathcal{W} \in \mathscr{W}(\mathcal{P}, \omega)} \leq q$. \square

References

1. Alimomeni, M., Safavi-Naini, R.: Guessing Secrety. In: Smith, A. (ed.) ICITS 2012. LNCS, vol. 7412, pp. 1–13. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-32284-6_1
2. Berkovits, S.: How to broadcast a secret. In: Davies, D.W. (ed.) EUROCRYPT 1991. LNCS, vol. 547, pp. 535–541. Springer, Heidelberg (1991). https://doi.org/10.1007/3-540-46416-6_50
3. Blundo, C., Cresti, A.: Space requirements for broadcast encryption. In: De Santis, A. (ed.) EUROCRYPT 1994. LNCS, vol. 950, pp. 287–298. Springer, Heidelberg (1995). <https://doi.org/10.1007/BFb0053444>

4. Blundo, C., Mattos, L.A.F., Stinson, D.R.: Trade-offs between communication and storage in unconditionally secure schemes for broadcast encryption and interactive key distribution. In: Koblitz, N. (ed.) CRYPTO 1996. LNCS, vol. 1109, pp. 387–400. Springer, Heidelberg (1996). https://doi.org/10.1007/3-540-68697-5_29
5. Boneh, D., Gentry, C., Waters, B.: Collusion resistant broadcast encryption with short ciphertexts and private keys. In: Shoup, V. (ed.) CRYPTO 2005. LNCS, vol. 3621, pp. 258–275. Springer, Heidelberg (2005). https://doi.org/10.1007/11535218_16
6. Chen, H., Ling, S., Padró, C., Wang, H., Xing, C.: Key predistribution schemes and one-time broadcast encryption schemes from algebraic geometry codes. In: Parker, M.G. (ed.) IMACC 2009. LNCS, vol. 5921, pp. 263–277. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-10868-6_16
7. Cover, T.M., Thomas, J.A.: Elements of Information Theory. Wiley-Interscience, 2nd edn. July 2006
8. Csiszár, I., Koerner, J.: Information Theory: Coding Theorems for Discrete Memoryless Systems, 2nd edn. Cambridge University Press, Cambridge (2011)
9. Dodis, Y., Fazio, N.: Public key broadcast encryption for stateless receivers. In: Feigenbaum, J. (ed.) DRM 2002. LNCS, vol. 2696, pp. 61–80. Springer, Heidelberg (2003). https://doi.org/10.1007/978-3-540-44993-5_5
10. Dodis, Y., Smith, A.: Entropic security and the encryption of high entropy messages. In: Kilian, J. (ed.) TCC 2005. LNCS, vol. 3378, pp. 556–577. Springer, Heidelberg (2005). https://doi.org/10.1007/978-3-540-30576-7_30
11. Fiat, A., Naor, M.: Broadcast encryption. In: Stinson, D.R. (ed.) CRYPTO 1993. LNCS, vol. 773, pp. 480–491. Springer, Heidelberg (1994). https://doi.org/10.1007/3-540-48329-2_40
12. Garay, J.A., Staddon, J., Wool, A.: Long-lived broadcast encryption. In: Bellare, M. (ed.) CRYPTO 2000. LNCS, vol. 1880, pp. 333–352. Springer, Heidelberg (2000). https://doi.org/10.1007/3-540-44598-6_21
13. Gentry, C., Waters, B.: Adaptive security in broadcast encryption systems (with Short Ciphertexts). In: Joux, A. (ed.) EUROCRYPT 2009. LNCS, vol. 5479, pp. 171–188. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-01001-9_10
14. Iwamoto, M., Shikata, J.: Constructions of symmetric-key encryption with guessing secrecy. In: IEEE International Symposium on Information Theory 2015, pp. 725–729, June 2015
15. Iwamoto, M., Shikata, J.: Information theoretic security for encryption based on conditional rényi entropies. In: Padró, C. (ed.) ICITS 2013. LNCS, vol. 8317, pp. 103–121. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-04268-8_7
16. Kurosawa, K., Yoshida, T., Desmedt, Y., Burmester, M.: Some bounds and a construction for secure broadcast encryption. In: Ohta, K., Pei, D. (eds.) ASIACRYPT 1998. LNCS, vol. 1514, pp. 420–433. Springer, Heidelberg (1998). https://doi.org/10.1007/3-540-49649-1_33
17. Luby, M., Staddon, J.: Combinatorial bounds for broadcast encryption. In: Nyberg, K. (ed.) EUROCRYPT 1998. LNCS, vol. 1403, pp. 512–526. Springer, Heidelberg (1998). <https://doi.org/10.1007/BFb0054150>
18. Naor, D., Naor, M., Lotspiech, J.: Revocation and tracing schemes for stateless receivers. In: Kilian, J. (ed.) CRYPTO 2001. LNCS, vol. 2139, pp. 41–62. Springer, Heidelberg (2001). https://doi.org/10.1007/3-540-44647-8_3
19. Padró, C., Gracia, I., Martín, S.: Improving the trade-off between storage and communication in broadcast encryption schemes. Discrete Appl. Math. **143**(1–3), 213–220 (2004)

20. Padró, C., Gracia, I., Martín, S., Morillo, P.: Linear broadcast encryption schemes. *Discrete Appl. Math.* **128**(1), 223–238 (2003)
21. Phan, D.H., Pointcheval, D., Streffer, M.: Security notions for broadcast encryption. In: Lopez, J., Tsudik, G. (eds.) *ACNS 2011*. LNCS, vol. 6715, pp. 377–394. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-21554-4_22
22. Russell, A., Wang, H.: How to fool an unbounded adversary with a short key. In: Knudsen, L.R. (ed.) *EUROCRYPT 2002*. LNCS, vol. 2332, pp. 133–148. Springer, Heidelberg (2002). https://doi.org/10.1007/3-540-46035-7_9
23. Shannon, C.E.: Communication theory of secrecy systems. *Bell Syst. Tech. J.* **28**, 656–715 (1949)
24. Watanabe, Y., Hanaoka, G., Shikata, J.: Unconditionally secure revocable storage: tight bounds, optimal construction, and robustness. In: Nascimento, A.C.A., Barreto, P. (eds.) *ICITS 2016*. LNCS, vol. 10015, pp. 213–237. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-49175-2_11
25. Watanabe, Y., Shikata, J.: Unconditionally secure broadcast encryption schemes with trade-offs between communication and storage. *IEICE Trans.* **99–A**(6), 1097–1106 (2016)
26. Yamamoto, H.: Rate-distortion theory for the shannon cipher system. *IEEE Trans. Inf. Theor.* **43**(3), 827–835 (1997)

Contrast Optimal XOR Based Visual Cryptographic Schemes

Sabyasachi Dutta¹(✉) and Avishek Adhikari²

¹ R.C. Bose Centre for Cryptology and Security, Indian Statistical Institute,
203 B.T Road, Kolkata 700 108, India
saby.math@gmail.com

² Department of Pure Mathematics, Ballygunge Science College,
University of Calcutta, 35 Ballygunge Circular Road, Kolkata 700 019, India
avishek.adh@gmail.com

Abstract. OR-based Visual Cryptographic Schemes (OVCS) suffer from poor visual quality of the reconstructed image. XOR-based visual secret sharing (XVCS) can be thought of as an alternative where the relative contrast of the reconstructed image is much better. Moreover, it is possible to achieve optimum relative contrast equal to 1 in XVCS which is an impossibility in case of OVCS. Although there are examples of XVCSs where optimum relative contrast is achieved but to the best of our knowledge, this is the first theoretical work to find a necessary and sufficient condition for a XOR-based VCS to achieve optimum relative contrast equal to 1 in terms of the underlying access structure.

Keywords: Cumulative array · Relative contrast
Equivalent participants · Essential participants
Maximal forbidden sets · Visual secret sharing scheme

1 Introduction

A traditional Visual Cryptographic Scheme (VCS) for a set of n participants $\mathcal{P} = \{1, 2, \dots, n\}$ is a variant of secret sharing, that encodes a secret image SI into n shares which are distributed by the dealer among n participants (also known as parties) in the form of transparencies on which the shares are photocopied. Such shares have the property that only “qualified” subsets of participants can visually recover the secret image by carefully stacking the transparencies. The first VCS was proposed by Naor and Shamir [20] where they considered the threshold access structure. This concept has been extended in [1, 3, 7, 8] to general access structures.

The mathematical operation that lies beneath the physical implementation of the above mentioned schemes is the Boolean operation “OR”. However the major problems for any OR-based visual cryptographic scheme are the huge share size (pixel expansion) and very poor contrast of the reconstructed image. Several

papers have been published to minimize the pixel expansion and to maximize contrast. One may refer to [2, 4, 5, 9, 10, 12, 16, 21] for a detailed survey.

Arumugam et al. [6] introduced a VCS for a special type of access structure lying in between the threshold access structure and general access structure in the OR-model. They called it $(k, n)^*$ -VCS, to address the scenario where one participant is “essential” and he needs the help of any $k - 1$ parties other than him, to recover the secret image. Guo et al. [14] forwarded this idea to the concept of $(k, n)^*$ -VCS with t essential participants who require the collaboration of $k - t$ more parties from the rest of the set of parties. Note that the case when $t=0$ we have the scenario of a (k, n) -VCS where no participant is essential. The case $t = 1$ is the usual $(k, n)^*$ -VCS while $t = n$ leads to the (n, n) -VCS.

1.1 “XOR” Based VCS: An Alternative for “OR” Based VCS

OR based visual cryptographic schemes suffer from the low quality of the reconstructed image. Tuyls et al. [22] gave a VCS based on polarization of light where the underlying mathematical operation was the Boolean “XOR” operation. The polarization of light is done by inserting a liquid crystal layer into a liquid crystal display (LCD). The advantage is two-fold. First, the liquid crystal layer can be driven in an LCD. Secondly, since the voltage applied to the liquid crystal layer makes it possible to rotate the polarization of light entering the layer over a certain angle, it facilitates a practical updating mechanism. Thus unlike OR-based schemes where a participant has to carry a number of transcripts to update the shares, in a XOR-based VCS a party has to carry just one dedicated trusted device that has a display. For recovering the secret image the shares i.e., the liquid crystal layers are to be stacked together. Moreover, due to the rapid advancement of technology these devices are getting cheaper. It is a reasonable expectation that polarization based visual cryptographic schemes will be implemented in every light-weight cryptographic situation. In [23] the authors constructed a XOR based (n, n) -VCS and proved that a XOR based $(2, n)$ -VCS is equivalent to a binary code. There are also two different methods to realize the XOR operation in the field of visual cryptography. First one uses a Mach-Zehnder Interferometer [17] and the other one proposed in [24] needs a copy machine with the reversing function. One for further studies, may refer to [15, 18, 19, 25]. All these papers have considered the common property of non-monotonicity of the access structure, i.e., super-set of the minimal qualified set may not get the secret back if all of them stack their shares. However, it does not prohibit us to define a visual cryptographic scheme. For most of the practical scenarios, the access structure is generally a public information. That is, the participants have complete knowledge of the qualified sets and forbidden sets. Therefore if a qualified set of participants come together then any minimal qualified subset of it may produce the corresponding shares to reconstruct the secret image. Thus it is sufficient to restrict ourselves to the collection of all minimal qualified sets corresponding to the access structure. The first XOR-based VCS for general access structure was proposed by Liu et al. [19]. They repeatedly used the share generation algorithm for a $(2, 2)$ -VCS to generate the shares of

the participants for any access structure. However, their construction method is not via basis matrices. Moreover, their construction deviates from the traditional VCS in the sense that,

1. the participants may have to carry multiple share images;
2. due to the presence of multiple shares, at the time of revealing the secret, the participants have to know for which access structures they are going to submit which of their shares.

However their construction technique is novel and to the best of our knowledge there does not exist any other construction method other than [19] that constructs standard visual cryptographic scheme for the general access structure in the XOR model. So researches towards finding XOR-based VCS without these assumptions are important. Dutta et al. [11] gave an efficient technique to construct XOR-based $(k, n)^*$ -VCS with t essential parties. Their linear algebraic technique can further be exploited to construct XOR-based VCS for general access structures. Yang et al. [25] provided plethora of examples of basis matrices by proving that basis matrices for OR-based (k, n) -VCS can be used as basis matrices for XOR-based (k, n) -VCS. Fu et al. [13] theoretically proved a necessary condition for the optimality of pixel expansion of any visual cryptographic scheme in both OR and XOR models. They gave an algorithm for reducing the pixel expansion of any scheme. However, their findings are based on the existence of basis matrices realizing an access structure. They have not however, given any construction method to produce the basis matrices or distribution matrices capturing the access structure in the first place. Their algorithm is novel modulo the existence of the basis matrices.

1.2 Our Contribution

In the OR-based VCS (OVCS) the relative contrast is always less than $\frac{1}{2}$. Moreover, it follows from [20] that for any access structure if there is a minimal qualified set of size t then relative contrast can never be better than $\frac{1}{2^{t-1}}$. This is true for whatever construction method we adopt to realize OVCS, as long as it is deterministic. On the other hand, for XOR-based VCS (XVCS) there is a possibility of achieving optimal relative contrast = 1. With the help of combinatorial design *cumulative array* we show that if a given access structure is “OPTIMAL” then it is possible to construct XVCS with relative contrast = 1. We explicitly describe the construction method and prove the correctness of the construction. We further prove that if an access structure does not satisfy the *optimality* condition then there cannot be any construction method realizing XVCS on the access structure achieving relative contrast equal to one. There were examples of XVCS which showed optimum relative contrast is achievable but to the best of our knowledge, this is the first theoretical work to answer the question of achievability of optimum relative contrast in terms of the underlying access structure.

2 Prerequisites

2.1 The Model for Non-monotone XOR-VCS

We follow standard notations and symbols through out. Let $\mathcal{P} = \{1, 2, 3, \dots, n\}$ denote a set of participants. Let $2^{\mathcal{P}}$ denote the set of all subsets of \mathcal{P} . Let $\mathcal{Q} \subseteq 2^{\mathcal{P}}$ and $\mathcal{F} \subseteq 2^{\mathcal{P}}$, where $\mathcal{Q} \cap \mathcal{F} = \emptyset$, respectively denote the set of all qualified sets and the set of all forbidden sets. The pair $(\mathcal{Q}, \mathcal{F})$ constitutes an access structure on \mathcal{P} . In this paper, we consider $\mathcal{Q} = \mathcal{Q}_{min} = \{X \subseteq \mathcal{P} : B \in \mathcal{F} \forall B \subset X\}$, the collection of all minimal qualified sets of participants.

The collection of all maximal forbidden sets is denoted by $\mathcal{F}_{max} = \{F \in \mathcal{F} : \exists B \in \mathcal{Q}_{min} \ \& \ B \subset F \cup \{i\} \ \forall i \in \mathcal{P} - F\}$. Note that in this paper, we do not care about any subset $Y \in 2^{\mathcal{P}}$ such that $X \subset Y$, for some $X \in \mathcal{Q}_{min}$. We are interested only in the fact that the minimal qualified sets of parties can recover the secret image and the forbidden sets can not. This makes the access structure non-monotone. In this paper whenever we consider an access structure, it is implicit that we are interested in \mathcal{Q}_{min} and \mathcal{F}_{max} .

Example 1. Let $\mathcal{P} = \{1, 2, \dots, 6\}$ and let \mathcal{Q}_{min} consist of the following minimal qualified subsets of participants $B_1 = \{1, 2, 3\}$, $B_2 = \{1, 2, 5\}$, $B_3 = \{1, 3, 4\}$, $B_4 = \{1, 4, 5\}$, $B_5 = \{1, 5, 6\}$. Note that $\{1, 2, 4\}$ and $\{2, 3, 4, 5, 6\}$ are members of both \mathcal{F} and \mathcal{F}_{max} while $\{2, 4, 5, 6\}$ is a member of \mathcal{F} but not a member of \mathcal{F}_{max} .

Notations: Let S be an $n \times m$ Boolean matrix and $X \subseteq \mathcal{P} = \{1, 2, \dots, n\}$. By $S[X]$ we denote the matrix obtained by restricting the rows of S to the indices belonging to X . Further, for any $X \subseteq \mathcal{P}$ the vector obtained by applying the boolean ‘‘XOR’’ operation to the rows of $S[X]$ is denoted by S_X . The Hamming weight of the row vector which represents the number of ones in the vector (S_X) is denoted by $w(S_X)$, if the context is clear. Other short hand notations and abbreviations used are given below:

- VCS \longrightarrow visual cryptographic scheme.
- OVCS \longrightarrow OR-based visual cryptographic scheme.
- XVCS \longrightarrow XOR-based visual cryptographic scheme.
- CA \longrightarrow cumulative array.
- $(k, n)^*$ -VCS \longrightarrow (k, n) -threshold VCS with one fixed essential party who is present in every minimal qualified set along with any other $k - 1$ regular parties.
- t - $(k, n)^*$ -VCS \longrightarrow (k, n) -threshold VCS with t many fixed essential parties who are present in every minimal qualified set along with any other $k - t$ regular parties.
- $\mathbf{0}$ \longrightarrow bold-case 0 denotes the zero-vector.
- $\mathbf{1}$ \longrightarrow bold-case 1 denotes the vector with all entries equal to 1.
- Contrast-optimal XVCS \longrightarrow XVCS with relative contrast equal to 1 for every minimal qualified set.

We are now in a position to give definition of a Gen-NM-XVCS. Here, “NM” stands for non-monotone while X stands for XOR.

Definition 1. Let $\mathcal{P} = \{1, 2, 3, \dots, n\}$ be a set of participants. A Gen-NM-XVCS on \mathcal{P} is a visual cryptographic scheme such that the following two conditions hold:

1. Any minimal qualified set of participants can recover the secret image.
2. Any maximal forbidden set of participants does not have any information about the secret image.

Any visual cryptographic scheme can be implemented by means of distribution matrices. To be more specific, let n and m be two integers, where n represents the number of parties and m the pixel expansion, i.e., the parameter that specifies how many sub-pixels are needed in each share to encode a single pixel of the secret image. A scheme is usually defined by two collections of Boolean matrices.

Definition 2. (via Collection of Matrices)

Let $\mathcal{P} = \{1, 2, 3, \dots, n\}$ be a set of participants. Let $(\mathcal{Q}_{min}, \mathcal{F}_{max})$ be the access structure defined on \mathcal{P} . Let m and $\{h_X\}_{X \in \mathcal{Q}_{min}}$ be non-negative integers satisfying $1 \leq h_X \leq m$. Two collections of $n \times m$ binary matrices \mathcal{C}_0 and \mathcal{C}_1 realize a $(\mathcal{Q}_{min}, \mathcal{F}_{max})$ -NM-XVCS, if there exists $\{\alpha_X > 0 : X \in \mathcal{Q}_{min}\}$ such that

1. For any $S \in \mathcal{C}_0$, the XOR operation of the rows of $S[X]$ for any minimal qualified set X results in a vector v_0 satisfying $w(v_0) \leq h_X - \alpha_X \cdot m$.
2. For any $T \in \mathcal{C}_1$, the XOR operation of the rows of $T[X]$ for any minimal qualified set X results in a vector v_1 satisfying $w(v_1) \geq h_X$.
3. Any forbidden set $Y \in \mathcal{F}_{max}$ has no information on the shared image. Formally, the two collections of $|Y| \times m$ matrices \mathcal{D}_t , with $t \in \{0, 1\}$, obtained by restricting each $n \times m$ matrix in \mathcal{C}_t to rows indexed by Y are indistinguishable in the sense that they contain the same matrices with the same frequencies.

The symbols α_X and $\alpha_X \cdot m$ respectively denote the relative contrast and contrast of the recovered image reconstructed by the minimal qualified set X . We are considering only “Black and White” images in this paper. A white pixel is identified as 0 while a black pixel is identified as 1.

During share generation phase the dealer chooses randomly a matrix from \mathcal{C}_b , if the secret pixel is $b \in \{0, 1\}$, and gives the participant P_i the i -th row as the participant’s share for all i . When the dealer wants to share a black and white secret image then for each constituent pixel he repeatedly performs the above process till all the pixels are shared. Note that the dealer has to store huge collections of matrices \mathcal{C}_0 and \mathcal{C}_1 to share an image. To reduce the storage space, a Gen-NM-XVCS may also be modelled by introducing the concept of basis matrices. The formal definition is as follows:

Definition 3 (via Basis Matrices). A $(\mathcal{Q}_{min}, \mathcal{F}_{max})$ -NM-XVCS is realized using two $n \times m$ binary matrices S^0 and S^1 called basis matrices, if there exist two sets of non-negative real numbers $\{\alpha_X\}_{X \in \mathcal{Q}_{min}}$ and $\{t_X\}_{X \in \mathcal{Q}_{min}}$ such that the following two conditions hold:

1. (contrast condition) If $X \in \mathcal{Q}_{min}$, then S_X^0 , the “XOR” of the rows indexed by X of S^0 , satisfies $w(S_X^0) \leq t_X - \alpha_X \cdot m$; whereas, for S^1 it results in $w(S_X^1) \geq t_X$.
2. (security condition) If $Y = \{i_1, i_2, \dots, i_s\} \in \mathcal{F}$ then the two $s \times m$ matrices $S^0[Y]$ and $S^1[Y]$ obtained by restricting S^0 and S^1 respectively to rows i_1, i_2, \dots, i_s are identical up to a column permutation.

For any minimal qualified set $X \in \mathcal{Q}_{min}$, the relative contrast of the reconstructed image is given by $\alpha_X = \frac{w(S_X^1) - w(S_X^0)}{m}$, where m is the pixel expansion of the scheme. If the secret pixel is $b \in \{0, 1\}$ then the dealer gives a random permutation to the columns of S^b and distributes the rows of the resulting matrix as shares to the parties. Similarly if the secret pixel is black then the dealer repeats the same process with the matrix S^1 . The collections of matrices \mathcal{C}_0 and \mathcal{C}_1 , that one requires to realize a VCS may be thought of as the collection of all possible matrices obtained by giving all possible column permutations to the basis matrices S^0 and S^1 respectively. As a result, the dealer has to store only the two matrices S^0 and S^1 , making the scheme efficient space-wise.

2.2 Equivalent Parties and Simplification of Access Structures

We now discuss a technique which simplifies and reduces a class of more complex access structures into a simpler one. For that we need to first define the notion of *equivalent participants*. In words, equivalent parties are the parties who enjoy the same rights and hence they can be given identical shares without hampering the access structure of a secret sharing scheme. Hence given an access structure if we can identify the equivalent parties then they can be given the same shares and the access structure reduces to a much simpler one. One can treat the reduced access structure (which is simpler than the original one) as the given access structure and build schemes keeping in mind that ultimately while distributing the shares the equivalent parties receive the same shares. We start with the formal definition of equivalent participants.

Definition 4 (adapted from [19]). Let \mathcal{Q}_{min} and \mathcal{F}_{max} denote the collections of minimal qualified sets and maximal forbidden sets respectively on a set of parties $\mathcal{P} = \{1, 2, \dots, n\}$. If parties i and j satisfy that, for all $F \in \mathcal{F}_{max}$, $i \in F$ if and only if $j \in F$, then the parties i and j are called *equivalent participants* for the access structure.

Example 2. Let us consider 3-(4, 6)*-XVCS on the set of participants $\mathcal{P} = \{1, 2, \dots, 6\}$, where the first three parties are essential in the sense that they are present in each of the minimal qualified sets. However, they need the share of one more party to reconstruct the secret image. Here, $\mathcal{Q}_{min} = \{\{1, 2, 3, 4\}, \{1, 2, 3, 5\}, \{1, 2, 3, 6\}\}$ and $\mathcal{F}_{max} = \{\{1, 2, 3\}, \{1, 2, 4, 5, 6\}, \{1, 3, 4, 5, 6\}, \{2, 3, 4, 5, 6\}\}$. In this access structure 4, 5, 6 are equivalent parties.

It is easy to see that the relation ‘ \sim' defined on \mathcal{P} by $i \sim j$ if and only if i and j are equivalent parties, is an equivalence relation on \mathcal{P} . Thus, the set \mathcal{P} is partitioned into equivalence classes. An equivalence class of $i \in \mathcal{P}$ is the set $[i] = \{j \in \mathcal{P} : i \sim j\}$. For the sake of better representation we will denote the equivalence class $[p]$ by \tilde{p} . We now give the definition of the simplified access structure $\tilde{\mathcal{Q}}_{min}$ derived from \mathcal{Q}_{min} .

Definition 5 (adopted from [19]). Let $(\mathcal{Q}_{min}, \mathcal{F}_{max})$ be a given access structure on a set \mathcal{P} of parties. Let $\tilde{\mathcal{P}} = \{\tilde{p} : p \in \mathcal{P}\}$. We choose one single representative from an equivalence class that is, one party represents an equivalence class. We say $\tilde{\mathcal{Q}}_{min} = \{\{\tilde{p} \in \tilde{\mathcal{P}} : p \in B\} : B \in \mathcal{Q}_{min}\}$ as the simplified access structure of the given access structure. If $\tilde{\mathcal{Q}}_{min} = \mathcal{Q}_{min}$ then the access structure is called the most simplified access structure.

Remark 1. For $2 \leq k \leq n$, the threshold access structure corresponding to (k, n) -XVCS are already in the most simplified form. In other words, no two parties are equivalent.

Example 3. Continuing from Example 2, we see that $\tilde{\mathcal{P}} = \{1, 2, 3, \tilde{4}\}$ and $\tilde{\mathcal{Q}}_{min} = \{\{1, 2, 3, \tilde{4}\}\}$, where $\tilde{4} = [4] = \{4, 5, 6\}$. This $\tilde{\mathcal{Q}}_{min}$ is the most simplified form of the given access structure.

2.3 Cumulative Array for an Access Structure

So far we have seen a way to simplify certain access structures via the concept of equivalent parties. Let $(\mathcal{Q}_{min}, \mathcal{F}_{max})$ be the given access structure and suppose that the access structure be already in its *most simplified form*. Let $\mathcal{F}_{max} = \{F_1, F_2, \dots, F_t\}$. Let us now recall the idea of cumulative array (see [7]) for \mathcal{Q}_{min} . The cumulative array (CA) is an $n \times t$ Boolean matrix such that $CA(i, j) = 1$ if and only if $i \notin F_j$ where n is the number of participants.

Example 4. The cumulative array for 3-(4, 6)*-access structure from Example 2 is given by

Parties	$F_1 = \{123\}$	$F_2 = \{12456\}$	$F_3 = \{13456\}$	$F_4 = \{23456\}$
1	0	0	0	1
2	0	0	1	0
3	0	1	0	0
4	1	0	0	0
5	1	0	0	0
6	1	0	0	0

where $\{123\}$ means the set $\{1, 2, 3\}$, $\{12456\}$ means the set $\{1, 2, 4, 5, 6\}$ etc. We will sometimes denote a set in this form for brevity, when there is no scope for confusion.

It is not very hard to see the following necessary and sufficient condition for checking whether two participants i and j are equivalent or not using cumulative array.

Proposition 1. *Two parties i and j are equivalent if and only if i th row and j th row of the corresponding cumulative array are identical.*

Thus once an access structure is reduced, via equivalent parties, to its most simplified form then the associated cumulative array does not contain two identical rows.

Observation. An access structure is in its “most simplified form” if the corresponding CA has no identical rows. For example, the CA for a (k, n) -threshold VCS with $1 < k \leq n$ has no identical rows and hence is in most simplified form.

Observation. Every row of an CA contains at least one 1, otherwise the party (indexing the row) belongs to every maximal forbidden set and hence in no minimal qualified set. So the party can be deleted from the access structure. Moreover, if $B \in \mathcal{Q}_{min}$ then $CA[B]$ contains at least one 1 in every column, otherwise B becomes a subset of some maximal forbidden set and hence not qualified to recover the secret image.

3 Main Results

Definition 6. *A cumulative array for \mathcal{Q}_{min} (which is in its most simplified form) is called **OPTIMAL** if it satisfies the following property: for each minimal qualified set $\{i_1, i_2, \dots, i_k\}$, every column of the restricted array $CA[\{i_1, i_2, \dots, i_k\}]$ contains only odd many 1’s.*

For example, the cumulative array for an (n, n) -threshold VCS is *OPTIMAL* whereas for a (k, n) -threshold VCS with $k < n$, it is not *OPTIMAL*.

We now present an easy lemma to show the existence of a XOR-based VCS which achieves optimal relative contrast 1.

Lemma 1. *Let (S^0, S^1) be the basis matrices constructed by the method of Naor-Shamir as in [20] to realize an (n, n) -OVCS. Then (S^0, S^1) also realizes (n, n) -XVCS having pixel expansion 2^{n-1} and optimal relative contrast 1.*

Proof. : We recall that S^0 consists of all possible even columns of length n while S^1 consists of all possible odd columns of same length, as given in [20]. It is easy to see that S^0 and S^1 can be used to distribute shares to the participants in the XOR model. The only qualified set of parties is \mathcal{P} itself. Also, $w(XOR(S^1_{\mathcal{P}})) = 2^{n-1}$ and $w(XOR(S^0_{\mathcal{P}})) = 0$. It follows that the “contrast condition” of Definition 3 is satisfied. The “security condition” for both OR based and XOR based models is the same. Hence we have the result. We further observe that the Naor-Shamir construction admits pixel expansion equal to 2^{n-1} and the relative contrast is 1 which is maximum in XOR-based VCS.

Construction 1. *Let us consider an access structure, in its most simplified form, $(\mathcal{Q}_{min}, \mathcal{F}_{max})$ on a set of n parties \mathcal{P} . Let the corresponding CA be *OPTIMAL*. Let $\mathcal{F}_{max} = \{F_1, F_2, \dots, F_k\}$ i.e. we have k many maximal forbidden sets.*

Let $W_{(k,k)}$ and $B_{(k,k)}$ respectively denote the Naor-Shamir [20] white basis matrix and black basis matrix corresponding to a (k, k) -threshold access structure. Each of $W_{(k,k)}$ and $B_{(k,k)}$ is of size $k \times 2^{k-1}$, where the rows are indexed by the maximal forbidden sets $\{F_1, F_2, \dots, F_k\}$ with respect to which the CA is constructed.

Let us write $W_{(k,k)} = \begin{bmatrix} \dots & R_1^0 & \dots \\ \dots & R_2^0 & \dots \\ \dots & R_k^0 & \dots \end{bmatrix}$ and $B_{(k,k)} = \begin{bmatrix} \dots & R_1^1 & \dots \\ \dots & R_2^1 & \dots \\ \dots & R_k^1 & \dots \end{bmatrix}$, where R_i^0 denotes the i th row of $W_{(k,k)}$ and R_i^1 denotes the i th row of $B_{(k,k)}$. Notice that the rows are the shares of the participants.

Now construct S^0 (white basis matrix) and S^1 (black basis matrix) realizing the given access structure $(\mathcal{Q}_{min}, \mathcal{F}_{max})$ as follows:

The i -th row of $S^0 = \text{XOR}$ of those rows in $W_{(k,k)}$ for which $i \notin F_j = R_{j_1}^0 \oplus R_{j_2}^0 \oplus \dots \oplus R_{j_s}^0$,

where $R_{j_\alpha}^0$ s are those rows of $W_{(k,k)}$ such that $i \notin F_{j_1} \cup F_{j_2} \cup \dots \cup F_{j_s}$.

The i -th row of $S^1 = \text{XOR}$ of those rows in $B_{(k,k)}$ for which $i \notin F_j = R_{j_1}^1 \oplus R_{j_2}^1 \oplus \dots \oplus R_{j_s}^1$,

where $R_{j_\alpha}^1$ s are those rows of $B_{(k,k)}$ such that $i \notin F_{j_1} \cup F_{j_2} \cup \dots \cup F_{j_s}$.

Proposition 2. *Let $(\mathcal{Q}_{min}, \mathcal{F}_{max})$ be a non-monotone access structure (in its most simplified form) with OPTIMAL CA. The matrices (S^0, S^1) constructed in the above manner are indeed the basis matrices realizing XVCS on the given access structure. Moreover, maximum relative contrast equal to 1 is attained through this construction.*

Remark 2. Before proving the proposition, we point out that if the CA of an access structure (after reducing it to its most simplified form) is not Optimal then the above construction method does not admit basis matrices for XVCS. For example, let us consider the $(2, 3)$ -threshold access structure. The CA for the access structure is constructed with the help of $\mathcal{F}_{max} = \{\{1\}, \{2\}, \{3\}\}$ and is given by

Parties	$F_1 = \{1\}$	$F_2 = \{2\}$	$F_3 = \{3\}$
1	0	1	1
2	1	0	1
3	1	1	0

From the CA it is clear that the access structure is already in its most simplified form. Moreover, the CA is not Optimal as $CA[\{1, 2\}] = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix}$, which is the restriction of the CA to rows indexed by the minimal qualified set $\{1, 2\}$ contains an even column. The last column of this restricted CA is indexed by the maximal forbidden set $\{3\}$. Since the number of maximal forbidden sets is 3, we consider the Naor-Shamir basis matrices for $(3, 3)$ -VCS,

$W_{(3,3)} = \begin{bmatrix} 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix}$ and $B_{(3,3)} = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 \end{bmatrix}$. So by the above construction method we compute

$S^0 = \begin{bmatrix} 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix}$ and $S^1 = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix}$. Now it is easy to see that the contrast condition is not holding for any minimal qualified set.

Proof of Proposition 2: Let $|\mathcal{F}_{max}| = t$, i.e. the access structure in its most simplified form has t many maximal forbidden sets and also it is given that the corresponding CA is optimal. We observe that if X is a minimal qualified set and since the CA is OPTIMAL then every column of the restricted CA, viz. $CA[X]$ contains odd number of 1s. The contrast condition holds because XOR of the shares of the participants in X satisfy the following:
 XOR of the rows (in S^1) corresp. to the parties in $X = \text{XOR of all the rows in } B_{(t,t)} = (1, 1, \dots, 1) = 2^{t-1}$ tuple with all entries equal to 1

and

XOR of the rows (in S^0) corresp. to the parties in $X = \text{XOR of all the rows in } W_{(t,t)} = (0, 0, \dots, 0) = 2^{t-1}$ tuple with all entries equal to 0.

Hence,

wt.(XOR of the rows (in S^1) corresp. to the parties in X) – wt.(XOR of the rows (in S^0) corresp. to the parties in X) = $2^{t-1} - 0 = 2^{t-1}$, which gives optimal relative contrast 1.

To prove the security condition we need to show that if F is any maximal forbidden set then the restricted matrices $S^0[F]$ and $S^1[F]$ are equal, upto a column permutation. So let F be a maximal forbidden set. The restricted matrix $CA[F]$ contains an all zero column, say the r th column. Here we notice that F is the r th maximal forbidden set indexing the r th column of the CA. Now, by our construction method, the shares of the participants in F do not contain any information about the r th row of $W_{(t,t)}$ and $B_{(t,t)}$. Now from the security condition (see Lemma 1) of (t, t) -threshold scheme, $W_{(t,t)}$ minus the r th row is equal (upto a column permutation) to $B_{(t,t)}$ minus the r th row. Without loss of generality we can assume that the matrices are equal. If two matrices M and N are equal then so are the matrices M' and N' obtained by giving same row operations on M and N respectively. This result follows from the fact that giving a row operation on a matrix is equivalent to multiplying the matrix by an elementary matrix from the left. Hence, the result follows.

Example 5. Consider the access structure $\mathcal{Q}_{min} = \{123, 14\}$ on the set of four parties $\mathcal{P} = \{1, 2, 3, 4\}$. Thus, $\mathcal{F}_{max} = \{12, 13, 234\}$ and the corresponding CA is

Parties	$F_1 = \{12\}$	$F_2 = \{13\}$	$F_3 = \{234\}$
1	0	0	1
2	0	1	0
3	1	0	0
4	1	1	0

which shows that access structure is already in its most simplified form. Thus, $CA[\{123\}] = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix}$ and $CA[\{14\}] = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix}$ which show that the CA is OPTIMAL.

Since $|\mathcal{F}_{max}| = 3$, we consider the Naor-Shamir basis matrices for $(3, 3)$ -VCS,

$$W_{(3,3)} = \begin{bmatrix} 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix} \text{ and } B_{(3,3)} = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 \end{bmatrix}.$$

We can now construct the basis matrices using the method of Construction 1

$S^0 = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix}$ and $S^1 = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{bmatrix}$ which give optimal relative contrast XVCS on the given access structure.

This example shows that there exist optimal contrast XVCS without being of the (n, n) -threshold type. Other examples of access structures that can have optimal contrast XVCS include *star-graph access structures, any access structure with just two minimal qualified sets, $(k - 1)$ - $(k, n)^*$ type access structures, complete bipartite graph access structure.*

From Proposition 2 we see that if the cumulative array of an access structure (in its most simplified form) is OPTIMAL then there exists a contrast optimal XVCS realizing that access structure. We now ask the converse question, namely if we somehow know that there is an access structure on which it is possible to have contrast optimal XVCS then is it necessarily true that the corresponding cumulative array of the access structure (in its most simplified form) OPTIMAL? Notice that we are not restricting ourselves to one particular way of constructing basis matrices so that contrast optimality is achieved. We seek for the result for any arbitrary method of construction.

To rephrase, we are finding the truth value of the following statement: If an access structure (in its most simplified form) has non-Optimal CA then there is no construction technique which will give contrast-optimal XVCS realizing the access structure.

Let us first consider two examples to gain insight into the problem.

Example 6. Let us consider a $(2, 3)$ -XVCS on the set of parties $\mathcal{P} = \{1, 2, 3\}$. Thus $\mathcal{Q}_{min} = \{12, 13, 23\}$ and $\mathcal{F}_{max} = \{1, 2, 3\}$. The cumulative array for this access structure is given in Remark 2 which shows that the access structure is already in its most simplified form. Consider the restriction of the CA to the rows indexed by the minimal qualified set 12, $CA[\{12\}] = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix}$ and thus it is non-Optimal.

Suppose it is possible to construct basis matrices S^0 and S^1 which give contrast optimal XVCS.

Let $S^0 = \begin{bmatrix} \dots & R_1^0 & \dots \\ \dots & R_2^0 & \dots \\ \dots & R_3^0 & \dots \end{bmatrix}$ and $S^1 = \begin{bmatrix} \dots & R_1^1 & \dots \\ \dots & R_2^1 & \dots \\ \dots & R_3^1 & \dots \end{bmatrix}$. From the definition of relative contrast it follows that for contrast to be 1

$$\left. \begin{matrix} R_1^0 \oplus R_2^0 = \mathbf{0} \\ R_1^0 \oplus R_3^0 = \mathbf{0} \\ R_2^0 \oplus R_3^0 = \mathbf{0} \end{matrix} \right\} \quad \text{and} \quad \left. \begin{matrix} R_1^1 \oplus R_2^1 = \mathbf{1} \\ R_1^1 \oplus R_3^1 = \mathbf{1} \\ R_1^1 \oplus R_2^1 = \mathbf{1} \end{matrix} \right\}$$

where $\mathbf{0}$ denotes the tuple with all-zero entries and $\mathbf{1}$ denotes the tuple with all-one entries. Now, the last three equations

$$\left. \begin{matrix} R_1^1 \oplus R_2^1 = \mathbf{1} \\ R_1^1 \oplus R_3^1 = \mathbf{1} \\ R_1^1 \oplus R_2^1 = \mathbf{1} \end{matrix} \right\}$$

are inconsistent.

Hence, there can not be any construction method whatsoever that will give contrast-optimal XVCS on $(2, 3)$ -threshold access structure.

Another type of situation may occur which we explain in the next example.

Example 7. Let us consider a $(3, 4)$ -threshold access structure on the set of parties $\mathcal{P} = \{1, 2, 3, 4\}$.

Thus $\mathcal{Q}_{min} = \{123, 124, 134, 234\}$ and $\mathcal{F}_{max} = \{12, 13, 14, 23, 24, 34\}$. The CA for this access structure is given by

Parties	$F_1 = \{12\}$	$F_2 = \{13\}$	$F_3 = \{14\}$	$F_4 = \{23\}$	$F_5 = \{24\}$	$F_6 = \{34\}$
1	0	0	0	1	1	1
2	0	1	1	0	0	1
3	1	0	1	0	1	0
4	1	1	0	1	0	0

From the CA it is clear that the access structure is already in its most simplified form. Moreover, the CA is non-Optimal as $CA[\{1, 2, 3\}] = \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 \end{bmatrix}$, which is the restriction of the CA to rows indexed by the minimal qualified set $\{1, 2, 3\}$. It contains three columns with weight 2. Let us consider this minimal qualified set $\{123\}$ and take F_6 which has made the CA non-Optimal.

Suppose it is possible to construct basis matrices S^0 and S^1 which give contrast optimal $(3, 4)$ -XVCS.

$$\text{Let } S^0 = \begin{bmatrix} \dots & R_1^0 & \dots \\ \dots & R_2^0 & \dots \\ \dots & R_3^0 & \dots \\ \dots & R_4^0 & \dots \end{bmatrix} \text{ and } S^1 = \begin{bmatrix} \dots & R_1^1 & \dots \\ \dots & R_2^1 & \dots \\ \dots & R_3^1 & \dots \\ \dots & R_4^1 & \dots \end{bmatrix}.$$

Now $\{1\} \cup F_6 = \{134\}$ and $\{2\} \cup F_6 = \{234\}$ are minimal qualified sets and hence

$$\left. \begin{matrix} R_1^0 \oplus R_3^0 \oplus R_4^0 = \mathbf{0} \\ R_2^0 \oplus R_3^0 \oplus R_4^0 = \mathbf{0} \end{matrix} \right\} \text{ and } \left. \begin{matrix} R_1^1 \oplus R_3^1 \oplus R_4^1 = \mathbf{1} \\ R_2^1 \oplus R_3^1 \oplus R_4^1 = \mathbf{1} \end{matrix} \right\}$$

Moreover, we started with the minimal qualified set $\{123\}$ and therefore we have

$$R_1^0 \oplus R_2^0 \oplus R_3^0 = \mathbf{0} \quad \text{and} \quad R_1^1 \oplus R_2^1 \oplus R_3^1 = \mathbf{1}$$

Now, from the equations related to white pixel we get $R_3^0 = \mathbf{0}$ and from the equations related to black pixel we have $R_3^1 = \mathbf{1}$. Thus the third participant alone is able to recover the secret although he belongs to the forbidden set F_6 . This contradiction shows that there can not be a contrast-optimal $(3, 4)$ -XVCS.

Keeping the above examples in mind we now proceed to prove the following proposition.

Proposition 3. *If the cumulative array for a most simplified access structure $(\mathcal{Q}_{min}, \mathcal{F}_{max})$ is non-Optimal then there cannot be a contrast-optimal XVCS realizing the access structure.*

Proof: Suppose there exists a construction method by which it is possible to have relative contrast 1.

Let the basis matrices be $S^0 = \begin{bmatrix} \dots R_1^0 & \dots \\ \dots R_2^0 & \dots \\ \dots & \dots \\ \dots R_n^0 & \dots \end{bmatrix}$ and $S^1 = \begin{bmatrix} \dots R_1^1 & \dots \\ \dots R_2^1 & \dots \\ \dots & \dots \\ \dots R_n^1 & \dots \end{bmatrix}$, where n is number of participants.

Now we know that the corresponding CA is not Optimal and therefore there exists at least one minimal qualified set say, B such that $CA[B]$ contains at least one non-zero even column. Let $F \in \mathcal{F}_{max}$ be that maximal forbidden set for which the column is of non-zero even weight. Let the weight be $2k$ where $k \neq 0$. Suppose i_1, i_2, \dots, i_{2k} be the corresponding parties in B but not in F . Let $B = \{i_1, i_2, \dots, i_{2k}, j_1, j_2, \dots, j_s\}$ such that the first $2k$ many parties are in $B \setminus F$. We have used different symbols j_1, j_2, \dots, j_s to denote the other parties in B , because it is possible that $B = \{i_1, i_2, \dots, i_{2k}\}$ and there is no more parties in B (e.g. see Example 6).

Since F is maximal forbidden set and $i_1, i_2, \dots, i_{2k} \notin F$ therefore $F \cup \{i_1\}$ contains at least one minimal qualified set, $F \cup \{i_2\}$ contains at least one minimal qualified set, \dots , $F \cup \{i_{2k}\}$ contains at least one minimal qualified set. We note that these minimal qualified sets respectively contain i_1, i_2, \dots, i_{2k} . Let the minimal qualified sets be respectively,

$$\{f_1^1, f_2^1, \dots, f_{\alpha(1)}^1, i_1\}, \{f_1^2, f_2^2, \dots, f_{\alpha(2)}^2, i_2\}, \dots, \{f_1^{2k}, f_2^{2k}, \dots, f_{\alpha(2k)}^{2k}, i_{2k}\}.$$

Thus we must have the following sets of equations (and using the fact of optimal relative contrast)

$$\left. \begin{aligned} R_{f_1^0}^0 \oplus R_{f_2^0}^0 \oplus \dots \oplus R_{f_{\alpha(1)}^0}^0 \oplus R_{i_1}^0 &= \mathbf{0} \\ R_{f_1^0}^0 \oplus R_{f_2^0}^0 \oplus \dots \oplus R_{f_{\alpha(2)}^0}^0 \oplus R_{i_2}^0 &= \mathbf{0} \\ \dots\dots\dots \\ R_{f_1^0}^0 \oplus R_{f_2^0}^0 \oplus \dots \oplus R_{f_{\alpha(2k)}^0}^0 \oplus R_{i_{2k}}^0 &= \mathbf{0} \end{aligned} \right\}$$

and

$$\left. \begin{aligned} R_{f_1^1}^1 \oplus R_{f_2^1}^1 \oplus \dots \oplus R_{f_{\alpha(1)}^1}^1 \oplus R_{i_1}^1 &= \mathbf{1} \\ R_{f_1^1}^1 \oplus R_{f_2^1}^1 \oplus \dots \oplus R_{f_{\alpha(2)}^1}^1 \oplus R_{i_2}^1 &= \mathbf{1} \\ \dots\dots\dots \\ R_{f_1^1}^1 \oplus R_{f_2^1}^1 \oplus \dots \oplus R_{f_{\alpha(2k)}^1}^1 \oplus R_{i_{2k}}^1 &= \mathbf{1} \end{aligned} \right\}$$

Notice that there are $2k$ many equations in each set and the rows $R_{i_1}^0, R_{i_2}^0, \dots, R_{i_{2k}}^0$ corresponding to i_1, i_2, \dots, i_{2k} occur exactly once in each set. Rows other than these are shares corresponding to the parties in F .

Last we have two more sets of equations corresponding to the parties B ,

$$R_{i_1}^0 \oplus \dots \oplus R_{i_{2k}}^0 \oplus R_{j_1}^0 \oplus \dots \oplus R_{j_s}^0 = \mathbf{0}, \quad R_{i_1}^1 \oplus \dots \oplus R_{i_{2k}}^1 \oplus R_{j_1}^1 \oplus \dots \oplus R_{j_s}^1 = \mathbf{1}$$

We observe that $j_1, j_2, \dots, j_s \in F$ and there are $(2k + 1)$ (odd) many equations for each set. Adding modulo 2 i.e. taking XOR of all the equations corresponding

to white pixel 0 we observe that the rows indexed by i_1, i_2, \dots, i_{2k} are deleted. Same thing happens with the equations corresponding to black pixel 1. Whatever rows we are left with, all are indexed by the parties in F . There are only two possibilities. Either the equations have inconsistency (as in Example 6) or a forbidden set of parties are able to retrieve the secret image (as in Example 7). These contradictions show that we cannot have such basis matrices which give optimal contrast for an access structure whose CA is not Optimal. This completes the proof of the proposition.

By Construction 1, Propositions 2 and 3 we now conclude that

Theorem 1. *A necessary and sufficient condition for $(\mathcal{Q}_{min}, \mathcal{F}_{max})$ -XVCS to achieve optimum relative contrast 1 is that the corresponding cumulative array of \mathcal{Q}_{min} is **OPTIMAL**, where the access structure $(\mathcal{Q}_{min}, \mathcal{F}_{max})$ is in its most simplified form.*

From the first *Observation* following Proposition 1 and from Theorem 1 we now have the following corollary.

Corollary 1. *For $2 \leq k \leq n - 1$, there does not exist (k, n) -XVCS that can achieve optimal relative contrast 1.*

Acknowledgement. Research of the second author is partially supported 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.

References

1. Adhikari, A.: Linear algebraic techniques to construct monochrome visual cryptographic schemes for general access structure and its applications to color images. *Des. Codes Crypt.* **73**(3), 865–895 (2014)
2. Adhikari, A., Bose, M.: A new visual cryptographic scheme using latin squares. *IEICE Trans. Fundam. Electron. Commun. Comput. Sci.* **87**(5), 1198–1202 (2004)
3. Adhikari, A., Dutta, T.K., Roy, B.: A new black and white visual cryptographic scheme for general access structures. In: Canteaut, A., Viswanathan, K. (eds.) *INDOCRYPT 2004*. LNCS, vol. 3348, pp. 399–413. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-30556-9_31
4. Adhikari, A., Bose, M., Kumar, D., Roy, B.K.: Applications of partially balanced incomplete block designs in developing $(2, n)$ visual cryptographic schemes. *IEICE Trans.* **90-A**(5), 949–951 (2007)
5. Adhikari, A., Roy, B.: On some constructions of monochrome visual cryptographic schemes. In: 1st International Conference on Information Technology, IT 2008, pp. 1–6. IEEE (2008)
6. Arumugam, S., Lakshmanan, R., Nagar, A.K.: On $(k, n)^*$ -visual cryptography scheme. *Des. Codes Crypt.* **71**(1), 153–162 (2014)
7. Ateniese, G., Blundo, C., Santis, A.D., Stinson, D.R.: Visual cryptography for general access structures. *Inf. Comput.* **129**, 86–106 (1996)

8. Ateniese, G., Blundo, C., De Santis, A., Stinson, D.R.: Constructions and bounds for visual cryptography. In: Meyer, F., Monien, B. (eds.) ICALP 1996. LNCS, vol. 1099, pp. 416–428. Springer, Heidelberg (1996). https://doi.org/10.1007/3-540-61440-0_147
9. Blundo, C., D'arco, P., De Santis, A., Stinson, D.R.: Contrast optimal threshold visual cryptography. *SIAM J. Discrete Math.* **16**(2), 224–261 (2003)
10. Droste, S.: New results on visual cryptography. In: Kobitz, N. (ed.) CRYPTO 1996. LNCS, vol. 1109, pp. 401–415. Springer, Heidelberg (1996). https://doi.org/10.1007/3-540-68697-5_30
11. Dutta, S., Adhikari, A.: XOR based non-monotone t - (k, n) *-visual cryptographic schemes using linear algebra. In: Hui, L.C.K., Qing, S.H., Shi, E., Yiu, S.M. (eds.) ICICS 2014. LNCS, vol. 8958, pp. 230–242. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-21966-0_17
12. Dutta, S., Singh Rohit, R., Adhikari, A.: Constructions and analysis of some efficient t - (k, n) *-visual cryptographic schemes using linear algebraic techniques. *Des. Codes Crypt.* **80**(1), 165–196 (2016). Springer
13. Fu, Z., Yu, B.: Optimal pixel expansion of deterministic visual cryptography scheme. *Multimedia Tools Appl.* **73**(3), 1177–1193 (2014)
14. Guo, T., Liu, F., Wu, C.K., Ren, Y.W., Wang, W.: On (k, n) visual cryptography scheme with t essential parties. In: Padró, C. (ed.) ICITS 2013. LNCS, vol. 8317, pp. 56–68. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-04268-8_4
15. Hao, H., Shen, G., Zhengxin, F., Bin, Y., Wang, J.: General construction for XOR-based visual cryptography and its extended capability. *Multimedia Tools Appl.* **75**(21), 13883–13911 (2016)
16. Kafri, O., Keren, E.: Encryption of pictures and shapes by random grids. *Opt. Lett.* **12**(6), 377–379 (1987)
17. Lee, S.S., Na, J.C., Sohn, S.W., Park, C., Seo, D.H., Kim, S.J.: Visual cryptography based on an interferometric encryption technique. *ETRI J.* **24**(5), 373–380 (2002)
18. Liu, F., Wu, C.K., Lin, X.J.: Some extensions on threshold visual cryptography schemes. *Comput. J.* **53**, 107–119 (2010)
19. Liu, F., Wu, C., Lin, X.: Step construction of visual cryptography schemes. *IEEE Trans. Inf. Forensics Secur.* **5**, 27–38 (2010)
20. Naor, M., Shamir, A.: Visual cryptography. In: De Santis, A. (ed.) EUROCRYPT 1994. LNCS, vol. 950, pp. 1–12. Springer, Heidelberg (1995). <https://doi.org/10.1007/BFb0053419>
21. Shyu, S.J., Chen, M.C.: Optimum pixel expansions for threshold visual secret sharing schemes. *IEEE Trans. Inf. Forensics Secur.* **6**(3), pt. 2, 960–969 (2011)
22. Tuyls, P., Hollmann, H.D.L., Lint, H.H., Tolhuizen, L.: A polarisation based visual crypto system and its secret sharing schemes (2002), <http://eprint.iacr.org>
23. Tuyls, P., Hollmann, H., Lint, J., Tolhuizen, L.: Xor-based visual cryptography schemes. *Des. Codes Crypt.* **37**, 169–186 (2005)
24. Viet, D.Q., Kurosawa, K.: Almost ideal contrast visual cryptography with reversing. In: Okamoto, T. (ed.) CT-RSA 2004. LNCS, vol. 2964, pp. 353–365. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-24660-2_27
25. Yang, C.-N., Wang, D.-S.: Property analysis of XOR-based visual cryptography. *IEEE Trans. Circ. Syst. Video Technol.* **24**(2), 189–197 (2014)

Verifiably Multiplicative Secret Sharing

Maki Yoshida¹(✉) and Satoshi Obana²

¹ NICT, Tokyo, Japan

maki-yos@nict.go.jp

² Hosei University, Tokyo, Japan

Abstract. Barkol et al. (Journal of Cryptology, 2010) introduced the notion of *d-multiplicative* secret sharing (*d*-MSS), which allows the players to multiply shared *d* secrets by converting their shares locally into an additive sharing of the product, and proved that *d*-MSS among *n* players is possible if and only if no *d* unauthorized sets of players cover the whole set of players (type Q_d). Although this result implies some limitations on secret sharing in the context of MPC, the *d*-multiplicative property is still useful for simplifying complex tasks of MPC by computing the product of *d* field elements directly and non-interactively. In this paper, to further improve usefulness, we introduce and study the *verifiability* of multiplication, which is mainly formalized for the motivated applications of *d*-MSS. Informally, a *d*-MSS scheme is *verifiable* if the scheme enables the players to *locally* generate an *additive* sharing of proof that the summed value is the correct product of shared *d* secrets. First, we prove that verifiably *d*-MSS among *n* players is possible if no $d + 1$ unauthorized sets of players cover the whole set of players (type Q_{d+1}) where the error probability is zero. That is, a larger number of players *n* is required. In addition, in the proposed error-free scheme, the share size of a proof increases with the number of unauthorized sets. To achieve the optimal bound on *n* of *d*-MSS (type Q_d) efficiently, we accept an error probability. We prove that verifiably *d*-MSS among *n* players is possible if and only if no *d* unauthorized sets of players cover the whole set of players (type Q_d) where the error probability is *non-zero but is chosen arbitrarily*. In the proposed scheme, each share of a proof consists of only *two field elements*. From these results, we can see that there is a tradeoff between usability and correctness (i.e. either no additional players or no error). Because these schemes do not require any setup or interaction, we can freely select them as the situation demands.

1 Introduction

A secret sharing (SS) scheme is a method of sharing a secret among a set of *n* players so that some predefined authorized subsets of the players are able to recover the secret. The notion of *threshold* SS was introduced by Shamir [24] and Blakley [4] independently where the cardinality of any authorized set is larger than a given threshold. Later, Ito et al. [15] generalized this notion to a setting

where the authorized subsets are an arbitrary family of subsets of the players, called *access structures*.

SS is now used as a central building block in many cryptographic and distributed applications such as unconditionally secure multiparty computation (MPC) [1, 2, 5, 7]. In addition, for natural application to unconditionally secure MPC [5, 7], the *multiplicative* property of SS is essential. We therefore focus on information-theoretically secure SS in this paper.

Motivated by open problems in the area of MPC such as unconditionally secure MPC with minimal interaction, Barkol et al. (Journal of Cryptology, 2010 [3]) introduced *d-multiplicative* SS and studied the type of access structures for which such secret sharing schemes exist. A secret sharing scheme is *d-multiplicative* if the scheme allows the players to multiply shared d (rather than two) secrets by *locally* converting their shares into an *additive* sharing of the product. They proved that *d-multiplicative* schemes exist if and only if no d unauthorized sets of players cover the whole set of players (type Q_d). In particular, t -private *d-multiplicative* secret sharing among n players is possible only if $n > d \cdot t$ where *t-private* means that every set of t players is unauthorized. This result implies a limitation on the usefulness of SS in the context of MPC in the sense that a larger number of players n is required for maintaining the privacy level t as d increases. In other words, if we have a sufficient number of players, there is a possibility of simplifying complex tasks of MPC by computing the product of two or more elements directly and non-interactively without any setup.

In this paper, we aim to improve the usefulness of *d-multiplicative* SS (MSS) in the context of MPC while maintaining its advantages: no need for any interaction, any setup, or any computational assumption.

First, we introduce the notion of *verifiably d-multiplicative* SS, which is mainly formalized for the motivated applications of *d-MSS* given in [3]. In the motivated applications, each player adds random additive shares of 0 to each generated share and the receiver of the shares only obtains the summed value (i.e. the product). We therefore call a *d-multiplicative* scheme *verifiable* if the scheme enables the players to *locally* generate an *additive* sharing of a proof that the sum of shares (rather than each share) is correct. We expect that the verifiability can be used for making MPC secure in the presence of an active adversary by accepting the output only if the correctness is verified. A concrete application is beyond the scope of this paper and is a possible future work.

Secondly, we study the feasibility of verifiably *d-multiplicative* SS. We prove that verifiably *d-multiplicative* secret sharing is possible if the access structures of type Q_{d+1} where the privacy achieved is perfect and the error probability is zero. In the threshold case, type Q_{d+1} implies $n > (d + 1) \cdot t$. This means that we need to degrade the privacy level t or gather a larger number of players n . In addition, in the proposed error-free scheme, the share size of a proof increases with the number of unauthorized sets. A basic approach for overcoming this problem in the context of MPC is to require interaction among the players [20] or to use *verifiable secret sharing* [22], which relies on computationally secure

commitment with a common reference string. That is, the advantages of d -MSS are spoiled.

To achieve the optimal bounds on n of d -MSS (i.e., $n > d \cdot t$ for t -privacy, or type Q_d), we accept an error probability and prove that verifiably d -multiplicative schemes exist if and only if the access structure is of type Q_d , where the privacy achieved is perfect, the error probability is non-zero but chosen arbitrarily, and each share of the proof only consists of *two field elements*.

The interesting point of these results is that a secret sharing scheme itself is not necessarily verifiable or linear. We note that the same results can be also obtained for non-perfect privacy from the result on the (im)possibility of non-perfect d -MSS in [26].

2 Preliminaries

In this section, we recall the definition of multiplicative and private properties, some results on feasibility, and a motivated application given in [3].

2.1 Notations and Definitions

A secret sharing scheme involves a dealer and n players P_1, \dots, P_n , and specifies a randomized mapping from the secret s to an n -tuple of shares (s_1, \dots, s_n) , where the share s_i is given to player P_i . We assume that the secret is taken from a finite field \mathbb{F} . We also assume that all shares s_i are taken from a finite share domain \mathcal{S} . Let \mathcal{D} denote a discrete probability distribution from which the dealer's randomness is chosen. To share a secret $s \in \mathbb{F}$, the dealer chooses a random element $r \in \mathcal{D}$ and applies a sharing function $\text{SHARE} : \mathbb{F} \times \mathcal{D} \rightarrow \mathcal{S}^n$ to compute $\text{SHARE}(s, r) = (s_1, \dots, s_n)$. For $T \subseteq [n]$, let $\text{SHARE}(s, r)_T$ denote the restriction of $\text{SHARE}(s, r)$ to its T -entries.

Definition 1 (t -Private secret sharing [3]). *A secret sharing scheme is said to be t -private if for every set $T \subseteq [n]$ with $|T| = t$ and every pair of secrets $s, s' \in \mathbb{F}$, the random variables $\text{SHARE}(s, r)_T$ and $\text{SHARE}(s', r)_T$ induced by a random choice of $r \in \mathcal{D}$ are identically distributed.*

Definition 2 (d -Multiplicative secret sharing [3]). *We call a secret sharing scheme d -multiplicative if it satisfies the following d -multiplicative property. Let $s^{(1)}, \dots, s^{(d)} \in \mathbb{F}$ be d secrets, and $r^{(1)}, \dots, r^{(d)} \in \mathcal{D}$ be d elements in the support of \mathcal{D} . For $1 \leq j \leq d$, let $(s_1^{(j)}, \dots, s_n^{(j)}) = \text{SHARE}(s^{(j)}, r^{(j)})$. We require the existence of a function $\text{MULT} : [n] \times \mathcal{S}^d \rightarrow \mathbb{F}$ such that for all possible $s^{(j)}$ and $r^{(j)}$ as above, $\sum_{i=1}^n \text{MULT}(i, s_i^{(1)}, \dots, s_i^{(d)}) = \prod_{j=1}^d s^{(j)}$.*

To generalize our results from the threshold case to general access structures, we show the notations and definitions of such secret sharing given in [3]. In contrast to traditional secret sharing specifying a collection of authorized player sets, the complementary notion of an *adversary structure*, specifying a collection of *unauthorized* sets, is used for convenience in [3].

Definition 3 (Adversary structure [3]). An n -player adversary structure is a collection of sets $\mathcal{T} \subseteq 2^{[n]}$ that is closed under subsets; that is, if $T \in \mathcal{T}$ and $T' \subseteq T$ then $T' \in \mathcal{T}$. Let $\hat{\mathcal{T}}$ be the collection of maximal sets in \mathcal{T} (namely those that are not contained in any other set from \mathcal{T}).

Definition 4 (\mathcal{T} -Private secret sharing [3]). Let \mathcal{T} be an n -player adversary structure. A secret sharing scheme is said to be \mathcal{T} -private if every pair of secret $s, s' \in \mathbb{F}$ and every $T \in \mathcal{T}$, the random variables $\text{SHARE}(s, r)_T$ and $\text{SHARE}(s', r)_T$ induced by a random choice of $r \in \mathcal{D}$ are identically distributed.

Definition 5 (Adversary structure of type Q_d [3]). Let n, d be positive integers and \mathcal{T} be an n -player adversary structure. We say that \mathcal{T} is of type Q_d if for every d sets $T_1, \dots, T_d \in \mathcal{T}$ we have $T_1 \cup \dots \cup T_d \subset [n]$. That is, no d unauthorized sets cover the entire set of players.

The main result in [3] is a characterization of d -multiplicative secret sharing.

Theorem 1 (Theorem 4.6 in [3]). For any positive integers n, d and a n -player adversary structure \mathcal{T} , there exists a d -multiplicative \mathcal{T} -private secret sharing scheme if and only if \mathcal{T} is of type Q_d .

2.2 A Motivated Application

The motivated applications of the d -multiplicative property given in [3] are secure polynomial evaluation and *general* secure computation with minimal interaction. It has been shown that given a t -private d -multiplicative secret sharing for n players over \mathbb{F} , there exists a t -private n -server secure polynomial evaluation protocol for multi-variate polynomials of degree d over \mathbb{F} where the communication complexity is linear in the input length (see Lemma 3.1 in [3]). In addition, the generalization from polynomials to arbitrary functions can be obtained by using *randomizing polynomials* [16] which enables to represent an arbitrary function by a vector of (randomized) degree-3 polynomials [3].

For simplicity, we briefly introduce the simplest case: A polynomial is the form $x_1 \cdot x_2 \cdots x_d$; There are d clients, who holds inputs and wish to evaluate the polynomial without revealing their inputs each other, and n servers, who help perform the evaluation. Client j with $1 \leq j \leq d$ holds an input $s^{(j)}$ and every server only knows the identity of the polynomial. Informally, a protocol should satisfy the following correctness and privacy requirements.

Correctness: All clients output $s^{(1)} \cdots s^{(d)}$ (assuming that both client and servers follow the protocol).

t -Privacy: Any collusion involving a strict subset of the clients and at most t servers should not learn anything about the inputs of the other clients other than what follows from their own inputs and the output.

The formal definitions and security proof are not included in [3] (the related literatures [6, 12] are referred), and omitted here.

The t -private n -server protocol given in [3] proceeds as follows:

- Round 1: Client j , $1 \leq j \leq d$, shares his input $s^{(j)}$ by computing $\text{SHARE}(s^{(j)}, r^{(j)}) = (s_1^{(j)}, \dots, s_n^{(j)})$. After sharing his input, he sends the share $s_i^{(j)}$ to Server i . In addition, Client j distributes between the servers random additive shares of 0, namely it sends to Server i a field element $z_i^{(j)}$ such that the n elements $z_i^{(j)}$ are random subject to the restriction that they add up to 0, i.e., $\sum_{i=1}^n z_i^{(j)} = 0$.
- Round 2: Server i , $1 \leq i \leq n$, computes $y_i = \text{MULT}(i, s_i^{(1)}, \dots, s_i^{(d)}) + \sum_{j=1}^d z_i^{(j)}$, and sends y_i to all clients.
- Output: Each client computes and outputs $\sum_{i=1}^n y_i$. From the d -multiplicative property, this output is equal to $s^{(1)} \dots s^{(d)}$.

An important point to note here is that the generated shares y_i is randomized by additive shares of 0 and each client only obtains the summed value (i.e., the product). Thus, in this paper, the notion of verifiability is defined for the summed value rather than each share.

3 Verifiably Multiplicative Secret Sharing

We now define the verifiability of multiplication. We assume that malicious players who may behave arbitrary have the same structure as that against privacy. To verify the summed value rather than each additive share, we define a proof and its shares by vectors in \mathbb{F}^c for a positive integer c where the summation of two vectors $a = (a_1, \dots, a_c)$ and $b = (b_1, \dots, b_c)$ is performed by adding the corresponding components of the vectors, i.e., $a + b = (a_1 + b_1, \dots, a_c + b_c)$.

Definition 6 ((ϵ, d) -Verifiably multiplicative secret sharing). *Let c be a positive integer. A \mathcal{T} -private secret sharing scheme is said to be (ϵ, d) -verifiably multiplicative if the scheme is d -multiplicative and there are two functions $\text{PROOF} : [n] \times \mathcal{S}^d \rightarrow \mathbb{F}^c$ and $\text{VER} : \mathbb{F} \times \mathbb{F}^c \rightarrow \{1, 0\}$ that satisfy the following properties.*

Correctness: For $s^{(j)} \in \mathbb{F}$ and $r^{(j)} \in \mathcal{D}$ with $1 \leq j \leq d$, let $(s_1^{(j)}, \dots, s_n^{(j)}) = \text{SHARE}(s^{(j)}, r^{(j)})$, $m = \sum_{i=1}^n \text{MULT}(i, s_i^{(1)}, \dots, s_i^{(d)})$, and $\sigma = \sum_{i=1}^n \text{PROOF}(i, s_i^{(1)}, \dots, s_i^{(d)})$. Then, $\text{VER}(m, \sigma) = 1$.

Verifiability: An adversary that modifies any additive shares for any $T \in \mathcal{T}$ can cause a wrong value to be accepted as the product with probability at most ϵ . More formally, we define the experiment $\text{Exp}(s^{(1)}, \dots, s^{(d)}, T, \text{Adv})$ with some d secrets $s^{(1)}, \dots, s^{(d)} \in \mathbb{F}$, unauthorized set $T \in \mathcal{T}$, and interactive adversary Adv .

$\text{Exp}(s^{(1)}, \dots, s^{(d)}, T, \text{Adv})$:

1. For each j with $1 \leq j \leq d$, sample $r^{(j)} \leftarrow \mathcal{D}$ and generate $(s_1^{(j)}, \dots, s_n^{(j)}) = \text{SHARE}(s^{(j)}, r^{(j)})$.
2. Give $\{(s_i^{(1)}, \dots, s_i^{(d)}) \mid i \in T\}$ to Adv .

3. *Adv* outputs modified additive shares $m'_i \in \mathbb{F}$ and $\sigma'_i \in \mathbb{F}^c$ with $i \in T$. For $i \notin T$, we define $m'_i = \text{MULT}(i, s_i^{(1)}, \dots, s_i^{(d)})$ and $\sigma'_i = \text{PROOF}(i, s_i^{(1)}, \dots, s_i^{(d)})$.
4. Compute $m' = \sum_{i=1}^n m'_i$ and $\sigma' = \sum_{i=1}^n \sigma'_i$.
5. If $m' \neq s^{(1)} \dots s^{(d)}$ and $\text{VER}(m', \sigma') = 1$, then output 1 else 0.

We require that for any d secrets $s^{(1)}, \dots, s^{(d)} \in \mathbb{F}$, any unauthorized set $T \in \mathcal{T}$, and any unbounded adversary *Adv*,

$$\Pr[\text{Exp}(s^{(1)}, \dots, s^{(d)}, T, \text{Adv}) = 1] \leq \epsilon.$$

Given an (ϵ, d) -verifiably multiplicative t -private secret sharing scheme, we can make the motivated application correct in the presence of at most t malicious servers. Specifically, the protocol satisfies the following strong correctness.

t -Correctness: All clients output $s^{(1)} \dots s^{(d)}$ or \perp assuming at most t malicious servers. That is, an incorrect value is not accepted.

The protocol in Sect. 2 is modified as follows.

- Round 1: Client j distributes between the servers random additive shares of the zero-vector, namely it sends to Server i a vector $z_i^{(j)} \in \mathbb{F}^{c+1}$ such that the n vectors $z_i^{(j)}$ are random subject to the restriction that they add up to the vector with all components being 0, i.e., $\sum_{i=1}^n z_i^{(j)} = (0, \dots, 0)$.
- Round 2: Server i , $1 \leq i \leq n$, computes a vector $y_i = (\text{MULT}(i, s_i^{(1)}, \dots, s_i^{(d)}), \text{PROOF}(i, s_i^{(1)}, \dots, s_i^{(d)})) + \sum_{j=1}^d z_i^{(j)}$, and sends y_i to all clients.
- Output: Let $y_i = (m_i, \sigma_i)$. Each client computes $m = \sum_{i=1}^n m_i$ and $\sigma = \sum_{i=1}^n \sigma_i$. It outputs m if $\text{VER}(m, \sigma) = 1$, otherwise it outputs 0.

4 Feasibilities

Our main results are sufficient conditions for (ϵ, d) -verifiably multiplicative \mathcal{T} -private secret sharing to be possible. For the error-free case $\epsilon = 0$, the condition is stronger than that of the previous d -multiplicative \mathcal{T} -private secret sharing, which does not require the verifiability.

Theorem 2. *For any positive integers n, d , and an n -player adversary structure \mathcal{T} , there exists a $(0, d)$ -verifiably multiplicative \mathcal{T} -private secret sharing scheme if \mathcal{T} is of type Q_{d+1} where $c = |\hat{\mathcal{T}}|$ (every proof consists of $|\hat{\mathcal{T}}|$ elements of \mathbb{F}).*

Then, we prove that the condition can be weakened to the optimal one, i.e., that of the previous d -multiplicative \mathcal{T} -private secret sharing (type Q_d) by relaxing the requirement on the error probability to $\epsilon > 0$ that is chosen arbitrarily.

Theorem 3. *For any positive integers n, E, d , and an n -player adversary structure \mathcal{T} , there exists a secret sharing scheme that is $(1/|\mathbb{F}|^E, d)$ -verifiably multiplicative and \mathcal{T} -private if and only if \mathcal{T} is of type Q_d where $c = 2E$ (every proof consists of two elements of \mathbb{F}^E).*

We now prove Theorem 2.

Proof. (Theorem 2). We construct a $(0, d)$ -verifiably multiplicative \mathcal{T} -private scheme for n players from the CNF scheme in [15], which is given for general access structures. In the CNF scheme, to share a given secret s , for $T \in \hat{\mathcal{T}}$, r_T is randomly chosen from \mathbb{F} subject to the restriction that $\sum_{T \in \hat{\mathcal{T}}} r_T = s$. Each share s_i is the set $\{r_T | i \notin T\}$. We note that in the t -private CNF scheme, s_i consists of exactly ${}_{n-1}C_t$ field elements. The \mathcal{T} -privacy property follows from the fact that every set $T \in \hat{\mathcal{T}}$ jointly misses r_T and thus can learn no information about the secret. The d -multiplicative property is proven in [3] and a multiplication function MULT exists. Thus, we prove the existence of PROOF and VER. The key idea is to generate shares of the product for subsets of players $[n] \setminus T$ for every set of malicious players $T \in \mathcal{T}$ and check the equality of all recovered values. Any set of malicious players is contained by some $T \in \hat{\mathcal{T}}$. Thus, the value recovered from shares for $[n] \setminus T$ is correct, and the equality of all recovered values guarantees that the error-probability is zero. Based on this idea, we define PROOF and VER as follows. We number the subsets in $\hat{\mathcal{T}}$ from 1 to $|\hat{\mathcal{T}}|$. Let $s^{(1)}, \dots, s^{(d)}$ be secrets. For $1 \leq j \leq d$, let $r_T^{(j)}$ with $T \in \hat{\mathcal{T}}$ denote the additive parts of $s^{(j)}$. We write the product $s^{(1)} \dots s^{(d)} = (\sum_{T \in \hat{\mathcal{T}}} r_T^{(1)}) \dots (\sum_{T \in \hat{\mathcal{T}}} r_T^{(d)})$ as the sum of the $|\hat{\mathcal{T}}|^d$ monomials of the form $r_{T_{j_1}}^{(1)} \dots r_{T_{j_d}}^{(d)}$. For each $T_l \in \hat{\mathcal{T}}$, we partition the monomials into $n - |T_l|$ disjoint sets $X_{l,i}$ such that $i \in [n] \setminus T_l$ and all monomials in set $X_{l,i}$ is obtained from s_i . The possibility of partition follows from the fact that every monomial as above can be assigned to a set $X_{l,i}$ such that $i \notin T_{j_1} \cup \dots \cup T_{j_d} \cup T_l$. The existence of such i follows from the assumption that \mathcal{T} is of type Q_{d+1} . For each $1 \leq i \leq n$, PROOF(i, \cdot) outputs $\sigma_i = (\sigma_{i,1}, \dots, \sigma_{i,|\hat{\mathcal{T}}|}) \in \mathbb{F}^{|\hat{\mathcal{T}}|}$ where $\sigma_{i,l}$ is the sum of the monomials in $X_{l,i}$ if $i \notin T_l$, and otherwise 0. We note that if all players follow the scheme, then $\sigma = \sum \sigma_i$ is the vector with all components being $s^{(1)} \dots s^{(d)}$. We define the verification function VER(m, σ) to be 1 if and only if $\sigma = (m, \dots, m)$ holds. Even if malicious players T provide incorrect shares, there is a component σ_l with $T \subseteq T_l$ which is the correct value $s^{(1)} \dots s^{(d)}$. Thus, VER detects the existence of an incorrect value without error. \square

Next, we prepare a lemma for the proof of Theorem 3.

Lemma 1. *Given d -multiplicative \mathcal{T} -private secret sharing schemes for n players over \mathbb{F} and \mathbb{F}^E , there exists a $(1/|\mathbb{F}|^E, d)$ -verifiably multiplicative \mathcal{T} -private secret sharing scheme for n players where $c = 2E$ (every proof consists of two elements of \mathbb{F}^E).*

Proof. For notational convenience, we present the proof for the case $E = 1$. The generalization to an arbitrary $E > 1$ is shown later. Suppose there is a d -multiplicative \mathcal{T} -private secret sharing scheme for n players over \mathbb{F} and its multiplication function, denoted by SHARE' and MULT', with randomness domain \mathcal{D}' and share domain \mathcal{S}' . We show a method of constructing a $(1/|\mathbb{F}|, d)$ -verifiably multiplicative \mathcal{T} -private secret sharing scheme for n players (SHARE, MULT, PROOF, VER) with $c = 2$ from (SHARE', MULT').

The key idea is as follows: For the product $m = s^{(1)} \cdots s^{(d)}$, PROOF generates additive shares of $\alpha \in \mathbb{F}$ and those of $\beta = \alpha \cdot m$, and then VER checks whether $\alpha \cdot m = \beta$. A similar technique is used for detection of cheaters in secret sharing by Cabello *et al.* [9] in which m is replaced with the secret s itself and α and β are shared together with the secret. In contrast, in the scheme we present here, additive shares of α and β are not shared beforehand and are computed by using only the d -multiplicative property. We note that the d -multiplication property imposes no linearity requirement on SHARE itself. Thus, we need to convert non-additive shares of α into additive ones. To realize such conversion, we additionally share “1” for padding the product 1^{d-1} with α .

Specifically, we define $\text{SHARE} : \mathbb{F} \times \mathcal{D} \rightarrow \mathcal{S}$ as follows: $\mathcal{D} = \mathbb{F} \times \mathcal{D}^{/4}$, $\mathcal{S} = \mathbb{F}^4$, and $\text{SHARE}(s, (\alpha, r_1, r_2, r_3, r_4)) = (\text{SHARE}'(s, r_1), \text{SHARE}'(\alpha, r_2), \text{SHARE}'(\alpha \cdot s, r_3), \text{SHARE}'(1, r_4))$. That is, randomly chosen $\alpha \in \mathbb{F}$, $\gamma = \alpha \cdot s \in \mathbb{F}$, and $1 \in \mathbb{F}$ are additionally shared.

Let $s^{(1)}, \dots, s^{(d)}$ be d secrets. Let $\alpha^{(1)}, \dots, \alpha^{(d)}, \gamma^{(1)}, \dots, \gamma^{(d)}$ be chosen as the above, that is, $\gamma^{(j)} = \alpha^{(j)} \cdot s^{(j)}$. For $1 \leq i \leq n$ and $1 \leq j \leq d$, $s_i^{(j)} = (t_i^{(j)}, \alpha_i^{(j)}, \gamma_i^{(j)}, 1_i^{(j)})$ be the i -th share of $s^{(j)}$. We define $\text{MULT}(i, s_i^{(1)}, \dots, s_i^{(d)}) = \text{MULT}'(i, t_i^{(1)}, \dots, t_i^{(d)})$, that is, the same as the original scheme. Then, we define $\text{PROOF}(i, s_i^{(1)}, \dots, s_i^{(d)}) = (\text{MULT}'(i, \alpha_i^{(1)}, 1_i^{(2)}, \dots, 1_i^{(d)}), \text{MULT}'(i, \gamma_i^{(1)}, t_i^{(2)}, \dots, t_i^{(d)}))$, which consists of an additive share of $\alpha^{(1)} \cdot 1 \cdots 1$ and that of $\gamma^{(1)} \cdot s^{(2)} \cdots s^{(d)} = \alpha^{(1)} \cdot s^{(1)} \cdot s^{(2)} \cdots s^{(d)}$. For $m \in \mathbb{F}$ and $\sigma = (\sigma_1, \sigma_2) \in \mathbb{F}^2$, $\text{VER}(m, \sigma) = 1$ if and only if $m \cdot \sigma_1 = \sigma_2$.

Let $m_i = \text{MULT}(i, s_i^{(1)}, \dots, s_i^{(d)})$ and $\sigma_i = (\sigma_{i,1}, \sigma_{i,2}) = \text{PROOF}(i, s_i^{(1)}, \dots, s_i^{(d)})$. It is obvious that the correctness holds because $m = \sum m_i = s^{(1)} \cdots s^{(d)}$, $\sigma_1 = \sum \sigma_{i,1} = \alpha^{(1)} \cdot 1 \cdots 1 = \alpha$, and $\sigma_2 = \sum \sigma_{i,2} = \alpha^{(1)} \cdot s^{(1)} \cdot s^{(2)} \cdots s^{(d)}$.

In the following, we prove the verifiability. Let $T \in \mathcal{T}$. Let $\Delta_m = m - m'$, $\Delta_\alpha = \sigma_1 - \sigma'_1$, and $\Delta_\beta = \sigma_2 - \sigma'_2$ where m' and $\sigma' = (\sigma'_1, \sigma'_2)$ is computed in Step 4 in *Exp*. Adv can choose $(\Delta_m, \Delta_\alpha, \Delta_\beta)$ arbitrarily by modifying m'_i and σ'_i for $i \in T$ in Step 3 of *Exp*. The error occurs if $\Delta_m \neq 0$ and $\text{VER}(m + \Delta_m, (\sigma_1 + \Delta_\alpha, \sigma_2 + \Delta_\beta)) = 1$, that is, $m \cdot \Delta_\alpha + \alpha^{(1)} \cdot \Delta_m + (\Delta_m \cdot \Delta_\alpha - \Delta_\beta) = 0$. For every choice of $(\Delta_m, \Delta_\alpha, \Delta_\beta)$ with $\Delta_m \neq 0$, there is a unique $\alpha^{(1)} \in \mathbb{F}$ satisfying the above equation. Thus, for any d secrets $s^{(1)}, \dots, s^{(d)}$, any $T \in \mathcal{T}$, and any unbounded adversary Adv, the probability of VER outputting 1 is $1/|\mathbb{F}|$.

We can choose E arbitrarily by using an extension field \mathbb{F}^E instead of \mathbb{F} . SHARE shares $\alpha \in \mathbb{F}^E$, $\gamma = \alpha \cdot s \in \mathbb{F}^E$, and $1 \in \mathbb{F}^E$ by using a scheme for \mathbb{F}^E . PROOF generates additive shares in \mathbb{F}^E and VER checks the equality over \mathbb{F}^E . It is easy to show that $\epsilon = 1/|\mathbb{F}|^E$ holds for the modified scheme with almost a same proof. Therefore, we obtain arbitrarily chosen ϵ by choosing a degree of the extension E such that $E = \min\{E' \mid \epsilon \leq 1/|\mathbb{F}|^{E'}\}$. \square

Proof. (Theorem 3). The only-if part is obvious from Theorem 1. If \mathcal{T} is of type Q_d , then there is a d -multiplicative \mathcal{T} -private secret sharing scheme for n players over a finite field. From Lemma 1, the if-part follows. \square

5 Conclusion

In this paper, we have introduced the notion of (ϵ, d) -verifiably multiplicative \mathcal{T} -private secret sharing, and clarified the conditions under which such scheme exists. Namely, we have shown that $(0, d)$ -verifiably multiplicative \mathcal{T} -private secret sharing scheme exists if the adversary structure \mathcal{T} is of type Q_{d+1} , and that, for arbitrarily small $\epsilon > 0$, (ϵ, d) -verifiably multiplicative \mathcal{T} -private secret sharing scheme exists if the adversary structure \mathcal{T} is of type Q_d . These feasibility results were obtained by presenting constructions of (ϵ, d) -verifiably multiplicative and \mathcal{T} -private secret sharing with the corresponding parameters.

Since it has been shown in [3] that a d -multiplicative \mathcal{T} -private secret sharing scheme exists only if the adversary structure \mathcal{T} is of type Q_d , our proposed construction for $\epsilon > 0$ made it clear that an (ϵ, d) -verifiably multiplicative \mathcal{T} -private secret sharing scheme with $\epsilon > 0$ exists *if and only if* the adversary structure \mathcal{T} is of type Q_d .

However, it is not made clear whether $(0, d)$ -verifiably multiplicative \mathcal{T} -private secret sharing scheme can be constructed even when the adversary structure \mathcal{T} is of type Q_d . To clarify the necessary and sufficient condition for the existence of $(0, d)$ -verifiably multiplicative \mathcal{T} -private secret sharing scheme will be future challenge.

References

1. Araki, T., Furukawa, J., Lindell, Y., Nof, A., Ohara, K.: High-throughput semi-honest secure three-party computation with an honest majority. In: 23rd ACM Conference on Computer and Communications Security (ACM CCS 2016), pp. 805–817 (2016)
2. Araki, T., Barak, A., Furukawa, J., Lichter, T., Lindell, Y., Nof, A., Ohara, K., Watzman, A., Weinstein, O.: Optimized honest-majority MPC for malicious adversaries - breaking the 1 billion-gate per second barrier. In: 38th IEEE Symposium on Security and Privacy (S&P 2017), pp. 843–862 (2017)
3. Barkol, O., Ishai, Y., Weinreb, E.: On d -multiplicative secret sharing. *J. Cryptology* **23**(4), 580–593 (2010)
4. Blakley, G.R.: Safeguarding cryptographic keys. In: AFIPS 1979 National Computer Conference, vol. 48, pp. 313–317 (1979)
5. Ben-Or, M., Goldwasser, S., Wigderson, A.: Completeness theorems for non-cryptographic fault-tolerant distributed computation. In: The 20th Annual ACM Symposium on Theory of Computing, STOC 1988, pp. 1–10 (1988)
6. Canetti, R.: Security and composition of multiparty cryptographic protocols. *J. Cryptology* **13**(1), 143–202 (2000)
7. Chaum, D., Crépeau, C., Damgård, I.: Multiparty unconditionally secure protocols. In: The 20th Annual ACM Symposium on Theory of Computing, STOC 1988, pp. 11–19 (1988)
8. Carpentieri, M., De Santis, A., Vaccaro, U.: Size of shares and probability of cheating in threshold schemes. In: Helleseth, T. (ed.) EUROCRYPT 1993. LNCS, vol. 765, pp. 118–125. Springer, Heidelberg (1994). https://doi.org/10.1007/3-540-48285-7_10

9. Cabello, S., Padró, C., Sáez, G.: Secret sharing schemes with detection of cheaters for a general access structure. *Des. Codes Crypt.* **25**(2), 175–188 (2002)
10. Cramer, R., Damgård, I., Maurer, U.: General secure multi-party computation from any linear secret-sharing scheme. In: Preneel, B. (ed.) *EUROCRYPT 2000*. LNCS, vol. 1807, pp. 316–334. Springer, Heidelberg (2000). https://doi.org/10.1007/3-540-45539-6_22
11. Cramer, R., Dodis, Y., Fehr, S., Padró, C., Wichs, D.: Detection of algebraic manipulation with applications to robust secret sharing and fuzzy extractors. In: Smart, N. (ed.) *EUROCRYPT 2008*. LNCS, vol. 4965, pp. 471–488. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-78967-3_27
12. Goldreich, O.: *Foundations of Cryptography: Vol. 2, Basic Applications*. Cambridge University Press, New York (2004)
13. Goldwasser, S., Micali, S., Wigderson, A.: How to play any mental game, or a completeness theorem for protocols with an honest majority. In: *The 19th Annual ACM Symposium on Theory of Computing, STOC 1987*, pp. 218–229 (1987)
14. Hirt, M., Maurer, U.: Player simulation and general adversary structures in perfect multiparty computation. *J. Cryptology* **13**(1), 31–60 (2000)
15. Ito, M., Saito, A., Nishizeki, T.: Secret sharing scheme realizing general access structure. In: *IEEE Global Telecommunications Conference, Globecom 1987*, pp. 99–102 (1987)
16. Ishai, Y., Kushilevits, E.: Randomizing polynomials: a new representation with applications to round-efficient secure computation. In: *The 41st Annual Symposium on Foundations of Computer Science (FOCS2000)*, pp. 294–304 (2000)
17. Ishai, Y., Ostrovsky, R., Seyalioglu, H.: Identifying cheaters without an honest majority. In: Cramer, R. (ed.) *TCC 2012*. LNCS, vol. 7194, pp. 21–38. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-28914-9_2
18. Liu, M., Xiao, L., Zhang, Z.: Multiplicative linear secret sharing schemes based on connectivity of graphs. *IEEE Trans. Inf. Theory* **53**(11), 3973–3978 (2007)
19. Maurer, U.: Secure multi-party computation made simple. In: Cimato, S., Persiano, G., Galdi, C. (eds.) *SCN 2002*. LNCS, vol. 2576, pp. 14–28. Springer, Heidelberg (2003). https://doi.org/10.1007/3-540-36413-7_2
20. Hirt, M., Tschudi, D.: Efficient general-adversary multi-party computation. In: Sako, K., Sarkar, P. (eds.) *ASIACRYPT 2013*. LNCS, vol. 8270, pp. 181–200. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-42045-0_10
21. Patra, A., Choudhary, A., Rabin, T., Rangan, C.P.: The round complexity of verifiable secret sharing revisited. In: Halevi, S. (ed.) *CRYPTO 2009*. LNCS, vol. 5677, pp. 487–504. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-03356-8_29
22. Rabin, T., Ben-Or, M.: Verifiable secret sharing and multiparty protocols with honest majority. In: *The 21st Annual ACM Symposium on Theory of Computing, STOC 1989*, pp. 73–85 (1989)
23. Rogaway, P., Bellare, M.: Robust computational secret sharing and a unified account of classical secret-sharing goals. In: *The 14th ACM Conference on Computer and Communications Security, CCS 2007*, pp. 172–184 (2007)
24. Shamir, A.: How to share a secret. *Commun. ACM* **22**(11), 612–613 (1979)
25. Yao, A.C.: Protocols for secure computations. In: *The 23rd Annual Symposium on Foundations of Computer Science, FOCS 1982*, pp. 160–164 (1982)
26. Yoshida, M., Fujiwara, T.: On the impossibility of d -multiplicative non-perfect secret sharing. *IEICE Trans.* **98–A**(2), 767–770 (2015)

Round and Communication Efficient Unconditionally-Secure MPC with $t < n/3$ in Partially Synchronous Network

Ashish Choudhury^{1(✉)}, Arpita Patra², and Divya Ravi²

¹ International Institute of Information Technology, Bangalore, India
ashish.choudhury@iiitb.ac.in

² Indian Institute of Science, Bangalore, India
{arpita,divyar}@iisc.ac.in

Abstract. In this work, we study unconditionally-secure multi-party computation (MPC) tolerating $t < n/3$ corruptions, where n is the total number of parties involved. In this setting, it is well known that if the underlying network is *completely asynchronous*, then one can achieve only *statistical security*; moreover it is *impossible* to ensure *input provision* and consider inputs of all the honest parties. The best known statistically-secure asynchronous MPC (AMPC) with $t < n/3$ requires a communication of $\Omega(n^5)$ field elements per multiplication. We consider a *partially synchronous* setting, where the parties are assumed to be globally synchronized initially for few rounds and then the network becomes completely asynchronous. In such a setting, we present a MPC protocol, which requires $\mathcal{O}(n^2)$ communication per multiplication while ensuring input provision. Our MPC protocol relies on a new four round, communication efficient statistical *verifiable secret-sharing* (VSS) protocol with broadcast communication complexity *independent* of the number of secret-shared values.

1 Introduction

Threshold unconditionally-secure multiparty computation (MPC) is a fundamental problem in secure distributed computing [2, 8, 12, 26, 36, 38]. Informally, an MPC protocol enables a set of n mutually distrusting parties to jointly and securely compute a publicly known function f of their private inputs over some finite field \mathbb{F} , even in the presence of a *computationally unbounded active adversary* Adv , who can corrupt any t out of the n parties. Let the parties be connected by pair-wise secure (private and authentic) channels. Then in the *synchronous* communication setting, where the parties are assumed to be synchronized through a global clock, it is known that *perfectly-secure* MPC is possible if and only if $t < n/3$ [8]. If a common broadcast channel is also available to the parties

A. Choudhury—Financial support from Infosys Foundation acknowledged.

A. Patra—Work partially supported by INSPIRE Faculty Fellowship (DST/INSPIRE/04/2014/015727) from Department of Science & Technology, India.

in addition to the pair-wise secure channels, then one can tolerate upto $t < n/2$ corruptions, albeit with *statistical security*¹ [36]. The resilience bounds become different if one considers a completely *asynchronous* setting, where parties are not synchronized and messages can be arbitrarily delayed. Specifically, perfectly-secure asynchronous MPC (AMPC) is possible if and only if $t < n/4$ [7], while statistically-secure AMPC is possible if and only if $t < n/3$ [9].

Feasibility Results for Unconditionally-secure MPC: In any general MPC protocol [2–5, 8, 10, 12, 15, 20, 27, 36], the function f is usually expressed as an arithmetic circuit (consisting of addition and multiplication gates) over \mathbb{F} and then the protocol “securely” evaluates each gate in the circuit in a shared/distributed fashion. More specifically, each party secret-shares its inputs among the parties using a linear secret-sharing scheme (LSS) [17], say Shamir [37], with threshold² t . The parties then interact to maintain the following invariant for each gate: *given the gate inputs in a secret-shared fashion, the gate output is computed in a secret-shared fashion*. Finally the (shared) circuit output is publicly reconstructed. Intuitively, the privacy follows since each intermediate value in the above process remains secret-shared with threshold t . Due to the *linearity* of the LSS, the addition (linear) gates are evaluated *locally* by the parties. However, maintaining the above invariant for the multiplication (non-linear) gates requires interaction among the parties. The focus therefore is rightfully placed on measuring the communication complexity (namely the *total* number of field elements communicated) required to evaluate the multiplication gates in the circuit. In the recent past, a lot of work has been done to design communication-efficient MPC protocols; we summarize the relevant works here.

With $t < n/3$, [5] presents a perfectly-secure MPC protocol with $\mathcal{O}(n)$ amortized communication complexity³ per multiplication, while [10] presents a statistically-secure MPC protocol with $t < n/2$ with almost $\mathcal{O}(n)$ communication complexity per multiplication. Both these results are in the synchronous setting and require *non-constant* number of rounds of interaction among the parties. While the protocol of [5] requires $\Theta(n + \mathcal{D})$ rounds, the protocol of [10] requires $\Theta(n^2 + \mathcal{D})$ rounds, where \mathcal{D} denotes the *multiplicative depth* of the circuit.

A major drawback of the synchronous setting is that it does not model real life networks like the Internet accurately where it is very hard to ensure that the users are synchronized through a global clock and that there exists a strict

¹ The outcome of a perfectly-secure protocol is error-free, while a negligible error is allowed in a statistically-secure protocol.

² Informally such a scheme ensures that the shared value remains information-theoretically secure even if upto t shares are revealed. Shamir sharing of a secret with threshold t is done by selecting a random polynomial of degree at most t with the secret as the constant term and defining the individual shares as distinct evaluations of the polynomial.

³ The amortized communication complexity is derived under the assumption that the circuit is large enough so that the terms that are independent of the circuit size can be ignored.

a priori known upper bound on the message delivery. Real life networks can be modelled more appropriately by the asynchronous setting, where there are no known upper bounds and messages are delivered arbitrarily (the only guarantee given in this model is that the messages sent by the honest parties will reach to their destination eventually). Hence designing AMPC protocols is practically motivated. However, an inherent challenge in designing protocols in a completely asynchronous setting is that it is impossible to distinguish between a *slow*, but honest party (whose messages are delayed arbitrarily) and a corrupt party (who do not send any message at all). Hence in a completely asynchronous protocol, no party can afford to receive messages from all the n parties, as the wait may turn out to be an endless wait. So as soon a party receives messages from $n-t$ parties, it has to proceed to the next “step” of the protocol. However, in this process, messages from t potentially honest, but slow parties may get ignored. Due to this inherent phenomena, designing efficient AMPC protocols is a challenging task, as evident from the known feasibility results for AMPC protocols summarized below.

In a completely asynchronous setting, [34] presents a perfectly-secure AMPC protocol with $t < n/4$ and $\mathcal{O}(n^2)$ communication per multiplication, while [32] presents a statistically-secure AMPC with $t < n/3$ and $\mathcal{O}(n^5)$ communication per multiplication. As it is clear, there is a significant gap in the communication complexity of MPC and AMPC protocols. In addition, any AMPC protocol cannot ensure *input provision*, namely the inputs of *all* the honest parties may not be considered for the circuit evaluation, as this may turn out to be an endless wait and so inputs of upto t potentially honest parties may get ignored. With an aim to bridge the gap in the communication complexity of synchronous and asynchronous MPC and to enforce input provision, the works of [4, 14] motivate and consider *hybrid* asynchronous setting, where the network is assumed to be synchronized for few initial rounds and then it becomes completely asynchronous. This is a practically motivated communication setting, which has been well considered in the recent past for bridging the efficiency gap between synchronous and asynchronous protocols for various distributed computing tasks [4, 6, 14, 23, 30].

With $t < n/4$, a perfectly-secure hybrid MPC protocol with one synchronous round is presented in [14], with $\mathcal{O}(n)$ amortized communication complexity per multiplication. In [15], four MPC protocols in the hybrid setting are proposed with $t < n/3$; while two of these protocols are perfectly-secure, the remaining two are statistically-secure. These protocols are obtained by instantiating the efficient framework for unconditionally-secure MPC proposed in [15] with existing VSS schemes with $t < n/3$ (more on this later). Among the perfectly-secure protocols, the first one requires less number of synchronous rounds, namely⁴ (12, 3), but requires a higher communication of $\mathcal{O}(n^5)$ per multiplication. The second perfectly-secure protocol requires more number of synchronous

⁴ We say a protocol requires (r, r') (synchronous) rounds, if it requires a total of r rounds of interaction among the parties and out of these r rounds, r' rounds require broadcast by the parties, where $r' \leq r$.

rounds, namely (21, 7), but provides a better communication complexity of $\mathcal{O}(n^4)$ per multiplication. So a tradeoff is attained between the amount of synchrony required and communication achieved per multiplication. The statistically-secure hybrid protocols of [15] with $t < n/3$ retain the same communication complexity as their perfect counterparts, but reduces the number of synchronous rounds. Namely the first statistically-secure protocol requires (7, 2) rounds and $\mathcal{O}(n^5)$ communication per multiplication, the second statistically-secure protocol requires (16, 6) rounds and $\mathcal{O}(n^4)$ communication per multiplication. As it is clear from these results, with $t < n/3$, significant improvement in the communication complexity is not achieved, even if partial synchrony is provided in the network. Our goal is to design more efficient hybrid MPC protocol with $t < n/3$ using minimal level of synchrony.

Our Results. We present a hybrid MPC protocol with $t < n/3$. Our protocol is *statistically-secure*, requires (4, 3) synchronous rounds and $\mathcal{O}(n^2)$ communication per multiplication. Moreover, our protocol also ensures input provision. Our protocol outperforms the existing hybrid MPC protocols with $t < n/3$, both in terms of communication complexity as well as in terms of the number of synchronous rounds required in the protocol.

To design our protocol, we follow the standard offline-online paradigm, based on Beaver’s circuit-randomization technique [2] and which is now the de facto style of designing efficient MPC protocols [3–5, 10, 14, 15]. In this paradigm, an MPC protocol is divided into two phases, a *circuit-independent* offline phase and a *circuit-dependent* online phase. While the offline phase generates “raw data”, independent of the circuit and actual inputs for the computation, the online phase utilizes this raw data for the circuit evaluation. In a more detail, the offline phase generates random *multiplication triples* of the form (a, b, c) , Shamir-shared with threshold t ; here a, b are random and private and $c = ab$ holds. Later, using such triples, multiplication gates are evaluated in a shared fashion. For each multiplication gate, one multiplication triple from the offline phase is utilized and the multiplication gate is evaluated at the cost of publicly reconstructing two Shamir-shared values. Reconstructing a Shamir-shared valued (with threshold t) can be done efficiently with $t < n/3$ using the standard Reed-Solomon (RS) error correction [31], even in a *completely asynchronous* setting [7, 11]. Hence we shift the focus to design an efficient offline phase in the hybrid setting for generating multiplication triples. For this we follow the recent framework of [15], which shows how to efficiently generate Shamir-shared multiplication triples in offline phase, using *any* (polynomial based) *verifiable secret-sharing* (VSS) protocol [13] as a black-box. Informally, a VSS protocol allows a designated party called *dealer* (D) to *verifiably* Shamir-share a secret with threshold t . Thus at the end of the VSS protocol it is ensured that there exists some polynomial of degree at most t with the secret as the constant term and every share-holder has a distinct evaluation of this polynomial. Moreover this is ensured irrespective of whether the dealer is under the influence of the adversary or not. In addition, if the dealer is *honest* then it is ensured that the secret remains information-theoretically secure from t corrupted share-holders.

In this work, our proposed VSS protocol in the setting of $t < n/3$ is plugged into the framework of [15] and the result is a more efficient hybrid MPC protocol. Communication-wise, our VSS protocol stands out with an amortized overhead of $\mathcal{O}(n^2)$ per secret-shared value, whereas the best known bound is only $\mathcal{O}(n^3)$ [25, 28]. The improvement comes from the fact that our VSS protocol requires a broadcast complexity that is *independent* of the number of secrets shared, a property that is not achieved by the known constructions [25, 28]. To induce a better complexity over point-to-point channels, we use the best known *broadcast amplification* protocols (aka multi-valued broadcast protocols) [22] to simulate the broadcast invocations in the VSS protocols of [25, 28]. Informally, in a multi-valued protocol, broadcasting a “sufficiently large” message of size ℓ has communication complexity of $\mathcal{O}(n\ell)$ over point-to-point channels and a broadcast complexity of $\text{poly}(n)$. With $t < n/3$, the most efficient multi-valued broadcast protocol is due to [35]. The protocol requires a communication complexity of $\mathcal{O}(n\ell)$ over point-to-point channels and broadcast of n^2 bits for broadcasting an ℓ -bit message. Detailed analysis and comparison of our VSS with existing ones is deferred to the full version of the paper. In Fig. 1, we compare our MPC and VSS protocols with their previous best counter parts.

(a) Communication complexity (in bits) per multiplication of various AMPC and hybrid MPC protocols with $t < n/3$

Security	Underlying Network		VSS Deployed in the Offline Phase	Communication Complexity
	Offline Phase	Online Phase		
Statistical	Asynchronous	Asynchronous	[34]	$\mathcal{O}(n^5 \log \mathbb{F})$ [34]
Perfect	Hybrid (12, 3) rounds	Asynchronous	[28]	$\mathcal{O}(n^5 \log \mathbb{F})$ [15]
Perfect	Hybrid (21, 7) rounds	Asynchronous	[25]	$\mathcal{O}(n^4 \log \mathbb{F})$ [15]
Statistical	Hybrid (7, 2) rounds	Asynchronous	[28]	$\mathcal{O}(n^5 \log \mathbb{F})$ [15]
Statistical	Hybrid (16, 6) rounds	Asynchronous	[25]	$\mathcal{O}(n^4 \log \mathbb{F})$ [15]
Statistical	Hybrid (4, 3) rounds	Asynchronous	[This work]	$\mathcal{O}(n^2 \log \mathbb{F})$ [This work]

(b) Amortized communication complexity per shared-secret of the underlying VSS deployed in the offline phase.

Security	Network Type	Overhead
Statistical	Asynchronous	$\mathcal{O}(n^4)$ [34]
Perfect	Synchronous (7, 2) rounds	$\mathcal{O}(n^4)$ [28]
Perfect	Synchronous (16, 6) rounds	$\mathcal{O}(n^3)$ [25]
Statistical	Synchronous (4, 3) rounds	$\mathcal{O}(n^2)$ [This work]

Fig. 1. Comparison of our results with previous best results.

Other Related Work. In the *synchronous* setting, MPC protocols with $\mathcal{O}(n)$ communication per multiplication has been reported in [5] with perfect security and $t < n/3$ and in [10] with statistical security and $t < n/2$. These protocols deploy *non-robust* secret-sharing protocols in the player-elimination and dispute-control framework. The non-robustness of the underlying primitives inflates the round complexity of their offline phase to $\mathcal{O}(n)$ and $\mathcal{O}(n^2)$ respectively. The naive approach of adopting these protocols in hybrid setting will lead to protocols with $\mathcal{O}(n)$ or $\mathcal{O}(n^2)$ synchronous (broadcast) rounds to execute the offline phase. The online phase of these protocols can be executed asynchronously. Our hybrid

MPC protocol on the other hand requires only a constant number of synchronous broadcast rounds.

The reported works [18, 19] in the synchronous setting with polylogarithmic (in n) communication per gate (denoted as $\tilde{\mathcal{O}}(n)$)⁵ are only *non-optimally* resilient. While [19] works with $t < (\frac{1}{2} - \epsilon)n$ and provides statistical security, [18] works with $t < (\frac{1}{3} - \epsilon)n$ and provides perfect security, where $\epsilon > 0$. These protocols also evaluate the underlying circuit in a secret-shared fashion. However, instead of Shamir secret-sharing, they use packed secret-sharing [24] taking advantage of the presence of larger subset of honest parties (due to the non-optimal resilience). Due to the use of packed secret-sharing, “multiple” gates can be evaluated simultaneously by doing a fixed set of operations on the shares. However, this requires “special” structure from the underlying circuit being available at each layer, maintaining which, demands additional circuitry to be incorporated between different layers of the circuit. Evaluating the overall circuit using packed secret-sharing makes these protocols highly non-trivial and complex. It is not known how to adapt these protocols in a completely asynchronous or a partially synchronous setting. Specifically, it is not clear whether these protocols can be executed in a hybrid setting, with a *constant* number of synchronous rounds. Therefore, while treating VSS as an MPC functionality and evaluating the resultant “VSS circuit” using the MPC protocols of [18, 19] may lead to sublinear (namely $\tilde{\mathcal{O}}(n)$) overhead per secret-shared value, it is not clear if the resultant protocols runs with a *constant* number of synchronous rounds in hybrid setting.

New Techniques. Our VSS protocol is built upon a new primitive called *information checking with succinct proof of possession* (ICPoP) that takes motivation from *information checking protocol* (ICP) introduced in [16, 33, 36]. An ICP allows a D to privately authenticate some data for an intermediary INT, who can later publicly reveal this data and prove that it originated from D. On the other hand, in an ICPoP protocol INT gives a proof of possession publicly of the data originated from D, instead of publicly revealing the data. The proof preserves data privacy and is “succinct” i.e. its size is *independent* of the size of the data. The succinctness of the proof makes the broadcast complexity of our VSS protocol independent of the number of shared secrets. Our ICPoP also offers *transferability* that allows any designated party to take possession of INT’s authenticated (by D) data and to be able to give a proof of possession on the “behalf” of INT. The existing ICPs do not support transferability.

We next give a high level overview of our VSS. To share a secret s , we embed s in the constant term of a random bivariate polynomial $F(x, y)$ of degree t in x and y . Every party P_i then obtains a *row polynomial* $f_i(x) = F(x, \alpha_i)$. The parties then publicly verify whether the row polynomials of at least $n - t$ parties called VCORE define a unique bivariate polynomial. The standard way to do this is to perform the “pair-wise checking”, where every pair of parties (P_i, P_j)

⁵ The actual complexity (communication, computation and round) of these protocols are of the form $\mathcal{O}((\log^k n \cdot \text{poly}(\log |C|)) \cdot |C|) + \mathcal{O}(\text{poly}(n, \log |C|, \mathcal{D}))$, where \mathcal{D} is the multiplicative depth of the underlying circuit C .

is asked to verify the consistency of the common values on their respective polynomials and publicly complain if there is any inconsistency, in which case D publicly resolves the complaint by making the common value public [21, 25, 28]. This approach will lead to a broadcast complexity of $\mathcal{O}(n^2)$ per secret-shared value; instead we use a statistical protocol called Poly-Check (Sect. 4.1), adapted from [34], which performs the same task in parallel for ℓ secrets (and hence ℓ bivariate polynomials), but keeping the broadcast complexity independent of ℓ . Once VCORE is found, it is ensured that D has committed a unique $F(x, y)$ and the secret $F(0, 0)$ to the parties in VCORE. To enable the parties to obtain their shares, the goal will be to enable each party P_j to compute its *column polynomial* $g_j(y) = F(\alpha_j, y)$. For this each party $P_i \in \text{VCORE}$ transfers its common value on $g_j(y)$ (namely $f_i(\alpha_j)$) to P_j . To ensure that correct values are transferred, P_j publicly gives a proof of possession of all the transferred values originated from D via the intermediary parties in VCORE. This is done in parallel for ℓ secrets (and hence ℓ bivariate polynomials); the succinctness of the proof ensures that this step has broadcast complexity, independent of ℓ .

2 Network Model, Definitions and Existing Tools

We consider a set $\mathcal{P} = \{P_1, \dots, P_n\}$ of n parties, connected by pair-wise private and authentic channels. For simplicity we assume $n = 3t + 1$, so $t = \Theta(n)$. There exists a computationally unbounded adversary Adv who can maliciously corrupt any t parties and may force them to behave in any arbitrary fashion during the execution of a protocol. We assume the adversary to be static, who decides the set of corrupted parties at the beginning of the protocol execution. We assume a partially synchronous network, where the first four rounds are synchronous, after which the entire communication is done *asynchronously*. Moreover, we assume that the parties have access to a broadcast channel during the second, third and fourth synchronous round. Our protocols operate over a finite field \mathbb{F} , where $|\mathbb{F}| > 2n$. We assume that there exists $2n$ distinct non-zero elements $\alpha_1, \dots, \alpha_n, \beta_1, \dots, \beta_n$ in \mathbb{F} . Each element of \mathbb{F} can be represented by $\mathcal{O}(\log |\mathbb{F}|)$ bits. The communication complexity of any protocol is defined to be the total number of field elements communicated by the *honest* parties in that protocol. We denote the point-to-point communication complexity by $\mathcal{PC}()$ and the broadcast communication complexity as $\mathcal{BC}()$.

Without loss of generality, we assume that the parties want to securely compute the function $f : \mathbb{F}^n \rightarrow \mathbb{F}$ via an MPC protocol, where $f(x_1, \dots, x_n) = y$, such that $x_i \in \mathbb{F}$ is the input of P_i and every party is supposed to receive the output $y \in \mathbb{F}$. The function f is assumed to be represented by a publicly known arithmetic circuit C over \mathbb{F} . The circuit C consists of n input gates, two-input addition (linear) and multiplication (non-linear) gates, zero-input random gates (for generating random values during the computation) and one output gate. We denote by c_M and c_R the number of multiplication and random gates in C respectively. By $[X]$ and $[X, Y]$ for $Y \geq X$, we denote the sets $\{1, \dots, X\}$ and $\{X, X + 1, \dots, Y\}$, respectively. We use $i \in [k]$ to denote that i can take a value

from the set $\{1, 2 \dots k\}$. We will also require that $|\mathbb{F}| > 4n^4(c_M + c_R)(3t + 1)2^\kappa$ to ensure that the error-probability of our MPC protocol is at most $2^{-\kappa}$, for a given error parameter κ .

2.1 Definitions

Definition 1 (*d*-sharing [3, 5, 20]). A value $s \in \mathbb{F}$ is said to be *d*-shared if there exists a polynomial over \mathbb{F} , say $f(\cdot)$, of degree at most d , such that $f(0) = s$ and every (honest) party $P_i \in \mathcal{P}$ holds a share s_i of s , where $s_i = f(\alpha_i)$. We denote by $[s]_d$, the vector of shares of s corresponding to the (honest) parties in \mathcal{P} .

A vector $\mathbf{S} = (s^{(1)}, \dots, s^{(\ell)}) \in \mathbb{F}^\ell$ is said to be *d*-shared if each $s^{(i)}$ is *d*-shared. Note that *d*-sharings are *linear*: given $[a]_d$ and $[b]_d$, then $[a + b]_d = [a]_d + [b]_d$ and $[c \cdot a]_d = c \cdot [a]_d$ holds, for a public constant c . In general, given ℓ sharings $[x^{(1)}]_d, \dots, [x^{(\ell)}]_d$ and a public linear function $g : \mathbb{F}^\ell \rightarrow \mathbb{F}^m$, where $g(x^{(1)}, \dots, x^{(\ell)}) = (y^{(1)}, \dots, y^{(m)})$, then $g([x^{(1)}]_d, \dots, [x^{(\ell)}]_d) = ([y^{(1)}]_d, \dots, [y^{(m)}]_d)$. We say that the parties *locally compute* $([y^{(1)}]_d, \dots, [y^{(m)}]_d) = g([x^{(1)}]_d, \dots, [x^{(\ell)}]_d)$ to mean that every P_i (locally) computes $(y_i^{(1)}, \dots, y_i^{(m)}) = g(x_i^{(1)}, \dots, x_i^{(\ell)})$, where $y_i^{(l)}$ and $x_i^{(l)}$ denotes the i^{th} share of $y^{(l)}$ and $x^{(l)}$ respectively.

Definition 2 (Polynomial-based) Verifiable Secret Sharing (VSS) [3–5]).

Let $\mathbf{S} = (s^{(1)}, \dots, s^{(L)}) \in \mathbb{F}^L$ be a set of L values that a dealer $D \in \mathcal{P}$ wants to *t*-share among \mathcal{P} . Let Sh be a protocol for the n parties, where D has the input \mathbf{S} . Then Sh is a VSS scheme if the following holds for every possible Adv , on all possible inputs: **(1) Correctness:** If D is honest then \mathbf{S} is *t*-shared among \mathcal{P} at the end of Sh . Moreover even if D is corrupted there exists a set of L values, say $(\bar{s}^{(1)}, \dots, \bar{s}^{(L)})$, which is *t*-shared among \mathcal{P} at the end of Sh . **(2) Privacy:** If D is honest then Sh reveals no information about \mathbf{S} to Adv in the information-theoretic sense; i.e. Adv 's view is identically distributed for all possible \mathbf{S} .

If Sh satisfies all its properties without any error then it is called *perfectly-secure*. If the correctness is satisfied with probability at least $1 - \epsilon$, for a given error parameter ϵ , then it is called *statistically-secure*.

Unconditionally-secure MPC: Recent papers on efficient unconditionally-secure MPC follow a simpler “property based” security definition of secure MPC [3, 5, 10, 20], instead of the more rigorous “real-world/ideal-world” paradigm based definition [1, 29]. As our main goal is to provide an efficient VSS and MPC, to avoid blurring the main focus of the paper and to avoid additional technicalities, we also use the property based security definition. However, we confirm that using standard techniques, like the above efficient protocols, our MPC protocol can be also proved secure according to the simulation based definition. We defer the details to the full version of the paper.

Let $f : \mathbb{F}^n \rightarrow \mathbb{F}$ be a publicly known function and party P_i has input $x_i \in \mathbb{F}$. In any (unconditionally-secure) multiparty computation, each party P_i *t*-shares

its input. Let x_i be the value shared by P_i . If P_i is honest then $x_i = x_i$. The parties then compute f as $y = f(x_1, \dots, x_n)$ and everyone receives y .

Definition 3 (Unconditionally-secure MPC). *A protocol Π among the n parties securely computes f , if it satisfies the following for every possible Adv, on all possible inputs: (1) **Correctness:** All honest parties obtain y at the end of Π . (2) **Privacy:** Adv obtains no additional information about the inputs of the honest parties, other than what is inferred from the inputs of the corrupted parties and y . Protocol Π is called perfectly-secure if it satisfies all its properties without any error. If the correctness is satisfied with probability at least $1 - 2^{-\kappa}$, for a given error parameter κ , then Π is called statistically-secure.*

Information Checking with Succinct Proof of Possession (ICPoP): An ICPoP protocol involves three entities: a designated dealer $D \in \mathcal{P}$ who holds a set of L private values $\mathcal{S} = \{s^{(1)}, \dots, s^{(L)}\}$, an intermediary $\text{INT} \in \mathcal{P}$ and the set of parties \mathcal{P} acting as verifiers (note that D and INT will also play the role of verifiers, apart from their designated role of dealer and intermediary respectively). The protocol proceeds in three phases, each of which is implemented by a dedicated sub-protocol: (1) **Distribution Phase:** Here D , sends \mathcal{S} to INT along with some *auxiliary information*. For the purpose of verification, some *verification information* is additionally sent to each individual verifier. (2) **Authentication Phase:** This phase is initiated by INT who interacts with D and the verifiers to ensure that the information it received from D is consistent with the verification information distributed to the individual verifiers. If D wants it can publicly abort this phase, which is interpreted as if D is accusing INT of malicious behaviour. (3) **Revelation Phase:** This phase is carried out by INT and the verifiers in \mathcal{P} only if D has not aborted the previous phase. Here INT reveals a proof of possession of the values received from D . The verifiers in \mathcal{P} check this proof with respect to their verification information. Then based on certain criteria, each verifier either outputs `AcceptProof` (indicating that it accepts the proof) or `RejectProof` (indicating that it rejects the proof).

Definition 4 (ICPoP). *A triplet of protocols ($\text{Distr}, \text{AuthVal}, \text{RevealPoP}$) (implementing the distribution, authentication and revelation phase respectively) is a $(1 - \epsilon)$ -secure ICPoP, for an error parameter ϵ , if the following holds: (1) **ICPoP-Correctness1:** If D and INT are honest, then each honest verifier $P_i \in \mathcal{P}$ outputs `AcceptProof` at the end of `RevealPoP`. (2) **ICPoP-Correctness2:** If D is corrupted and INT is honest and if ICPoP proceeds to `RevealPoP`, then except with probability at most ϵ , all honest verifiers output `AcceptProof` at the end of `RevealPoP`. (3) **ICPoP-Correctness3:** If D is honest, INT is corrupted, ICPoP proceeds to `RevealPoP` and if the honest verifiers output `AcceptProof`, then except with probability at most ϵ , the proof produced by INT corresponds⁶ to the values in \mathcal{S} . (4) **ICPoP-Privacy:** If D and INT are honest, then information obtained by Adv during ICPoP is independent of \mathcal{S} .*

⁶ The interpretation of a proof corresponding to a set of values will be clear later during the formal presentation of our ICPoP.

(5) ICPoP-Succinctness of the Proof: *The size of the proof produced by INT during RevealPoP is independent of L .*

Properties of Polynomials: A bivariate polynomial $F(x, y)$ of degree at most t is of the form $F(x, y) = \sum_{i,j=0}^{i,j=t} r_{ij} x^i y^j$, where $r_{ij} \in \mathbb{F}$. Let $f_i(x) \stackrel{\text{def}}{=} F(x, \alpha_i)$, $g_i(y) \stackrel{\text{def}}{=} F(\alpha_i, y)$ for $i \in [n]$. We call $f_i(x)$ and $g_i(y)$ as *i th row polynomial* and *column polynomial* respectively of $F(x, y)$. We say that a row polynomial $\bar{f}_i(x)$ lies on a bivariate polynomial $F(x, y)$ of degree at most t if $F(x, \alpha_i) = \bar{f}_i(x)$ holds. Similarly we will say that a column polynomial $\bar{g}_i(y)$ lies on $F(x, y)$ if $F(\alpha_i, y) = \bar{g}_i(y)$ holds. We will use the following well known standard properties of bivariate and univariate polynomials.

Lemma 1 ([1, 11, 34]). *Let $f_1(x), \dots, f_\ell(x), g_1(y), \dots, g_\ell(y)$ be degree t univariate polynomials with $t + 1 \leq \ell \leq n$, such that $f_i(\alpha_j) = g_j(\alpha_i)$ holds for every $\alpha_i, \alpha_j \in \{\alpha_1, \dots, \alpha_\ell\}$. Then there exists a unique bivariate polynomial $\bar{F}(x, y)$ of degree t , such that $f_i(x)$ and $g_i(y)$ lie on $\bar{F}(x, y)$, for $i \in [\ell]$.*

Lemma 2 ([1, 11, 34]). *Let $f_1(x), \dots, f_\ell(x)$ be univariate polynomials of degree at most t where $t + 1 \leq \ell \leq n$. Let $F(x, y)$ and $G(x, y)$ be two bivariate polynomials of degree at most t , such that $f_i(x)$ lies on both $F(x, y)$ and $G(x, y)$ for each $i \in [\ell]$. Then $F(x, y) = G(x, y)$.*

Lemma 3 ([34]). *Let $G^{(1)}(x), \dots, G^{(L)}(x)$ be degree d polynomials and let $A(x) \stackrel{\text{def}}{=} eG^{(1)}(x) + \dots + e^L G^{(L)}(x)$, where e is a random value from $\mathbb{F} \setminus \{0\}$. Let a tuple $(\gamma, v_1, v_2, \dots, v_L)$ be such that $v_i \neq G^{(i)}(\gamma)$ for some $i \in [L]$. Then except with probability at most $\frac{L-2}{|\mathbb{F}|-1}$, the condition $A(\gamma) \neq ev_1 + \dots + e^L v_L$ holds.*

Lemma 4 ([34]). *Let $h^{(0)}(y), \dots, h^{(L)}(y)$ be $L+1$ polynomials and r be a random value from $\mathbb{F} \setminus \{0\}$. Let $h_{\text{com}}(y) \stackrel{\text{def}}{=} h^{(0)}(y) + rh^{(1)}(y) + \dots + r^L h^{(L)}(y)$. If at least one of $h^{(0)}(y), \dots, h^{(L)}(y)$ has degree more than t , then except with probability at most $\frac{L}{|\mathbb{F}|}$, the polynomial $h_{\text{com}}(y)$ will have degree more than t .*

3 Efficient ICPoP

We present a $(1 - \epsilon)$ -secure ICPoP protocol, where $|\mathcal{S}| = L = \ell \times \text{pack}$, with $\ell \geq 1$ and $1 \leq \text{pack} \leq n - t$; moreover $\epsilon = \max\{\frac{n\ell}{|\mathbb{F}|-1}, \frac{n(n-1)}{|\mathbb{F}|-\text{pack}}\}$. The protocol has communication complexity $\mathcal{PC}(\mathcal{O}(n\ell))$ and $\mathcal{BC}(\mathcal{O}(n))$. Hence the broadcast complexity is *independent* of ℓ . Our ICPoP is similar to the asynchronous ICP of [33], adapted to the synchronous setting with the following differences: in ICP the whole \mathcal{S} is revealed during the revelation phase, as only its authenticity is required during the revelation phase. We require INT to be able to publicly prove the possession of \mathcal{S} while maintaining its privacy. Hence the auxiliary information distributed in our ICPoP differs and also used differently; the details follow.

Let $\mathcal{S} = \{(s^{(1,1)}, \dots, s^{(1,\text{pack})}), \dots, (s^{(\ell,1)}, \dots, s^{(\ell,\text{pack})})\}$. During the distribution phase, D embeds the values $(s^{(k,1)}, \dots, s^{(k,\text{pack})})$ for $k \in [\ell]$ in a random degree d *secret-encoding* polynomial $G^{(k)}(x)$ at $x = \beta_1, \dots, \beta_{\text{pack}}$, where $d = \text{pack} + t - 1$. In addition, D picks a *masking set* \mathcal{M} , consisting of $2 \cdot \text{pack}$ random values $\{(m^{(1,1)}, \dots, m^{(1,\text{pack})}), (m^{(2,1)}, \dots, m^{(2,\text{pack})})\}$, which are embedded in two random degree d polynomials $H^{(1)}(x)$ and $H^{(2)}(x)$ respectively at $x = \beta_1, \dots, \beta_{\text{pack}}$; we call these polynomials as *masking polynomials*. The polynomials are sent to INT , while each verifier P_i receives the values $v_{1,i}, \dots, v_{\ell,i}, m_{1,i}, m_{2,i}$ of these polynomials at a secret evaluation point γ_i . This distribution achieves **ICPoP-Privacy**, as each secret-encoding polynomial has degree d and adversary may get at most t values on these polynomials; so it will lack pack values on each polynomial to uniquely interpolate them.

During revelation phase, to give a proof of possession of \mathcal{S} , INT produces a random linear combination of the values in $\mathcal{S} \cup \mathcal{M}$ by making public a random linear combiner, say e and a linear combination $C(x) \stackrel{\text{def}}{=} eH^{(1)}(x) + e^2H^{(2)}(x) + e^3G^{(1)}(x) + \dots + e^{\ell+2}G^{(\ell)}(x)$. The values $C(\beta_1), \dots, C(\beta_{\text{pack}})$ define pack linear combinations of $\mathcal{S} \cup \mathcal{M}$ with respect to e . The pair $(e, C(x))$ is considered as a *proof of possession* of \mathcal{S} (union \mathcal{M}) and verified as follows: each verifier locally verifies if the corresponding linear combination $em_{1,i} + e^2m_{2,i} + e^3v_{1,i} + \dots + e^{\ell+2}v_{\ell,i}$ satisfies $C(x)$ at $x = \gamma_i$ and accordingly broadcast an **Accept** or a **Reject** message. If more than t verifiers broadcast **Accept** then the proof $(e, C(x))$ is said to be accepted, otherwise it is rejected. The proof will always be accepted for an *honest* D and INT , implying **ICPoP-Correctness1**. The size of the proof is $\mathcal{O}(n)$ (as $d = \mathcal{O}(n)$), which is independent of ℓ , implying **ICPoP-Succinctness of the Proof**. No additional information about the secret-encoding polynomials is revealed from $C(x)$, thanks to the masking polynomials. If D is *honest* and INT is *corrupted* then the evaluation points of the honest verifiers will be private. So if INT gives a proof of possession of $\mathcal{S}^* \cup \mathcal{M}^* \neq \mathcal{S} \cup \mathcal{M}$ by revealing a linear combination of $\mathcal{S}^* \cup \mathcal{M}^*$ through $(e, C^*(x))$ where $C^*(x) \neq C(x)$, then with high probability, every honest verifier will reject the proof. This is because the corresponding linear combination of the values possessed by the honest verifiers will fail to satisfy $C^*(x)$; this implies **ICPoP-Correctness 3**.

The above mechanism, however, fails to achieve **ICPoP-Correctness 2**, as a *corrupted* D can distribute “inconsistent” polynomials and values to an *honest* INT and honest verifiers respectively; later on the proof produced by INT will be rejected by every honest verifier. To verify the consistency of the distributed information, during the authentication phase, INT “challenges” D by making public a random linear combination $A(x)$ of the received polynomials. In response, D either instructs to abort the protocol or continue, after verifying whether the $A(x)$ polynomial satisfies the corresponding random linear combination of the values held by each verifier. The idea here is that if D distributed inconsistent data, then with very high probability, any random linear combination of the distributed polynomials would fail to satisfy the corresponding linear combination of the values given to the honest verifiers. And this will be locally learned by the honest verifiers after $A(x)$ is made public. So if D still instructs

to continue the protocol, then clearly D is corrupted; so later even if the proof produced in the revelation phase turns out to be inconsistent with the information held by the honest verifiers, the proof is accepted by adding an additional acceptance condition to deal with this particular case. We stress that the additional acceptance condition never gets satisfied for an *honest* D and a *corrupted* INT. The privacy of the secret-encoding polynomials is still preserved during the authentication phase (for an honest INT and D), thanks to the masking polynomials⁷. The formal steps of ICPoP are given in Fig. 3. In the protocol, if the output is **AcceptProof** then we additionally let the parties output **pack** linear combinations of the values in $\mathcal{S} \cup \mathcal{M}$ possessed by INT; looking ahead this will be useful in our VSS. In Fig. 2 we give a pictorial representation of the values distributed and revealed in ICPoP.

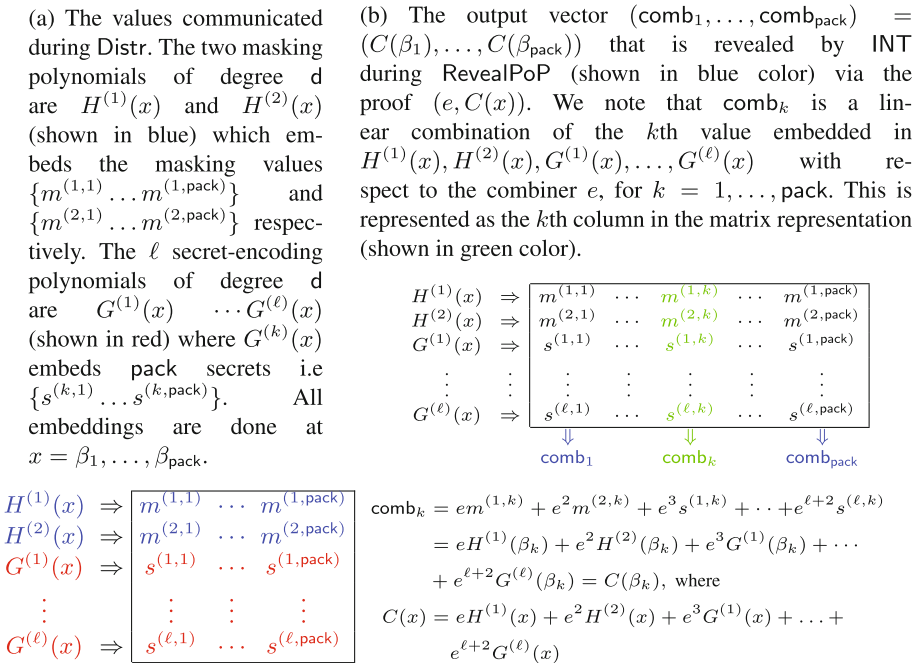


Fig. 2. Pictorial representation of the information generated and communicated during ICPoP protocol.

In ICPoP, the *correspondence* between a proof and a set of values is defined as follows: Let $\mathcal{S} = \{(s^{(1,1)}, \dots, s^{(1,pack)}), \dots, (s^{(\ell,1)}, \dots, s^{(\ell,pack)})\}$ and $\mathcal{M} = \{(m^{(1,1)}, \dots, m^{(1,pack)}), (m^{(2,1)}, \dots, m^{(2,pack)})\}$. We say that a proof $(e, C(x))$ *corresponds* to $\mathcal{S} \cup \mathcal{M}$ if $C(x)$ embeds linear combination of $\mathcal{S} \cup \mathcal{M}$ with respect to e

⁷ This explains the need for two masking polynomials: one is used to preserve the privacy of the secret-encoding polynomials during the authentication phase while the other is used to maintain the privacy during the revelation phase.

at $x = \beta_1, \dots, \beta_{\text{pack}}$; i.e. if $C(\beta_i) = em^{(1,i)} + e^2m^{(2,i)} + e^3s^{(1,i)} + \dots + e^{(\ell+2)}s^{(\ell,i)}$ holds for $i \in [\text{pack}]$.

We state the properties of our ICPoP and the final theorem. We give proofs for the important properties; the other proofs are simple and will appear in the full version.

Lemma 5 (ICPoP-Correctness1). *If D and INT are honest then each honest verifier $P_i \in \mathcal{P}$ outputs `AcceptProof` along with $(C(\beta_1), \dots, C(\beta_{\text{pack}}))$ at the end of `RevealPoP`.*

Lemma 6 (ICPoP-Correctness2). *If D is corrupt and INT is honest, and if ICPoP proceeds to `RevealPoP`, then all honest verifiers output `AcceptProof`, except with probability at most $\frac{n\ell}{|\mathbb{F}|-1}$.*

Proof. We claim that if INT is *honest* and ICPoP proceeds to `RevealPoP`, then an *honest* verifier P_i broadcasts `Accept`, except with probability at most $\frac{\ell}{|\mathbb{F}|-1}$. Assuming that the claim is true, from the union bound it follows that the probability any honest verifier fails to broadcast an `Accept` message is at most $\frac{n\ell}{|\mathbb{F}|-1}$, as the number of honest parties is upper bounded by n . This ensures that there will be more than t `Accept` messages broadcasted by honest verifiers implying that each honest verifier outputs `AcceptProof` at the end of `RevealPoP`.

We next proceed to prove our claim. For this we focus on a designated *honest* verifier P_i and consider the relationship that holds between the polynomials $\overline{G}^{(1)}(x), \dots, \overline{G}^{(\ell)}(x), \overline{H}^{(1)}(x), \overline{H}^{(2)}(x)$ distributed by a *corrupted* D to INT and the tuple $(\overline{\gamma}_i, \overline{v}_{1,i}, \overline{v}_{2,i}, \dots, \overline{v}_{\ell,i}, \overline{m}_{1,i}, \overline{m}_{2,i})$ distributed by D to P_i . We have two cases:

- $\overline{v}_{k,i} = \overline{G}^{(k)}(\overline{\gamma}_i)$ for each $k \in [\ell]$ and $\overline{m}_{1,i} = \overline{H}^{(1)}(\overline{\gamma}_i), \overline{m}_{2,i} = \overline{H}^{(2)}(\overline{\gamma}_i)$: In this case, the claim is true without any error as P_i will find that condition C1 is true for the $C(x)$ polynomial during `RevealPoP`.
- At least one of the following holds — either $\overline{v}_{k,i} \neq \overline{G}^{(k)}(\overline{\gamma}_i)$ for some $k \in [\ell]$ or $\overline{m}_{1,i} \neq \overline{H}^{(1)}(\overline{\gamma}_i)$ or $\overline{m}_{2,i} \neq \overline{H}^{(2)}(\overline{\gamma}_i)$: In this case, $A(\overline{\gamma}_i) \neq d\overline{m}_{1,i} + d^2\overline{m}_{2,i} + d^3\overline{v}_{1,i} + d^4\overline{v}_{2,i} + \dots + d^{\ell+2}\overline{v}_{\ell,i}$ holds, except with probability at most $\frac{\ell}{|\mathbb{F}|-1}$ (follows from Lemma 3 by substituting $L = \ell + 2$). So clearly the verifier P_i will find that condition C2 is true during `RevealPoP`

Lemma 7 (ICPoP-Correctness3). *If D is honest, INT is corrupted, ICPoP proceeds to `RevealPoP` and if the honest verifiers output `AcceptProof`, then except with probability at most $\frac{nd}{|\mathbb{F}|-\text{pack}}$, the proof produced by INT corresponds to the values in $\mathcal{S} \cup \mathcal{M}$.*

Lemma 8 (ICPoP-Privacy). *If D and INT are honest, then the information obtained by Adv during ICPoP is independent of the values in \mathcal{S} .*

Theorem 1. *Protocols (Distr, AuthVal, RevealPoP) constitute a $(1 - \epsilon)$ -secure ICPoP for $L = \ell \times \text{pack}$ values with $\ell \geq 1$ and $1 \leq \text{pack} \leq n - t$, where $\epsilon = \max\{\frac{n\ell}{|\mathbb{F}|-1}, \frac{nd}{|\mathbb{F}|-\text{pack}}\}$ and $d = \text{pack} + t - 1$. The protocol has communication complexity $\mathcal{P}(\mathcal{O}(n\ell))$ and $\mathcal{BC}(\mathcal{O}(n))$.*

$\text{ICPoP}(\mathcal{D}, \text{INT}, \mathcal{P}, \ell, \text{pack}, \mathcal{S}) : \mathcal{S} = \{(s^{(1,1)}, \dots, s^{(1,\text{pack})}), \dots, (s^{(\ell,1)}, \dots, s^{(\ell,\text{pack})})\}$
$\text{Distr}(\mathcal{D}, \text{INT}, \mathcal{P}, \ell, \text{pack}, \mathcal{S} \cup \mathcal{M})$
<p>Round 1:</p> <ul style="list-style-type: none"> - \mathcal{D} defines a <i>masking set</i> $\mathcal{M} \stackrel{\text{def}}{=} \{(m^{(1,1)}, \dots, m^{(1,\text{pack})}), (m^{(2,1)}, \dots, m^{(2,\text{pack})})\}$ consisting of $2 \cdot \text{pack}$ random elements from \mathbb{F}. Let $d \stackrel{\text{def}}{=} \text{pack} + t - 1$. Dealer \mathcal{D} selects ℓ random <i>secret-encoding polynomials</i> $G^{(1)}(x), G^{(2)}(x), \dots, G^{(\ell)}(x)$ of degree at most d, such that $G^{(k)}(\beta_1) = s^{(k,1)}, \dots, G^{(k)}(\beta_{\text{pack}}) = s^{(k,\text{pack})}$ for $k \in [\ell]$. In addition, \mathcal{D} selects two random <i>masking polynomials</i> $H^{(1)}(x), H^{(2)}(x)$ of degree d, such that $H^{(k)}(\beta_1) = m^{(k,1)}, \dots, H^{(k)}(\beta_{\text{pack}}) = m^{(k,\text{pack})}$ for $k \in [2]$. For each verifier $P_i \in \mathcal{P}$, dealer \mathcal{D} selects a random <i>evaluation point</i> γ_i such that $\gamma_i \in \mathbb{F} \setminus \{\beta_1, \dots, \beta_{\text{pack}}\}$. - \mathcal{D} gives $\mathcal{S} \cup \mathcal{M}$ to INT by sending $G^{(1)}(x), \dots, G^{(\ell)}(x), H^{(1)}(x)$ and $H^{(2)}(x)$ to INT. To each verifier $P_i \in \mathcal{P}$, dealer \mathcal{D} sends $(\gamma_i, v_{1,i}, v_{2,i}, \dots, v_{\ell,i}, m_{1,i}, m_{2,i})$, where $v_{k,i} \stackrel{\text{def}}{=} G^{(k)}(\gamma_i)$ for $k \in [\ell]$ and $m_{k,i} \stackrel{\text{def}}{=} H^{(k)}(\gamma_i)$ for $k \in [2]$. <p>Local Computation by INT: Let $\overline{G}^{(1)}(x), \dots, \overline{G}^{(\ell)}(x), \overline{H}^{(1)}(x)$ and $\overline{H}^{(2)}(x)$ be the polynomials received from \mathcal{D} (if \mathcal{D} is honest then these will be the same polynomials as selected by \mathcal{D}). INT sets $\overline{\mathcal{S}} = \{(\overline{s}^{(1,1)}, \dots, \overline{s}^{(1,\text{pack})}), \dots, (\overline{s}^{(\ell,1)}, \dots, \overline{s}^{(\ell,\text{pack})})\}$ and $\overline{\mathcal{M}} = \{(\overline{m}^{(1,1)}, \dots, \overline{m}^{(1,\text{pack})}), (\overline{m}^{(2,1)}, \dots, \overline{m}^{(2,\text{pack})})\}$, where $\overline{s}^{(k,1)} = \overline{G}^{(k)}(\beta_1), \dots, \overline{s}^{(k,\text{pack})} = \overline{G}^{(k)}(\beta_{\text{pack}})$ for $k \in [\ell]$ and $\overline{m}^{(k,1)} = \overline{H}^{(k)}(\beta_1), \dots, \overline{m}^{(k,\text{pack})} = \overline{H}^{(k)}(\beta_{\text{pack}})$ for $k \in [2]$; $\overline{\mathcal{S}} \cup \overline{\mathcal{M}}$ are considered to be <i>received</i> by INT from \mathcal{D}.</p> <p>Local Computation Each Verifier P_i: Let $(\overline{\gamma}_i, \overline{v}_{1,i}, \overline{v}_{2,i}, \dots, \overline{v}_{\ell,i}, \overline{m}_{1,i}, \overline{m}_{2,i})$ be the tuple received from \mathcal{D} (if \mathcal{D} is honest then this will be the same tuple as computed by \mathcal{D}).</p>
$\text{AuthVal}(\mathcal{D}, \text{INT}, \mathcal{P}, \ell, \text{pack}, \overline{\mathcal{S}} \cup \overline{\mathcal{M}})$
<p>Round 1: INT selects a random element $d \in \mathbb{F} \setminus \{0\}$ and broadcasts $(d, A(x))$, where $A(x) \stackrel{\text{def}}{=} d\overline{H}^{(1)}(x) + d^2\overline{H}^{(2)}(x) + d^3\overline{G}^{(1)}(x) + d^4\overline{G}^{(2)}(x) + \dots + d^{\ell+2}\overline{G}^{(\ell)}(x)$.</p> <p>Round 2: Upon receiving $(d, A(x))$ from the broadcast of INT, \mathcal{D} checks if $A(\gamma_i) = dm_{1,i} + d^2m_{2,i} + d^3v_{1,i} + d^4v_{2,i} + \dots + d^{\ell+2}v_{\ell,i}$ holds for every $P_i \in \mathcal{P}$. If not then it broadcasts an Abort message, else it broadcasts an OK message.</p>
<p>RevealPoP($\mathcal{D}, \text{INT}, \mathcal{P}, \ell, \text{pack}, \overline{\mathcal{S}} \cup \overline{\mathcal{M}}$): Executed only if \mathcal{D} broadcasted OK during AuthVal.</p> <p>Round 1: INT chooses a random element $e \in \mathbb{F} \setminus \{0\}$ and broadcasts $(e, C(x))$ as a <i>proof of possession</i> of $\overline{\mathcal{S}} \cup \overline{\mathcal{M}}$, where $C(x) \stackrel{\text{def}}{=} e\overline{H}^{(1)}(x) + e^2\overline{H}^{(2)}(x) + e^3\overline{G}^{(1)}(x) + e^4\overline{G}^{(2)}(x) + \dots + e^{\ell+2}\overline{G}^{(\ell)}(x)$.</p> <p>Round 2: Upon receiving the broadcast of $(e, C(x))$ from INT, every verifier $P_i \in \mathcal{P}$ locally verifies the following conditions:</p> <ul style="list-style-type: none"> - $C(\overline{\gamma}_i) \stackrel{?}{=} e\overline{m}_{1,i} + e^2\overline{m}_{2,i} + e^3\overline{v}_{1,i} + \dots + e^{\ell+2}\overline{v}_{\ell,i}$ — we call this condition C1. - $A(\overline{\gamma}_i) \neq d\overline{m}_{1,i} + d^2\overline{m}_{2,i} + d^3\overline{v}_{1,i} + d^4\overline{v}_{2,i} + \dots + d^{\ell+2}\overline{v}_{\ell,i}$ holds during AuthVal — we call this condition C2. <p>Verifier P_i broadcasts Accept if condition C1 or C2 is true for P_i, else it broadcasts Reject.</p> <p>Output Determination: If more than t verifiers broadcast Accept then each verifier P_i outputs AcceptProof along with the vector $(\text{comb}_1, \dots, \text{comb}_{\text{pack}}) \stackrel{\text{def}}{=} (C(\beta_1), \dots, C(\beta_{\text{pack}}))$, else each verifier P_i outputs RejectProof.</p>

Fig. 3. Efficient ICPoP protocol where $\ell \geq 1$ and $1 \leq \text{pack} \leq n - t$.

Proof. The properties of ICPoP follow from Lemmas 5–8. We next prove the communication complexity. During *Distr*, D sends $\ell + 2$ polynomials of degree d to INT and a tuple of $\ell + 3$ values to each individual verifier. During *AuthVal* a polynomial of degree d is broadcasted by INT and D broadcasts either an *OK* or *Abort* message. During *RevealPoP*, INT broadcasts a polynomial of degree d and each individual verifier broadcasts either an *Accept* or a *Reject* message. So overall the protocol has communication complexity $\mathcal{PC}(\mathcal{O}(n\ell))$ and $\mathcal{BC}(\mathcal{O}(n))$, as $d = \mathcal{O}(n)$. This also proves the **ICPoP-Succinctness of the Proof** property, as the size of the proof is independent of ℓ .

Transferability of ICPoP: In our VSS protocol we will use ICPoP as follows: after receiving $\mathcal{S} \cup \mathcal{M}$ from D via the secret-encoding and masking polynomials, INT will send these polynomials (and hence $\mathcal{S} \cup \mathcal{M}$) to another designated party, say $P_R \in \mathcal{P}$ (if INT is corrupted then it can send incorrect polynomials to P_R). Later on, party P_R will act as an INT and produce a proof of possession of $\mathcal{S} \cup \mathcal{M}$, which got “transferred” to P_R from INT; the proof gets verified with respect to the verification information held by the verifiers. This transfer of $\mathcal{S} \cup \mathcal{M}$ will satisfy all the properties of ICPoP, imagining P_R as the new INT. Specifically if D is *honest* and both INT and P_R are *honest*, then the privacy will hold. Moreover if P_R produces a proof of possession of incorrect sets (this can be the case if either INT or P_R is corrupted), then the proof gets rejected. If D is *corrupted* and both INT and P_R are honest then the proof given by P_R will be accepted.

4 Statistical VSS with a Quadratic Overhead

We present a 4-round VSS protocol *Sh* to t -share $\ell \times (n - t) = \Theta(n\ell)$ values with communication complexity $\mathcal{PC}(\mathcal{O}(n^3\ell))$ and $\mathcal{BC}(\mathcal{O}(n^3))$. So for sufficiently large ℓ , the broadcast complexity will be *independent* of ℓ . For simplicity, we will present a 5-round statistical VSS protocol *Sh-Single* for sharing a single secret. We will then explain how to reduce the number of rounds of *Sh-Single* from five to four. Finally we extend this four round *Sh-Single* to get *Sh*. We first discuss a protocol *Poly-Check* adapted from [34], used in our VSS.

4.1 Verifiably Distributing Values on Bivariate Polynomials of Degree at Most t

In our VSS protocol we will come across the following situation: D will select L bivariate polynomials $F^{(1)}(x, y), \dots, F^{(L)}(x, y)$, each of degree at most t and send the i th row polynomials $f_i^{(1)}(x), \dots, f_i^{(L)}(x)$ of $F^{(1)}(x, y), \dots, F^{(L)}(x, y)$ respectively to each P_i ; we stress that the corresponding column polynomials are retained by D . The parties now want to publicly verify if there is a set of at least $t + 1$ *honest* parties, who received row polynomials, lying on L unique bivariate polynomials of degree at most t without revealing any additional information about the polynomials. For this we use a two round protocol *Poly-Check* (see Fig. 4), which is adapted from an asynchronous protocol for the same purpose, presented in [34].

In the protocol Poly-Check, there is a designated *verifier* V , who challenges D to broadcast a random linear combination of the n column polynomials of all the bivariate polynomials selected by D . Specifically V provides a challenge combiner, say r and in response D makes public a linear combination of its column polynomials with respect to r ; to maintain the privacy of the column polynomials, this linear combination is blinded by a random degree t *blinding polynomial* $B(y)$, selected by D , with each party P_i having a value on this polynomial. Corresponding to the linear combination of the column polynomials produced by D , each party P_i makes public a linear combination of n values of all its row polynomials, with respect to the combiner r , which is blinded by the value of $B(y)$ possessed by it. The idea here is the following: if indeed there exists a set of $t + 1$ honest parties that we are looking for, then the values of the row polynomials possessed by these parties will define degree t column polynomials. And these column and row polynomials will be “pair-wise consistent”. Based on this idea we check if the blinded linear combination of the column polynomials produced by D is of degree t . Moreover it is also checked if there exists a *witness set* $\mathcal{W}^{(V)}$ of at least $2t + 1$ parties, such that their blinded linear combination of row polynomial values satisfies the linear combination produced by D . If any one of the above conditions is not satisfied the parties output \perp , otherwise they output $\mathcal{W}^{(V)}$. It is ensured that if V is *honest*, then except with probability $\frac{nL}{|\mathbb{F}|}$, the honest parties in $\mathcal{W}^{(V)}$ constitute the desired set of row polynomial holders. The properties of Poly-Check are stated in Lemma 9; we refer to [34] for the complete proof.

Lemma 9 (Properties of Protocol Poly-Check [34]). *In protocol Poly-Check, the following holds:*

- If D is honest then every honest party outputs a $\mathcal{W}^{(V)}$ set which includes all the honest parties. Moreover the row polynomials of the honest parties in $\mathcal{W}^{(V)}$ will lie on $F^{(1)}(x, y), \dots, F^{(L)}(x, y)$. Furthermore Adv gets no additional information about $F^{(1)}(x, y), \dots, F^{(L)}(x, y)$ in the protocol.
- If D is corrupted and V is honest and if the parties output a $\mathcal{W}^{(V)}$, then except with probability at most $\frac{nL}{|\mathbb{F}|}$, there exists L bivariate polynomials, say $\overline{F}^{(1)}(x, y), \dots, \overline{F}^{(L)}(x, y)$, of degree at most t , such that the row polynomials of the honest parties in $\mathcal{W}^{(V)}$ lie on $\overline{F}^{(1)}(x, y), \dots, \overline{F}^{(L)}(x, y)$.
- The protocol requires two rounds and has communication complexity $\mathcal{BC}(\mathcal{O}(n))$.

4.2 Five Round Statistical VSS for a Single Secret

To t -share s , D selects a random *secret-carrying* bivariate polynomial $F(x, y)$ of degree at most t such that $s = F(0, 0)$. The i th row polynomial $f_i(x)$ of $F(x, y)$ is given to each P_i . We stress that *only* the row polynomials are distributed. The parties then verify the consistency of the distributed polynomials by publicly verifying the existence of a set VCORE of at least $2t + 1$ parties, such that

Poly-Check(D, V, \mathcal{P} , L , $\{F^{(1)}(x, y), \dots, F^{(L)}(x, y), B(y)\}$, $\{\bar{f}_i^{(1)}(x), \dots, \bar{f}_i^{(L)}(x), \bar{b}_i\}_{i \in [n]}$)

Round 1: Verifier V selects a random combiner $r \in \mathbb{F} \setminus \{0\}$ and broadcasts r .

Round 2: The parties on receiving r from the broadcast of V do the following:

- D broadcasts the polynomial $E(y) \stackrel{\text{def}}{=} B(y) + r g_1^{(1)}(y) + r^2 g_2^{(1)}(y) + \dots + r^n g_n^{(1)}(y) + r^{(n+1)} g_1^{(2)}(y) + r^{(n+2)} g_2^{(2)}(y) + \dots + r^{2n} g_n^{(2)}(y) + \dots + r^{(L-1)n+1} g_1^{(L)}(y) + r^{(L-1)n+2} g_2^{(L)}(y) + \dots + r^{Ln} g_n^{(L)}(y)$. Here $g_i^{(k)}(y) = F^{(k)}(\alpha_i, y)$ for $k \in [L]$ and $i \in [n]$.

- Each party $P_i \in \mathcal{P}$ (including D) broadcasts the linear combination $e_i \stackrel{\text{def}}{=} \bar{b}_i + r \bar{f}_i^{(1)}(\alpha_1) + r^2 \bar{f}_i^{(1)}(\alpha_2) + \dots + r^n \bar{f}_i^{(1)}(\alpha_n) + r^{(n+1)} \bar{f}_i^{(2)}(\alpha_1) + r^{(n+2)} \bar{f}_i^{(2)}(\alpha_2) + \dots + r^{2n} \bar{f}_i^{(2)}(\alpha_n) + \dots + r^{(L-1)n+1} \bar{f}_i^{(L)}(\alpha_1) + r^{(L-1)n+2} \bar{f}_i^{(L)}(\alpha_2) + \dots + r^{Ln} \bar{f}_i^{(L)}(\alpha_n)$

Output determination: If $E(y)$ has degree more than t then each party $P_j \in \mathcal{P}$ outputs \perp and terminate. Else each party $P_j \in \mathcal{P}$ creates a *witness set* $\mathcal{W}^{(V)}$, initialized to \emptyset and then does the following:

- Include party P_i to $\mathcal{W}^{(V)}$ if the relation $E(\alpha_i) \stackrel{?}{=} e_i$ is true.
- If $|\mathcal{W}^{(V)}| \geq 2t + 1$ then P_j outputs $\mathcal{W}^{(V)}$, else P_j outputs \perp .

Fig. 4. Checking the consistency of row polynomials distributed by D under the supervision of a designated verifier V. The inputs for (an honest) D are L secret bivariate polynomials $F^{(1)}(x, y), \dots, F^{(L)}(x, y)$ of degree at most t and a secret blinding polynomial $B(y)$ of degree at most t . The inputs for (an honest) party P_i are L row polynomials $\bar{f}_i^{(1)}(x), \dots, \bar{f}_i^{(L)}(x)$ of degree at most t and a share \bar{b}_i of blinding polynomial. If D and P_i are honest then these values are private and $\bar{f}_i^{(k)}(x) = F^{(k)}(x, \alpha_i)$ and $\bar{b}_i = B(\alpha_i)$ will hold for each $k \in [L]$.

the row polynomials of the *honest* parties in VCORE lie on a unique bivariate polynomial, say $\bar{F}(x, y)$, of degree at most t . For this, n instances of Poly-Check are executed (one on the behalf of each party playing the role of the designated verifier V) and it is verified if there is common subset of at least $2t + 1$ parties, present across all the generated witness sets. As there will be at least one instance of Poly-Check executed on the behalf of an honest verifier, clearly the common subset of $2t + 1$ parties satisfies the properties of VCORE. To maintain the privacy of the row polynomials during the Poly-Check instances, n independent *blinding polynomials* are used by D, one for each instance. If a VCORE is found, then we say that D has “committed” the secret $\bar{s} = \bar{F}(0, 0)$ to the parties in VCORE via their row polynomials and the next goal will be to ensure that each party P_j obtains its column polynomial $\bar{g}_j(y)$ of $\bar{F}(x, y)$; party P_j can then output its share $\bar{s}_j = \bar{g}_j(0)$ of \bar{s} and hence \bar{s} will be t -shared via $\bar{F}(x, 0)$. If D is *honest* then $\bar{F}(x, y) = F(x, y)$ will hold (and hence $\bar{s} = s$), as VCORE will include all the honest parties.

To enable P_j obtain $\bar{g}_j(y)$, each $P_i \in \text{VCORE}$ can send the common point $\bar{f}_i(\alpha_j)$ on $\bar{g}_j(y)$ to P_j , where $\bar{f}_i(\alpha_j)$ denotes the j th value on the i th row polynomial received by P_i (if D is honest then $\bar{f}_i(\alpha_j) = f_i(\alpha_j)$ holds). The honest

parties in VCORE will always send the correct values; however the corrupted parties may send incorrect values. Due to insufficient redundancy in the received $\bar{f}_i(\alpha_j)$ values, party P_j cannot error-correct them (for this we require $|\text{VCORE}|$ to be of size at least $3t+1$). The way out is that P_j gives a *proof of possession* of the $\bar{f}_i(\alpha_j)$ values received from the parties P_i in VCORE. Namely the values on the row polynomials are initially distributed by D by executing instances of *Distr*. There will be n^2 such instances and instance *Distr* $_{ij}$ is executed to distribute $f_i(\alpha_j)$ to P_i , considering P_i as an INT; the corresponding instances *AuthVal* $_{ij}$ are also executed and it is ensured that the *AuthVal* instances, involving any party from VCORE as an INT, is not aborted by D. Now when a party P_i in VCORE sends $\bar{f}_i(\alpha_j)$ to P_j , party P_j acts as an INT and publicly gives a proof of possession of $\bar{f}_i(\alpha_j)$ by executing an instance *RevealPoP* $_{ji}$ of *RevealPoP*. The idea is to use the *transferability* property of *ICPoP* to identify the incorrectly transferred values. Namely if D is *honest* and an incorrect $\bar{f}_i(\alpha_j)$ is transferred to P_j , then the corresponding proof gets rejected during *RevealPoP* $_{ji}$ and P_j discard such values.

Unfortunately, if D is *corrupted* then the above mechanism alone is not sufficient for P_j to robustly reconstruct $\bar{g}_j(y)$. Because a *corrupted* P_i in VCORE can then transfer an incorrect $\bar{f}_i(\alpha_j)$ to P_j and still the proof will get accepted; this is because if *both* D and INT are corrupted, then INT will know the full auxiliary and verification information involved in *ICPoP*. As a result, P_j will end up not reconstructing a degree t column polynomial from the received $\bar{f}_i(\alpha_j)$ values. To deal with this particular case, we ensure that the \mathcal{M} sets used by D in the *ICPoP* instances have a similar “structure” as the corresponding \mathcal{S} sets. Specifically, D selects two random *masking* bivariate polynomials $M^{(1)}(x, y)$ and $M^{(2)}(x, y)$ each of degree at most t . Let $m_i^{(1)}(x), m_i^{(2)}(x)$ denote the corresponding row polynomials. The instances *Distr* $_{ij}$ are executed by setting $\mathcal{S}_{ij} = \{f_i(\alpha_j)\}$ and $\mathcal{M}_{ij} = \{m_i^{(1)}(\alpha_j), m_i^{(2)}(\alpha_j)\}$ (thus $\ell = 1$ and $\text{pack} = 1$ in these instances). The corresponding *AuthVal* $_{ij}$ instances are executed with $\bar{\mathcal{S}}_{ij} = \{\bar{f}_i(\alpha_j)\}$ and $\bar{\mathcal{M}}_{ij} = \{\bar{m}_i^{(1)}(\alpha_j), \bar{m}_i^{(2)}(\alpha_j)\}$, which denotes the \mathcal{S} and \mathcal{M} sets respectively received by P_i during *Distr* $_{ij}$ (if D is honest then these will be the same as \mathcal{S}_{ij} and \mathcal{M}_{ij}). The existence of VCORE will now imply that D has committed a secret-carrying polynomial, say $\bar{F}(x, y)$ and two masking bivariate polynomials, say $\bar{M}^{(1)}(x, y), \bar{M}^{(2)}(x, y)$ to the parties in VCORE, where all these polynomials have degree at most t . It follows that any linear combination of the column polynomials $\bar{F}(\alpha_j, y), \bar{M}^{(1)}(\alpha_j, y)$ and $\bar{M}^{(2)}(\alpha_j, y)$ will be a degree t univariate polynomial. And this property is used by P_j to identify the correctly transferred $\bar{\mathcal{S}}_{ij} \cup \bar{\mathcal{M}}_{ij}$ sets. Namely the values in the transferred $\bar{\mathcal{S}}_{ij} \cup \bar{\mathcal{M}}_{ij}$ sets should lie on degree t univariate polynomials and hence any random linear combination of these sets should also lie on a degree t polynomial. Based on this observation, party P_j selects a *common* random combiner, say e_j , for *all* the transferred $\bar{\mathcal{S}}_{ij} \cup \bar{\mathcal{M}}_{ij}$ sets and publicly reveals a linear combination of these $\bar{\mathcal{S}}_{ij} \cup \bar{\mathcal{M}}_{ij}$ sets via the *RevealPoP* $_{ji}$ instances. It is then publicly verified if these linearly combined values lie on a degree t polynomial. If not then it implies that D is

corrupted and it is discarded; see Figs. 5 and 6 for the formal details. For the ease of understanding, a pictorial representation of the information distributed in Sh-Single is given in Fig. 7.

We state the properties of Sh-Single and formally prove the correctness of Sh-Single in case of a corrupt dealer by means of a claim below. The remaining proofs will appear in the full version.

Lemma 10. *If D is honest then except with probability at most $\frac{n^3 t}{|\mathbb{F}|-1}$, it is not discarded during Sh-Single.*

Lemma 11 (Correctness for an honest D). *If D is honest then except with probability at most $\frac{n^3 t}{|\mathbb{F}|-1}$, the value s is t -shared at the end of Sh-Single.*

Lemma 12. *Let $\bar{f}_i(x), \bar{m}_i^{(1)}(x)$ and $\bar{m}_i^{(2)}(x)$ be the row polynomials defined by the values in $\bar{S}_{ij} \cup \bar{M}_{ij}$ received by party $P_i \in \mathcal{P}$ from D for $j \in [n]$. If D is corrupted and a VCORE is formed during Sh-Single then except with probability at most $\frac{3n^2}{|\mathbb{F}|}$, there exist bivariate polynomials, say $\bar{F}(x, y), \bar{M}^{(1)}(x, y)$ and $\bar{M}^{(2)}(x, y)$, each of degree at most t , such that for each honest $P_i \in \text{VCORE}$, the polynomials $\bar{f}_i(x), \bar{m}_i^{(1)}(x)$ and $\bar{m}_i^{(2)}(x)$ lie on $\bar{F}(x, y), \bar{M}^{(1)}(x, y)$ and $\bar{M}^{(2)}(x, y)$ respectively.*

Proof. From the definition, $\text{VCORE} = \mathcal{W}^{(P_1)} \cap \mathcal{W}^{(P_2)} \cap \dots \cap \mathcal{W}^{(P_n)}$ and $|\text{VCORE}| \geq 2t+1$. This ensures that there are at least $t+1$ common honest parties in VCORE, say HVCORE. Consider an honest party $P_j \in \mathcal{P}$, playing the role of the verifier V in the instance Poly-Check $^{(P_j)}$. It follows from Lemma 9 (by substituting $L = 3$) that for the instance Poly-Check $^{(P_j)}$, except with probability at most $\frac{3n}{|\mathbb{F}|}$, the row polynomials $\bar{f}_i(x), \bar{m}_i^{(1)}(x)$ and $\bar{m}_i^{(2)}(x)$ of the parties $P_i \in \text{HVCORE}$ together lie on three unique bivariate polynomials, say $\bar{F}(x, y), \bar{M}^{(1)}(x, y)$ and $\bar{M}^{(2)}(x, y)$ respectively of degree at most t . The same will be true with respect to every other instance Poly-Check $^{(P_k)}$, corresponding to every other honest verifier $P_k \neq P_j$. Moreover, the set of three bivariate polynomials defined via each of these instances of Poly-Check will be the same, namely $\bar{F}(x, y), \bar{M}^{(1)}(x, y)$ and $\bar{M}^{(2)}(x, y)$ respectively. This follows from Lemma 2 (by substituting $\ell = |\text{HVCORE}|$) and the fact that $|\text{HVCORE}| \geq t+1$. The lemma now follows from the union bound and the fact that there are $\Theta(n)$ honest parties, playing the role of V .

Lemma 13 (Correctness for a corrupted D). *If D is corrupted and not discarded during Sh-Single, then there exists some value, say \bar{s} , such that except with probability at most $\frac{n^3}{|\mathbb{F}|-1}$, \bar{s} is t -shared at the end of Sh-Single.*

Proof. If a corrupted D is not discarded then it implies that a set VCORE with $|\text{VCORE}| \geq 2t+1$ is constructed during Sh-Single. Let HVCORE be the set of honest parties in VCORE; clearly $|\text{HVCORE}| \geq t+1$. From Lemma 12 it follows

Sh-Single(D, \mathcal{P} , s)

Round 1: Dealer D does the following:

- Select a random *secret-carrying bivariate polynomial* $F(x, y)$ of degree at most t with $F(0, 0) = s$. Select two random *masking bivariate polynomials* $M^{(1)}(x, y)$ and $M^{(2)}(x, y)$, each of degree at most t . In addition select n random *blinding univariate polynomials* $B^{(P_1)}(y), \dots, B^{(P_n)}(y)$, each of degree at most t , where $B^{(P_i)}$ is associated with party $P_i \in \mathcal{P}$. Corresponding to each $P_i \in \mathcal{P}$, compute row polynomials $f_i(x) \stackrel{\text{def}}{=} F(x, \alpha_i)$, $m_i^{(1)}(x) \stackrel{\text{def}}{=} M^{(1)}(x, \alpha_i)$, $m_i^{(2)}(x) \stackrel{\text{def}}{=} M^{(2)}(x, \alpha_i)$ and share-vector $(b_i^{(P_1)}, \dots, b_i^{(P_n)})$ of blinding polynomials, where $b_i^{(P_j)} \stackrel{\text{def}}{=} B^{(P_j)}(\alpha_i)$ for $j \in [n]$. Let $S_{ij} \stackrel{\text{def}}{=} \{f_i(\alpha_j)\}$ and $\mathcal{M}_{ij} \stackrel{\text{def}}{=} \{m_i^{(1)}(\alpha_j), m_i^{(2)}(\alpha_j)\}$ for $i, j \in [n]$.
- To each $P_i \in \mathcal{P}$, send $(b_i^{(P_1)}, \dots, b_i^{(P_n)})$. In addition, for $j \in [n]$, execute an instance $\text{Distr}(D, P_i, \mathcal{P}, 1, 1, S_{ij} \cup \mathcal{M}_{ij})$ of Distr to give $S_{ij} \cup \mathcal{M}_{ij}$ to P_i , considering P_i as an INT. Let Distr_{ij} denote the corresponding instance of Distr .

Round 2: Each $P_i \in \mathcal{P}$ (including D) does the following: let $\overline{S}_{ij} = \{\overline{f}_{ij}\}$ and $\overline{\mathcal{M}}_{ij} = \{\overline{m}_{ij}^{(1)}, \overline{m}_{ij}^{(2)}\}$ be the secret and masking set respectively received from D in Distr_{ij} . In addition, let $(\overline{b}_i^{(P_1)}, \dots, \overline{b}_i^{(P_n)})$ denote the vector received^a from D. Let $\overline{f}_i(x)$, $\overline{m}_i^{(1)}(x)$ and $\overline{m}_i^{(2)}(x)$ be the polynomials defined by the points $\{(\alpha_j, \overline{f}_{ij})\}_{j \in [n]}$, $\{(\alpha_j, \overline{m}_{ij}^{(1)})\}_{j \in [n]}$ and $\{(\alpha_j, \overline{m}_{ij}^{(2)})\}_{j \in [n]}$ respectively. If these polynomials are not of degree t then P_i broadcasts (Abort, P_i) , else it does the following:

- Transfer $\overline{S}_{ij} \cup \overline{\mathcal{M}}_{ij}$ to P_j by sending all the information received from D in the instance Distr_{ij} .
- As an INT, execute the steps of Round 1 of an instance $\text{AuthVal}(D, P_i, \mathcal{P}, 1, 1, \overline{S}_{ij} \cup \overline{\mathcal{M}}_{ij})$ of AuthVal , corresponding to the instance Distr_{ij} , for $j \in [n]$. Let this instance of AuthVal be denoted as AuthVal_{ij} .
- As a verifier V, execute the steps of Round 1 of an instance $\text{Poly-Check}(D, P_i, \mathcal{P}, 3, \{M^{(1)}(x, y), M^{(2)}(x, y), F(x, y), B^{(P_i)}(y)\}, \{\overline{m}_j^{(1)}(x), \overline{m}_j^{(2)}(x), \overline{f}_j(x), \overline{b}_j^{(P_i)}\}_{j \in [n]})$ of Poly-Check ; denote this instance as $\text{Poly-Check}^{(P_i)}$.

Round 3: Each $P_i \in \mathcal{P}$ (including D) does the following: If (Abort, \star) message is received from the broadcast of more than t parties then discard D and abort Sh-Single . Else P_i does the following:

- Corresponding to each $j, k \in [n]$, participate as a verifier during Round 2 of AuthVal , in the instances AuthVal_{jk}
- Execute the steps of Round 2 of Poly-Check , corresponding to the instances $\text{Poly-Check}^{(P_1)}, \dots, \text{Poly-Check}^{(P_n)}$.
- **[Additional steps, If $P_i = D$]** — In addition to the above steps, P_i executes the following steps if P_i is D:
 - As a D, execute the steps of Round 2 of AuthVal , corresponding to the instances AuthVal_{jk} for each $j, k \in [n]$.
 - As a D, execute the steps of Round 2 of Poly-Check , corresponding to $\text{Poly-Check}^{(P_1)}, \dots, \text{Poly-Check}^{(P_n)}$.

^a If D is honest then $\overline{S}_{ij} = S_{ij}$, $\overline{\mathcal{M}}_{ij} = \mathcal{M}_{ij}$ and $(\overline{b}_i^{(P_1)}, \dots, \overline{b}_i^{(P_n)}) = (b_i^{(P_1)}, \dots, b_i^{(P_n)})$.

Fig. 5. VSS for sharing a single secret: Part I.

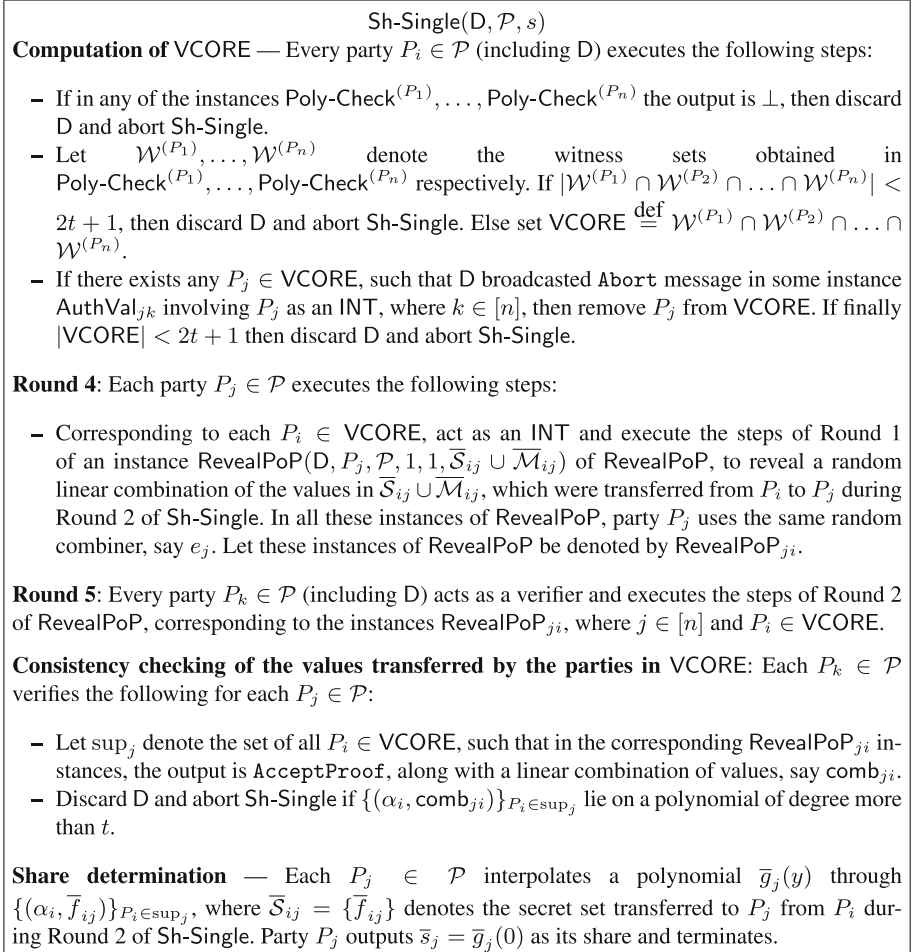


Fig. 6. VSS for sharing a single secret: Part II.

that except with probability at most $\frac{3n^2}{|\mathbb{F}|}$, the row polynomials $\overline{f}_i(x), \overline{m}_i^{(1)}(x)$ and $\overline{m}_i^{(2)}(x)$ of the parties in HVCORE lie on unique bivariate polynomials, say $\overline{F}(x, y), \overline{M}^{(1)}(x, y)$ and $\overline{M}^{(2)}(x, y)$ of degree at most t . We define $\overline{s} \stackrel{\text{def}}{=} \overline{F}(0, 0)$ and claim that \overline{s} is t -shared via the polynomial $\overline{f}_0(x) \stackrel{\text{def}}{=} \overline{F}(x, 0)$, with each honest P_j holding the share $\overline{s}_j \stackrel{\text{def}}{=} \overline{F}(\alpha_j, 0)$. To prove our claim, we show that each honest party P_j outputs its degree t univariate polynomial $\overline{g}_j(y) \stackrel{\text{def}}{=} \overline{F}(\alpha_j, y)$ except with probability at most $\frac{n^2}{|\mathbb{F}|-1}$; this ensures that P_j obtains the correct share, as $\overline{s}_j = \overline{g}_j(0)$. For this, we further need to show that the $\overline{\mathcal{S}}_{ij}$ set transferred by each party $P_i \in \text{sup}_j$ to P_j contains the value $\overline{g}_j(\alpha_i)$.

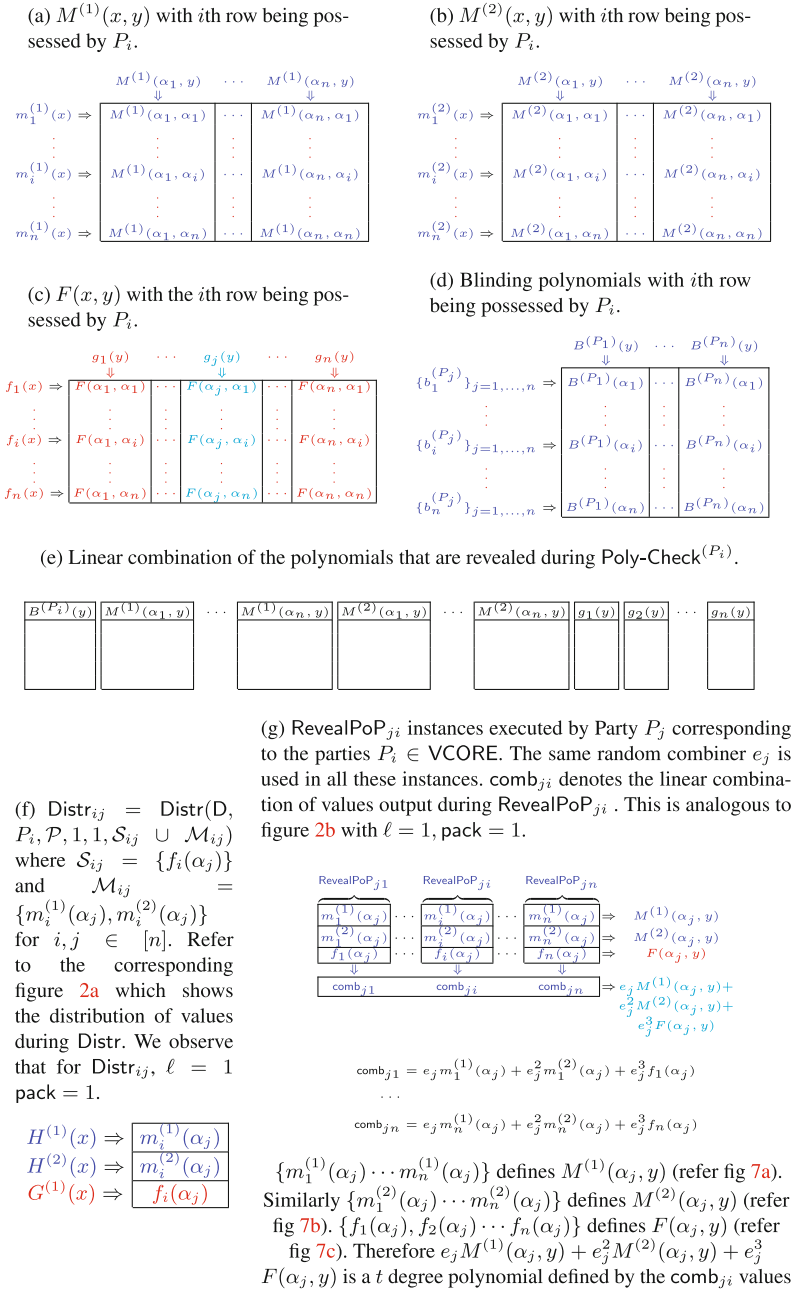


Fig. 7. Pictorial representation of the values distributed in Sh-Single protocol.

Consider an honest P_j . Notice that $\text{sup}_j \subseteq \text{VCORE}$. We first argue that every $P_i \in \text{HVCORE}$ is present in sup_j , except with probability at most $\frac{n^2}{|\mathbb{F}|-1}$. This is because there are $\Theta(n)$ such parties P_i and in each corresponding RevealPoP_{ji} instance, the output is AcceptProof , which follows from Lemma 6 (by substituting $\ell = 1$). Now consider the set of values $\overline{\mathcal{S}}_{ij} = \{\overline{f}_{ij}\}$ and $\overline{\mathcal{M}}_{ij} = \{\overline{m}_{ij}^{(1)}, \overline{m}_{ij}^{(2)}\}$ transferred by the parties $P_i \in \text{HVCORE}$ to P_j . Since $\overline{f}_{ij} = \overline{f}_i(\alpha_j) = \overline{g}_j(\alpha_i)$ holds, it follows that the values $\{\overline{f}_{ij}\}_{P_i \in \text{HVCORE}}$ define the degree t univariate polynomial $\overline{g}_j(y)$. Similarly the values $\{\overline{m}_{ij}^{(1)}\}_{P_i \in \text{HVCORE}}$ and $\{\overline{m}_{ij}^{(2)}\}_{P_i \in \text{HVCORE}}$ define degree t univariate polynomials $\overline{M}^{(1)}(y, \alpha_j)$ and $\overline{M}^{(2)}(y, \alpha_j)$ respectively. To complete the proof, we argue that except with probability at most $\frac{2}{|\mathbb{F}|}$, the values in the $\overline{\mathcal{S}}_{ij}$ and $\overline{\mathcal{M}}_{ij}$ set transferred by a *corrupted* party $P_i \in \text{sup}_j$ lie on $\overline{g}_j(y)$, $\overline{M}^{(1)}(y, \alpha_j)$ and $\overline{M}^{(2)}(y, \alpha_j)$ respectively. This is because the combiner e_j selected by the honest P_j in the RevealPoP_{ji} instances corresponding to the parties in sup_j is truly random and unknown to the adversary in advance, when the $\overline{\mathcal{S}}_{ij}$ and $\overline{\mathcal{M}}_{ij}$ sets are transferred to P_j . The rest follows from Lemma 4 (by substituting $L = 2$) and the fact that the values $\{\text{comb}_{ji}\}_{P_i \in \text{sup}_j}$ lie on a polynomial of degree at most t (otherwise D would have been discarded), say $\text{comb}_j(y)$, where $\text{comb}_j(y) \stackrel{\text{def}}{=} e_j \overline{M}^{(1)}(y, \alpha_j) + e_j^2 \overline{M}^{(2)}(y, \alpha_j) + e_j^3 \overline{g}_j(y)$. As there can be n^2 pair of parties involving a corrupted party, it follows by the union bound that except with probability at most $\frac{2n^2}{|\mathbb{F}|}$, the corrupted parties in VCORE transfer the correct values to the honest parties.

As each honest P_j correctly obtains its column polynomial except with probability at most $\frac{n^2}{|\mathbb{F}|-1}$ and as there are $\Theta(n)$ such honest parties, it follows that except with probability at most $\frac{n^3}{|\mathbb{F}|-1}$, the value \overline{s} is t -shared.

Lemma 14 (Privacy). *In protocol Sh-Single, the value s remains information theoretically secure.*

Theorem 2. *Sh-Single is a five round VSS protocol for a single secret, satisfying the requirements of VSS except with probability $\frac{n^3 t}{|\mathbb{F}|-1}$. The protocol has communication complexity $\mathcal{PC}(\mathcal{O}(n^3))$ and $\mathcal{BC}(\mathcal{O}(n^3))$.*

Proof. The properties of VSS follow from Lemmas 11–14. In the protocol n^2 instances of ICPoP (with $\ell = 1$, $\text{pack} = 1$) and n instances of Poly-Check (each with $L = 3$) are executed. The rest follows from the communication complexity of ICPoP (Theorem 1) and Poly-Check (Lemma 9).

From Five Rounds to Four Rounds: In Sh-Single, the instances of RevealPoP which start getting executed during Round 4 can be instead instantiated during Round 3 itself. Namely irrespective of the formation of VCORE , each party P_j starts executing the instance RevealPoP_{ji} corresponding to *each* party $P_i \in \mathcal{P}$, based on the set of values in $\overline{\mathcal{S}}_{ij} \cup \overline{\mathcal{M}}_{ij}$ which were transferred to P_j by P_i

during Round 2. Next VCORE is computed and if P_i is found not to be present in VCORE, then the instance RevealPoP_{j_i} can be halted; otherwise the remaining steps of the RevealPoP_{j_i} instance are executed during Round 4. Based on this modification, Sh-Single now requires four rounds, the rest of the properties remain same.

Sharing $\ell \times (n - t)$ Secrets: To share $\ell \times (n - t)$ secrets, the underlying instances of Distr , AuthVal and RevealPoP are executed to deal with $\ell \times \text{pack}$ values simultaneously, where $\text{pack} = n - t$. The steps for consistency checking of the values transferred by the parties in VCORE are also generalized to deal with $\ell \times (n - t)$ values. With these modifications, we get a four round Sh for sharing $\ell(n - t)$ values. The properties of Sh follow in a straight forward fashion from the corresponding properties of Sh-Single , taking into account that the underlying instances of ICPoP that are executed deal with $\ell \times (n - t)$ values. We state the theorem below. The proof will appear in the full version.

Theorem 3. *Sh is a four round VSS for $\ell \times (n - t)$ values, with an error probability of $\max\{\frac{n^3(n-1)}{|\mathbb{F}|-(n-t)}, \frac{n^3\ell}{|\mathbb{F}|-1}\}$. The protocol has communication complexity $\mathcal{PC}(\mathcal{O}(n^3\ell))$ and $\mathcal{BC}(\mathcal{O}(n^3))$.*

5 Efficient Statistical MPC Protocol

Using Sh , we design a statistical MPC protocol in the partially synchronous setting. The protocol is designed in the offline-online paradigm, where in the offline phase, the parties generate t -sharing of random and private multiplication triples of the form (a, b, c) , where $c = ab$. Later in the online phase, these triples are used for the shared evaluation of the circuit using the standard Beaver multiplication triple based technique [2, 3, 5, 14]. For designing the offline phase protocol, we use the protocol Sh and deploy the efficient framework of [15]. The shared evaluation of the circuit is done in a completely asynchronous fashion in the online phase. We get the following theorem. The complete description of the protocol and the proof will appear in the full version.

Theorem 4. *Let $f : \mathbb{F}^n \rightarrow \mathbb{F}$ be a function expressed as an arithmetic circuit over a finite field \mathbb{F} , consisting of c_M and c_R multiplication and random gates respectively. Assuming that the first four communication rounds are synchronous broadcast rounds after which the entire communication is asynchronous, there exists a statistical MPC protocol to securely compute f , provided $|\mathbb{F}| \geq 4n^4(c_M + c_R)(3t + 1)2^\kappa$ for a given error parameter κ . The protocol has communication complexity $\mathcal{PC}(\mathcal{O}(n^2(c_M + c_R) + n^4))$ and $\mathcal{BC}(\mathcal{O}(n^4))$.*

References

1. Asharov, G., Lindell, Y.: A full proof of the BGW protocol for perfectly-secure multiparty computation. *J. Cryptol.* **30**(1), 58–151 (2017)
2. Beaver, D.: Efficient multiparty protocols using circuit randomization. In: Feigenbaum, J. (ed.) *CRYPTO 1991*. LNCS, vol. 576, pp. 420–432. Springer, Heidelberg (1992). https://doi.org/10.1007/3-540-46766-1_34
3. Beerliová-Trubíniová, Z., Hirt, M.: Efficient multi-party computation with dispute control. In: Halevi, S., Rabin, T. (eds.) *TCC 2006*. LNCS, vol. 3876, pp. 305–328. Springer, Heidelberg (2006). https://doi.org/10.1007/11681878_16
4. Beerliová-Trubíniová, Z., Hirt, M.: Simple and efficient perfectly-secure asynchronous MPC. In: Kurosawa, K. (ed.) *ASIACRYPT 2007*. LNCS, vol. 4833, pp. 376–392. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-76900-2_23
5. Beerliová-Trubíniová, Z., Hirt, M.: Perfectly-secure MPC with linear communication complexity. In: Canetti, R. (ed.) *TCC 2008*. LNCS, vol. 4948, pp. 213–230. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-78524-8_13
6. Beerliová-Trubíniová, Z., Hirt, M., Nielsen, J.B.: On the theoretical gap between synchronous and asynchronous MPC protocols. In: *Proceedings of the PODC*, pp. 211–218. ACM (2010)
7. Ben-Or, M., Canetti, R., Goldreich, O.: Asynchronous secure computation. In: *Proceedings of the STOC*, pp. 52–61. ACM (1993)
8. Ben-Or, M., Goldwasser, S., Wigderson, A.: Completeness theorems for non-cryptographic fault-tolerant distributed computation (Extended Abstract). In: *Proceedings of the STOC*, pp. 1–10. ACM (1988)
9. Ben-Or, M., Kelmer, B., Rabin, T.: Asynchronous secure computations with optimal resilience (Extended Abstract). In: *Proceedings of the PODC*, pp. 183–192. ACM (1994)
10. Ben-Sasson, E., Fehr, S., Ostrovsky, R.: Near-Linear unconditionally-secure multiparty computation with a dishonest minority. In: Safavi-Naini, R., Canetti, R. (eds.) *CRYPTO 2012*. LNCS, vol. 7417, pp. 663–680. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-32009-5_39
11. Canetti, R.: *Studies in Secure Multiparty Computation and Applications*. Ph.D. thesis, Weizmann Institute, Israel (1995)
12. Chaum, D., Crépeau, C., Damgård, I.: Multiparty unconditionally secure protocols (Extended Abstract). In: *STOC*, pp. 11–19. ACM (1988)
13. Chor, B., Goldwasser, S., Micali, S., Awerbuch, B.: Verifiable secret sharing and achieving simultaneity in the presence of faults. In: *FOCS*, pp. 383–395. IEEE Computer Society (1985)
14. Choudhury, A., Hirt, M., Patra, A.: Asynchronous multiparty computation with linear communication complexity. In: Afek, Y. (ed.) *DISC 2013*. LNCS, vol. 8205, pp. 388–402. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-41527-2_27
15. Choudhury, A., Patra, A.: An efficient framework for unconditionally secure multiparty computation. *IEEE Trans. Inf. Theor.* **63**(1), 428–468 (2017)
16. Cramer, R., Damgård, I., Dziembowski, S., Hirt, M., Rabin, T.: Efficient multiparty computations secure against an adaptive adversary. In: Stern, J. (ed.) *EUROCRYPT 1999*. LNCS, vol. 1592, pp. 311–326. Springer, Heidelberg (1999). https://doi.org/10.1007/3-540-48910-X_22

17. Cramer, R., Damgård, I., Maurer, U.: General secure multi-party computation from any linear secret-sharing scheme. In: Preneel, B. (ed.) EUROCRYPT 2000. LNCS, vol. 1807, pp. 316–334. Springer, Heidelberg (2000). https://doi.org/10.1007/3-540-45539-6_22
18. Damgård, I., Ishai, Y., Krøigaard, M.: Perfectly secure multiparty computation and the computational overhead of cryptography. In: Gilbert, H. (ed.) EUROCRYPT 2010. LNCS, vol. 6110, pp. 445–465. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-13190-5_23
19. Damgård, I., Ishai, Y., Krøigaard, M., Nielsen, J.B., Smith, A.: Scalable multiparty computation with nearly optimal work and resilience. In: Wagner, D. (ed.) CRYPTO 2008. LNCS, vol. 5157, pp. 241–261. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-85174-5_14
20. Damgård, I., Nielsen, J.B.: Scalable and unconditionally secure multiparty computation. In: Menezes, A. (ed.) CRYPTO 2007. LNCS, vol. 4622, pp. 572–590. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-74143-5_32
21. Fitzi, M., Garay, J., Gollakota, S., Rangan, C.P., Srinathan, K.: Round-optimal and efficient verifiable secret sharing. In: Halevi, S., Rabin, T. (eds.) TCC 2006. LNCS, vol. 3876, pp. 329–342. Springer, Heidelberg (2006). https://doi.org/10.1007/11681878_17
22. Fitzi, M., Hirt, M.: Optimally efficient multi-valued byzantine agreement. In: PODC, pp. 163–168. ACM Press (2006)
23. Fitzi, M., Nielsen, J.B.: On the number of synchronous rounds sufficient for authenticated Byzantine agreement. In: Keidar, I. (ed.) DISC 2009. LNCS, vol. 5805, pp. 449–463. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-04355-0_46
24. Franklin, M.K., Yung, M.: Communication complexity of secure computation (Extended Abstract). In: STOC, pp. 699–710. ACM (1992)
25. Gennaro, R., Ishai, Y., Kushilevitz, E., Rabin, T.: The round complexity of verifiable secret sharing and secure multicast. In: STOC, pp. 580–589. ACM (2001)
26. Goldreich, O., Micali, S., Wigderson, A.: How to play any mental game or a completeness theorem for protocols with honest majority. In: STOC, pp. 218–229. ACM (1987)
27. Hirt, M., Maurer, U., Przydatek, B.: Efficient secure multi-party computation. In: Okamoto, T. (ed.) ASIACRYPT 2000. LNCS, vol. 1976, pp. 143–161. Springer, Heidelberg (2000). https://doi.org/10.1007/3-540-44448-3_12
28. Katz, J., Koo, C.Y., Kumaresan, R.: Improving the round complexity of VSS in point-to-point networks. *Inf. Comput.* **207**(8), 889–899 (2009)
29. Kushilevitz, E., Lindell, Y., Rabin, T.: Information-theoretically secure protocols and security under composition. *SIAM J. Comput.* **39**(5), 2090–2112 (2010)
30. Lynch, N.A.: *Distributed Algorithms*. Morgan Kaufmann, San Francisco (1996)
31. McEliece, R.J., Sarwate, D.V.: On sharing secrets and reed-solomon codes. *Commun. ACM* **24**(9), 583–584 (1981)
32. Patra, A., Choudhary, A., Pandu Rangan, C.: Efficient statistical asynchronous verifiable secret sharing and multiparty computation with optimal resilience. *IACR Cryptology ePrint Archive*, 2009:492 (2009)
33. Patra, A., Choudhury, A., Pandu Rangan, C.: Asynchronous Byzantine agreement with optimal resilience. *Distrib. Comput.* **27**(2), 111–146 (2014)
34. Patra, A., Choudhury, A., Pandu Rangan, C.: Efficient asynchronous verifiable secret sharing and multiparty computation. *J. Cryptology* **28**(1), 49–109 (2015)

35. Patra, A.: Error-free multi-valued broadcast and Byzantine agreement with optimal communication complexity. In: Fernández Anta, A., Lipari, G., Roy, M. (eds.) OPODIS 2011. LNCS, vol. 7109, pp. 34–49. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-25873-2_4
36. Rabin, T., Ben-Or, M.: Verifiable secret sharing and multiparty protocols with honest majority (Extended Abstract). In: STOC, pp. 73–85. ACM (1989)
37. Shamir, A.: How to share a secret. *Commun. ACM* **22**(11), 612–613 (1979)
38. Yao, A.C.: Protocols for secure computations. In: FOCS, pp. 160–164. IEEE Computer Society (1982)

Catching MPC Cheaters: Identification and Openability

Robert Cunningham¹, Benjamin Fuller², and Sophia Yakubov¹(✉)

¹ MIT Lincoln Laboratory, Lexington, USA
{rkc,sophia.yakubov}@ll.mit.edu

² University of Connecticut, Mansfield, USA
benjamin.fuller@uconn.edu

Abstract. Secure multi-party computation (MPC) protocols do not completely prevent malicious parties from cheating or disrupting the computation. We augment MPC with three new properties to discourage cheating. First is a strengthening of *identifiable abort*, called *completely identifiable abort*, where all parties who do not follow the protocol will be identified as cheaters by each honest party. The second is *completely identifiable auditability*, which means that a third party can determine whether the computation was performed correctly (and who cheated if it was not). The third is *openability*, which means that a distinguished coalition of parties can recover the MPC inputs.

We construct the first (efficient) MPC protocol achieving these properties. Our scheme is built on top of the SPDZ protocol (Damgård et al., Crypto 2012), which leverages an offline (computation-independent) pre-processing phase to speed up the online computation. Our protocol is *optimistic*, retaining online SPDZ efficiency when no one cheats. If cheating does occur, each honest party performs only local computation to identify cheaters.

Our main technical tool is a new locally identifiable secret sharing scheme (as defined by Ishai, Ostrovsky, and Zikas (TCC 2012)) which we call *commitment enhanced secret sharing* or CESS.

The work of Baum, Damgård, and Orlandi (SCN 2014) introduces the concept of auditability, which allows a third party to verify that the computation was executed correctly, but not to identify the cheaters if it was not. We enable the third party to identify the cheaters by augmenting the scheme with CESS. We add openability through the use of verifiable encryption and specialized zero-knowledge proofs.

Approved for public release: distribution unlimited. This material is based upon work supported under Air Force Contract No. FA8721-05-C-0002 and/or FA8702-15-D-0001. Any opinions, findings, conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the U.S. Air Force.

1 Introduction

Secure multi-party computation (MPC) allows multiple parties to evaluate a function of their secret inputs while maintaining their privacy. In this work, we focus on preventing malicious behavior that is not prevented by the guarantees of traditional MPC.

Completely Identifiable Abort. In traditional MPC, a malicious party cannot cause the computation to return an incorrect output, but it *can* cheat by deviating from the protocol and causing a termination with an error, known as an abort.¹ Since the cheater remains anonymous, it does not face any consequences for its actions. There is no point in recomputing the function, as the honest parties do not know who to exclude. In order to avoid such stalemates, it is desirable to be able to identify the cheaters. An *MPC with identifiable abort* [7, 10, 19, 29] guarantees that all honest parties agree on a subset of cheating parties.² We introduce MPC with *completely* identifiable abort, which guarantees that all honest parties agree on *all* cheating parties.

Completely Identifiable Auditability. Baum et al. [2] define *auditability*, which enables any third party to verify the correctness of an MPC given a public transcript of the computation, and the output it produced. We introduce *completely identifiable* auditability, which allows the third party auditor to identify all the cheating parties if the computation was not carried out correctly.

Openability. In traditional MPC, parties are free to provide any well formed input they want. Many applications require that the inputs be measurement values from the real world. However, there is no binding between each party's input and the real measurement value, and parties can lie about their measurements. These lies may change the output of the MPC to one that does not match reality; if this occurs, it is desirable to identify the party responsible. We enable the recovery of MPC inputs by an *opening coalition* and call this *openability*.

To ensure that openability does not break the security of the underlying MPC, opening coalitions must contain at least one party that did not participate in the MPC. We call this distinguished party a *judge* J . This judge's role may be shared by more than one party.

1.1 Our Construction

We extend SPDZ [13, 14], which is one of the fastest known MPC protocols; it leverages an inefficient offline pre-processing phase to enable quick online computation. The online portion of SPDZ is very efficient, using fast information-theoretic tools. For n users, the online communication cost is $O(n^2m)$ messages where m is the number of multiplications evaluated.

¹ In this work we consider an arbitrary number of malicious parties. In this setting, it is impossible to guarantee termination without error [11].

² Cheater identification gained popularity in the areas of secret sharing [7, 18, 29] and pay television [10].

Like SPDZ, our construction is secure in the presence of a malicious adversary statically making arbitrarily many corruptions. Our protocol is *optimistic*; if cheating does not occur, the online communication and computation are very efficient. Our online protocol has only twice the online complexity of SPDZ. In particular, the communication complexity remains $O(n^2m)$. If cheating does occur, each party must perform an additional local computation whose complexity is $O(nm)$ in order to identify the cheaters. The additional local work uses computationally secure tools, which are slower.

Starting Point: SPDZ. SPDZ leverages Beaver triples [4], which are pre-computed during the offline phase. Each input is additively secret shared; the computation then proceeds gate by gate. Additions are computed by each party locally. Multiplications use Beaver triples, and require two values to be reconstructed (which requires two broadcast messages from each participant). To prevent malicious parties from providing incorrect shares during these reconstructions, SPDZ uses a linear MAC of the form $\text{MAC}(x) = \alpha x$, where α is the MAC key which is secret shared amongst all the parties. The linear MAC shares follow the computation, and are checked at the end of the computation to detect whether any cheating took place.

Adding Completely Identifiable Abort. We design a new locally identifiable secret sharing scheme (as defined by [18]) which we call *commitment-enhanced secret sharing (CESS)*. A locally identifiable secret sharing scheme is a secret sharing scheme where, after reconstruction, all honest parties agree on the set of parties that modified their shares [18].

Each CESS share contains an additive share (as in SPDZ), and additionally includes linearly homomorphic commitments to *every* additive share.³ Each CESS share also contains the decommitment value for the commitment to the corresponding additive share. This conceptually simple change of giving each party a commitment to every additive share allows identification of cheaters. When CESS shares are used for computation, since we use linearly homomorphic commitments, if cheating occurs, each honest party can use the homomorphism of the commitment scheme to transform the input share commitments of each other party into a commitment to that party's output share. All parties whose claimed output shares do not match their output share commitments are identified as cheaters.

On the Use of Broadcast. Unlike SPDZ, our protocol requires a fully secure broadcast channel. Secure (or *authenticated*) broadcast is a very expensive primitive; constructions typically require $O(n^2)$ messages, and use public key primitives. Secure broadcast is used by our protocol *only* in the INPUT operation, a total of n times.

The reason we can use broadcast so sparingly is that, because our CESS shares are checkable, there is no need to securely broadcast them. Instead, we implement a specific, *optimistic* share broadcast protocol. Since the validity of each decommitment can be checked, the parties send one another their respective

³ We use Pedersen commitments [23] to enable efficient zero-knowledge proofs.

additive shares and decommitments. Any honest party P_i receiving an incorrect additive share and decommitment from P_j can send help requests to all other parties asking for the correct values. If any honest parties received correct values from P_j , they will forward those to P_i , solving the problem. Otherwise, all honest parties will agree that P_j cheated, and abort the protocol. If cheating does not occur, each CESS reconstruction takes only n^2 messages. (If cheating does occur, up to an additional $2n(n-1)t$ messages may be required, where t is the number of corrupt parties).

Related Work on SPDZ with Identifiable Abort. Recent works by Baum et al. [3] and Spini and Fehr [26] add identifiable abort to the SPDZ protocol.⁴ In Table 1, we compare our efficiency to theirs, both in the case that cheating does not occur and in the case that it does. We improve on the number of broadcast messages for both schemes in all cases.

Baum et al. augment SPDZ with identifiable abort using a homomorphic information-theoretic signature scheme. In the event of cheating, our scheme relies on computational techniques to identify the misbehaving party, while the scheme of Baum et al. uses information-theoretic techniques.⁵ Our use of computational techniques necessitates a larger field (not necessary in Baum et al. [3]). This makes the length of each message depend logarithmically on the security parameter. Baum et al. [3] (and Spini and Fehr [26]) assume a broadcast channel, which may mitigate the efficiency advantage of the smaller field. Implementing broadcast often uses signatures, which already adds a logarithmic dependence in the security parameter to all messages.

Spini and Fehr [26] take a different approach. Their approach is based on dispute control. If no cheating occurs, they retain the exact online efficiency of SPDZ (while our protocol is slower by a factor of 2). If cheating does occur, they have two forms of identification. Their protocol can ensure that each honest party knows the identity of some cheating party by doubling the cost of SPDZ. At an additional cost, they can ensure that all honest parties *agree* on the identity of some cheating party. Spini and Fehr require multiple rounds of blame assignment, in contrast to our conceptually simple approach.

If the prospect of being identified is likely to discourage all cheating in the first place, the Spini and Fehr protocol achieves better efficiency. Our protocol is more prudent if cheating may still occur, for example in the setting of multiple independent malicious actors. Also, as we discuss next, our protocol allows for an outside observer to identify cheaters. Public identification is not possible using the protocol of Spini and Fehr.

Auditability. Openability relies on a property called *auditability*, introduced by Baum, Damgård, and Orlandi [2]. They add a public transcript τ to SPDZ

⁴ An alternative approach uses bitcoin to introduce financial repercussions for cheating [1, 21].

⁵ We use the Pedersen commitment scheme, which is information-theoretically hiding but only computationally binding. So, computational assumptions are only necessary for the correctness of cheater identification.

Table 1. Online Complexity of Protocols with Identifiable Abort. If cheating does occur we consider the worst case complexity of the protocol. Spini and Fehr move through multiple different protocols of decreasing efficiency, we assume if cheating occurs they are forced to use the most expensive protocol.

		cheating?	our scheme	Baum et. al	Spini and Fehr
# messages	broadcast (bc)	no	n	$n(n + 2m + 2)$	$O(n^2)$
	point-to-point		$2(m + 1)n(n - 1)$	$2mn(n - 1)$	$O(mn)$
	broadcast (bc)	yes (worst case)	n	$n(n + 2m + 2)$	$O(n^2)$
	point-to-point		$2(m + 1)n(n - 1)(2t + 1)$	$2mn(n - 1)$	$O(mn)$
public key operations/party		no	none, except in broadcast	none, except in broadcast	none, except in broadcast
		yes (worst case)	$O(nm)$	none, except in broadcast	$O(nm)$

(modeled as a public append-only bulletin board) to allow external parties to check protocol correctness even if all participants are malicious. The public transcript τ contains Pedersen commitments [23] to each precomputed Beaver triple value and input i_i .

The transcript τ contains all values reconstructed during the computation; namely the Beaver triple differences. Though τ does not contain Pedersen commitments to intermediate computation values or to the output, these commitments can be derived using the linear homomorphism of Pedersen commitments and the posted Beaver triple differences. An auditor holding a transcript τ and the evaluation circuit C can derive a Pedersen commitment c_{out} to the correct computation output out . The auditor can then check that c_{out} is indeed a commitment to the claimed output out' .

Completely Identifiable Auditability. Our construction includes commitments for each input *share*, while the construction of Baum et al. [2] includes a single commitment for each *input*. This increases the number of committed values, but does not affect the online communication complexity. The additional commitments enable *completely identifiable* auditability, meaning that in addition to public auditing of the protocol correctness, we can support public identification of cheaters. We also rely on auditability to add openness; openness is unachievable without an audit check to ensure that the transcript is well formed.

Adding Openability. An openable MPC protocol allows a distinguished *opening coalition* to recover the computation inputs. This is useful in case there is cause to doubt the truthfulness of these inputs. One might think that having commitments to the inputs would be enough — each party can the “open” by providing the decommitment, and anyone who does not cooperate is identified as a cheater. However, we cannot rely on the input owners to cooperate with the opening. While it is tempting to simply identify parties who do not cooperate as cheaters, they might actually be honest. The adversary might be blocking their messages, or their opening values might have been destroyed by the event that caused doubts about the input veracity. (An example of such a situation

is discussed in more detail in Sect. 1.2; in the case of a satellite collision, the collision may have destroyed the data stored on the satellites involved).

Our approach to building openability is to encrypt each MPC input and prove consistency with the public transcript τ of the MPC protocol. This encryption scheme must have two properties: (1) the message must only be readable by an allowed coalition, and (2) the proof of consistency must be efficient. A threshold encryption scheme splits the secret key between a coalition of parties and only the entire coalition (or an allowed subset) can jointly decrypt a ciphertext. A verifiable encryption scheme allows efficient proofs about the underlying plaintext [6, 8, 12, 15].

We design a new threshold verifiable encryption scheme which, to the best of our knowledge, is the first *universally composable* [9] threshold verifiable encryption scheme. Our scheme uses a variant of the scheme described by Camenisch and Shoup [8]. The scheme of Camenisch and Shoup cannot be universally composable, since a ciphertext commits to the underlying plaintext in a perfectly binding way. Simulating decryption would involve breaking this commitment, which even a powerful simulator cannot do. We avoid this problem using a layer of secret sharing and commitments that are only *computationally* binding.

Note that, in our construction, it is possible to open some inputs while maintaining the privacy of others. This is very useful in the case when there is cause to suspect some parties of lying, but not others.

1.2 A Motivating Example

In this section we present satellite conjunction analysis [17] as a motivating use case for our augmented MPC.⁶ Those readers who are convinced of the need to catch MPC cheaters may proceed to Sect. 1.2.

Multiple government organizations and companies own satellites. The purpose of many satellites is secret, so organizations are not willing to share their trajectories. However, there is risk to not sharing trajectory information. The active Iridium 33 satellite collided with the inactive Cosmos 2251 satellite in 2009 [20] creating significant debris which endangered other satellites [28]. To avoid such catastrophes, the organizations want to jointly compute whether collisions will occur without revealing satellite trajectories. As a result of the computation, parties should learn only whether a collision will take place, and who the involved organizations are. Hemenway et al. observed that MPC enables such a joint and private computation [16, 17].

In traditional MPC protocols malicious parties cannot affect the output of the computation (other than by changing their inputs). However, malicious parties can cause the computation to abort — to terminate with an error — without ever being identified as the culprit. Imagine that some malicious organization wants to cause a satellite collision. All it would have to do is aim its satellite at another, prevent the MPC from completing every time it is run, sit back and wait! Because no culprit in an abort can be identified, the malicious organization

⁶ Other sensitive applications include economic markets [5] and elections [2].

would not be caught until it is too late. In order to avoid this, we augment MPC with completely identifiable abort.

Satellites generally reside in one of three bands: low-earth orbit (LEO), medium-earth orbit (MEO), or geosynchronous orbit (GEO). Collisions between functioning objects at different levels are unlikely; however, if a collision occurs at one level, the resulting debris may collide with objects at other levels. Suppose that the above satellite collision computation is performed by all organizations with objects in medium earth orbit. Organizations with satellites in low earth orbit are also affected by the results of the computation, even though they don't participate, since a collision in medium earth orbit could cause debris to fly into low earth orbit, potentially damaging the satellites there. For convenience we will refer to one of the organizations owning satellites in low earth orbit as Leo. Leo wants to be able to determine whether the medium earth orbit computation was performed correctly even if *all* of the organizations involved in it might have malicious intentions, so as to determine the risk to his own satellites. Given a transcript τ of the MPC, any external organization such as Leo should be able to *audit* the correctness of the computation, as described by Baum et al. [2].

Now, imagine that Leo performed the audit, and determined that the MPC was performed correctly. However, the next day, a collision occurs and debris destroys one of Leo's satellites! This could only have occurred if one of the organizations participating in the MPC provided incorrect inputs to the computation. In such a situation, it would be crucial to be able to determine who is responsible. We achieve this property by adding openability. Once inputs are opened, of the organizations whose satellites were involved in the collision, whoever's claimed input trajectory does not intersect with the collision location is the one to blame.

Complete identifiable abort, completely identifiable auditability and openability make MPC much more appealing, especially for high-risk applications such as determining the likelihood of satellite collisions. The increased accountability dis-incentivizes cheating, and increases all parties' trust in the computation output.

Organization. The rest of the paper is organized as follows. In Sect. 2 we describe our augmented MPC definitions. In Sect. 3 we introduce *commitment-enhanced secret sharing*, which is crucial for our construction. In Sect. 4, we describe how commitment-enhanced secret sharing can be used in MPC. In Sect. 5 we construct our MPC protocol with completely identifiable abort. In Sect. 6 we introduce a universally composable threshold verifiable encryption scheme. Finally, in Sect. 7 we add openability using threshold verifiable encryption.

2 Definitions

Notation. Throughout this work we implicitly consider a sequence of protocols parameterized by a security parameter k . For notational clarity we usually omit

k (except in the cryptographic building block descriptions in Appendix C of the full version of this paper), but it is implied that all algorithms take k as input.

All of our MPC protocols consider arithmetic circuits over p -order fields, where p is a large Sophie-Germain prime (that is, $q = 2p + 1$ is also a prime). \mathbb{Z}_p refers to the field $\{0, \dots, p - 1\}$; \mathbb{Z}_p^* refers to $\mathbb{Z}_p \setminus \{0\} = \{1, \dots, p - 1\}$. \mathbb{QR}_p refers to $\{x^2 \bmod p : x \in \mathbb{Z}_p^*\}$ (quadratic residues modulo p). An element g of a group G is a generator of that group if $\forall x \in G, \exists a$ such that $g^a = x$.

Model. We implicitly assume two available functionalities: a broadcast channel, and an append-only bulletin board. We assume the availability of a secure broadcast channel for unit cost. If a broadcast channel is not naturally available, it can be implemented using digital signatures. We make very sparing use of the broadcast channel; in fact, values need only be broadcast once per input. This is because all other values that may need to be broadcast are secret shares, and we do not require the full power of a secure broadcast channels for those, as discussed in Sect. 1.1.

There are several ways to implement an append-only public bulletin board. One simple way is using a public server against which privacy is desirable (so, this server cannot simply be used to perform the computation in question), but which is trusted to behave semi-honestly. Another way, which does not require trust in an additional third party, is using a blockchain (but without necessarily using proofs of work which rely on an honest majority). Put very simply, every post p to the bulletin board is broadcast together with a signature σ by the posting party on p and a hash of the previous post (or posts, if there were simultaneous posts broadcast). The use of the public bulletin board in our protocol is unusual in that it is public knowledge who needs to be providing a post at which point in the protocol. Thus, omitting a post contributed by a party would not result in a valid bulletin board transcript. Chaining the posts together by signing the posts together with hashes of previous posts ensures that parties' posts cannot be replayed from protocol execution to protocol execution.

Multi-Party Computation (MPC). Consider n parties (P_1, \dots, P_n) each of whom has a secret input (in_1, \dots, in_n). *Secure Multi-Party Computation* (MPC) allows them to compute a joint function $C(in_1, \dots, in_n) = out$ on their values, where C is a circuit representing the function. As a result of this computation, all of the parties learn the output out , but no party learns anything else about others' inputs.

This privacy guarantee should hold even if some parties are adversarially controlled, meaning that they are trying to learn something about other parties' inputs. Different MPC protocols maintain their security in the presence of different numbers and types of adversarially controlled parties. In this paper, we consider security in the presence of arbitrarily many adversarial parties, chosen statically (meaning that the adversarial parties are fixed before the protocol begins, but it could be that all parties participating in the protocol are adversarial). Adversarial parties run in probabilistic polynomial time and can act maliciously, meaning that they can deviate arbitrarily from the specified protocol.

The security requirement of MPC is formally defined with respect to an *ideal functionality*, wherein a trusted third party receives inputs from everyone, performs the computation internally, and then distributes the output. When interacting with this ideal functionality, no party learns more than their own input and the output, since those are the only values it sees. For an MPC protocol to be secure, there must exist an efficient simulator that, given the view of all adversarial parties in an ideal execution (meaning their input and the output), can produce a view that is indistinguishable from a real protocol execution view.⁷

Intuitively, the two most important properties of an MPC protocol (both implied by this definition) are *correctness* and *privacy*. Informally, an MPC protocol π satisfies *correctness* if for all inputs (in_1, \dots, in_n) and circuits C where $C(in_1, \dots, in_n) = out$, the protocol π returns out when evaluating C on inputs in_1, \dots, in_n . An MPC protocol π satisfies *privacy* if no party P_i can learn anything about the inputs of any other party, other than what is revealed by out .

Another desirable property is *fairness*; fairness means that if one party learns the output, so do all parties. In the setting where the majority of parties may be adversarial, fairness is known to be unachievable [11]. So, we instead consider *security with abort*, a weaker notion of security that allows an adversary to violate fairness by causing an abort. The ideal functionality for secure MPC with abort is shown in Fig. 1.

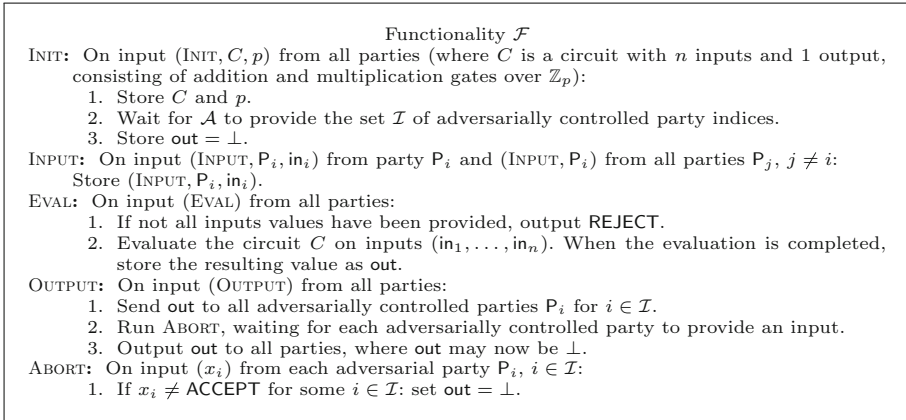


Fig. 1. Ideal functionality for MPC.

MPC with Identifiable Abort. The ideal MPC functionality given in Fig. 1 implies that if any malicious parties attempt to cause the computation to return anything other than the correct output, the protocol aborts (returns \perp). The honest parties are left knowing something went wrong — however, they do not

⁷ We call the list of protocol messages the *view* of the protocol. We use the word transcript or τ to refer to the public information used for auditing (following the notation of [2]).

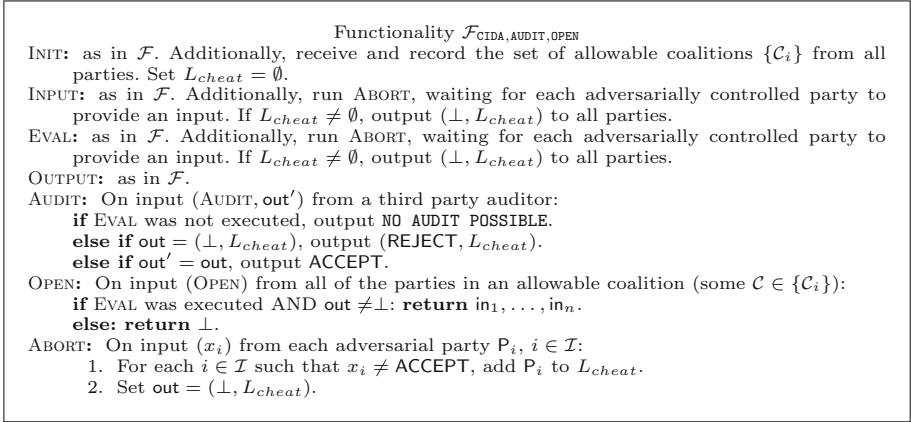


Fig. 2. Ideal functionality for openable and auditable MPC with completely identifiable abort

learn *what* went wrong, or which of the other parties are to blame. *MPC with identifiable abort*, defined by Ishai, Ostrovsky and Zikas [19], ensures that when an abort occurs all the honest parties agree on the identity of *at least one* malicious party P_i . We extend the definition of Ishai et al. [19], defining MPC with *completely identifiable abort* as MPC which ensures that when an abort occurs all honest parties agree on the identities of *all* parties who deviated from the protocol. More formally, Fig. 2 describes the ideal functionality $\mathcal{F}_{\text{CIDA}}$ for MPC with completely identifiable abort. $\mathcal{F}_{\text{CIDA}}$ is simply $\mathcal{F}_{\text{CIDA, AUDIT, OPEN}}$ without the AUDIT and OPEN commands.

Auditability. Any MPC which supports arbitrarily many adversarially controlled parties enables all honest parties to determine whether the protocol was executed correctly. It is also useful to allow any third party to inspect some evidence of the computation and arrive at the same conclusion. Baum, Damgard and Orlandi [2] introduce *auditability* to MPC; they describe a protocol where, given the circuit C being evaluated, a presumed output out' and a public transcript τ updated throughout the computation, any third party can audit the computation and ascertain that it was performed correctly with output out'. In this setting, we model the MPC as also outputting τ .

More formally, Baum et al. introduce the AUDIT algorithm. AUDIT takes in the public transcript τ which is created during computation, the circuit C which was evaluated, and the computation output out, and returns a 0 or a 1, depending on whether the computation was correct.

Definition 1 (Auditable Correctness [2]). *An MPC protocol satisfies Auditable Correctness if for all circuits C and for all potential outputs out,*

- $\text{AUDIT}(\tau, C, \text{out}) = 1$ with overwhelming probability if for some inputs $\text{in}_1, \dots, \text{in}_n$, $C(\text{in}_1, \dots, \text{in}_n) = \text{out}$ and τ is a transcript of the MPC evaluation of C on $\text{in}_1, \dots, \text{in}_n$, and

- $\text{AUDIT}(\tau, C, \text{out}) = 0$ with overwhelming probability if for all inputs $\text{in}_1, \dots, \text{in}_n$, $C(\text{in}_1, \dots, \text{in}_n) \neq \text{out}$ or τ is not a valid transcript of an MPC evaluation of C on inputs $\text{in}_1, \dots, \text{in}_n$.

Completely Identifiable Auditability. We add *completely identifiable* auditability, which allows a third party to identify all cheaters if the protocol was not executed correctly. Informally, we say that an MPC protocol satisfies *Completely Identifiable Auditable Correctness* if it satisfies auditable correctness and, when AUDIT outputs 0, it additionally outputs L_{cheat} (a list of all cheaters). Figure 2 describes this enhanced AUDIT protocol.

While auditability makes the computation execution more transparent, it does not provide any check on the veracity of the computation inputs. As motivated in the introduction, a correct computation on false inputs can be catastrophic. To address this issue, we define *openability* next.

Openability. In extreme cases, it may be necessary to open the inputs of an MPC evaluation (see Sect. 1.2 for a motivating example). Of course, inputs should not be recoverable by any one party; this would violate the privacy guarantees of MPC. However, we can define *allowable coalitions*, or *groups* of parties who we trust not to abuse this privilege. In this context, one might want several additional players that we will call judges $\{J_i\}$. A judge J_i notionally has the power to determine that an opening is justified. We include multiple judges to compensate in case some of the parties who participated in the MPC do not cooperate. This is something we need to account for, since if party P_i knows that it will be identified as a liar, it will not cooperate with an input opening. Two reasonable examples of allowable opening coalitions might be all the parties from the MPC together with any judge party ($\{P_1, \dots, P_n, J_i\}$), or some t of the parties together with two judges ($\{P_{i_1}, \dots, P_{i_t}, J_i, J_j\}$).

More formally, we introduce the protocol OPEN executed jointly by an allowable opening coalition. OPEN takes in a transcript τ , and returns $(\text{in}_1, \dots, \text{in}_n)$. We require that the OPEN protocol be *sound*, as described in Definition 2. Notice that the transcript τ also needs to be *hiding*, meaning that it shouldn't reveal any information about the values being computed on. However, this property is implied by the privacy requirement of MPC, and does not need to be explicitly restated.

Definition 2 (Opening Soundness). *We say that an MPC protocol satisfies Opening Soundness if for all circuits C and for all inputs $\text{in}_1, \dots, \text{in}_n$, for all MPC evaluations of C on $\text{in}_1, \dots, \text{in}_n$ resulting in output out and transcript τ (where all participants may be malicious), the probability that $\text{AUDIT}(\tau, C, \text{out}) = 1$ and $\text{OPEN}(\tau) \neq (\text{in}_1, \dots, \text{in}_n)$ is negligible.*

Figure 2 describes the ideal functionality $\mathcal{F}_{\text{CIDA, AUDIT, OPEN}}$ of such a protocol. For OPEN to work for only allowable coalitions, such coalitions (and their associated cryptographic identity) must be known when EVAL is executed.

3 Commitment-Enhanced Secret Sharing

This section describes a new locally identifiable secret sharing scheme which we call *commitment-enhanced secret sharing (CESS)*. CESS is our main building block to add completely identifiable abort to MPC. We first describe basic secret sharing, and then describe locally identifiable secret sharing (LISS) before proceeding to describe CESS.

Secret Sharing. Secret sharing was introduced by Shamir [25]. A t -out-of- n sharing of a secret x is an encoding of the secret into n pieces, or *shares*, such that any t shares together can be used to reconstruct the secret x , but fewer than t shares give no information at all about x . A secret sharing scheme consists of two algorithms: SHARE and REC.

- $\text{SHARE}(x) \rightarrow (s_1, \dots, s_n)$ takes in a secret x and produces the n secret shares.
- $\text{REC}(s_{i_1}, \dots, s_{i_t}) \rightarrow \tilde{x}$ takes in t secret shares and returns the reconstructed secret \tilde{x} .

For n -out-of- n secret sharing, a simple scheme called additive secret sharing (SS_{Add}) can be used. $\text{SS}_{\text{Add}}.\text{SHARE}(x)$ generates $n - 1$ random elements s_1, \dots, s_{n-1} in some additive group, and computes the n th share as $s_n = x - (s_1 + \dots + s_{n-1})$. Any $n - 1$ shares appear completely random; however, the sum of all n shares gives the secret x . Additive secret sharing has some linear properties: a shared value x can be multiplied by a constant, or added to another shared value x' , by separately operating on the individual shares. We use the notation $[x]_{\mathcal{P}_j}$ to denote the additive secret share of element x belonging to party \mathcal{P}_j .

Shamir t -out-of- n secret sharing ($\text{SS}_{\text{Shamir}}$) uses degree- $(t - 1)$ polynomials over some field. $\text{SS}_{\text{Shamir}}.\text{SHARE}(x)$ generates a random degree- $(t - 1)$ polynomial f with x as its y -intercept; each share s_i is a point $(x_i, f(x_i))$ on the polynomial (with $x_i \neq 0$). For simplicity, we fix $x_i = i$. Any t shares can be used to interpolate the polynomial, reconstructing x . Any fewer than t shares give no information about x .

Looking ahead, our MPC protocols are presented using additive secret sharing, but can be trivially extended to use Shamir secret sharing if a t -out-of- n sharing (for some $t < n - 1$) is desired.

3.1 Locally Identifiable Secret Sharing (LISS)

Secret sharing provides confidentiality. However, there are no guarantees that the reconstruction protocol REC returns the correct secret in the presence of malicious parties. Robust secret sharing guarantees reconstruction correctness in the presence of active adversaries [27].⁸ It is also useful to identify the parties

⁸ Robust secret sharing does not require security in the presence of a malicious dealer. This is in contrast to verifiable secret sharing [24]. Looking ahead, the reason we do not require security against a malicious dealer is that dealing is done via MPC in the preprocessing phase.

that provided incorrect shares; this is known as an identifiable secret sharing [22]. Identifiable secret sharing becomes impossible when a majority of parties are adversarial [18, Theorem 3]. However, a slightly weaker task is possible in the presence of an adversarial majority: *honest parties* can agree on the set of parties who provided incorrect shares, but cannot prove it to a third party who did not hold one of the shares. This is known as *locally identifiable secret sharing (LISS)*. We modify the inputs to the reconstruction algorithm REC of a LISS to also include the index i of the party performing the reconstruction; if that party P_i is honest, it has the additional knowledge that the share s_i has not been tampered with. Definition 3 is taken from [18, Definition 4].

Definition 3 (Locally Identifiable Secret Sharing). *An n -out-of- n secret sharing scheme is locally identifiable if it satisfies two requirements: unanimity, meaning that all honest parties should agree on either a correct reconstruction or on the correct set of cheating parties (L_{cheat}), and predictable failure, meaning that the output of the reconstruct algorithm should be simulatable if it does not return the correct secret. Predictable failure ensures that the output of the reconstruction algorithm does not reveal anything about the secret, unless it correctly returns the secret. We give more rigorous descriptions of unanimity and predictable failure below.*

Unanimity. *For any probabilistic polynomial time adversary \mathcal{A} and for any secret x , the probability of \mathcal{A} 's success in the following game is negligible:*

1. $(s_1, \dots, s_n) \leftarrow \text{SHARE}(x)$.
2. \mathcal{A} outputs a set $\mathcal{I} \subsetneq \{1, \dots, n\}$ of adversarial party indices. Let $H = \{1, \dots, n\} \setminus \mathcal{I}$ be the set of honest party indices.
3. \mathcal{A} receives s_i for $i \in \mathcal{I}$.
4. \mathcal{A} selects some $B \subseteq \mathcal{I}$, and outputs s'_i for $i \in B$, where $s'_i \neq s_i$.
5. Let \tilde{x}_i be the value reconstructed by each party P_i , for $i \in H$, with the assumption that s_i is correct. That is, each party P_i runs $\tilde{x}_i \leftarrow \text{REC}(i, t_1, \dots, t_n)$ (where $t_j = s'_j$ if $j \in B$ and $t_j = s_j$ otherwise).

The adversary \mathcal{A} succeeds unless:

1. All honest parties reconstruct the correct secret ($\tilde{x}_i = x$ for all $i \in H$), or
2. All honest parties agree on the set of cheating players ($\tilde{x}_i = (\text{REJECT}, L_{\text{cheat}} = B)$ for all $i \in H$).

Predictable Failure. *There exists an algorithm SIMREC such that for any probabilistic polynomial time adversary \mathcal{A} and for any secret x , the probability of \mathcal{A} 's success in the following game is negligible:*

1. $(s_1, \dots, s_n) \leftarrow \text{SHARE}(x)$.
2. \mathcal{A} outputs a set $\mathcal{I} \subsetneq \{1, \dots, n\}$ of adversarial party indices. Let $H = \{1, \dots, n\} \setminus \mathcal{I}$ be the set of honest party indices.
3. \mathcal{A} receives s_i for $i \in \mathcal{I}$.
4. \mathcal{A} selects some $B \subseteq \mathcal{I}$, and outputs s'_i for $i \in B$, where $s'_i \neq s_i$.

5. $\text{simout} \leftarrow \text{SIMREC}(\mathcal{I}, B, \{s_i\}_{i \in \mathcal{I}}, \{s'_i\}_{i \in B})$.
6. $\tilde{x}_i \leftarrow \text{REC}(i, t_1, \dots, t_n)$ for $i \in \{1, \dots, n\}$, where $t_j = s'_j$ if $j \in B$ and $t_j = s_j$ otherwise.

The adversary \mathcal{A} succeeds unless:

1. $\text{simout} = \text{success}$ and $\tilde{x}_i = x$ for all $i \in \{1, \dots, n\}$, or
2. $\text{simout} = \{\tilde{x}_i\}_{i \in \mathcal{I}}$ (where \tilde{x}_i is either a reconstructed value, or $(\text{REJECT}, L_{\text{cheat}})$).

Our LISS Construction. In order to support cheater identification, we introduce the *commitment-enhanced secret sharing (CESS)* scheme. A CESS of a secret x is based on an additive secret sharing of x . The i th CESS share additionally includes a Pedersen commitment (described in Appendix C of the full version of this paper) to *each* additive share, as well as the decommitment value for the i th commitment. The decommitment values contained in the CESS shares can be viewed as an additive secret sharing of one global decommitment value r_x . The product of the commitments will itself be a valid commitment \mathbf{c}_x to the secret x , and the sum of the individual decommitments will be the corresponding decommitment value. We use the following notation to denote a CESS share of x belonging to party P_i :

$$\langle x \rangle_{P_i} \stackrel{\text{def}}{=} ([x]_{P_i}, [r_x]_{P_i}, (\mathbf{c}_{x,1}, \dots, \mathbf{c}_{x,n})),$$

where

- $[x]_{P_i}$ is the additive secret share of x belonging to P_i ,
- $[r_x]_{P_i}$ is the decommitment value for $\mathbf{c}_{x,i}$ (equivalently, the additive secret share of the decommitment value r_x for \mathbf{c}_x) belonging to P_i , and
- $\mathbf{c}_{x,i}$ is the Pedersen commitment $\text{pc}([x]_{P_i}, [r_x]_{P_i})$ to value $[x]_{P_i}$ with decommitment value $[r_x]_{P_i}$ (as described in Appendix C of the full version of this paper).

We informally refer to a CESS share as an $\langle \rangle$ -share. Notice that each $\langle \rangle$ -share contains $O(n)$ elements, which makes it large and unwieldy. However, the commitments, which make up the bulk of the $\langle \rangle$ -share, do not ever need to be communicated in order to execute reconstruction CESS.REC , since they are replicated in every share. The reconstruction algorithm CESS.REC only receives the additive secret shares, together with one party's local copy of the commitment values $(\mathbf{c}_{x,1}, \dots, \mathbf{c}_{x,n})$. CESS.REC is described in Fig. 3.

We additionally describe *private reconstruction* CESS.PRIVREC (Fig. 4), which describes how a value can be reconstructed by only one party, while everyone still agrees on the cheaters' identities.⁹ The sole object of all but one parties performing CESS.PRIVREC is to compile the correct L_{cheat} , not to reconstruct

⁹ We do not extend the definition of a locally identifiable secret sharing scheme to support private opening; rather, we just describe the functionality. We leave a formal definition to future work.

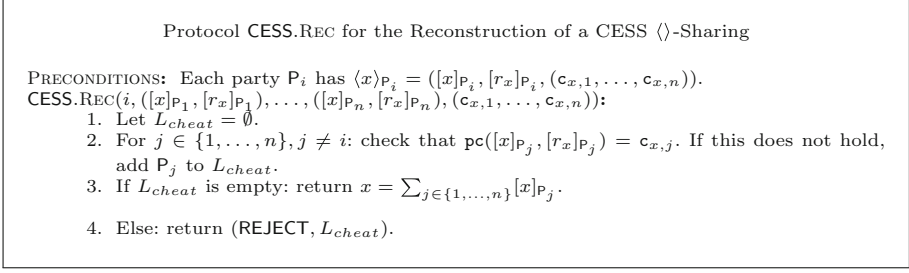


Fig. 3. Protocol REC for the reconstruction of a $\langle \rangle$ -Sharing

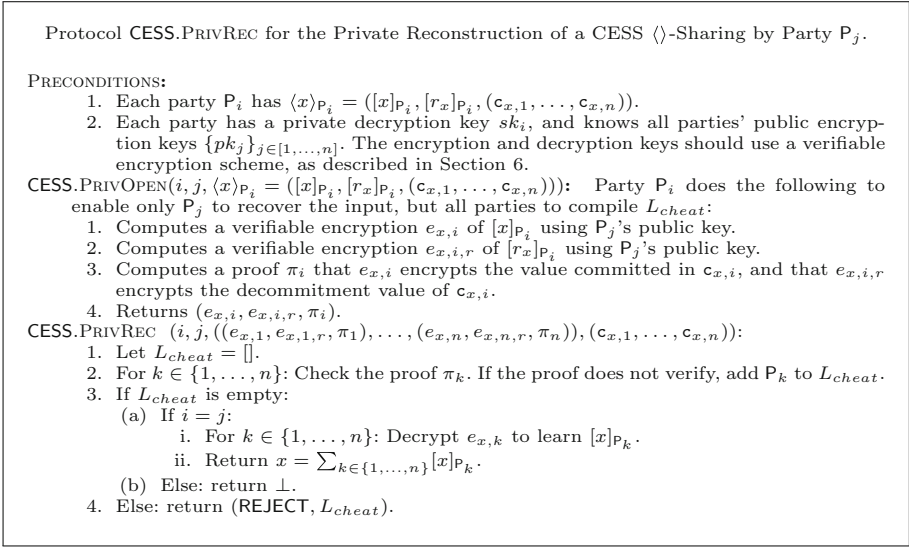


Fig. 4. Protocol PRIVREC for the private reconstruction of a $\langle \rangle$ -Sharing

the value in question. In order to support private reconstruction CESS.PRIVREC, the shares can't be used for reconstruction directly; rather, related values must be derived from the shares. We call this process CESS.PRIVOPEN.

Theorem 1. *Assuming that the commitment scheme \mathbf{p} is secure, the CESS scheme is a locally identifiable secret sharing scheme (LISS).*

Proof. The CESS scheme achieves unanimity. In order to succeed, the adversary \mathcal{A} would have to provide an incorrect additive share of the secret $[x]_{P_i}'$ or an incorrect additive share of the decommitment value $[r_x]_{P_i}'$ for every corrupt party that tampers with their share ($P_i, i \in B$). (Notice that we do not consider the commitments to be a tamperable part of the sharing, since they are never communicated.) In order to avoid having honest parties add P_i to the list of

cheaters L_{cheat} , the adversary must supply $[x]_{P_i}'$ and $[r_x]_{P_i}'$ such that

$$\text{pc}([x]_{P_i}', [r_x]_{P_i}') = c_{x,i} = \text{pc}([x]_{P_i}, [r_x]_{P_i}),$$

which violates the binding property of the Pedersen commitment scheme.

The CESS scheme also achieves predictable failure. The reconstruction simulator SIMREC simply checks the decommitments provided by all of the adversarial parties. If P_i 's decommitment does not verify, SIMREC adds P_i to L_{cheat} ; it then returns (REJECT, L_{cheat}). If all of the decommitments do verify (meaning that none of the shares could have been altered), SIMREC returns success.

4 Adapting CESS for Use with SPDZ

The CESS scheme as described in Sect. 3 isn't quite ready to be used in MPC. Firstly, the CESS reconstruction algorithm REC requires each party to compute n commitments to assemble the list of cheaters L_{cheat} , whether cheating occurred at all or not. This is inefficient, and we remedy it. Secondly, we need to homomorphically compute on CESS shares.

Augmenting CESS with MACs. It would be nice for each party to be able to begin reconstruction by performing an efficient check to determine if cheating occurred, and only proceed with the expensive computation of L_{cheat} when cheating is detected. We can employ the linear MACs from Damgård et al. [14] to detect cheating. The linear MACs consist of $\text{MAC}(x) = \alpha x$, where α is an additively secret-shared MAC key. $\text{MAC}(x)$ is then itself additively secret-shared. MACs can be checked without reconstructing the MAC key α , as described in Fig. 5.

We use the following notation to denote a MAC-augmented CESS (CESS_{MAC}) share of x belonging to party P_i :

$$\langle\langle x \rangle\rangle_{P_i} \stackrel{\text{def}}{=} ([x]_{P_i}, [r_x]_{P_i}, (c_{x,1}, \dots, c_{x,n}), [\text{MAC}(x)]_{P_i}),$$

where $c_{x,i} = \text{pc}([x]_{P_i}, [r_x]_{P_i})$, all of $(c_{x,1}, \dots, c_{x,n})$ is public, and each party P_i is separately assumed to hold an additive share of the secret MAC key α . The reconstruction algorithm $\text{CESS}_{\text{MAC}}.\text{REC}$, executed interactively by the parties, is shown in Fig. 6.

This remains a locally identifiable secret sharing scheme, because a cheating party would have to cause the MAC to verify in order to avoid detection, which they can only do with negligible probability, as shown by Damgård et al. [13].

Note that a party *can* cause MACCHECK to fail, while being honest about all other values, without being identified. If a party does this, $\text{CESS}_{\text{MAC}}.\text{REC}$ will still produce the correct output, but the parties will be forced to execute the more expensive $\text{CESS}.\text{REC}$. In this situation, our cheating party forces the participants to waste computational resources; but, since the reconstruction still succeeds, we do not require that they be identified.

In later sections, we will describe how multiple $\langle\langle \cdot \rangle\rangle$ -sharings are dealt with throughout our MPC protocol. If it is desired, steps 2 through 4 in Fig. 6 can be

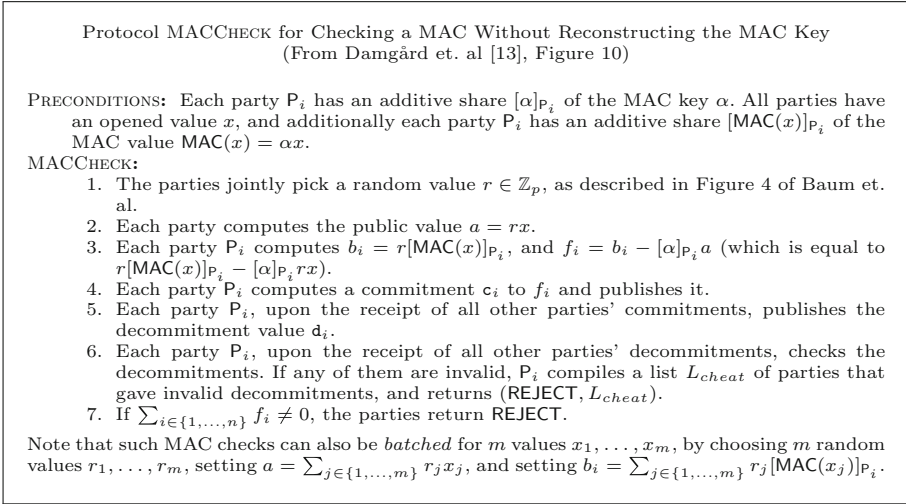


Fig. 5. Protocol MACCHECK for checking a MAC without reconstructing the key

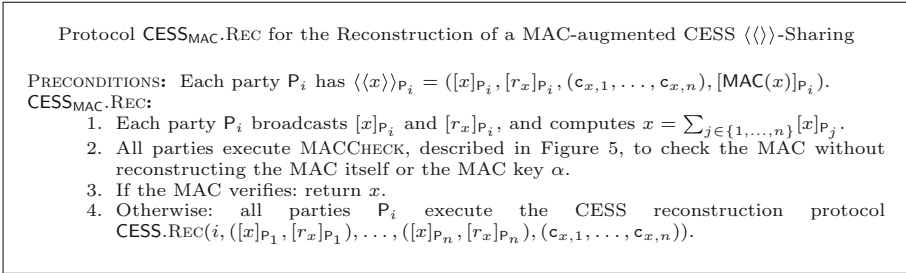


Fig. 6. Protocol REC for the reconstruction of a $\langle\langle\rangle\rangle$ -Sharing

postponed, and then performed in batch form. If the MAC verifies, each party's MAC check communication overhead is independent of the number of sharings being verified.

Computing on Commitment-Enhanced Secret Shares. Finally, in order to use MAC-augmented CESS (CESS_{MAC}) in MPC, we need to describe how to compute on shares. Once we can compute CESS_{MAC} shares, the locally identifiable property of CESS_{MAC} will be used to provide completely identifiable abort.

Linear computations on CESS_{MAC} shares can be performed locally, as shown below, since both additive shares and Pedersen commitments are linearly homomorphic.

- To add a constant ϵ to $\langle\langle x \rangle\rangle_{P_i} = ([x]_{P_i}, [r_x]_{P_i}, (c_{x,1}, \dots, c_{x,n}), [\text{MAC}(x)]_{P_i})$, first compute $[x + \epsilon]_{P_i}$ as

$$[x + \epsilon]_{P_1} = [x]_{P_1} + \epsilon \text{ and } [x + \epsilon]_{P_i} = [x]_{P_i} \text{ for } i \neq 1.$$

Then compute

$$\langle\langle x + \epsilon \rangle\rangle_{P_i} = ([x + \epsilon]_{P_i}, [r_x]_{P_i},$$

$$(\mathbf{c}_{x,1} \mathbf{pc}(\epsilon, 0), \mathbf{c}_{x,2}, \dots, \mathbf{c}_{x,n}), [\text{MAC}(x)]_{P_i} + \epsilon[\alpha]_{P_i}).$$

- To add $\langle\langle x \rangle\rangle_{P_i} = ([x]_{P_i}, [r_x]_{P_i}, (\mathbf{c}_{x,1}, \dots, \mathbf{c}_{x,n}), [\text{MAC}(x)]_{P_i})$ and $\langle\langle y \rangle\rangle_{P_i} = ([y]_{P_i}, [r_y]_{P_i}, (\mathbf{c}_{y,1}, \dots, \mathbf{c}_{y,n}), [\text{MAC}(y)]_{P_i})$, compute

$$\langle\langle x + y \rangle\rangle_{P_i} = ([x]_{P_i} + [y]_{P_i}, [r_x]_{P_i} + [r_y]_{P_i},$$

$$(\mathbf{c}_{x,1} \mathbf{c}_{y,1}, \dots, \mathbf{c}_{x,n} \mathbf{c}_{y,n}), [\text{MAC}(x)]_{P_i} + [\text{MAC}(y)]_{P_i}).$$

- To multiply $\langle\langle x \rangle\rangle_{P_i} = ([x]_{P_i}, [r_x]_{P_i}, (\mathbf{c}_{x,1}, \dots, \mathbf{c}_{x,n}), [\text{MAC}(x)]_{P_i})$ by a constant ϵ , compute

$$\langle\langle \epsilon x \rangle\rangle_{P_i} = (\epsilon[x]_{P_i}, \epsilon[r_x]_{P_i}, (\mathbf{c}_{x,1}^\epsilon, \dots, \mathbf{c}_{x,n}^\epsilon), \epsilon[\text{MAC}(x)]_{P_i}).$$

Beaver triples are a commonly used technique in MPC [4]. A Beaver triple consists of secret sharings (computed during the preprocessing phase) of values a , b and c such that $ab = c$. Each Beaver triple allows a single multiplication to be efficiently computed during the online phase. Beaver triples can be augmented for CESS_{MAC} . Given a Beaver triple $\langle\langle a \rangle\rangle$, $\langle\langle b \rangle\rangle$ and $\langle\langle c \rangle\rangle$, the multiplication of $\langle\langle x \rangle\rangle$ and $\langle\langle y \rangle\rangle$ can be done as follows:

- To multiply $\langle\langle x \rangle\rangle_{P_i}$ by $\langle\langle y \rangle\rangle_{P_i}$:
 1. Open the sharings $\langle\langle \epsilon \rangle\rangle_{P_i} = \langle\langle x - a \rangle\rangle_{P_i}$ and $\langle\langle \delta \rangle\rangle_{P_i} = \langle\langle y - b \rangle\rangle_{P_i}$ to obtain the difference values ϵ and δ .
 2. Compute the product $\langle\langle xy \rangle\rangle_{P_i} = \langle\langle c + \delta a + \epsilon b + \epsilon \delta \rangle\rangle_{P_i}$ by performing the linear operations as described above.

5 Malicious-Majority MPC with Identifiable Abort

In the previous two sections, we introduced CESS_{MAC} (a locally-identifiable secret sharing scheme) and showed how to compute on it. In this section, we build an efficient MPC scheme with completely identifiable abort on top of CESS_{MAC} . As discussed in the introduction, we augment the SPDZ protocol.

In the setup phase INIT of SPDZ, shares of random values and Beaver triples are generated ahead of time (using slower somewhat-homomorphic encryption techniques), and are then used to facilitate fast multiplications throughout the on-line computation. For our construction, we need a setup functionality $\mathcal{F}_{\text{SETUP}}$ that generates $\langle\langle \cdot \rangle\rangle$ sharings of random numbers and Beaver triples.¹⁰ We describe a secure instantiation π_{SETUP} of $\mathcal{F}_{\text{SETUP}}$ in Appendix A of the full version of this paper.

¹⁰ The SPDZ protocol generates the same number of shared values. However, their sharings only contain an additive secret sharing and a linear MAC. The size of $\langle\langle \cdot \rangle\rangle$ -shares grows linearly with the number of players, while SPDZ shares have a constant size for a fixed security parameter.

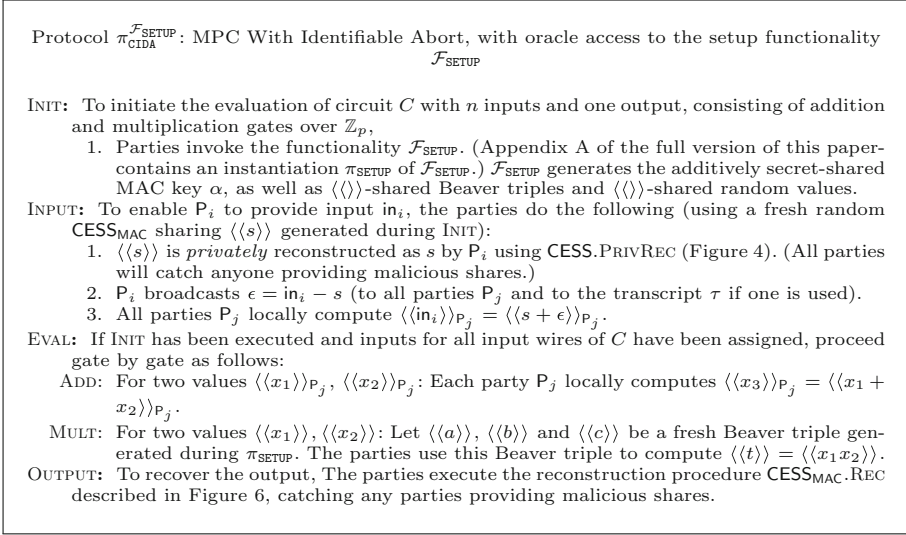


Fig. 7. MPC with completely identifiable abort

Figure 7 gives a slightly simplified illustration of our protocol. The simplification comes from our modular usage of the $\langle\langle\rangle\rangle$ -share reconstruction protocol REC, so that cheating detection and cheater identification is performed with every reconstruction. During EVAL, the only communication involved is the reconstruction of two values ϵ and δ . Using the reconstruction procedure $\text{CESS}_{\text{MAC}}.\text{REC}$ described in Fig. 6, any parties providing malicious shares will be caught.

Theorem 2. *Assuming that the discrete log problem (DLP) is hard in the Pedersen commitment group $\mathbb{Q}\mathbb{R}_q$, the protocol $\pi_{\text{CIDA}}^{\mathcal{F}_{\text{SETUP}}}$ with oracle access to the functionality $\mathcal{F}_{\text{SETUP}}$ is a UC-secure implementation of the functionality $\mathcal{F}_{\text{CIDA}}$.*

Informally, Theorem 2 holds because after running INIT, the only messages sent are (1) a single value broadcast during INPUT, and (2) reconstructions of $\langle\langle\rangle\rangle$ -sharings. Since the $\langle\langle\rangle\rangle$ -sharing scheme (CESS_{MAC}) is a locally identifiable secret sharing scheme, adversarially controlled parties are not be able to change any shared values without the honest parties identifying their malicious behavior. The value broadcast during INPUT defines the input in question, and inconsistencies with that value will also be detected during reconstructions. A formal proof of Theorem 2 appears in the full version of this paper.

By the universal composition theorem [9], this implies that a UC-secure implementation π_{SETUP} of $\mathcal{F}_{\text{SETUP}}$ gives a UC-secure implementation $\pi_{\text{CIDA}}^{\pi_{\text{SETUP}}}$ of $\mathcal{F}_{\text{CIDA}}$, simply by replacing the call to $\mathcal{F}_{\text{SETUP}}$ with a call to π_{SETUP} .

Optimistic Protocol. The cheater detection and identification inherent in the CESS_{MAC} openings of EVAL can be safely postponed to OUTPUT. That way,

cheating detection (MACCHECK) is batched in such a way that the communication required is independent of the number of multiplications performed, as described in Fig. 5. If MACCHECK reveals that cheating occurred, the parties will finally perform all of the relevant computations on the CESS_{MAC} commitments.

To make the protocol optimistic, parties must save all received shares of the difference values ϵ and δ from Beaver triple multiplications performed throughout the computation.¹¹ This adds an $O(nm)$ storage overhead for each party, where n is the number of parties and m is the number of multiplications in the computation. However, this does not asymptotically increase the storage requirements, because each party must store $O(nm)$ secret-shared Beaver triples anyway, which are generated during π_{SETUP} .

6 UC Threshold Verifiable Encryption

To achieve openability, we leverage *threshold verifiable encryption*. Verifiable encryption schemes support efficient zero-knowledge proofs on ciphertexts; *threshold* verifiable encryption schemes additionally require (at least a threshold number of parties in) a coalition \mathcal{C} in order to perform decryption.

We leverage a modified version of the verifiable encryption scheme described by Camenisch and Shoup [8].¹² Our modifications consist solely of removing elements from the ciphertext, so the modified scheme naturally inherits CPA security of the original (but not its CCA security). The modified version of their scheme consists of the following algorithms:

VER.KEYGEN(1^k):

1. Let $n = pq$ where $p = 2p' + 1$ and $q = 2q' + 1$, and p' and q' are k -bit primes.
2. Let $h = 1 + n$.
3. Choose random $g' \in Z_{n^2}^*$, set $g = (g')^{2n} \bmod n^2$.
4. Choose a random $sk \in \{1, \dots, \lfloor (n^2)/4 \rfloor\}$.
5. Let $pk = g^{sk} \bmod n^2$.
6. Return (pk, sk)

VER.ENC(pk, x):

1. Choose a random $r \in [n/4]$.
2. $e = (g^r \bmod n^2, pk^r h^x \bmod n^2)$.
3. Return the ciphertext e .

VER.DEC($sk, e = (u, v)$):

1. $h^x = v/(u^{sk}) \bmod n^2$.
2. Compute x (this is possible for $h = 1 + n$).
3. Return the plaintext x .

¹¹ Note that if a public transcript τ is maintained, it contains all of these difference values.

¹² Their scheme is secure against chosen ciphertext attacks, which is unnecessary for our purposes.

This encryption scheme is verifiable, because statements about the underlying plaintext can be proven using efficient zero-knowledge proofs (Appendix C of the full version of this paper). It can also be instantiated as a threshold verifiable encryption scheme, by secret-sharing the secret key among a coalition \mathcal{C} , and performing decryption in a distributed way using that secret shared key.

However, this candidate threshold verifiable encryption scheme is not universally composable. Informally, for each honest party, the simulator needs to produce an encryption e_i of their input in_i without knowing in_i . Since the encryption scheme is perfectly binding, the simulator would be unable to produce encryptions that decrypt to the correct inputs.

To overcome this problem, we add a layer of secret sharing and commitments to secret shares. The augmented construction is as follows:

THRESHVER.KEYGEN($1^k, i$):

Run $(pk_i, sk_i) \leftarrow \text{VER.KEYGEN}(1^k)$.

THRESHVER.ENC($\{pk_i\}_{i \in \mathcal{C}}, x$):

1. Additively share x into $|\mathcal{C}|$ values, denoted $[x]$.
2. For each $i \in \mathcal{C}$:
 - (a) Choose a random value r_i and compute the commitment $\mathbf{c}_i = \text{pc}([x]_{\mathcal{P}_i}, r_i)$.
 - (b) Encrypt $[x]_{\mathcal{P}_i}$ as $e_i \leftarrow \text{VER.ENC}(pk_i, [x]_{\mathcal{P}_i})$.
 - (c) Encrypt r_i as $e_{i,r} \leftarrow \text{VER.ENC}(pk_i, r_i)$.
 - (d) Compute the following non-interactive zero-knowledge proofs (Appendix C of the full version of this paper):
 - i. Proof π_i that e_i encrypts the value in \mathbf{c}_i .
 - ii. Proof $\pi_{i,r}$ that $e_{i,r}$ encrypts the decommitment value r_i for \mathbf{c}_i .
3. Return $e = \{\mathbf{c}_i, e_i, e_{i,r}, \pi_i, \pi_{i,r}\}_{i \in \mathcal{C}}$.

THRESHVER.DEC(e):

1. Each party \mathcal{P}_i ($i \in \mathcal{C}$):
 - (a) Decrypts e_i and $e_{i,r}$: $[x]_{\mathcal{P}_i} = \text{VER.DEC}(sk_i, e_i)$, $r_i = \text{VER.DEC}(sk_i, e_{i,r})$.
 - (b) Sends $[x]_{\mathcal{P}_i}$ and r_i to all other parties.
2. All parties check that $\mathbf{c}_i = \text{pc}([x]_{\mathcal{P}_i}, r_i)$. If not, REJECT.
3. All parties reconstruct x using $[x]$.

Informally, no party can cause incorrect decryption without a REJECT because that would involve breaking the binding property of the Pedersen commitment scheme.

However, a UC simulator can force decryption to return a desired value. The simulator only needs to open a commitment to an appropriate share of x , not an encryption. Since Pedersen commitments are only computationally binding, the simulator can commit to an arbitrary value, and break the binding property (using a trapdoor) to open to the share of x .¹³

¹³ The simulator chooses the generators used in the Pedersen commitment scheme when selecting the CRS; he does so in such a way that he knows their discrete log relationship, which serves as his trapdoor.

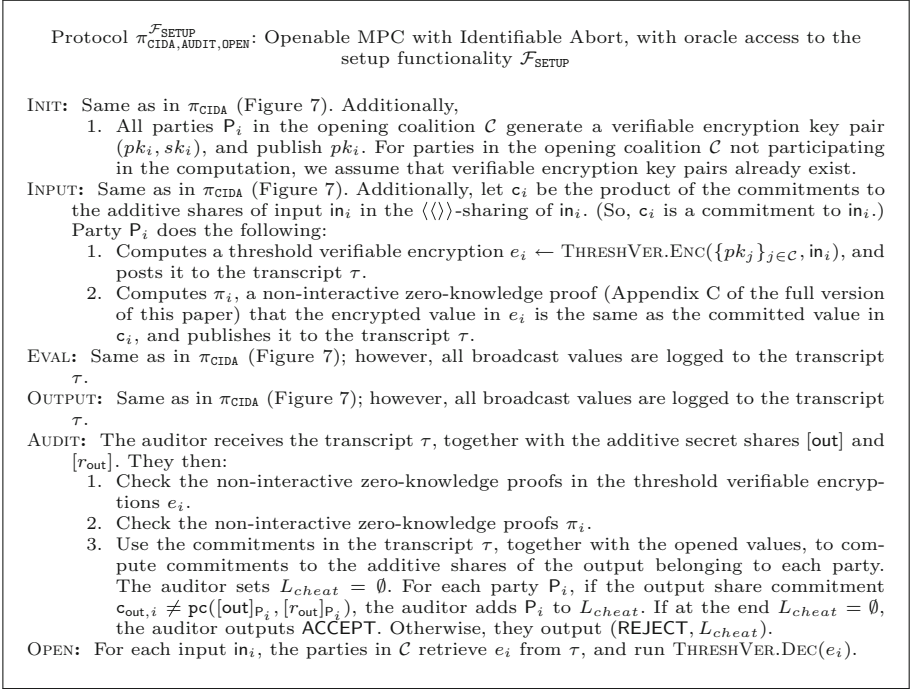


Fig. 8. Openable MPC with identifiable abort

Recall that a verifiable encryption scheme supports proofs about the underlying plaintext. Because multiple encryption public keys are now involved, we compute proofs relative to the commitments c_i , not the ciphertexts e_i . A commitment to the value x can be computed by taking the product of the additive share commitments: $c = \prod_{i \in \mathcal{C}} c_i$. Proofs like those described in Appendix C of the full version of this paper can then be done with respect to c .

Efficiency. Notice that this encryption scheme is very inefficient: a ciphertext consists of $O(|\mathcal{C}|)$ elements, as opposed to $O(1)$ elements as in the scheme of Camenisch and Shoup [8]. However, as we show in Sect. 7, we only use this scheme to encrypt computation inputs. This additional work is independent of the size of the computed circuit f .

7 Openable Auditable MPC

In this section, we augment our construction from Sect. 5 with completely identifiable auditability and openability. Completely identifiable auditability is achieved by logging all public values (including commitments from setup and publicly opened difference values) from the construction in Sect. 5 to the public transcript τ . As in Baum et al. [2], the input share commitments together with

the public values can be used to obtain a commitment to the output shares, and those commitments can be checked against the claimed output share and decommitment values.

In order to add openability, we leverage threshold verifiable encryption (Sect. 6). The augmented construction is shown in Fig. 8. Informally, each party encrypts their input in such a way that the opening coalition can decrypt it, publishes the resulting ciphertext to the transcript τ , and proves that this ciphertext encrypts the input used in the computation.

Theorem 3. *Assuming (a) that the discrete log problem (DLP) is hard in the Pedersen commitment group \mathbb{QR}_q , (b) a secure NIZKP scheme, and (c) that $(\text{THRESHVER.KEYGEN}, \text{THRESHVER.ENC}, \text{THRESHVER.DEC})$ is a semantically secure verifiable encryption scheme, the protocol $\pi_{\text{CIDA,AUDIT,OPEN}}^{\mathcal{F}_{\text{SETUP}}}$ with oracle access to the functionality $\mathcal{F}_{\text{SETUP}}$ is a UC-secure implementation of the functionality $\mathcal{F}_{\text{CIDA,AUDIT,OPEN}}$.*

Informally, Theorem 3 holds because the zero-knowledge proofs in INPUT prove that encryptions to valid shares of the input values are decryptable by the opening coalition \mathcal{C} . A proof of Theorem 3 appears in the full version of this paper.

Efficiency. To achieve completely identifiable auditability, no additional values need to be computed at all. As stated above, values that were previously broadcast are now additionally posted to the transcript.

Openability requires one additional threshold verifiable encryption (Sect. 6) to each input. Each threshold verifiable encryption entails one additive secret-sharing (to the opening coalition \mathcal{C}), and two verifiable encryptions and a commitment for each share. Additionally, $2|\mathcal{C}| + 1$ non-interactive zero-knowledge proofs (described in Appendix C of the full version of this paper) are required, where $|\mathcal{C}|$ is the size of opening coalition. This cost is small, and independent of the computation.

Acknowledgements. © 2016 Massachusetts Institute of Technology. Delivered to the US Government with Unlimited Rights, as defined in DFARS Part 252.227-7013 or 7014 (Feb 2014). Notwithstanding any copyright notice, U.S. Government rights in this work are defined by DFARS 252.227-7013 or DFARS 252.227-7014 as detailed above. Use of this work other than as specifically authorized by the U.S. Government may violate any copyrights that exist in this work. The work of Benjamin Fuller was done in part at MIT Lincoln Laboratory.

The authors would like to thank Carsten Baum, Mayank Varia, Samuel Yeom, and Arkady Yerukhimovich for helpful discussion.

References

1. Andrychowicz, M., Dziembowski, S., Malinowski, D., Mazurek, L.: Secure multi-party computations on Bitcoin. *Commun. ACM* **59**(4), 76–84 (2016)
2. Baum, C., Damgård, I., Orlandi, C.: Publicly auditable secure multi-party computation. In: Abdalla, M., De Prisco, R. (eds.) *SCN 2014*. LNCS, vol. 8642, pp. 175–196. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-10879-7_11
3. Baum, C., Orsini, E., Scholl, P.: Efficient secure multiparty computation with identifiable abort. *Cryptology ePrint Archive*, Report 2016/187 (2016). <http://eprint.iacr.org/2016/187>
4. Beaver, D.: Efficient multiparty protocols using circuit randomization. In: Feigenbaum, J. (ed.) *CRYPTO 1991*. LNCS, vol. 576, pp. 420–432. Springer, Heidelberg (1992). https://doi.org/10.1007/3-540-46766-1_34
5. Bogetoft, P., Christensen, D.L., Damgård, I., Geisler, M., Jakobsen, T., Krøigaard, M., Nielsen, J.D., Nielsen, J.B., Nielsen, K., Pagter, J., Schwartzbach, M., Toft, T.: Secure multiparty computation goes live. In: Dingleline, R., Golle, P. (eds.) *FC 2009*. LNCS, vol. 5628, pp. 325–343. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-03549-4_20
6. Boudot, F.: Efficient proofs that a committed number lies in an interval. In: Preneel, B. (ed.) *EUROCRYPT 2000*. LNCS, vol. 1807, pp. 431–444. Springer, Heidelberg (2000). https://doi.org/10.1007/3-540-45539-6_31
7. Brickell, E.F., Stinson, D.R.: The detection of cheaters in threshold schemes. In: Goldwasser, S. (ed.) *CRYPTO 1988*. LNCS, vol. 403, pp. 564–577. Springer, New York (1990). https://doi.org/10.1007/0-387-34799-2_40
8. Camenisch, J., Shoup, V.: Practical verifiable encryption and decryption of discrete logarithms. In: Boneh, D. (ed.) *CRYPTO 2003*. LNCS, vol. 2729, pp. 126–144. Springer, Heidelberg (2003). https://doi.org/10.1007/978-3-540-45146-4_8
9. Canetti, R.: Universally composable security: A new paradigm for cryptographic protocols. *Cryptology ePrint Archive*, Report 2000/067 (2000). <http://eprint.iacr.org/2000/067>
10. Chor, B., Fiat, A., Naor, M.: Tracing traitors. In: Desmedt, Y.G. (ed.) *CRYPTO 1994*. LNCS, vol. 839, pp. 257–270. Springer, Heidelberg (1994). https://doi.org/10.1007/3-540-48658-5_25
11. Cleve, R.: Limits on the security of coin flips when half the processors are faulty. In: *Proceedings of the Eighteenth Annual ACM Symposium on Theory of Computing*, STOC 1986, pp. 364–369. ACM, New York (1986)
12. Damgård, I., Fujisaki, E.: A statistically-hiding integer commitment scheme based on groups with hidden order. In: Zheng, Y. (ed.) *ASIACRYPT 2002*. LNCS, vol. 2501, pp. 125–142. Springer, Heidelberg (2002). https://doi.org/10.1007/3-540-36178-2_8
13. Damgård, I., Keller, M., Larraia, E., Pastro, V., Scholl, P., Smart, N.P.: Practical covertly secure MPC for dishonest majority – or: breaking the SPDZ limits. In: Crampton, J., Jajodia, S., Mayes, K. (eds.) *ESORICS 2013*. LNCS, vol. 8134, pp. 1–18. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-40203-6_1
14. Damgård, I., Pastro, V., Smart, N., Zakarias, S.: Multiparty computation from somewhat homomorphic encryption. In: Safavi-Naini, R., Canetti, R. (eds.) *CRYPTO 2012*. LNCS, vol. 7417, pp. 643–662. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-32009-5_38
15. Fujisaki, E., Okamoto, T.: Statistical zero knowledge protocols to prove modular polynomial relations. In: Kaliski, B.S. (ed.) *CRYPTO 1997*. LNCS, vol. 1294, pp. 16–30. Springer, Heidelberg (1997). <https://doi.org/10.1007/BFb0052225>

16. Hemenway, B., Lu, S., Ostrovsky, R., IV, W.W.: High-precision secure computation of satellite collision probabilities. Cryptology ePrint Archive, Report 2016/319 (2016). <http://eprint.iacr.org/2016/319>
17. Hemenway, B., Welser, W.I., Baiocchi, D.: Achieving higher-fidelity conjunction analyses using cryptography to improve information sharing. Technical report (2014). http://www.rand.org/pubs/research_reports/RR344.html
18. Ishai, Y., Ostrovsky, R., Sevalioglu, H.: Identifying cheaters without an honest majority. In: Cramer, R. (ed.) TCC 2012. LNCS, vol. 7194, pp. 21–38. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-28914-9_2
19. Ishai, Y., Ostrovsky, R., Zikas, V.: Secure multi-party computation with identifiable abort. In: Garay, J.A., Gennaro, R. (eds.) CRYPTO 2014. LNCS, vol. 8617, pp. 369–386. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-662-44381-1_21
20. Jakhu, R.S.: Iridium-Cosmos collision and its implications for space operations. In: Schrogl, K.U., Rathgeber, W., Baranes, B., Venet C. (eds.) Yearbook on Space Policy 2008/2009. Yearbook on Space Policy, pp. 254–275. Springer, Vienna (2010). https://doi.org/10.1007/978-3-7091-0318-0_10
21. Kumaresan, R., Bentov, I.: How to use Bitcoin to incentivize correct computations. In: Ahn, G.-J., Yung, M., Li, N. (eds.) ACM CCS 14: 21st Conference on Computer and Communications Security, 3–7 November 2014, pp. 30–41. ACM Press, Scottsdale (2014)
22. Kurosawa, K., Obana, S., Ogata, W.: t -cheater identifiable (k, n) threshold secret sharing schemes. In: Coppersmith, D. (ed.) CRYPTO 1995. LNCS, vol. 963, pp. 410–423. Springer, Heidelberg (1995). https://doi.org/10.1007/3-540-44750-4_33
23. Pedersen, T.P.: Non-interactive and information-theoretic secure verifiable secret sharing. In: Feigenbaum, J. (ed.) CRYPTO 1991. LNCS, vol. 576, pp. 129–140. Springer, Heidelberg (1992). https://doi.org/10.1007/3-540-46766-1_9
24. Rabin, T., Ben-Or, M.: Verifiable secret sharing and multiparty protocols with honest majority (extended abstract). In: 21st Annual ACM Symposium on Theory of Computing, 15–17 May 1989, pp. 73–85. ACM Press, Seattle (1989)
25. Shamir, A.: How to share a secret. Commun. Assoc. Comput. Mach. **22**(11), 612–613 (1979)
26. Spini, G., Fehr, S.: Cheater detection in SPDZ multiparty computation. In: Nascimento, A.C.A., Barreto, P. (eds.) ICITS 2016. LNCS, vol. 10015, pp. 151–176. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-49175-2_8
27. Tompa, M., Woll, H.: How to share a secret with cheaters. In: Odlyzko, A.M. (ed.) CRYPTO 1986. LNCS, vol. 263, pp. 261–265. Springer, Heidelberg (1987). https://doi.org/10.1007/3-540-47721-7_20
28. Wright, D.: Colliding satellites: consequences and implications. Union Concerned Scientists **26**, 1–10 (2009)
29. Wu, T.-C., Wu, T.-S.: Cheating detection and cheater identification in secret sharing schemes. In: IEE Proceedings - Computers and Digital Techniques, vol. 142, pp. 367–369. IET (1995)

Secure Grouping Protocol Using a Deck of Cards

Yuji Hashimoto^{1,2(✉)}, Kazumasa Shinagawa^{2,3}, Koji Nuida², Masaki Inamura¹,
and Goichiro Hanaoka²

¹ Tokyo Denki University, Saitama, Japan
yuji.ewwwd@icloud.com

² National Institute of Advanced Industrial Science and Technology, Tokyo, Japan

³ Tokyo Institute of Technology, Tokyo, Japan

Abstract. We consider a problem, which we call *secure grouping*, of dividing a number of parties into some subsets (groups) in the following manner: Each party has to know the other members of his/her group, while he/she may not know anything about how the remaining parties are divided (except for certain public predetermined constraints, such as the number of parties in each group). In this paper, we construct an information-theoretically secure protocol using a deck of physical cards to solve the problem, which is jointly executable by the parties themselves without a trusted third party. Despite the non-triviality and the potential usefulness of the secure grouping, our proposed protocol is fairly simple to describe and execute. Our protocol is based on algebraic properties of conjugate permutations. A key ingredient of our protocol is our new techniques to apply multiplication and inverse operations to hidden permutations (i.e., those encoded by using face-down cards), which would be of independent interest and would have various potential applications.




1 Introduction



Multiparty computation (MPC) is a cryptographic technology that enables two or more parties to jointly compute a multivariate function from their local inputs, in such a way that each party knows the party's local input/output pair but may not know anything about other parties' local inputs and outputs except for those implied by the party's own input/output pair only. A direction in the study of MPC, which has recently been an active branch in this area, is so-called *card-based protocols* [1–21], where protocols for MPC are supposed to use a deck of *physical cards* instead of usual electronic computers. In a card-based protocol, private information is usually encoded by using face-down cards with mutually indistinguishable back sides, and randomness is introduced by applying *shuffle operations* to some face-down cards. A typical property is that, in contrast to ordinary computer-based MPC where each party may execute a program at local environment (hence the security has to rely on certain cryptographic techniques, some of which may be only computationally secure), a card-based protocol is supposed to be executed at a public place where the parties can simply monitor and prevent the other parties' adversarial behaviors without any

cryptographic machinery. Consequently, it is usual that card-based protocols provide information-theoretic security.

For card-based protocols, it is known that every function is at least securely computable when ignoring possibly expensive computational costs [1, 10]. On the other hand, many efficient card-based protocols specialized to some typical problems have been also investigated. In those previous studies, the target problem to be solved by card-based protocols was usually a type of problem that already had an efficient computer-based counterpart, such as the case of card-based Millionaires' Problem [12]; see the Related Works paragraph below for further details. In contrast, in this paper we deal with a new type of interesting problem described below, which we call *secure grouping*; for this problem, even a computer-based solution (except ones yielded by naively applying general-purpose MPC protocols) has not been known to the authors' best knowledge.

The secure grouping is defined as the problem of dividing a number of parties into some subsets (called *groups*) in the following manner: Each party has to know the other members of his/her group, while he/she may not know anything about how the remaining parties are divided, except for certain public predetermined constraints such as the number of parties in each group. For instance, suppose that there are six parties, say, Parties 1, 2, ..., 6, and they wish to randomly divide themselves into three pairs. Some examples of the possibilities are (12, 34, and 56), (14, 26, and 35), (16, 23, and 45), etc. Then the goal is to generate one of the all possibilities uniformly at random, while each party has to know who is the partner but may not know about the other two parties.

It is worth emphasizing that such a secure grouping cannot be achieved by a simple lottery; namely, when each of the six parties in the example above picks up one of the two s, two s, and two s, there seems to be no simple way for every party to know the other party having the same card without revealing any party's card to the remaining parties. This suggests that secure grouping is really a non-trivial problem. We also note that our setting of secure grouping covers various situations, such as the case where n parties wish to randomly select two distinguished persons (like "Werewolves" in the famous Werewolf game) in such a way that only the distinguished persons themselves know who are the distinguished persons; or the dealer in a card game wishes to randomly select a partner from the other players in such a way that only the dealer and the partner him/herself know who is the dealer's partner¹. The flexibility of secure grouping would be interesting and be potentially useful.

Our Contributions. In this paper, we propose a card-based protocol to solve the problem of secure grouping explained above. As opposed to usual card-based protocols where two kinds of cards (e.g.,  and ) are used, here we use different cards (with indistinguishable back sides) whose front sides are numbers

¹ In some card games, the dealer announces one of the cards (e.g., "♠ 8") and then the player having this card becomes the dealer's partner. However, now the dealer cannot know who is the partner, though the partner him/herself can know that he/she is the dealer's partner; hence the condition of secure grouping is not achieved.

$\boxed{1}, \boxed{2}, \dots$, which we call *number cards*. A face-down card with front side \boxed{k} is called a *commitment* of k . By a rough estimate, our proposed protocol uses approximately $3dn$ number cards where n is the number of parties to be divided into groups and d is the maximal number of parties in a group.

One of our main ideas is to utilize some algebraic properties of conjugate permutations (see Sect. 3 for details). To intuitively explain, here we consider a case of dividing seven parties into two pairs and one triple. In this case, we deal with permutations of $1, 2, \dots, 7$, where a permutation σ is encoded as the sequence of number cards with front sides $\sigma^{-1}(1), \sigma^{-1}(2), \dots, \sigma^{-1}(7)$ ². Now we note that a grouping like $(ab, cd, \text{ and } efg)$ can be represented by a permutation of the form $\tau = (a, b)(c, d)(e, f, g)$, which means that τ exchanges a and b , exchanges c and d , and changes e, f, g cyclically to f, g, e , respectively. Then the problem of secure grouping is reduced to generating uniformly at random, in a committed form (i.g., each number card is faced down), a permutation ρ of the same “type” $(*, *)(*, *)(*, *, *)$ and also the square ρ^2 of the permutation; once commitments of such ρ and ρ^2 are obtained, each party, say Party i , can know the other two (or fewer) parties in his/her group by picking up the i -th face-down cards for ρ and ρ^2 . For example, when $\rho = (1, 5)(3, 6)(2, 7, 4)$, the commitments to ρ and ρ^2 are given by

$$\rho = \boxed{5} \boxed{4} \boxed{6} \boxed{7} \boxed{1} \boxed{3} \boxed{2} \quad \text{and} \quad \rho^2 = \boxed{1} \boxed{7} \boxed{3} \boxed{2} \boxed{5} \boxed{6} \boxed{4}$$

(where each card is actually faced down), and then

- for example, Party 4 obtains $\boxed{7}$ and $\boxed{2}$, which tells that Parties 7 and 2 are the other members of the group of size $3 = 2 + 1$;
- while Party 6 obtains $\boxed{3}$ and $\boxed{6}$ (the party’s own number), which tells that Party 3 is the other member of the group of size $2 = 1 + 1$.

We note that, when the sizes of the groups are at most d , a similar process can be done by using permutations $\rho, \rho^2, \dots, \rho^{d-1}$. Moreover, group theory ensures that the process of randomly shuffling the seven numbers appearing in a given permutation τ without changing the type is equivalent to taking a conjugate permutation $\sigma^{-1}\tau\sigma$ with random permutation σ of the seven numbers. Then the latter problem can be solved by using a protocol for computing multiplication and inverse of permutations in a committed form; this protocol (see Sect. 3 for details) is also a part of our contribution in this paper, which would be of independent interest. Secure grouping is now achieved by combining these ideas. See Sect. 4 for details.

The “plain” protocol explained above is seemingly applicable only to “simple” types of secure grouping where the parties have “symmetric” roles and the groups with the same number of members have “symmetric” roles as well. Nevertheless, in fact the idea of the protocol is also applicable to more complex types

² Note that this sequence of number cards is obtained by moving, for each $k = 1, 2, \dots, 7$, the k -th card \boxed{k} to the $\sigma(k)$ -th position. For example, if $\sigma(k) = k + 1$ for $1 \leq k \leq 6$ and $\sigma(7) = 1$, then the resulting card sequence is $\boxed{7} \boxed{1} \boxed{2} \boxed{3} \boxed{4} \boxed{5} \boxed{6}$.

of secure grouping. For example, in the aforementioned case of selecting two distinguished persons, we can use secure grouping of type $(*,*)(*)(*)\cdots(*)$ and then the only group with two members specifies the two distinguished persons. On the other hand, in another aforementioned case of choosing a partner of the dealer (numbered as Player 1), we can use our secure grouping protocol starting from a permutation $(1,2)(3)(4)\cdots(n)$ and then shuffling all the numbers *except the number 1* (i.e., the random permutation σ is chosen with constraint $\sigma(1) = 1$); now the resulting permutation ρ is of the form $(1,k)(*)(*)\cdots(*)$, the number k on the card picked up by Player 1 (dealer) specifies the partner, and the partner will pick up the card $\boxed{1}$ which tells that he/she is the dealer's partner. Moreover, we can also handle the cases where the groups with equal numbers of parties have to be mutually distinguished, by appropriately introducing some dummy number cards indicating the "names" of groups and then shuffling all the numbers except for dummy numbers. These examples show the flexibility of our proposed protocol.

Related Works. It is known that every function can be securely computed based on a deck of cards [1, 10]. Besides researches for improving general-purpose protocols, the other important direction is to investigate efficient card-based protocols customized to some useful applications: for example, the problem of generating secret permutations without fixed points [1, 3], *secure voting* [6, 17], and *Millionaires' Problem* [12]. In the early research of card-based cryptography, Crépeau and Kilian [1] constructed a protocol that randomly selects a permutation with no fixed point without revealing which one was selected. It has an application for e.g., exchanging gifts among multiple players in which each player does not receive his/her own gift. Recently, Ishikawa et al. [3] introduced a new shuffle called a *Pile-Scramble Shuffle* to improve the protocol in [1]. We use the Pile-Scramble Shuffles in the construction of our protocols. For the secure voting, Mizuki, Asiedu, and Sone [6] constructed a protocol for two candidates, which takes n bits as inputs and outputs the sum of the inputs. Recently, Shinagawa et al. [17] constructed a secure voting protocol for multiple candidates based on a new type of cards. For the Millionaires' Problem, Nakai et al. [12] constructed a protocol, which takes two strings x, y as inputs and outputs a bit indicating whether $x > y$ or not.

2 Preliminaries

In this section we prepare necessary tools to construct our secure grouping protocol. We suppose that a distinct number from 1 to n is assigned to each player in advance, where n is the total number of players, and the correspondence between the numbers and the players is publicly known. We identify a player with the assigned number. Throughout this paper, S_n denotes the group of permutations on the set $\{1, 2, \dots, n\}$ of numbers.

2.1 Definitions and Properties About Permutations

In this subsection, we describe some definitions related to permutations and look at their properties.

Definition 1 (cyclic permutation). A permutation τ is called a cyclic permutation if there are a unique integer $r > 1$ and distinct numbers i_1, i_2, \dots, i_r satisfying the following conditions:

- We have $\tau(i_1) = i_2, \dots, \tau(i_{r-1}) = i_r$, and $\tau(i_r) = i_1$.
- We have $\tau(k) = k$ for any number k different from i_1, i_2, \dots, i_r .

In this case, we call the permutation τ a cycle of length r and write it as (i_1, i_2, \dots, i_r) (or simply $(i_1 i_2 \dots i_r)$ if no ambiguity occurs).

In the case above, the set $\{i_1, i_2, \dots, i_r\}$ is called the *cyclic area* of the cyclic permutation $\tau = (i_1, i_2, \dots, i_r)$. For example, the permutation $\tau \in S_4$ given by $(\tau(1), \tau(2), \tau(3), \tau(4)) = (1, 4, 2, 3)$ is a cycle (243) of length three with cyclic area $\{2, 3, 4\}$, while $\sigma \in S_4$ given by $(\sigma(1), \sigma(2), \sigma(3), \sigma(4)) = (2, 1, 4, 3)$ is not a cyclic permutation.

We say that two cyclic permutations σ, τ with cyclic areas C_σ, C_τ , respectively, are *disjoint* if $C_\sigma \cap C_\tau = \emptyset$. For example, the two cyclic permutations (123) and (45) are disjoint, while (264) and (345) are not disjoint. We note that disjoint cyclic permutations are commutative in the group of permutations.

The following fact about permutations is well-known.

Proposition 1. Any permutation is uniquely represented by the product of disjoint cyclic permutations.

For example, the permutation $\tau \in S_6$ given by $\tau(1) = 2, \tau(2) = 3, \tau(3) = 1, \tau(4) = 4, \tau(5) = 6$, and $\tau(6) = 5$ is decomposed into disjoint cycles as $\tau = (123)(56)$. We also note that, it is convenient to consider as if a permutation σ virtually involves “cycle (k) of length one” when $\sigma(k) = k$; by using the abused notation, the permutation $\tau \in S_6$ above can be also represented by $\tau = (123)(4)(56)$.

Next we define the *type of permutation*. Type of permutation τ is the data of how many cycles of each length are present in the decomposition of τ into disjoint cycles as above.

Definition 2 (type of permutation). Let $\tau \in S_n$, which is decomposed into disjoint cycles (including the virtual “cycles of length one” as mentioned above). For each $i = 1, 2, \dots, n$, let m_i denote the number of cycles of length i in the decomposition of τ . Then we say that τ is of type $\langle 1^{m_1}, 2^{m_2}, \dots, n^{m_n} \rangle$; here the terms i^{m_i} with $m_i = 0$ may be omitted in the notation.

Note that $\langle 1^{m_1}, 2^{m_2}, \dots, n^{m_n} \rangle$ can be also viewed as the set of permutations of type $\langle 1^{m_1}, 2^{m_2}, \dots, n^{m_n} \rangle$. For example the permutation $\tau = (13)(25)(798) = (13)(25)(4)(6)(798) \in S_9$ belongs to the set $\langle 1^2, 2^2, 3^1 \rangle$.

2.2 Number Cards

We use cards with numbers written on the front since these are convenient for treating permutations of numbers $1, 2, \dots, n$ directly³. We call the cards *number cards* and write them as below.

$$\boxed{1} \boxed{2} \cdots \boxed{n}$$

The backs of number cards are indistinguishable. We denote the back of a number card by $\boxed{?}$. A face-down card is called *commitment*, and an operation to flip a face-down card into a face-up card is called *open*. Using the number cards, permutations in S_n are represented by a card sequence (x_1, x_2, \dots, x_n) in a certain way explained later.

We also use the term “permutation” as an *operation* for card sequences. That is, we say “applying a permutation σ to a card sequence x ” in the sense that rearranging x according to σ , formally defined as follows.

Definition 3 (applying a permutation to a card sequence). *Let $\sigma \in S_n$ be a permutation and let $x = (x_1, x_2, \dots, x_n)$ be a card sequence. We define a card sequence $\sigma(x)$ obtained by applying the permutation σ to the sequence x by*

$$\sigma(x) := (x_{\sigma^{-1}(1)}, x_{\sigma^{-1}(2)}, \dots, x_{\sigma^{-1}(n)}).$$

In other words this operation moves each i -th card to the $\sigma(i)$ -th position. For example, when $\sigma = (13)(265)(4)(7) \in S_7$ and $x = (x_1, \dots, x_7)$, we have $\sigma(x) = (x_3, x_5, x_1, x_4, x_6, x_2, x_7)$. For the special case, the identity permutation $\text{id}_n \in S_n$ is the identity operation such that a card sequence (x_1, x_2, \dots, x_n) is moved to a card sequence (x_1, x_2, \dots, x_n) itself.

Definition 4 (card sequence representing a permutation). *Let $\sigma \in S_n$ be a permutation. We define the card sequence for permutation σ to be the card sequence $\sigma(\boxed{1}, \boxed{2}, \dots, \boxed{n})$ obtained by applying σ to the card sequence $x = (x_1, x_2, \dots, x_n)$ with $x_i = \boxed{i}, i = 1, 2, \dots, n$.*

For example a permutation $\tau = (12)(34)(567) \in S_7$ is represented by the following card sequence

$$\boxed{2} \boxed{1} \boxed{4} \boxed{3} \boxed{7} \boxed{5} \boxed{6} .$$

³ Usually, we define coding rules such as $\boxed{\clubsuit} \boxed{\heartsuit} = 0$ and $\boxed{\heartsuit} \boxed{\clubsuit} = 1$ since the card-based protocol normally uses Boolean values. If the usual Boolean encoding rule is used instead of the number cards, the secure grouping protocol can still be executed. In the case the number of cards increases $2^{\lceil \log_2 n \rceil}$ times larger.

2.3 Pile-Scramble Shuffle

A *shuffle*, which is an operation to apply a random permutation chosen from some distribution, plays an important role in card-based cryptography. While different types of shuffles are proposed and used for various applications, we use one of the shuffles called *Pile-Scramble Shuffles*. It is proposed by Ishikawa et al. [3] and believed to be an “efficient shuffle” since it has an easy implementation by e.g., utilizing physical envelopes.

Definition 5 (Pile-Scramble Shuffle). *Let $n \geq 1$ be any integer. The Pile-Scramble Shuffle of degree n is the operation that takes a card sequence $x = (x_1, x_2, \dots, x_n)$ and outputs $r(x) = (x_{r^{-1}(1)}, x_{r^{-1}(2)}, \dots, x_{r^{-1}(n)})$ where $r \in S_n$ is a random permutation and hidden from all parties.*

Pile-Scramble Shuffle is described by using the following notation:

$$\|\boxed{?}\boxed{?}\dots\boxed{?}\|(x) \rightarrow \boxed{?}\boxed{?}\dots\boxed{?}(r(x)).$$

We also define a similar operation for the case where each component x_i of x is not a single card but some other object, such as a collection of multiple cards.

3 Permutation Randomizing Protocol

In this section, we present a new protocol called *permutation randomizing protocol* which is used as the main building block in our secure grouping protocol. This section is our main technical contribution part. In the simplest situation for our protocol, given an input permutation τ that is publicly known, this protocol outputs a committed card sequence representing a random permutation of the same type as τ . We emphasize that this functionality cannot be achieved by using naive shuffles since the Pile-Scramble Shuffle in general changes the type of a permutation. Therefore, we need to realize an operation on permutations that does not change the type. The key mathematical fact here is that any permutation ρ that is conjugate to a permutation τ has the same type as τ . More precisely, we utilize the following well-known property in group theory:

Lemma 1. *Let $\pi \in S_n$ be any permutation, which is expressed as the decomposition into disjoint cyclic permutations. Let $\nu \in S_n$, and let π' denote the permutation obtained by changing each number j appearing in the expression of π to the number $\nu^{-1}(j)$. Then we have $\pi' = \nu^{-1}\pi\nu$.*

Proof. Let $a \in \{1, 2, \dots, n\}$ and let $b := \pi'(a)$. Then b is (cyclically) next to a in the expression of π' as the decomposition into disjoint cyclic permutations. By the definition of π' , this implies that $\nu(b)$ is (cyclically) next to $\nu(a)$ in the expression of π , which means that $\pi(\nu(a)) = \nu(b)$. Hence we have $\nu^{-1}\pi\nu(a) = \nu^{-1}(\nu(b)) = b$, therefore π' and $\nu^{-1}\pi\nu$ are equal as permutations.

3.1 Permutation Division Protocol

Here we propose a protocol, called the *permutation division protocol*, which is the main ingredient of our permutation randomizing protocol. Given committed card sequences for permutations $v, w \in S_n$ as inputs, this protocol outputs the committed card sequence for permutation $v^{-1}w \in S_n$. As explained later, this protocol enables us to generate a committed card sequence for a permutation $\sigma^{-1}\tau\sigma$ as in Lemma 1 from given card sequences for σ, τ .

This protocol is composed of four steps as follows. Here, for any permutation x , we write “ (x) ” to mean that the displayed card sequence in a figure is the committed card sequence for x , while we also write x to indicate that the displayed card sequence is the opened card sequence for “ x ”.

1. Arrange the committed card sequences for v and w as in the figure below.

$$\begin{array}{c} \boxed{?} \boxed{?} \dots \boxed{?} \quad (v) \\ \boxed{?} \boxed{?} \dots \boxed{?} \quad (w) \end{array}$$

2. Apply Pile-Scramble Shuffle to the first and the second rows simultaneously,

$$\left\| \begin{array}{c} \boxed{?} \boxed{?} \dots \boxed{?} \\ \boxed{?} \boxed{?} \dots \boxed{?} \end{array} \right\| \begin{array}{c} (v) \\ (w) \end{array} \rightarrow \begin{array}{c} \boxed{?} \boxed{?} \dots \boxed{?} \\ \boxed{?} \boxed{?} \dots \boxed{?} \end{array} \begin{array}{c} (rv) \\ (rw) \end{array}$$

where $r \in S_n$ is a uniformly random permutation.

3. Open the first row, which reveals the permutation rv . Then apply the permutation $(rv)^{-1} = v^{-1}r^{-1}$ to the second row. More precisely, the latter operation can be efficiently performed by rearranging the n columns of the two rows in a way that the first row becomes the sequence $(1, 2, \dots, n)$ representing $\text{id}_n \in S_n$ where $\boxed{*}$ denote a face-up card having some $i \in \{1, 2, \dots, n\}$.

$$\begin{array}{c} \boxed{*} \boxed{*} \dots \boxed{*} \quad rv \\ \boxed{?} \boxed{?} \dots \boxed{?} \quad (rw) \end{array} \rightarrow \begin{array}{c} \boxed{1} \boxed{2} \dots \boxed{n} \quad \text{id}_n \\ \boxed{?} \boxed{?} \dots \boxed{?} \quad (v^{-1}r^{-1}rw) \end{array}$$

4. Output the second row (note that now $v^{-1}r^{-1}rw = v^{-1}w$).

$$\boxed{?} \boxed{?} \dots \boxed{?} \quad (v^{-1}w)$$

The correctness of our protocol has been explained above. On the other hand, the following property holds for the security of our protocol.

Proposition 2. *The distribution of the only data available during the protocol, which is the card sequence for $rv \in S_n$ opened at Step 3, is uniform and is independent of v and w .*

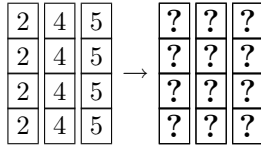
Proof. Indeed, for any $u \in S_n$, the number of the possible choice of the uniformly random r that satisfies $rv = u$ is 1 (i.e., $r = uv^{-1}$). Hence, the permutation rv appearing at Step 3 is uniformly random and independent of v, w , as desired.

3.2 Permutation Randomizing Protocol

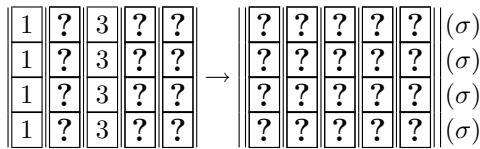
Here we describe our permutation randomizing protocol. Given an input permutation τ that is publicly known, this protocol outputs a committed card sequence representing a random permutation of the same type as τ . In addition to the degree n of permutations, our protocol in a general form takes an integer $k \geq 1$ (which is the number of input permutations) and a subset I of $\{1, 2, \dots, n\}$ as public parameters; we call the set I as the *fixing set* of our protocol. By introducing the fixing set, we can, for example, use our secure grouping protocol starting from a permutation $(1, 2)(3)(4) \cdots (n)$ and then shuffling all the numbers *except the number 1* (i.e., the random permutation σ is chosen with constraint $\sigma(1) = 1$). Such a generalized setting for the protocol here is required in our construction of the secure grouping protocol that flexibly covers various situations.

Let $\tau_1, \tau_2, \dots, \tau_k \in S_n$ be publicly known inputs for the protocol. Then our permutation randomizing protocol with fixing set I is performed as follows. In the figures below, we consider an example where $n = 5$, $k = 2$, and $I = \{1, 3\}$.

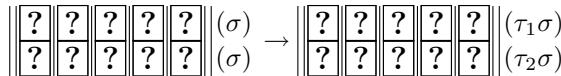
1. Arrange $2k$ times the opened cards for numbers in $\{1, 2, \dots, n\} \setminus I$ in increasing order, and face down the cards.



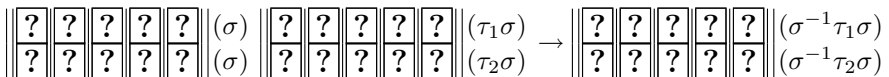
2. Apply Pile-Scramble Shuffle to the $2k$ rows simultaneously.
3. For each of $2k$ rows, insert the opened cards for numbers in I to the row in a way that the number card a for $a \in I$ is at the a -th column. Then face down all the inserted cards. Note that the resulting committed card sequences represent the same (partially shuffled) permutation in S_n , say σ .



4. For each $i = 1, 2, \dots, k$, apply the permutation τ_i to one of the committed card sequences for σ generated above.



5. For each $i = 1, 2, \dots, k$, perform the permutation division protocol for committed card sequences for σ and $\tau_i \sigma$. Then output the resulting sequences.



We note that the (committed) permutation σ generated in Step 3 is a uniformly random permutation in S_n satisfying that $\sigma(j) = j$ for every $j \in I$. For the security of the protocol, the following property is deduced straightforwardly from Proposition 2.

Proposition 3. *The distribution of the only data available during the protocol, which is the k card sequences opened during the permutation division protocols at Step 5, is uniform and is independent of the permutations σ and $\sigma^{-1}\tau_i\sigma$ for $i = 1, 2, \dots, k$.*

4 Secure Grouping Protocol

In this section we present a construction of a secure grouping protocol, which is based on the permutation randomizing protocol described above. See also “Our Contributions” paragraph in the introduction for an intuitive idea of our construction of the protocol.

4.1 Our Setting for Grouping

Before presenting our proposed secure grouping protocol, here we clarify our setting for the grouping problem. We suppose that there are n players, indexed by numbers $1, 2, \dots, n$, to be divided into groups. In our secure grouping protocol, the number of groups with k members for each $k \geq 1$, denoted by $\mathcal{M}(k)$, should be determined in advance and is treated as public information. Note that the integers $\mathcal{M}(k)$ satisfy that $\mathcal{M}(k) \geq 0$ for each $k \geq 1$ and $\sum_{k \geq 1} \mathcal{M}(k) = n$. We may express \mathcal{M} by the sequence $(\mathcal{M}(1), \mathcal{M}(2), \dots, \mathcal{M}(k))$ where k is the maximal integer satisfying $\mathcal{M}(k) > 0$.

Our protocol can also handle a certain kind of constraints on the groupings, specified in the following manner. For each integer $k \geq 1$, let \mathcal{C}_k be a (possibly empty) set of non-empty subsets of $\{1, 2, \dots, n\}$. Let \mathcal{C} be the sequence of $\mathcal{C}_1, \mathcal{C}_2, \dots$. The meaning of a constraint \mathcal{C} is the following:

- For each $k \geq 1$ and each $C \in \mathcal{C}_k$, the players in C must belong to the same group of size k .
- For any $k, k' \geq 1$, $C \in \mathcal{C}_k$, and $C' \in \mathcal{C}_{k'}$, if $C \neq C'$, then the players in C and the players in C' must belong to different groups.

Accordingly, the sets $\mathcal{C}_1, \mathcal{C}_2, \dots$ must satisfy the following conditions:

- For any $C \in \mathcal{C}_k$, we have $1 \leq |C| \leq k$.
- For any $C \in \mathcal{C}_k$ and $C' \in \mathcal{C}_{k'}$ with $k \neq k'$, the subsets C, C' of $\{1, 2, \dots, n\}$ must be (different and) disjoint with each other.
- For any $C, C' \in \mathcal{C}_k$, C and C' must be disjoint unless these are equal.
- For any $k \geq 1$, we have $|\mathcal{C}_k| \leq \mathcal{M}(k)$.

Such a constraint \mathcal{C} should also be specified in advance and is also treated as public information in our proposed protocol.

We define a *grouping* of n players to be a partition \mathcal{G} of $\{1, 2, \dots, n\}$, that is, a set of disjoint non-empty subsets of $\{1, 2, \dots, n\}$ satisfying that the union of all sets in \mathcal{G} is $\{1, 2, \dots, n\}$. For each $k \geq 1$, let \mathcal{G}_k denote the (possibly empty) sets of all $A \in \mathcal{G}$ with $|A| = k$. We say that a grouping \mathcal{G} *satisfies* a constraint $(\mathcal{M}, \mathcal{C})$, if the followings hold:

- We have $|\mathcal{G}_k| = \mathcal{M}(k)$ for any $k \geq 1$.
- If $k \geq 1$ and $C \in \mathcal{C}_k$, then there is a unique group A in \mathcal{G}_k satisfying $C \subset A$; we sometimes write this group A as $A[C]$.
- If $k \geq 1$ and $C, C' \in \mathcal{C}_k$ are different, then we have $A[C] \neq A[C']$.

Note that the conditions for \mathcal{C} and \mathcal{M} introduced above ensure that the constraint can be satisfied by at least one grouping. In our proposed secure grouping protocol, each player $P \in \{1, 2, \dots, n\}$ will only receive the information on the (unique) set $A \in \mathcal{G}$ with $P \in A$; we sometimes write this group A as $A[P]$. We give some examples of the situation above for the sake of explanation.

Example 1. *We consider a case of grouping of nine players into three groups with three members, with constraints that Players 8 and 9 want to be in the same group while Player 1 does not want to be in the same group as them. This situation can be expressed by $\mathcal{M} = (0, 0, 3)$, $\mathcal{C}_1 = \mathcal{C}_2 = \emptyset$, and $\mathcal{C}_3 = \{\{1\}, \{8, 9\}\}$. Then an example of a grouping is given by $\mathcal{G} = \mathcal{G}_3 = \{\{1, 4, 6\}, \{2, 5, 7\}, \{3, 8, 9\}\}$.*

Example 2. *We consider a situation to classify five players into two distinguished persons and three ordinary persons in the following manner: each distinguished person is told who is the other distinguished person; while each ordinary person is not told who are the distinguished persons, nor who are the other ordinary persons. This situation can be realized by treating each of the three ordinary persons as an individual group of size one consisting of him/herself alone, while treating the two distinguished persons naturally as a (unique) group of size two. Accordingly, we set $\mathcal{M} = (3, 1)$ and set each \mathcal{C}_k to be an empty set. Then an example of a grouping is given by $\mathcal{G} = \{\{2\}, \{4\}, \{5\}, \{1, 3\}\}$ (hence $\mathcal{G}_1 = \{\{2\}, \{4\}, \{5\}\}$ and $\mathcal{G}_2 = \{\{1, 3\}\}$); this means that Players 2, 4, and 5 are ordinary persons, and Players 1 and 3 are the distinguished persons.*

Example 3. *We consider a slightly more complicated situation where seven players are classified into two “Role A” players, one “Role B” player, two “Role C” players, and two ordinary players. The additional requirements are as follows:*

- Each player with Role A and each player with Role C are told his/her own role, are told who is the other player with the same role as him/herself, but are told nothing about the remaining players’ roles.
- The player with Role B and each ordinary player are told his/her own role, but are told nothing about the remaining players’ roles.

In contrast to Example 2 where the ordinary and the distinguished persons can be distinguished just by the sizes of the groups (one for the former, and two for the latter), here we should distinguish Role B from the ordinary players (both would be represented by size-one groups) and Role C from Role A (both would be represented by size-two groups).

A solution is to introduce dummy indices 8 representing “Role B” and 9 representing “Role C”. Namely, we divide the nine numbers into one group consisting of the dummy index 8 and a player’s index (who becomes “Role B”), one group consisting of the dummy index 9 and two players’ indices (who become “Role C”), two groups consisting of a player’s index only (who becomes “ordinary player”), and one group consisting of two players’ indices only (who become “Role A”). Accordingly, we set the constraint to be $\mathcal{M} = (2, 2, 1)$, $\mathcal{C}_1 = \emptyset$, $\mathcal{C}_2 = \{\{8\}\}$, and $\mathcal{C}_3 = \{\{9\}\}$. An example of a grouping is given by $\mathcal{G}_1 = \{\{1\}, \{6\}\}$, $\mathcal{G}_2 = \{\{2, 7\}, \{4, 8\}\}$, and $\mathcal{G}_3 = \{\{3, 5, 9\}\}$; this means that Players 1 and 6 are ordinary players, Players 2 and 7 are the Role A players, Player 4 is the Role B player, and Players 3 and 5 are the Role C players. We note that similar ideas to introduce dummy indices representing “names of groups” can be applied to the case of more complicated groupings.

4.2 Secure Grouping Protocol for Simpler Case

Before describing our proposed secure grouping protocol in a general form, here we consider a simpler case with empty constraints, that is, the sets \mathcal{C}_k for specifying constraints for the groupings are all empty. This case includes the case mentioned in Example 2 above.

Here we suppose that the number n of players for the secure grouping and the group size function \mathcal{M} (as well as the empty constraint sets \mathcal{C}_k) are determined in advance and are public information. As a pre-computation part of the protocol, the players compute a permutation $\tau \in S_n$ as follows; note that this τ is also a public information, therefore the computation of τ does not need any secure computation protocol. Let k denote the maximal integer with $\mathcal{M}(k) > 0$. First, the players compute integers a_0, a_1, \dots, a_{k-1} recursively by $a_0 := 0$ and $a_i := a_{i-1} + i \cdot \mathcal{M}(i)$ for $1 \leq i \leq k - 1$. Then the players define τ to be the product of cyclic permutations

$$(a_{i-1} + (j - 1)i + 1 \ a_{i-1} + (j - 1)i + 2 \ \cdots \ a_{i-1} + (j - 1)i + i)$$

for all $1 \leq i \leq k$ and $1 \leq j \leq \mathcal{M}(i)$. We note that this permutation τ is of type $\langle r_1^{\mathcal{M}(r_1)}, r_2^{\mathcal{M}(r_2)}, \dots, r_\ell^{\mathcal{M}(r_\ell)} \rangle$ where r_1, r_2, \dots, r_ℓ are the integers at which the function \mathcal{M} takes a positive value. For example, if $\mathcal{M} = (3, 2, 0, 1)$, then we have

$$\tau = (1)(2)(3)(4 \ 5)(6 \ 7)(8 \ 9 \ 10 \ 11) = (4 \ 5)(6 \ 7)(8 \ 9 \ 10 \ 11) \in \langle 1^3, 2^2, 4^1 \rangle .$$

We also note that, our protocol below utilizes the permutation randomizing protocol introduced in Sect. 3 with empty fixing set $I = \emptyset$ as a sub-protocol. This sub-protocol is given a number of publicly known permutations

$\tau_1, \tau_2, \dots, \tau_\ell \in S_n$ as inputs, and outputs committed card sequences for permutations $\rho_1, \rho_2, \dots, \rho_\ell \in S_n$, where $\rho_i = \sigma^{-1}\tau_i\sigma$ with common and uniformly random permutation $\sigma \in S_n$ for each $1 \leq i \leq \ell$.

Then, given the data above including the permutation τ , the main part of our secure grouping protocol is executed as follows, where k denotes as above the maximal integer with $\mathcal{M}(k) > 0$ (which is equal to the maximal length of cyclic permutations involved in τ):

1. The players jointly execute the permutation randomizing protocol (with empty fixing set $I = \emptyset$) for input permutations $\tau, \tau^2, \dots, \tau^{k-1}$, and obtain the committed card sequences $x[\rho], x[\rho^2], \dots, x[\rho^{k-1}]$ for permutations $\rho, \rho^2, \dots, \rho^{k-1}$ with $\rho = \sigma^{-1}\tau\sigma$ (note that $\sigma^{-1}\tau^j\sigma = (\sigma^{-1}\tau\sigma)^j$ for any j).
2. Each Player i picks the i -th card $x[\rho^j]_i$ of the card sequence $x[\rho^j]$ for all $1 \leq j \leq k-1$. Then the numbers (except the number i itself) written on the front of these $k-1$ cards (that may be duplicated) show the other players in Player i 's group.

For example, if $\tau \in S_{11}$ is as above and $\sigma = (1\ 8)(2\ 6\ 3\ 7\ 10)(4\ 11) \in S_{11}$ is chosen in the protocol, then we have $k = 4$, $\rho = (1\ 9\ 7\ 4)(2\ 3)(5\ 11)$, and the card sequences satisfy

$$\begin{aligned} \text{fronts of } x[\rho] &= \boxed{4}\ \boxed{3}\ \boxed{2}\ \boxed{7}\ \boxed{11}\ \boxed{6}\ \boxed{9}\ \boxed{8}\ \boxed{1}\ \boxed{10}\ \boxed{5} \ , \\ \text{fronts of } x[\rho^2] &= \boxed{7}\ \boxed{2}\ \boxed{3}\ \boxed{9}\ \boxed{5}\ \boxed{6}\ \boxed{1}\ \boxed{8}\ \boxed{4}\ \boxed{10}\ \boxed{11} \ , \\ \text{fronts of } x[\rho^3] &= \boxed{9}\ \boxed{3}\ \boxed{2}\ \boxed{1}\ \boxed{11}\ \boxed{6}\ \boxed{4}\ \boxed{8}\ \boxed{7}\ \boxed{10}\ \boxed{5} \ . \end{aligned}$$

Then Player 3 takes the cards $\boxed{2}$, $\boxed{3}$, and $\boxed{2}$, therefore the player's group is $\{2, 3\}$. On the other hand, Player 4 takes the cards $\boxed{7}$, $\boxed{9}$, and $\boxed{1}$, therefore the player's group is $\{1, 4, 7, 9\}$.

4.3 Secure Grouping Protocol for General Case

From now, we describe our secure grouping protocol in a general case where the constraint set \mathcal{C}_k may be non-empty. We note that these sets \mathcal{C}_k are also determined in advance and publicly known. Now the pre-computation part to determine a public permutation $\tau \in S_n$ is executed as follows, where k denotes the maximal integer with $\mathcal{M}(k) > 0$:

- Initialize τ and auxiliary counters B by $\tau \leftarrow \text{id}_n$ and $B \leftarrow \{1, 2, \dots, n\} \setminus \bigcup_{j=1}^k \bigcup_{A \in \mathcal{C}_j} A$. Then do the following for each $\lambda = 1, 2, \dots, k$:
 - Do the following for each $\mu = 1, 2, \dots, \mathcal{M}(\lambda)$:
 - * If \mathcal{C}_λ contains a set, say $C = \{a_1, a_2, \dots, a_\ell\}$, then update τ and B by $\tau \leftarrow \tau \cdot (a_1\ a_2\ \dots\ a_\ell\ b_1\ b_2\ \dots\ b_{\lambda-\ell})$ and $B \leftarrow B \setminus \{b_1, b_2, \dots, b_{\lambda-\ell}\}$, where $b_1, b_2, \dots, b_{\lambda-\ell}$ are the first $\lambda - \ell$ elements of the set B ; and then remove the set C from \mathcal{C}_λ .

* If \mathcal{C}_λ is empty, then update τ and B by $\tau \leftarrow \tau \cdot (b_1 b_2 \cdots b_\lambda)$ and $B \leftarrow B \setminus \{b_1, b_2, \dots, b_\lambda\}$, where $b_1, b_2, \dots, b_\lambda$ are the first λ elements of the set B .

This procedure is constructed to ensure that the resulting τ is a permutation in S_n and satisfies the constraint $(\mathcal{M}, \mathcal{C})$. For example, if $n = 9$, $\mathcal{M} = (2, 2, 1)$ and \mathcal{C} are as in Example 3, then the computation above is performed as follows:

(Initialize) $\tau = \text{id}_9$, $\mathcal{C}_1 = \emptyset$, $\mathcal{C}_2 = \{\{8\}\}$, $\mathcal{C}_3 = \{\{9\}\}$, $B = \{1, 2, 3, 4, 5, 6, 7\}$
 $\rightarrow (\lambda = 1, \mu = 1) \tau = (1) = \text{id}_9$, $\mathcal{C}_1 = \emptyset$, $\mathcal{C}_2 = \{\{8\}\}$, $\mathcal{C}_3 = \{\{9\}\}$,
 $B = \{2, 3, 4, 5, 6, 7\}$
 $\rightarrow (\lambda = 1, \mu = 2) \tau = (2) = \text{id}_9$, $\mathcal{C}_1 = \emptyset$, $\mathcal{C}_2 = \{\{8\}\}$, $\mathcal{C}_3 = \{\{9\}\}$,
 $B = \{3, 4, 5, 6, 7\}$
 $\rightarrow (\lambda = 2, \mu = 1) \tau = (8\ 3)$, $\mathcal{C}_1 = \mathcal{C}_2 = \emptyset$, $\mathcal{C}_3 = \{\{9\}\}$, $B = \{4, 5, 6, 7\}$
 $\rightarrow (\lambda = 2, \mu = 2) \tau = (8\ 3)(4\ 5)$, $\mathcal{C}_1 = \mathcal{C}_2 = \emptyset$, $\mathcal{C}_3 = \{\{9\}\}$, $B = \{6, 7\}$
 $\rightarrow (\lambda = 3, \mu = 1) \tau = (8\ 3)(4\ 5)(9\ 6\ 7)$, $\mathcal{C}_1 = \mathcal{C}_2 = \mathcal{C}_3 = \emptyset$, $B = \emptyset$

We also note that, our protocol below utilizes the permutation randomizing protocol with fixing set $I = \bigcup_{j=1}^k \bigcup_{A \in \mathcal{C}_j} A \subset \{1, 2, \dots, n\}$ introduced in Sect. 3. This sub-protocol is given publicly known permutations $\tau_1, \tau_2, \dots, \tau_\ell \in S_n$ as inputs, and outputs committed card sequences for permutations $\rho_1, \rho_2, \dots, \rho_\ell \in S_n$, where for each i , $\rho_i = \sigma^{-1} \tau_i \sigma$ with common and uniformly random permutation $\sigma \in S_n$ satisfying that $\sigma(a) = a$ for every $a \in I$.

Then, given the data above including the permutation τ , the main part of our secure grouping protocol is executed as follows, where k denotes as above the maximal integer with $\mathcal{M}(k) > 0$:

1. The players jointly execute the permutation randomizing protocol with fixing set I for input permutations $\tau, \tau^2, \dots, \tau^{k-1}$, and obtain the committed card sequences $x[\rho], x[\rho^2], \dots, x[\rho^{k-1}]$ for permutations $\rho, \rho^2, \dots, \rho^{k-1}$ with $\rho = \sigma^{-1} \tau \sigma$ (note that $\sigma^{-1} \tau^j \sigma = (\sigma^{-1} \tau \sigma)^j$ for any j).
2. Each Player i picks the i -th card $x[\rho^j]_i$ of the card sequence $x[\rho^j]$ for all $1 \leq j \leq k - 1$. Then the numbers (except the number i itself) written on the front of these $k - 1$ cards (that may be duplicated) show the other players in Player i 's group.

We note that, if $\mathcal{C}_i = \emptyset$ for any $1 \leq i \leq k$, then the protocol above coincides with the protocol described in Sect. 4.2.

5 Proofs of Correctness and Security

In this section, we prove the correctness and the security of our proposed secure grouping protocol.

5.1 Proof of Correctness

In this subsection, we prove the correctness of our secure grouping protocol as follows:

Theorem 1. *Let $(\mathcal{M}, \mathcal{C})$ be a possible constraint for our protocol. Then our secure grouping protocol with constraint $(\mathcal{M}, \mathcal{C})$ generates each grouping \mathcal{G} satisfying the constraint $(\mathcal{M}, \mathcal{C})$ with equal probability.*

To prove the theorem, we introduce some auxiliary definitions. First, let $\pi \in S_n$ be a permutation and let $\pi = \pi_1 \pi_2 \cdots \pi_\ell$ be the decomposition of π into disjoint cyclic permutations π_1, \dots, π_ℓ , where the cyclic permutations of length 1 are also included in the decomposition. Then we define the grouping $\mathcal{G}[\pi]$ specified by π to be the set of the cyclic areas of the cyclic permutations $\pi_1, \pi_2, \dots, \pi_\ell$. For example, if $\pi = (1\ 5)(4)(2\ 6\ 3) \in S_6$, then $\mathcal{G}[\pi] = \{\{4\}, \{1, 5\}, \{2, 3, 6\}\}$.

Secondly, we say that a permutation $\pi \in S_n$ satisfies the constraint $(\mathcal{M}, \mathcal{C})$, if the following conditions are satisfied:

- Let $r_1 < r_2 < \cdots < r_L$ be all the positive integers with $\mathcal{M}(r_i) > 0$. Then $\pi \in \langle r_1^{\mathcal{M}(r_1)}, r_2^{\mathcal{M}(r_2)}, \dots, r_L^{\mathcal{M}(r_L)} \rangle$.
- Let $k \geq 1$ and $C = \{a_1, a_2, \dots, a_h\} \in \mathcal{C}_k$ (we assume that the elements a_1, a_2, \dots, a_h of any set $C \in \mathcal{C}_k$ are always written in increasing order, in our argument below as well as the construction of the secure grouping algorithm). Then the numbers a_1, a_2, \dots, a_h are involved in the cyclic area of the same cyclic permutation in the decomposition of π , say π_i , and we have $\pi_i(a_j) = a_{j+1}$ for any $1 \leq j \leq h - 1$.

We note that, if $\pi \in S_n$ satisfies the constraint $(\mathcal{M}, \mathcal{C})$, then the grouping $\mathcal{G}[\pi]$ satisfies the constraint $(\mathcal{M}, \mathcal{C})$ as well. We note also that, by the construction, the permutation $\tau \in S_n$ computed in the pre-computation part of our secure grouping protocol with constraint $(\mathcal{M}, \mathcal{C})$ satisfies the constraint $(\mathcal{M}, \mathcal{C})$ in the sense above.

Now we show the following property:

Lemma 2. *Let $\rho \in S_n$ be the permutation generated (in the committed form) in our secure grouping protocol. Then the output of our secure grouping protocol is $\mathcal{G}[\rho]$.*

Proof. Let k denote the integer specified in the construction of the protocol. Let $i \in \{1, 2, \dots, n\}$, and let ρ_i denote the cyclic permutation in the decomposition of ρ whose cyclic area contains i . Then, by the definition of the card sequence representing a permutation, the numbers written on the cards obtained by Player i at the end of the protocol are $(\rho^j)^{-1}(i) = \rho^{-j}(i) = \rho_i^{-j}(i)$ for $j = 1, 2, \dots, k - 1$. Moreover, by the definition of k , the length of the cyclic permutation ρ_i is at most k , therefore the set of those numbers $\rho_i^{-j}(i)$ for $j = 1, 2, \dots, k - 1$ together with the number i itself is equal to the group in $\mathcal{G}[\rho]$ containing i , the latter being the cyclic area of ρ_i by definition. Hence the claim holds.

By Lemmas 1, 2 and the fact that the (partially shuffled) permutation $\sigma \in S_n$ generated in the permutation randomizing protocol fixes each element of the fixing set I , it follows that the output of our secure grouping algorithm is a grouping satisfying the given constraint $(\mathcal{M}, \mathcal{C})$.

Moreover, since the permutation $\sigma \in S_n$ generated in the permutation randomizing protocol is chosen uniformly at random from all the permutations in S_n that fixes every element of I , the following property is deduced straightforwardly by Lemma 1:

Lemma 3. *Given the input $\tau, \tau^2, \dots, \tau^{k-1}$ for the permutation randomizing protocol executed internally in our secure grouping protocol, the (committed) permutations $\rho, \rho^2, \dots, \rho^{k-1}$ corresponding to the output of the permutation randomizing protocol satisfy that ρ is uniformly random over the set of all permutations in S_n satisfying the constraint $(\mathcal{M}, \mathcal{C})$.*

On the other hand, the following property is deduced straightforwardly by the definition of the grouping $\mathcal{G}[\pi]$ specified by a permutation π :

Lemma 4. *Let $(\mathcal{M}, \mathcal{C})$ be a given constraint. For any grouping \mathcal{G} satisfying the constraint $(\mathcal{M}, \mathcal{C})$, the number of permutations π that satisfies the constraint $(\mathcal{M}, \mathcal{C})$ and satisfies $\mathcal{G}[\pi] = \mathcal{G}$ is independent of the choice of \mathcal{G} .*

Now our claim follows by combining the last two lemmas: Namely, for any two groupings $\mathcal{G}, \mathcal{G}'$ satisfying the constraint $(\mathcal{M}, \mathcal{C})$, the number of permutations ρ satisfying the constraint $(\mathcal{M}, \mathcal{C})$ that specifies the grouping \mathcal{G} is equal to the number of those permutations that specifies the grouping \mathcal{G}' , and those permutations ρ are chosen with equal probability. This completes the proof.

5.2 Proof of Security

In this subsection, we prove the security of our secure grouping protocol as follows:

Theorem 2. *Let $(\mathcal{M}, \mathcal{C})$ be a possible constraint for our secure grouping protocol. Let \mathcal{G} denote the grouping which is the output of our secure grouping protocol with constraint $(\mathcal{M}, \mathcal{C})$. Then, for any Player i , the information obtained by the player during the protocol is independent of the groups $A \in \mathcal{G}$ that do not contain i .*

To prove the theorem, we first note that, the argument in the proof of Lemma 2 implies that the output of Player i in the secure grouping protocol is the sequence of numbers $(\rho^{-1}(i), \rho^{-2}(i), \dots, \rho^{-(k-1)}(i))$, where ρ is the permutation generated (in the committed form) in the protocol. Let ρ_i denote the unique cyclic permutation involved in ρ that contains the number i . Then, by using the output above, Player i can recover not only the cyclic area of ρ_i (which is an *unordered* set) but also the whole of the cyclic permutation ρ_i itself. Therefore, the information obtained by Player i during the protocol is the cyclic permutation ρ_i as well as the card sequences that are opened during the permutation randomizing protocol. Moreover, Proposition 3 implies that the latter cards opened

during the permutation randomizing protocol provides essentially no information, therefore it suffices to concern the information on the cyclic permutation ρ_i only.

Now the following property is deduced straightforwardly by the definition of the grouping $\mathcal{G}[\pi]$ specified by a permutation π :

Lemma 5. *Let i and ρ_i be as above. Let \mathcal{G}' and \mathcal{G}'' be any grouping satisfying the constraint $(\mathcal{M}, \mathcal{C})$, in which the group including i is equal to the cyclic area of ρ_i . Then, among the permutations $\tilde{\rho}$ whose decomposition into disjoint cyclic permutations involves ρ_i , the number of those permutations that satisfies $\mathcal{G}[\tilde{\rho}] = \mathcal{G}'$ is equal to the number of those permutations $\tilde{\rho}$ that satisfies $\mathcal{G}[\tilde{\rho}] = \mathcal{G}''$.*

Since the choice of the permutation ρ is uniformly random, it follows by Lemma 5 that the conditional distribution of the grouping \mathcal{G} generated by our secure grouping algorithm, except the group including i , conditioned on the choice of the cyclic permutation ρ_i is still the uniform distribution. This completes the proof.

Acknowledgement. We thank the members of Shin-Akarui-Angou-Benkyou-Kai for their helpful comments. In particular we would like to thank Shuichi Katsumata for his helpful comments. A part of this work is supported by JST CREST grant number JPMJCR1688.

References

1. Crépeau, C., Kilian, J.: Discreet solitary games. In: Stinson, D.R. (ed.) CRYPTO 1993. LNCS, vol. 773, pp. 319–330. Springer, Heidelberg (1994). https://doi.org/10.1007/3-540-48329-2_27
2. Boer, B.: More efficient match-making and satisfiability *The Five Card Trick*. In: Quisquater, J.-J., Vandewalle, J. (eds.) EUROCRYPT 1989. LNCS, vol. 434, pp. 208–217. Springer, Heidelberg (1990). https://doi.org/10.1007/3-540-46885-4_23
3. Ishikawa, R., Chida, E., Mizuki, T.: Efficient card-based protocols for generating a hidden random permutation without fixed points. In: Calude, C.S., Dinneen, M.J. (eds.) UCNC 2015. LNCS, vol. 9252, pp. 215–226. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-21819-9_16
4. Koch, A., Walzer, S., Härtel, K.: Card-based cryptographic protocols using a minimal number of cards. In: Iwata, T., Cheon, J.H. (eds.) ASIACRYPT 2015. LNCS, vol. 9452, pp. 783–807. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-662-48797-6_32
5. Mizuki, T.: Efficient and secure multiparty computations using a standard deck of playing cards. In: Foresti, S., Persiano, G. (eds.) CANS 2016. LNCS, vol. 10052, pp. 484–499. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-48965-0_29
6. Mizuki, T., Asiedu, I.K., Sone, H.: Voting with a logarithmic number of cards. In: Mauri, G., Dennunzio, A., Manzoni, L., Porreca, A.E. (eds.) UCNC 2013. LNCS, vol. 7956, pp. 162–173. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-39074-6_16
7. Mizuki, T., Kumamoto, M., Sone, H.: The five-card trick can be done with four cards. In: Wang, X., Sako, K. (eds.) ASIACRYPT 2012. LNCS, vol. 7658, pp. 598–606. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-34961-4_36

8. Mizuki, T., Shizuya, H.: A formalization of card-based cryptographic protocols via abstract machine. *Int. J. Inf. Sec.* **13**(1), 15–23 (2014)
9. Mizuki, T., Shizuya, H.: Practical card-based cryptography. In: Ferro, A., Luccio, F., Widmayer, P. (eds.) *FUN 2014*. LNCS, vol. 8496, pp. 313–324. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-07890-8_27
10. Mizuki, T., Sone, H.: Six-card secure AND and four-card secure XOR. In: Deng, X., Hopcroft, J.E., Xue, J. (eds.) *FAW 2009*. LNCS, vol. 5598, pp. 358–369. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-02270-8_36
11. Mizuki, T., Uchiike, F., Sone, H.: Securely computing XOR with 10 cards. *Australas. J. Comb.* **36**, 279–293 (2006)
12. Nakai, T., Tokushige, Y., Misawa, Y., Iwamoto, M., Ohta, K.: Efficient card-based cryptographic protocols for millionaires' problem utilizing private permutations. In: Foresti, S., Persiano, G. (eds.) *CANS 2016*. LNCS, vol. 10052, pp. 500–517. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-48965-0_30
13. Niemi, V., Renvall, A.: Secure multiparty computations without computers. *Theor. Comput. Sci.* **191**(1–2), 173–183 (1998)
14. Nishida, T., Hayashi, Y., Mizuki, T., Sone, H.: Card-based protocols for any boolean function. In: Jain, R., Jain, S., Stephan, F. (eds.) *TAMC 2015*. LNCS, vol. 9076, pp. 110–121. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-17142-5_11
15. Nishida, T., Mizuki, T., Sone, H.: Securely computing the three-input majority function with eight cards. In: Dediu, A.-H., Martín-Vide, C., Truthe, B., Vega-Rodríguez, M.A. (eds.) *TPNC 2013*. LNCS, vol. 8273, pp. 193–204. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-45008-2_16
16. Nishimura, A., Hayashi, Y., Mizuki, T., Sone, H.: An implementation of non-uniform shuffle for secure multi-party computation. In: *Proceedings of the 3rd ACM International Workshop on ASIA Public-Key Cryptography, AsiaPKC@AsiaCCS*, Xi'an, China, 30 May–03 June 2016, pp. 49–55 (2016)
17. Shinagawa, K., Mizuki, T., Schuldt, J.C.N., Nuida, K., Kanayama, N., Nishide, T., Hanaoka, G., Okamoto, E.: Multi-party computation with small shuffle complexity using regular polygon cards. In: Au, M.-H., Miyaji, A. (eds.) *ProvSec 2015*. LNCS, vol. 9451, pp. 127–146. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-26059-4_7
18. Shinagawa, K., Mizuki, T., Schuldt, J., Nuida, K., Kanayama, N., Nishide, T., Hanaoka, G., Okamoto, E.: Secure multi-party computation using polarizing cards. In: Tanaka, K., Suga, Y. (eds.) *IWSEC 2015*. LNCS, vol. 9241, pp. 281–297. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-22425-1_17
19. Shinagawa, K., Mizuki, T., Schuldt, J.C.N., Nuida, K., Kanayama, N., Nishide, T., Hanaoka, G., Okamoto, E.: Secure computation protocols using polarizing cards. *IEICE Trans.* **99-A**(6), 1122–1131 (2016)
20. Stiglic, A.: Computations with a deck of cards. *Theor. Comput. Sci.* **259**(1–2), 671–678 (2001)
21. Ueda, I., Nishimura, A., Hayashi, Y., Mizuki, T., Sone, H.: How to implement a random bisection cut. In: Martín-Vide, C., Mizuki, T., Vega-Rodríguez, M.A. (eds.) *TPNC 2016*. LNCS, vol. 10071, pp. 58–69. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-49001-4_5

Four Cards Are Sufficient for a Card-Based Three-Input Voting Protocol Utilizing Private Permutations

Takeshi Nakai, Satoshi Shirouchi, Mitsugu Iwamoto^(✉), and Kazuo Ohta

Department of Informatics, The University of Electro-Communications, Chofu, Japan
{t-nakai,s.shirouchi,mitsugu,kazuo.ohta}@uec.ac.jp

Abstract. The card-based cryptographic protocol is a variant of multi-party computation that enables us to compute a certain function securely by using playing cards. In existing card-based cryptographic protocols, a special operation of cards called a *shuffle* is used to achieve the information-theoretic security. Recently, card-based cryptographic protocols have been reconsidered from the viewpoint of multi-party computations. In this direction, a new model of card-based cryptographic protocol including a new assumption called *Private Permutations* (PP, for short) is introduced and succeeds in constructing efficient protocols for the millionaires' protocol. In this paper, we construct efficient card-based cryptographic OR and XOR protocols based on the existing AND protocol. Furthermore, by unifying AND and OR protocols, it is shown that a majority voting protocol with three inputs is efficiently obtained. Our construction requires only *four* cards thanks to PPs, whereas the previous work requires eight cards.

Keywords: Card-based cryptographic protocols
Multi-party computation · Logic gates · Majority voting
Private permutation

1 Introduction

1.1 Background

It is known that multi-party computation can be realized by a deck of playing cards [1], referred to as *card-based cryptographic protocols* (card-based protocol, for short). An important feature of the card-based protocols is that every operation (including the randomization) of cards is performed in front of all the players, which is free from the semi-honest model. For instance, an important randomization technique called the *random bisection cut*, which is a type of shuffling technique, requires the player who holds the bisected cards between the two stacks with the same number of cards an unspecified number of times. This is acceptable as an operation of cards, but it is infeasible from the viewpoint of information-theoretically secure multi-party computation in a rigorous sense

because the number of shuffles can be counted by the player and it could be known to all the players if the shuffle operations were to be recorded by video or other means.

To resolve this problem, a modified model of card-based protocols was studied in [3] from the viewpoint of multi-party computation. In this model, each player is allowed to have private randomness, and to operate the cards behind the player's back. This assumption is called *Private Permutation (PP)*. Furthermore, communication complexity is taken into account in [3] as an efficiency measure, which was not studied in the previous work. Under the modified model, a shuffle is broken into two PPs and a communication, and efficient protocols for the millionaires' problem [4] were proposed. Furthermore, it is proposed in [2] that an efficient card-based AND protocol is realized by utilizing the idea of PPs implicitly (e.g., see Epilogue B in [2]). However, XOR and OR protocols based on PPs are not proposed.

Observing the card-based millionaires' protocol [3] and AND protocol [2] based on PPs, we can reduce the number of cards thanks to the representation of the player's input not by the cards but by the player's actions on the cards. For instance, in the 3-card AND protocol [2], which will be explained in Protocol 2 below, Bob's input is represented by replacing the card on the left or the right. PPs represent a natural assumption for playing cards, but it is worthwhile to note that a semi-honest model must be assumed if we utilize PPs.

1.2 Contributions and Organization of This Paper

In addition to the 3-card AND protocol [2] described in Sect. 3.1, we propose the following two logic operations that are more efficient than previous work:

- 3-card OR (in Sect. 3.2), and
- 2-card XOR (in Sect. 3.3).

The interesting points of the proposed protocols are not only that we can substantially reduce the number of cards, but we can also simultaneously realize AND and OR operations. This simultaneous realization enables us to implement the card-based majority voting protocol with three inputs using *four* cards, which is the main contribution of this paper. Note that in the previous work based on shuffles [8] eight cards are necessary to implement the majority voting protocol. It is possible to extend our protocol to the majority voting with n (≥ 3) players, which will be presented in the final version of this paper.

The rest of this paper is organized as follows: We describe in Sect. 2 the previous work [2,3] and the efficiency measure summarized in Table 1 based on the previous work. Our first contribution is the efficient card-based cryptographic protocols based on PPs explained in Sect. 3, where we propose 2-card XOR and 3-card OR protocols in addition to 3-card AND in [2].

Observing the relations for $a, b \in \{0, 1\}$,

$$a \wedge b = 1 \iff a + b \geq 2$$

$$a \vee b = 1 \iff a + b \geq 1$$

and using the results in Sect. 3, we can construct the three-input majority voting protocol using *four* cards, which is the second contribution of this paper. This result reduces the previous result with eight cards by half.






In Sect. 5, we discuss the extension of our idea to a card-based cryptographic protocol using rotation of cards. Specifically, several card-based cryptographic protocols exploit the difference between a card and its rotation, e.g.,  and . Such protocols have succeeded in reducing the number of cards compared to the previous protocols [10, 11]. We show that our proposed techniques based on PPs succeed in reducing the numbers of cards compared to [10, 11].





Table 1. Comparison between previous work and our results

Protocol	References	# of PPs	# of comm.	# of Cards
AND	Mizuki–Kumamoto–Sone [6]	5	3	4
	[2] [Sect. 3.1]	2	1	3
OR	Mizuki–Kumamoto–Sone [6]	5	3	4
	This work [Sect. 3.2]	2	1	3
XOR	Mizuki–Sone [5]	3	2	4
	This work [Sect. 3.3]	2	1	2
Majority Voting with 3 inputs	Nishida–Mizuki–Sone [8]	5	3	8
	This work [Sect. 4]	3	2	4

2 Preliminaries

2.1 Shuffle and Private Permutation

We use two cards  and , and we assume that the cards with the same suit cannot be distinguished from each other. The backs of all cards are indistinguishable, and we denote this by .

In most of the previous work, e.g., [5, 6, 8, 9], card-based protocols consist of three operations such as *permutation*, *reverse*, and *shuffle*. Permutation is the permutation of cards *in public*, where every player knows what type of permutation was used. Reverse means turning over the cards, e.g.,  \mapsto  and  \mapsto .

The most important operation among these three operations is the shuffle, which is a *probabilistic* permutation of face-down cards *in public*. The shuffle must not to leak the information of the permutation used in the shuffle to any player (even the player who performs the shuffle). One of the important shuffle operations is called *random bisection cut* [5], which is described as follows:

For a positive integer v , prepare $2v$ cards. Divide these cards exactly by half, and denote them by $\mathbf{u}_0, \mathbf{u}_1$. Specifically, we denote

$$\overbrace{\boxed{?} \boxed{?} \cdots \boxed{?}}^{v \text{ cards}} \overbrace{\boxed{?} \boxed{?} \cdots \boxed{?}}^{v \text{ cards}}.$$

$=:\mathbf{u}_0$ $=:\mathbf{u}_1$

Then a player randomly swaps cards between the two card stacks (e.g., shuffles the bisected cards) an unspecified number of times. This is considered to be equivalent to randomly choosing one of the following $2v$ cards, i.e., swap \mathbf{u}_0 and \mathbf{u}_1 or not, with probability $1/2$:

$$\underbrace{\boxed{?} \boxed{?} \cdots \boxed{?}}_{\mathbf{u}_0} \underbrace{\boxed{?} \boxed{?} \cdots \boxed{?}}_{\mathbf{u}_1} \text{ or } \underbrace{\boxed{?} \boxed{?} \cdots \boxed{?}}_{\mathbf{u}_1} \underbrace{\boxed{?} \boxed{?} \cdots \boxed{?}}_{\mathbf{u}_0} \tag{1}$$

Note again that no player must know which card in (1) is chosen.

The shuffle operation has been traditionally used in card-based cryptographic protocols, but it has a merit and a demerit. Its merit is that every operation can be *made public* and it is difficult for any player to cheat if we allow the shuffle operation, which enables us to eliminate the semi-honest assumption. Meanwhile, the requirement of public randomization using the shuffle is actually *infeasible* to realize from the information-theoretic viewpoint in the rigorous sense although the shuffle is *physically* realized. For instance, we believe that shuffling the cards many times in front of all players can hide the permutation of the cards, but if we can shoot a video of the shuffle operation, the video will clarify how many times the shuffles were performed.

To remove such infeasibility, [3] proposed a modified model for the card-based protocols. The idea of [3] is that we regard the card-based cryptographic protocols as a type of multi-party computation (MPC). The modified model in [3] consists of the following four operations:

- *Permutation*: Permutation of face-down cards in public.
- *Reverse*: Turning over a card.
- *Private Permutation (PP)*: Permutation of cards in private, e.g., behind the player’s back. Private randomness is available.
- *Communication*: Handing over cards to the other player.

The PPs corresponds to the private randomness in MPC. By introducing PPs, shuffles can be divided into communications and PPs. For instance, a random bisection cut can be regarded as two PPs and a communication between two players, say Alice and Bob, as shown in Protocol 1. In Protocol 1, PPs are used in steps (1) and (3). The output of Protocol 1 is $(\mathbf{u}_{r_A \oplus r_B}, \mathbf{u}_{1-r_A \oplus r_B})$, where \oplus denotes XOR. Because r_A (r_B , resp.) is not disclosed to Bob (Alice, resp.), all players including Alice and Bob cannot know the result $(\mathbf{u}_{r_A \oplus r_B}, \mathbf{u}_{1-r_A \oplus r_B})$ until the cards are opened.

By introducing PPs, every card-based cryptographic protocol becomes theoretically feasible, and it is easy to discuss the card-based cryptographic protocol

Protocol 1. Random Bisection Cut [3]

- (1) Alice chooses $r_A \in \{0, 1\}$ with probability $1/2$ privately, and switches the order of the bisected cards $(\mathbf{u}_0, \mathbf{u}_1)$ if $r_A = 1$ behind her back, or leaves the same if $r_A = 0$. Then, she has $(\mathbf{u}_{r_A}, \mathbf{u}_{1-r_A})$ behind her back.
 - (2) Alice sends the cards $(\mathbf{u}_{r_A}, \mathbf{u}_{1-r_A})$ to Bob.
 - (3) Bob chooses $r_B \in \{0, 1\}$ with probability $1/2$ privately, and switches the order of the received cards $(\mathbf{u}_{r_A}, \mathbf{u}_{1-r_A})$ if $r_B = 1$ behind his back, or leaves the same if $r_B = 0$.
-

from the viewpoint of MPC. Instead, in compensation for the theoretical feasibility, card-based protocols based on PPs apply the semi-honest model because the PP requires *private* operations. The reconsideration of card-based protocols based on PPs was initiated by [3], but the idea of PP can be seen implicitly in [2], which will be explained in the 3-card AND protocol as shown in Protocol 2.

2.2 Efficiency Measures

In [3], the authors evaluated the efficiency of card-based cryptographic protocols by the number of PPs, communications, and cards, which correspond to the amount of randomness, communication complexity, and memory size, respectively, in ordinary MPC. In Table 1, a shuffle in previous work is counted as one communications and two PPs. We adopt the same efficiency measures in this paper.

3 Proposed Protocols for Logical Gates

Starting from the card-based cryptographic protocol for an AND gate with three cards [2], we propose 3-card OR and 2-card XOR protocols. In this section let a and b be the binary inputs of Alice and Bob, respectively.

3.1 Basic Idea: PPs and Inputs by the Player's Actions

In the Epilogue in [2] (Solution B), the 3-card AND protocol is proposed as shown in Protocol 2¹. Table 2 shows the correspondence between cards at step (2) and the output of the protocol. Subscripts of \clubsuit and \heartsuit indicate the player who had the card originally². We can observe that the PP introduced in [3] was implicitly used in steps (1) and (2) of Protocol 2.

We also note that Bob's input at the step (3) in Protocol 2 is not represented by the suit of the card but is represented by the action taken by Bob, i.e., Bob's value corresponds to his choice of left or right where he places his \clubsuit . In this

¹ Slightly modified for later discussion, but essentially the same as the protocol in [2].

² Hereafter, we remove the frame of cards for simplicity.

Protocol 2. Three-card AND Protocol [2]

Inputs: Alice has $a \in \{0, 1\}$, and Bob has $b \in \{0, 1\}$.

Setup: Alice has a pair of $\clubsuit\heartsuit$. Bob has one \clubsuit .

- (1) Alice chooses \clubsuit if $a = 0$, otherwise \heartsuit , behind her back. She sends the face-down card to Bob.
- (2) If $b = 0$, Bob puts \clubsuit to the left of the card he received. Otherwise, he places \clubsuit to the right of the card he received.
- (3) Bob reveals the left card to Alice. If this card is \clubsuit , then $a \wedge b = 0$. Otherwise, $a \wedge b = 1$. He discards the card on the right.

Table 2. Three-card AND protocol

a	b	Step (2)	Output
0	0	$\clubsuit_{\text{Bob}} \clubsuit_{\text{Alice}}$	0 (\clubsuit_{Bob})
0	1	$\clubsuit_{\text{Alice}} \clubsuit_{\text{Bob}}$	0 (\clubsuit_{Alice})
1	0	$\clubsuit_{\text{Bob}} \heartsuit_{\text{Alice}}$	0 (\clubsuit_{Bob})
1	1	$\heartsuit_{\text{Alice}} \clubsuit_{\text{Bob}}$	1 ($\heartsuit_{\text{Alice}}$)

study, we utilize this idea to express a player’s input by his/her action, and succeed in reducing the number of cards compared to previous work.

Security Proof of 3-card AND protocol: We present a brief overview of the security proof for Protocol 2, which will be useful to understand the security of the protocols proposed hereafter.

Since we compute AND, the player who inputs 1 can uniquely determine the other player’s input at the end of the protocol. Meanwhile, for the player who inputs 0, no information must leak out to the player, which we have to check. When Alice inputs $a = 0$ (\clubsuit), the output is either \clubsuit_{Alice} or \clubsuit_{Bob} , which is opened by Bob and is indistinguishable to Alice. When Bob inputs $b = 0$, he places his \clubsuit on the left, and he simply shows his card to Alice. Hence, he obtains no information on Alice’s input, which is discarded at the end of the protocol.

It is clear that no information is obtained by the players other than Alice and Bob (if such players exist) because the only information they can obtain is the output. □

3.2 Three-Card OR Protocol

Although the concept of PPs is implicitly used in [2], but this paper only concentrated on the construction of card-based AND protocols, and no card-based protocols were shown for the other logical gates. Hereafter, we show card-based protocols for computing OR and XOR, which are realized with three and two cards, respectively.

To construct card-based OR protocols, we should recall De Morgan’s law: $a \vee b = \neg(\neg a \wedge \neg b)$. Using this identity, the card-based OR protocol can be

Protocol 3. Three-Card OR Protocol

Inputs: Alice has $a \in \{0, 1\}$, and Bob has $b \in \{0, 1\}$.

Setup: Alice has a pair of $\clubsuit\heartsuit$. Bob has one \clubsuit .

- (1) Alice chooses \heartsuit if $a = 0$, otherwise \clubsuit , behind her back. She sends the face-down card to Bob.
 - (2) If $b = 0$, Bob puts \clubsuit to the right of the card he received. Otherwise, he places \clubsuit to the left of the card he received.
 - (3) Bob reveals the left card to Alice. If this card is \heartsuit , then $a \wedge b = 0$. Otherwise, $a \wedge b = 1$. He discards the card on the right.
-

Table 3. Three-Card OR Protocol

a	b	Step (2)	Output
0	0	$\heartsuit_{\text{Alice}} \clubsuit_{\text{Bob}}$	0 ($\heartsuit_{\text{Alice}}$)
0	1	$\clubsuit_{\text{Bob}} \heartsuit_{\text{Alice}}$	1 (\clubsuit_{Bob})
1	0	$\clubsuit_{\text{Alice}} \heartsuit_{\text{Bob}}$	1 (\clubsuit_{Alice})
1	1	$\clubsuit_{\text{Bob}} \clubsuit_{\text{Alice}}$	1 (\clubsuit_{Bob})

obtained by negating Alice’s input, Bob’s input, and the output. Specifically, when Alice inputs $a = 0$, she should use \heartsuit (otherwise, \clubsuit), and when Bob inputs $b = 0$, he should place \clubsuit to the *right* of the card he received. Finally, the output should be negated. Then, we have Protocol 3, where the different parts from Protocol 2 are underlined.

The relation among the inputs, the pair of cards at the end of step (2), and the output is shown in Table 3. Security proof is not necessary since this protocols is essentially the same as Protocol 2.

3.3 Two-Card XOR Protocol

The proposed 2-card XOR protocol is shown in Protocol 4. In this protocol, PPs are used in steps (1) and (2). The relationships among the inputs, the pair of cards at the end of step (2), and the output are shown in Table 4.

Protocol 4. Two-card XOR Protocol

Inputs: Alice has $a \in \{0, 1\}$, and Bob has $b \in \{0, 1\}$.

Initial Setting: Alice has a pair of $\clubsuit\heartsuit$. Bob has no card.

- (1) Alice prepares $\clubsuit\heartsuit$ if $a = 0$, otherwise $\heartsuit\clubsuit$. She sends the two face-down cards to Bob.
 - (2) Bob puts the cards behind his back and switches their order if $b = 1$ or leaves the same if $b = 0$.
 - (3) Bob opens the cards. If they are $\clubsuit\heartsuit$, $a \oplus b = 0$. Otherwise, i.e., $\heartsuit\clubsuit$, $a \oplus b = 1$.
-

Table 4. Two-Card XOR Protocol

a	b	Step (2)	Output
0	0	♣ ♥	0 (♣♥)
0	1	♣ ♥	1 (♥♣)
1	0	♥ ♣	1 (♥♣)
1	1	♥ ♣	0 (♣♥)

Security of Two-card XOR Protocol: For Alice and Bob, there is no information to be kept secret because, if the value of XOR and one of the two inputs are given, the other input is uniquely determined. Furthermore, no information except for the output is known to the players other than Alice and Bob.

It is clear that no information is obtained by the players other than Alice and Bob (if such players exist) because the only information they can obtain is the output. \square

4 Majority Voting Protocol with Four Cards

Based on the observations on the three-card AND/OR protocols, we propose a new card-based majority voting protocol with three inputs that uses only four cards. Consider the scenario such that Alice, Bob, and Carol have their binary values a , b , and c , respectively, and they want to know the result of majority voting without revealing their individual inputs.

Two types realizations of such a majority voting protocol can be considered. One realization is computing the summation $s := a + b + c$ and then output s , which tells us which is the majority [7]. The other realization is to output 0 if the majority is 0, otherwise output 1 [8]. In this study, we focus on the latter since it is more secure and theoretically interesting. Specifically, we want to compute the following function $\text{maj}(a, b, c) \in \{0, 1\}$ securely:

$$\text{maj}(a, b, c) = \begin{cases} 0, & \text{if } a + b + c \leq 1 \\ 1, & \text{if } a + b + c \geq 2. \end{cases} \quad (2)$$

4.1 Idea Behind Our Majority Voting Protocol with Three-Inputs

Assume that Alice, Bob, and Carol vote a , b , and c , respectively, in this order. We focus on the Carol's vote $c \in \{0, 1\}$.

In the case of $c = 0$, the following relationship holds.

$$a + b + c \geq 2 \iff a + b \geq 2 \iff a \wedge b = 1 \quad (3)$$

This relationship implies that $a \wedge b$ is the result of the majority voting when $c = 0$.

Meanwhile, in the case of $c = 1$, we have the following relationship:

$$a + b + c \geq 2 \iff a + b \geq 1 \iff a \vee b = 1 \tag{4}$$

Hence, $a \vee b$ is the result of the majority voting when $c = 1$.

Summarizing, we have

$$\text{maj}(a, b, c) = \begin{cases} a \wedge b, & \text{if } c = 0 \\ a \vee b, & \text{if } c = 1, \end{cases} \tag{5}$$

which can be calculated securely if we can merge the AND and OR protocols in Protocols 2 and 3, respectively. In fact, such unification is possible by using four cards, which will be explained in the next subsection.

4.2 Unifying AND and OR Operations

Modification of Three-Card OR Protocol. Because the 3-card AND and OR protocols in Protocols 2 and 3, respectively, are essentially the same based on the De Morgan’s law, and hence, they have a symmetric form. From this observation, we design a unified AND/OR protocol where $a \wedge b$ and $a \vee b$ result in the left and right cards, respectively, for inputs $a, b \in \{0, 1\}$.

To obtain the unified protocol, the formats of the outputs of Protocols 2 and 3 must be the same. Then, we exchange ♣ and ♥ in Protocol 3. Moreover, we swap the left and right cards in the step (2) of Protocol 3 in order to make $a \vee b$ place on the right. Then, we obtain Protocol 5 from Protocol 3. The relationships among the inputs, the pair of cards at the end of step (2), and the output are shown in Table 5.

Protocol 5. Modified Three-Card OR Protocol

Inputs: Alice has $a \in \{0, 1\}$, and Bob has $b \in \{0, 1\}$.

Setup: Alice has one ♣ and one ♥, and Bob has one ♥.

- (1) If $a = 0$, Alice selects ♣ behind her back. Otherwise, she selects ♥ behind her back. Then she sends the face-down card she selected to Bob.
 - (2) If $b = 0$, Bob places ♥ to the left of the card he received behind his back. Otherwise, he places ♥ to the right behind his back.
 - (3) Bob opens the card on the right. If this card is ♣, output $a \vee b = 0$. Otherwise, output $a \vee b = 1$. He discards the card on the left.
-

Table 5. Modified Three-Card OR Protocol

a	b	Step (2)	Output
0	0	♥ _{Bob} ♣ _{Alice}	0 (♣ _{Alice})
0	1	♣ _{Alice} ♥ _{Bob}	1 (♥ _{Bob})
1	0	♥ _{Bob} ♥ _{Alice}	1 (♥ _{Alice})
1	1	♥ _{Alice} ♥ _{Bob}	1 (♥ _{Bob})

Protocol 6. Four-card AND/OR protocol

Inputs: Alice has $a \in \{0, 1\}$, and Bob has $b \in \{0, 1\}$.**Setup:** Each of Alice and Bob has $\clubsuit\heartsuit$.

- (1) If $a = 0$, then Alice sends face-down \clubsuit . Otherwise, she sends face-down \heartsuit to Bob.
 - (2) If $b = 0$, Bob places \clubsuit behind his back to the left of the card he received. Otherwise, he places \heartsuit behind his back to the right.
 - (3) The left card represents $a \wedge b$, and the right card represents $a \vee b$. If it is \clubsuit , the output is 0; otherwise output 1.
-

Protocol 7. Majority Voting Protocol with Three Inputs

Inputs: Alice has $a \in \{0, 1\}$, Bob has $b \in \{0, 1\}$, and Carol has $c \in \{0, 1\}$.**Setup:** Alice and Bob have a pair $\clubsuit\heartsuit$. Carol does not have any card.

- (1) Alice chooses \clubsuit behind her back if $a = 0$, otherwise, she chooses \heartsuit behind her back. Then, she sends the face-down card she chose to Bob.
 - (2) Bob places \clubsuit behind his back to the left side of the card from Alice if $b = 0$, otherwise he places \heartsuit behind his back to the right. Then, the left of the cards represents $a \wedge b$, the right represents $a \vee b$.
 - (3) Bob sends face-down the pair of cards to Carol.
 - (4) If $c = 0$, Carol selects the left card. Otherwise, she selects the right.
 - (5) Carol opens the card she selected. If the opened card is \clubsuit , then the result is $\text{maj}(a, b, c) = 0$ (i.e., $a + b + c \leq 1$), otherwise the result is $\text{maj}(a, b, c) = 1$ (i.e., $a + b + c \geq 2$). She discards the card that is not opened.
-

Four-Card AND/OR Protocol. Observe that the right card and the left card are discarded at the end of the protocol in both Protocols 2 and 5, respectively. We also observe that Bob has \clubsuit and \heartsuit at step (1) in both Protocols 2 and 5, respectively. From these observations, we can merge Protocols 2 and 5 by letting Bob have \clubsuit and \heartsuit in the initial setup. Then, we can implement the results of AND and OR simultaneously in a one card-based protocol, as shown in Protocol 6.

We show in the next section that the 4-card AND/OR protocol is useful in calculating the majority voting with only *four* cards.

4.3 Proposed Four-Card Majority Voting Protocol

Based on the 4-card AND/OR protocol, it is easy to compute the majority voting protocol. First, Alice and Bob computes $a \wedge b$ and $a \vee b$ simultaneously, where the result is concealed. Then, Carol chooses $a \wedge b$ or $a \vee b$ depending on $c = 0$ or $c = 1$, respectively, behind her back. The detailed algorithm is shown in Protocol 7. Table 6 shows the pair of cards at the end of step (2) and the output.

Table 6. Majority Voting with Three-Inputs

a	b	c	Step (2)	Output
0	0	0	♣ _{Bob} ♣ _{Alice}	0 (♣ _{Bob})
0	1	0	♣ _{Alice} ♥ _{Bob}	0 (♣ _{Alice})
1	0	0	♣ _{Bob} ♥ _{Alice}	0 (♣ _{Bob})
1	1	0	♥ _{Alice} ♥ _{Bob}	1 (♥ _{Alice})
0	0	1	♣ _{Bob} ♣ _{Alice}	0 (♣ _{Alice})
0	1	1	♣ _{Alice} ♥ _{Bob}	1 (♥ _{Bob})
1	0	1	♣ _{Bob} ♥ _{Alice}	1 (♥ _{Alice})
1	1	1	♥ _{Alice} ♥ _{Bob}	1 (♥ _{Bob})

Protocol 8. Two-Card AND Protocol Using Rotation of Cards

Inputs: Alice has $a \in \{0, 1\}$, and Bob has $b \in \{0, 1\}$.

Setup: Each of Alice and Bob has ♣.

- (1) Alice chooses ♣ if $a = 0$, ♣ otherwise, and send its back to Bob.
 - (2) If $b = 0$, Bob places behind his back ♣ to the left of the card he received, otherwise places ♣ to the right.
 - (3) Bob opens the left card. We have $a \wedge b = 0$ if this card is ♣, otherwise we have $a \wedge b = 1$.
-

Table 7. Comparison with the Previous Work Using Rotation of Cards

Protocols	References	# of PPs	# of Comm.	# of Cards
AND	Shinagawa et al. [11]	3	2	3
	This work: Modification of Protocol 2	2	1	2
OR	Shinagawa et al. [11]	3	2	3
	This work: Modification of Protocol 3	2	1	2
XOR	Mizuki–Shizuya [10]	3	2	2
	This work: Modification of Protocol 4	2	1	1

5 Concluding Remarks

In this paper, we proposed a card-based cryptographic protocol for computing XOR and OR operations based on the model of card-based the protocol presented in [3], i.e., permutation, reverse, private permutation (PP), and communication. In the proposed protocols, we succeeded in realizing card-based cryptographic protocols by representing the participants’ inputs not only by suit of cards but also by the player’s actions.

Our idea is so powerful that it can be applied to the other types of card-base cryptographic protocols. For instance, our idea can reduce the number of

cards for the card-based protocols using ♣ and ♠ to represent numbers. For instance, the 3-card AND protocol in Protocol 2 can be realized with *two* cards, as shown in Protocol 8. Table 7 shows a comparison between previous work and the proposed protocols if our idea is applied to these card-based cryptographic protocols.

Because our AND and OR protocols can be simultaneously realized, we show that the majority voting protocol with three inputs can be realized with *four* cards, which halves the previous majority voting protocol with eight cards. Note that the proposed majority voting protocol can be extended to a majority voting protocol with an arbitrary number of players, which will be presented in the final version of this paper.

Acknowledgement. The authors are grateful to the anonymous reviewers for their helpful comments. They also would like to thank Prof. Takaaki Mizuki for drawing the authors' attention to [2]. This work was partially supported by JSPS KAKENHI Grant Numbers JP15H02710 and JP17H01752.

References

1. Boer, B.: More efficient match-making and satisfiability *The Five Card Trick*. In: Quisquater, J.-J., Vandewalle, J. (eds.) EUROCRYPT 1989. LNCS, vol. 434, pp. 208–217. Springer, Heidelberg (1990). https://doi.org/10.1007/3-540-46885-4_23
2. Marcedone, A., Wen, W., Shi, E.: Secure Dating with Four or Fewer Cards. IACR ePrint Archive, 1031 (2015), <https://eprint.iacr.org/2015/1031>
3. Nakai, T., Tokushige, Y., Misawa, Y., Iwamoto, M., Ohta, K.: Efficient card-based cryptographic protocols for millionaires' problem utilizing private permutations. In: Foresti, S., Persiano, G. (eds.) CANS 2016. LNCS, vol. 10052, pp. 500–517. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-48965-0_30
4. Yao, A.: Protocols for secure computations. In: IEEE Symposium on FOCS, vol. 23, pp. 160–164. IEEE (1982)
5. Mizuki, T., Sone, H.: Six-card secure AND and four-card secure XOR. In: Deng, X., Hopcroft, J.E., Xue, J. (eds.) FAW 2009. LNCS, vol. 5598, pp. 358–369. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-02270-8_36
6. Mizuki, T., Kumamoto, M., Sone, H.: The five-card trick can be done with four cards. In: Wang, X., Sako, K. (eds.) ASIACRYPT 2012. LNCS, vol. 7658, pp. 598–606. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-34961-4_36
7. Mizuki, T., Asiedu, I.K., Sone, H.: Voting with a logarithmic number of cards. In: Mauri, G., Denunzio, A., Manzoni, L., Porreca, A.E. (eds.) UCNC 2013. LNCS, vol. 7956, pp. 162–173. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-39074-6_16
8. Nishida, T., Mizuki, T., Sone, H.: Securely computing the three-input majority function with eight cards. In: Dediu, A.-H., Martín-Vide, C., Truthe, B., Vega-Rodríguez, M.A. (eds.) TPNAC 2013. LNCS, vol. 8273, pp. 193–204. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-45008-2_16
9. Koch, A., Walzer, S., Härtel, K.: Card-based cryptographic protocols using a minimal number of cards. In: Iwata, T., Cheon, J.H. (eds.) ASIACRYPT 2015. LNCS, vol. 9452, pp. 783–807. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-662-48797-6_32

10. Mizuki, T., Shizuya, H.: Practical card-based cryptography. In: Ferro, A., Luccio, F., Widmayer, P. (eds.) FUN 2014. LNCS, vol. 8496, pp. 313–324. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-07890-8_27
11. Shinagawa, K., Nuida, K., Nishide, T., Hanaoka, G., Okamoto, E.: Committed AND protocol using three cards with more handy shuffle. In: ISITA 2016, pp. 736–738 (2016)

Single-Shot Secure Quantum Network Coding for General Multiple Unicast Network with Free Public Communication

Go Kato¹(✉), Masaki Owari², and Masahito Hayashi^{3,4}

¹ NTT Communication Science Laboratories, NTT Corporation, Tokyo, Japan
kato.go@lab.ntt.co.jp

² Department of Computer Science, Faculty of Informatics, Shizuoka University,
Shizuoka, Japan

masakiowari@inf.shizuoka.ac.jp

³ Graduate School of Mathematics, Nagoya University, Nagoya, Japan
masahito@math.nagoya-u.ac.jp

⁴ Centre for Quantum Technologies, National University of Singapore, Singapore,
Singapore

Abstract. Based on a secure classical network code, we propose a general method for constructing a secure quantum network code in the multiple unicast setting under restricted eavesdropper's power. This protocol certainly transmits quantum states when there is no attack. We also show the secrecy with shared randomness as additional resource from the secrecy and the recoverability of the corresponding secure classical network code. Our protocol does not require verification process, which ensures single-shot security.

Keywords: Secrecy · Quantum state · Network coding
Multiple unicast · General network

1 Introduction

Quantum information processing offers information tasks that overwhelm the conventional information technologies. Some of them require transmission of quantum state. For example, quantum communication protocols with exponentially small communication complexity must meet this requirement [1, 2]. When multiple users use quantum information processing, they need to be linked with each other via a quantum network. For efficient use of a quantum network, quantum network coding is needed. To meet the demand, the paper [18] initiated the study of quantum network coding with the butterfly network as a typical example. Under this example, the paper [19] clarified the importance of prior entanglement in a quantum network code. Kobayashi et al. [20] discussed a method for generating GHZ-type states via quantum network coding. Leung et al. [21] investigated several types of networks when classical communication is allowed. Based on these studies, Kobayashi et al. [22] made a code to transmit

quantum states based on a linear classical network code. Then, Kobayashi et al. [23] generalized the result to the case with non-linear network code. However, no existing study has discussed the security for the quantum network code when an adversary attacks the quantum network. Since the improvement of the security is one of the most essential requirements for developing quantum networks, the security analysis is strongly required for quantum network codes. Indeed, it is possible to check the security in these existing methods by verifying the non-existence of the eavesdropper. However, the verification requires us to repeat the same quantum state transmission several times. Hence, it is impossible to guarantee the security under a single transmission in the simple application of these existing methods. Therefore, it is needed to propose a quantum network code that guarantees its security. That is, our aim is a natural extension of classical secure network coding.

On the other hand, for a classical network, Ahlswede et al. [29] started the study of network coding. Then, Cai et al. [3] initiated to address the security of network code, and pointed out that the network coding enhances the security. Currently, many papers [4–16] have already studied the security for network codes. In these studies, the security was shown against wiretapping on a part of the channels. Hence, it is strongly needed to propose a quantum network code whose security is guaranteed under the similar setting. To see our contribution, we explain the characteristics of a quantum network. Studies on classical network coding have most often discussed the unicast setting, in which, we discuss the one-to-one communication via the network. Even in the unicast setting, there are many examples of network codes that overcome the routing, as numerically reported in [28, Sect. 3]. However, in the quantum setting, it is not easy to find such an example in the unicast setting. As another formulation, studies on classical network coding often focus on the multicast setting, in which one sender sends information to multiple receivers. However, in the quantum setting, this is impossible due to the no-cloning theorem. Hence, we discuss the multiple-unicast setting, which has multiple pairs composing of a sender and receiver. However, the multiple-unicast setting has not been well examined even in the classical case, i.e., it has been discussed only in a few papers such as Agarwal et al. [17] with the classical case.

In this paper, we generally construct a quantum network code in the multiple-unicast setting whose security is guaranteed. Our code is canonically constructed from a classical network code in the multiple-unicast setting, and it certainly transmits quantum states when there is no attack. Our main issue is the secrecy of the transmitted quantum states when Eve attacks only edges in the subset E_A of the set of edges of the given network. That is, we show the secrecy of our quantum network code when the secrecy and recoverability of the corresponding classical network are shown against Eve's attack on the subset E_A of edges. That is, we clarify the relation between quantum secrecy and the pair of classical secrecy and classical recoverability in the network coding. We also give several examples for such secure quantum network codes. Indeed, it is not so easy to satisfy this condition for the corresponding classical network. Hence, we allow several nodes

in the network to share common randomness, which is called shared randomness. Since a quantum channel is much more expensive than a classical public channel, we assume that any amount of the classical public channel can be freely used. Under this assumption, the transmission of a quantum state is equivalent to sharing a maximally entangled state via quantum teleportation [27]. Hence, we show the reliability of the transmission by proving that an entangled state can be shared by sending entanglement halves from sink nodes. Our general construction covers the previous code for the butterfly network in [34].

Here, we emphasize the difference between our offered security from the conventional quantum security like quantum key distribution (QKD). In QKD, we repeat the same type of quantum communication several times, which enables us to verify the non-existence of the eavesdropper and to ensure the security. However, our security analysis does not require such repetitive quantum communications, i.e., such a verification process because we assume that the eavesdropper wiretaps only a part of the channels. Since this kind of security analysis holds even under the single-shot setting, we call it single-shot security.

The remaining part of this paper is organized as follows. Section 2 prepares several knowledges for secure classical network coding including secrecy and recoverability. Section 3 provides our general construction of secure quantum network code and states its security theorem. Appendix A shows the security theorem. Appendix B provides precise constructions of the matrices appearing in the main body.

2 Preparation from Secure Classical Network Coding

In this section, we introduce classical network coding and its security analysis which is necessary for analyzing the security of quantum network codes in the next section.

2.1 Classical Linear Multiple-Unicast Network Coding

As a preparation for our general treatment of quantum linear multiple-unicast network codes, we treat a classical linear multiple-unicast network code with shared-randomness, i.e., impose the linearity condition on the operations on the all nodes. In the classical setting of network coding, the network is given as a directed graph (\tilde{V}, \tilde{E}) , where the set of vertices \tilde{V} expresses the set of nodes and the set of edges \tilde{E} expresses the set of communication channels, i.e., the set of packets. When a single character in \mathbb{F}_q is transmitted from a vertex $u \in \tilde{V}$ to another vertex $v \in \tilde{V}$ via a channel, the channel is expressed as $(u, v) \in \tilde{E}$ in the directed graph, where \mathbb{F}_q is the finite field whose order is a power q of the prime p . We denote the number of edges $|\tilde{E}|$ as N . The transmission on the edge is done in the order of the number assigned to the edges.

Since our setting is multiple-unicast, there are (not necessarily distinctive) n pairs of a source node and a terminal node; that is, the single source or terminal node may appear multiple times in the set of pairs. That is, the purpose of

this network is transmitting the messages from the respective source node to the respective terminal node. In this network, n messages are required to be transmitted. To realize the secrecy, some of nodes share common randomness, and a node sharing common randomness is called a shared-randomness node. Here, let n' be the number of shared-randomness. In this setting, a source node may be required to transmit information to plural terminal nodes, and plural source nodes may be required to transmit information to an identical terminal node. Thus, our setting includes the unicast setting as well. We denote the sets of source nodes, terminal nodes, and shared-randomness nodes by V_S , V_T , and V_{SR} .

However, to express the protocol systematically, we need to assign a single quantum system to each message. Hence, we virtually introduce input vertices, output vertices, and shared-randomness vertices, where an input vertex transmits only one message, an output vertex receives only one message, and a shared-randomness vertex generates only one random number. Then, we denote the sets of input vertices, output vertices, and shared-randomness vertices by V_I , V_O , and V_R , respectively. Hence, $|V_I| = |V_O| = n$, and $|V_R| = n'$. We label input vertices as $i_1, \dots, i_n \in V_I$, and the corresponding output vertices as $o_1, \dots, o_n \in V_O$. We similarly label shared-randomness vertices as $r_1, \dots, r_{n'} \in V_R$. Thus, we have

$$\begin{aligned} V_S &= \{v \in V \mid \exists v_i \in V_I \text{ s.t. } (v_i, v) \in E\}, \\ V_{SR} &= \{v \in V \mid \exists v_r \in V_R \text{ s.t. } (v_r, v) \in E\}, \\ V_T &= \{v \in V \mid \exists v_o \in V_O \text{ s.t. } (v, v_o) \in E\}. \end{aligned} \quad (1)$$

The set of all vertices are given as $V := \tilde{V} \cup V_I \cup V_O \cup V_R$, where these sets have no intersection.

An input vertex is connected to a source node via an edge, which are called an input edge. Similarly, an output vertex is connected to a terminal node (a shared-randomness node) via an edge, which is called an output edge (a shared-randomness edge). In contrast, shared-randomness vertex is connected to multiple shared-randomness nodes via edges, which are called shared-randomness edges. Then, we denote the sets of input edges, output edges, and shared-randomness edges by E_I , E_O , and E_R , respectively. Thus, $|E_I| = |E_O| = n$. We denote the number of edges connecting the shared-randomness vertex r_j by l_j . Thus, the number $|E_R|$ of shared-randomness edges is $l := \sum_{j=1}^{n'} l_j$. We also have

$$\begin{aligned} E_I &= \{(u, v) \in E \mid u \in V_I, v \in V\}, \\ E_O &= \{(u, v) \in E \mid u \in V, v \in V_O\}, \\ E_R &= \{(u, v) \in E \mid u \in V_R, v \in V\}. \end{aligned} \quad (2)$$

The set of all edges are given as $E := \tilde{E} \cup E_I \cup E_O \cup E_R$. These sets have no intersection, so, $|E| = N + 2n + l$. These numbers are summarized in Table 1.

To define ordering on edges, we define a map \mathbf{e} from $\{1, \dots, N + 2n + l\}$ to E as follows: $\mathbf{e}(1), \dots, \mathbf{e}(n)$ are input edges, where $\mathbf{e}(j)$ is an input

Table 1. Characteristic numbers of the network coding. Notations undefined here will be defined later.

n	No. of input edges. $ E_I $
	No. of output edges. $ E_O $
	No. of input vertices. $ V_I $
	No. of output vertices. $ V_O $
l	No. of shared-randomness edges. $ E_R $
n'	No. of shared-randomness. $ V_R $
N	No. of other edges. $ \tilde{E} $
h	No. of edges attacked by Eve. $ E_A $
h'	No. of protected edges. $ E_P $

edge going out from an input vertex i_j ; $\mathbf{e}(n+1), \dots, \mathbf{e}(n+l)$ are shared-randomness edges, where $\mathbf{e}\left(n + \sum_{j=1}^{k-1} l_j + 1\right), \dots, \mathbf{e}\left(n + \sum_{j=1}^k l_j\right)$ going out from r_k ; and $\mathbf{e}(n+l+1), \dots, \mathbf{e}(n+l+N)$ are edges in the directed graph (\tilde{V}, \tilde{E}) that originally appears in the classical setting of network coding. Finally, $\mathbf{e}(n+l+N+1), \dots, \mathbf{e}(2n+l+N)$ are output edges, where $\mathbf{e}(n+l+N+j)$ is an output edge going into an output vertex o_j . Here, we assume that our network and ordering satisfy the following connectivity condition: For all $n+l+1 \leq i \leq N+2n+l$, there exists $j < i$ such that $\mathbf{e}(j)$ is connected to $\mathbf{e}(i)$.

For an edge \mathbf{e} , we denote its input and output vertices by $\mathbf{v}_I(\mathbf{e})$ and $\mathbf{v}_O(\mathbf{e})$. Thus, we have $\mathbf{e} = (\mathbf{v}_I(\mathbf{e}), \mathbf{v}_O(\mathbf{e}))$. Therefore, at time $t = i$, the random variable $Y_i \in \mathbb{F}_q$ is inputted from the vertex $\mathbf{v}_I(\mathbf{e}(i))$ to the edge $\mathbf{e}(i)$. Hence, it is transferred to the vertex $\mathbf{v}_O(\mathbf{e}(i))$. The set $\mathbf{I}(i)$ is defined as the set of natural numbers identifying the edges that have transferred their information to the vertex $\mathbf{v}_I(\mathbf{e}(i))$ before the time $t = i$:

$$\mathbf{I}(i) := \{j \in \mathbb{N} | j < i, \exists v \in V, \text{ s.t. } \mathbf{e}(j) = (v, \mathbf{v}_I(\mathbf{e}(i)))\}. \quad (3)$$

Since there exists $j < i$ such that $\mathbf{e}(i)$ is connected to $\mathbf{e}(j)$ via $\mathbf{v}_I(\mathbf{e}(i))$, the set $\mathbf{I}(i)$ is not empty for all $n+l+1 \leq i \leq N+2n+l$. Since we impose the linearity condition on the operations on all the nodes, the random variable Y_i is given as a linear combination of the random variables $\{Y_j\}_{j \in \mathbf{I}(i)}$ as $Y_i = \sum_{j \in \mathbf{I}(i)} \theta_{ij} Y_j$, where $\theta_{ij} \in \mathbb{F}_q$. That is, the set $\{\theta_{ij}\}_{i \in \{1, \dots, |E|\}, j \in \mathbf{I}(i)}$ completely determines the linear multiple-unicast coding on a given network. For convenience, we define $\theta_{ij} = 0$ for $j \notin \mathbf{I}(i)$ so that we have

$$Y_i = \sum_{j < i} \theta_{ij} Y_j. \quad (4)$$

Example 1. Figure 1 depicts an example of a vertex and connecting edges, where the edges $\mathbf{e}(2)$, $\mathbf{e}(5)$, and $\mathbf{e}(7)$ go into the vertex and the edges $\mathbf{e}(4)$ and $\mathbf{e}(8)$

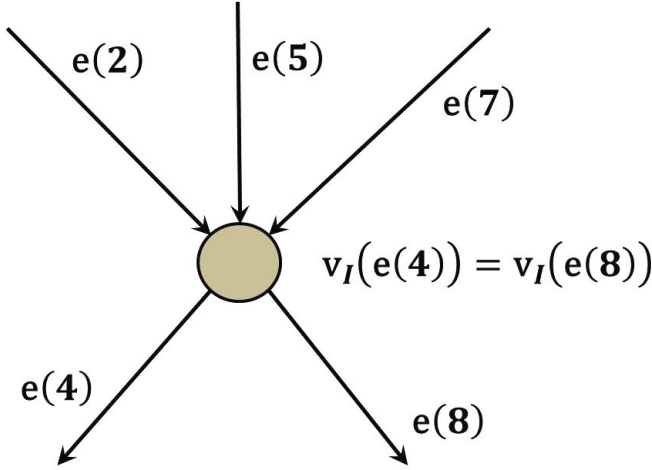


Fig. 1. Example of a vertex and connecting edges

go out from the vertex. Hence, the vertex can be written as $\mathbf{v}_I(\mathbf{e}(4))$ as well as $\mathbf{v}_I(\mathbf{e}(8))$, that is, $\mathbf{v}_I(\mathbf{e}(4)) = \mathbf{v}_I(\mathbf{e}(8))$. At time $t = 4$, the information from $\mathbf{e}(2)$ has arrived, but the information from $\mathbf{e}(5)$ and $\mathbf{e}(7)$ has not yet. Since $\mathbf{e}(1)$ and $\mathbf{e}(3)$, which do not appear in Fig. 1, do not connect to $\mathbf{v}_I(\mathbf{e}(4))$, the operation on $\mathbf{v}_I(\mathbf{e}(4))$ is determined by θ_{42} , and $\{\theta_{4i}\}_{i < 4}$ can be written as

$$\{\theta_{4i}\}_{i < 4} := (0, \theta_{42}, 0).$$

Similarly, at time $t = 8$, information from $\mathbf{e}(2)$, $\mathbf{e}(5)$, and $\mathbf{e}(7)$ has arrived at $\mathbf{v}_I(\mathbf{e}(8))$. Thus, the information transferred through $\mathbf{e}(8)$ can be written as $\sum_{j < 8} \theta_{8j} Y_j$, where Y_j is information transferred through $\mathbf{e}(j)$, and $\{\theta_{8j}\}_{j < 8}$ can be written down as

$$\{\theta_{8j}\}_{j < 8} := (0, \theta_{82}, 0, 0, \theta_{85}, 0, \theta_{87}).$$

Due to the linear structure given in (4), the random variables Y_i is given as a linear combination of the messages $\mathbf{A} := (A_1, \dots, A_n)$ generated in input vertices and the shared random number $\mathbf{B} := (B_1, \dots, B_{n'})$ generated in shared-randomness vertices. For simplicity, combining these random variables, we define the random vector $\mathbf{X}' := (\mathbf{A}, \mathbf{B})$. Hence, there uniquely exists an \mathbb{F}_q -valued $(N + 2n + l) \times (n + n')$ matrix $M_0 = (m_0(i, k))$ such that

$$Y_i = \sum_{k=1}^{n+n'} m_0(i, k) X_k. \quad (5)$$

The concrete construction of M_0 is given in Appendix B.1. Since $\mathbf{e}(i)$ for $N + n + l + 1 \leq i \leq N + 2n + l$ is an output edge, the corresponding coefficients

$m_0(i, k)$ must satisfy

$$\{m_0(n+l+N+i', k)\}_{k=1}^{n+n'} = (\overbrace{0, \dots, 0}^{i'-1}, 1, \overbrace{0, \dots, 0}^{n+n'-i'}) \tag{6}$$

for $1 \leq i' \leq n$. In other words, we define a multiple-unicast network code by the following condition:

Definition 1. A network code $\{\theta_{ij}\}_{i \in \{1, \dots, |E|\}, j \in \mathbf{I}(i)}$ is called a multiple-unicast network code if the coefficients $\{m_0(i, k)\}_{i,k}$ satisfy Eq.(6), where the coefficients $\{m_0(i, k)\}_{i,k}$ are defined by Eq. (5).

2.2 Secrecy of Classical Multiple-Unicast Network Code

In this subsection, we present the results regarding the secrecy of classical network codes, which is necessary to derive the main results regarding quantum network codes. Although there are a lot of existing works on secrecy of classical network coding [3, 5–16], they do not discuss the case when an adversary called “Eve” contaminates a part of the network as well as wiretap a part of the network. Only the paper [33] discusses such an adversary, though its analysis is limited to the unicast case.

To discuss this kind of secrecy, we define $E_A \subset \tilde{E}$ and h as the set of edges attacked by Eve, and the size of E_A , i.e., $h := |E_A|$, respectively. Eve is assumed to be able to eavesdrop and contaminate the information on all the edges in E_A . Eve also knows the network structure, i.e., the topology of network and all the coefficients θ .

To number these edges, we define an increasing function $\varsigma(j) \in \mathbb{N}$ so that E_A can be written as $E_A = \{\mathbf{e}(\varsigma(j))\}_{j=1}^h$. At the j -th attack, Eve attacks the edge $\mathbf{e}(\varsigma(j))$, i.e., wiretaps the random variable $Z_j := Y_{\varsigma(j)}$, and injects the random variable C_j to the vertex $\mathbf{v}_O(\mathbf{e}(\varsigma(j)))$ instead of Z_j . Then, to treat the vector $\mathbf{C} := (C_1, \dots, C_h)$ jointly with \mathbf{A}, \mathbf{B} , we redefine the vector $\mathbf{X} := (\mathbf{A}, \mathbf{B}, \mathbf{C})$.

Due to the linear structure of the network, there uniquely exists an \mathbb{F}_q -valued $(N+2n+l) \times (n+n'+h)$ matrix M satisfying the following. Eve’s attack changes the input information Y_i of the edge $\mathbf{e}(i)$ to

$$Y_i = \sum_{k=1}^{n+n'+h} m(i, k) X_k. \tag{7}$$

The concrete construction of M is given in Appendix B.2.

Then, we denote the output information of the edge $\mathbf{e}(i)$ by Y'_i . Thus,

$$Y'_i := \begin{cases} C_j & \text{when there exists } j \text{ satisfying } i = \varsigma(j) \\ Y_i & \text{otherwise.} \end{cases} \tag{8}$$

Hence, defining the \mathbb{F}_q -valued $(N+2n+l) \times (n+n'+h)$ matrix M' as

$$m'(i, k) := \begin{cases} \delta_{k, n+n'+j} & \text{when there exists } j \text{ satisfying } i = \varsigma(j) \\ m(i, k) & \text{otherwise,} \end{cases} \tag{9}$$

we have

$$Y'_i = \sum_{k=1}^{n+n'+h} m'(i, k) X_k. \quad (10)$$

Here, we categorize her attack into three types: *simple attack*, *deterministic attack* and *probabilistic attack*:

Simple attack: A simple attack is an attack in which Eve just deterministically chooses her input value $\mathbf{C} = \{X_{n+n'+c}\}_{c=1}^h$ as a constant. Hence, her input is independent from her information $\{Z_k\}_{k=1}^j$.

Deterministic attack: A deterministic attack is defined by a set of functions $\{g_j\}_{j=1}^h$:

$$g_j : \mathbb{F}_q^j \rightarrow \mathbb{F}_q.$$

Hence, Eve determines her j -th input value $X_{n+n'+j} = C_j$ as

$$C_j = g_j \left(\{Z_k\}_{k=1}^j \right) = g_j \left(\left\{ \sum_{k=1}^{n+n'+i-1} m(\varsigma(i), k) X_k \right\}_{i=1}^j \right).$$

From the definition of $\{g_j\}_{j=1}^h$, there exists $\prod_{j=1}^h q^{q^j}$ different deterministic attacks. We write the set of all deterministic attacks as \mathcal{G} . Note that a simple attack is also a deterministic attack.

Probabilistic attack: A probabilistic attack is an attack in which Eve probabilistically chooses one of the deterministic attacks $\{g_j\}_{j=1}^h$ and applies it. Hence, a probabilistic attack is determined by a probability distribution $P_{\mathbf{G}} \left(\{g_j\}_{j=1}^h \right)$ on the set of all deterministic attacks \mathcal{G} , where \mathbf{G} is the corresponding random variable. Note that a deterministic attack $\{g_j\}_{j=1}^h$ is a special probabilistic attack whose probability distribution satisfies $P_{\mathbf{G}} \left(\{g_j\}_{j=1}^h \right) = 1$ and $P_{\mathbf{G}} \left(\{g'_j\}_{j=1}^h \right) = 0$ for an arbitrary deterministic attack $\{g'_j\}_{j=1}^h \neq \{g_j\}_{j=1}^h$.

However, any case can be reduced to a simple attack with $\mathbf{C} = 0$ as follows. Eve's information is given as $\{Z_k\}_{k=1}^h$ when C_j is given by the function g_j . Now, we denote Eve's information with $\mathbf{C} = 0$ by $\{\tilde{Z}_k\}_{k=1}^h$. Due to the linearity of the network, we have

$$Z_k = \tilde{Z}_k + \sum_{j=1}^h m(\varsigma(k), n+n'+j) g_j(\{Z_k\}_{k=1}^j). \quad (11)$$

This fact leads to the following lemma.

Lemma 1 ([33, Theorem 1]). *Any deterministic attack with function $\{g_j\}_{j=1}^h$ can be reduced to the simple attack with $\mathbf{C} = 0$. Since any probabilistic attack is given as a probabilistic mixture of deterministic attacks, it can also be reduced to the simple attack with $\mathbf{C} = 0$.*

Now, for given E_A and the function ς , we define a matrix M_ς as a $h \times (n + n' + h)$ matrix whose elements are given by $\{m(\varsigma(j), k)\}_{1 \leq j \leq h, 1 \leq k \leq n+n'+h}$. We further define submatrices of M_ς as $M_\varsigma = (M_{\varsigma,1}, M_{\varsigma,2}, M_{\varsigma,3})$ where the sizes of $M_{\varsigma,1}$, $M_{\varsigma,2}$, and $M_{\varsigma,3}$ are $h \times n$, $h \times m$, and $h \times h$, respectively. For $\mathbf{x} = (\mathbf{a}, \mathbf{b}, \mathbf{c}) \in \mathbb{F}_q^{n+n'+h}$, the condition

$$z_i = \sum_{k=1}^{n+n'+h} m_\varsigma(i, k) x_k \tag{12}$$

can be rewritten as

$$\mathbf{z} = M_{\varsigma,1}\mathbf{a} + M_{\varsigma,2}\mathbf{b} + M_{\varsigma,3}\mathbf{c}. \tag{13}$$

Lemma 2. *Secrecy holds for Eve’s attack on E_A if and only if the following condition holds: For any vector $\mathbf{a} \in \mathbb{F}_q^n$, there exists $\mathbf{b}(\mathbf{a}) \in \mathbb{F}_q^m$ such that*

$$M_{\varsigma,1}\mathbf{a} = M_{\varsigma,2}\mathbf{b}(\mathbf{a}). \tag{14}$$

Proof. Due to Lemma 1, it is enough to discuss the case with $\mathbf{C} = 0$. When secrecy holds, $\{M_{\varsigma,2}\mathbf{b} | \mathbf{b} \in \mathbb{F}_q^m\} = \{M_{\varsigma,1}\mathbf{a} + M_{\varsigma,2}\mathbf{b} | \mathbf{b} \in \mathbb{F}_q^m\}$ for any $\mathbf{a} \in \mathbb{F}_q^n$. The latter set contains $M_{\varsigma,2}\mathbf{b}$, which ensures the existence of $\mathbf{b}(\mathbf{a})$.

When such a vector $\mathbf{b}(\mathbf{a})$ exists, due to the uniformity of \mathbf{B} , the distribution of $M_{\varsigma,2}\mathbf{B}$ is the same as that of $M_{\varsigma,1}\mathbf{a} + M_{\varsigma,2}\mathbf{B}$, which implies the secrecy.

2.3 Recoverability Against Eve’s Attack

For our analysis of quantum network coding, we need to discuss the recoverability of a classical network code against Eve’s attack in addition to the secrecy when the receiver is considered to be one party. That is, we assume that Bob can access all the shared random variables and a subset of edges $E_P \subset E$ dependently of the set E_A and the network structure, i.e., the matrix M' . Under this assumption, we require that Bob can recover the original message correctly. Notice that this kind of recoverability does not imply the recoverability of our multiple-unicast setting.

Notice that we do not assume the condition $E_P \cap E_A = \emptyset$. For the subset E_P , we define the monotone increasing function $\iota : \{1, \dots, |E_P|\} \rightarrow \{1, \dots, N + 2n + l\}$ as it satisfies $E_P = \{e(\iota(i))\}_{i=1}^{h'}$, where $h' := |E_P|$. Then, the information $\{Y_{\iota(i)}\}_{i=1}^{h'}$ on E_P can be written as

$$Y_{\iota(i)} = \sum_{k=1}^{n+n'+h} m'_\iota(i, k) X_k, \tag{15}$$

where $m'_\iota(i, k) := m'(\iota(i), k)$. Then, we define the concept of the recoverability as follows.

Table 2. Summary of matrices

Matrix	Input system	Output system	Equation
M_0	Messages, shared random variables	Inputs of all edges = outputs of all edges	(5)
M	Messages, shared random variables, Eve's input	Inputs of all edges	(7)
M'	Messages, shared random variables, Eve's input	Outputs of all edges	(9)
M_ζ	Messages, shared random variables, Eve's input	Inputs of attacked edges	(12)
M'_l	Messages, shared random variables, Eve's input	Outputs of protected edges	(15)

Definition 2. A subset E_P of edges is called recoverable for M' when for any vector $\mathbf{b} \in \mathbb{F}_q^{n'}$ there exists a function $f_b : \mathbb{F}_q^{h'} \rightarrow \mathbb{F}_q^n$ such that

$$f_b (M'_l \cdot (\mathbf{a}, \mathbf{b}, \mathbf{c})^T) = \mathbf{a} \tag{16}$$

for any $\mathbf{a} \in \mathbb{F}_q^n$ and $\mathbf{c} \in \mathbb{F}_q^h$.

The function f_b is nothing but a decoder of the input \mathbf{a} from the information on E_P . Since condition (16) does not depend on the choice of \mathbf{c} , it guarantees the recoverability even when Eve chooses \mathbf{c} depending on her observed information. Overall, the defined matrices in this section are summarized in Table 2.

3 Secure Quantum Network Coding for General Network

3.1 Coding Scheme

In this section, we treat quantum network coding based on the results for classical network coding in the previous section. Quantum network coding can be categorized by the type of classical communication allowed [18–23]. In this paper, we consider the case when classical communication is freely available. In this case, it is known that for an arbitrary classical multiple-unicast code on an arbitrary classical network, there exists a corresponding quantum multiple-unicast network code on the corresponding quantum network [22, 23]. We start this subsection by extending this known result to the case when shared randomness is employed.

As we explained in the previous section, we start with a graph (\tilde{V}, \tilde{E}) corresponding to the quantum network. That is, the set of vertices \tilde{V} and the set of edges \tilde{E} represent nodes and quantum channels, respectively, where the quantum channels can send one quantum system of dimension q from one node to another. As mentioned above, we assume that classical communication is freely available.

That is, each node can freely send classical information. To describe the input systems and the output systems, we additionally consider the set of input vertices $V_I = \{v_I(i)\}_{i=1}^n$ and the set of output vertices $V_O = \{v_O(i)\}_{i=1}^n$. Since we employ shared randomness, we also use the set of shared-randomness vertices V_R . To connect them to the vertices in \tilde{V} , we consider input, output and shared-randomness edges (E_I, E_O, E_R) defined by Eq. (2). Therefore, in the following, we address the vertices $V := \tilde{V} \cup V_I \cup V_O \cup V_R$ and $E := \tilde{E} \cup E_I \cup E_O \cup E_R$.

Then, the purpose of multiple-unicast quantum network code is to send an arbitrary quantum state on \mathbb{C}^q from a source node having $v_I(i)$ to a terminal node having $v_O(i)$ for all i through the quantum network simultaneously. Since classical communication to terminal nodes is free, this task is equivalent to constructing the maximally entangled state on $\mathbb{C}^q \otimes \mathbb{C}^q$ between a source node having $v_I(i)$ and a terminal node having $v_O(i)$ for all i .

As we explained in the previous section, in classical network coding, an edge has a label indicating the time ordering: $E = \{\mathbf{e}(i)\}_{i=1}^{N+2n+l}$. At time i , the edge $\mathbf{e}(i)$ transfers the information Y_i , which is coded from the information Y_j transferred from all the edges $\mathbf{e}(j)$ with $j \in \mathbf{I}(i)$; here, the encoding is given by Eq.(4) with $\{\theta_{ij}\}_{j \in \mathbf{I}(i)}$. Hence, the classical network code is characterized by the label $\mathbf{e}(j)$ and the encoding $\{\theta_{ij}\}_{i \in \{1, \dots, |E|\}, j \in \mathbf{I}(i)}$.

We write the set of all edges except shared-randomness edges as $E_q := E \setminus E_R$. In the corresponding quantum network, for all edges $e \in E_q$, there exist a q -dimensional Hilbert space having a computational basis $\{|k\rangle\}_{k \in \mathbb{F}_q}$. Since all edges have a label $\mathbf{e}(j)$, we write a Hilbert space corresponding to the edge $\mathbf{e}(j)$ as \mathcal{H}_j . For $1 \leq j \leq n$, \mathcal{H}_j is an input Hilbert space on the source node $\mathbf{v}_O(\mathbf{e}(j))$. Similarly, for $N + n + l + 1 \leq j \leq N + 2n + l$, \mathcal{H}_j is an output Hilbert space on the terminal node $\mathbf{v}_I(\mathbf{e}(j))$. On the other hand, for $n + l + 1 \leq j \leq n + l + N$, \mathcal{H}_j is a Hilbert space that is sent from the node $\mathbf{v}_I(\mathbf{e}(j))$ to the node $\mathbf{v}_O(\mathbf{e}(j))$ through the quantum channel $\mathbf{e}(j)$ at time j . Initially, we set the state to be transmitted in each input edge. Then, for $n + l + 1 \leq j \leq N + 2n + l$, the states of Hilbert spaces \mathcal{H}_j are initially in the state $|0\rangle$. The shared randomness is still classical shared randomness in the quantum network coding; therefore, no Hilbert space corresponds to a shared-randomness edge. On the other hand, when a shared-randomness node in \tilde{V} has an incoming shared-randomness edge $\mathbf{e}(j)$ for $n \leq j \leq n + l$, it receives the common randomness b_k at time j , where $n + \sum_{j'=1}^{k-1} l_{j'} + 1 \leq j \leq n + \sum_{j'=1}^k l_{j'}$. Then, the shared-randomness node operates a controlled unitary depending on b_k later.

Before presenting a quantum network coding protocol, we give the notations used in it. We write the shared randomness as $\mathbf{b} = (b_1, \dots, b_{n'}) \in \mathbb{F}_q^{n'}$. From \mathbf{b} , we further define $\mathbf{b}' = (b'_{n+1}, \dots, b'_{n+l}) \in \mathbb{F}_q^l$ by the relation $b'_j = b_k$ for $n + \sum_{j=1}^{k-1} l_j + 1 \leq j \leq n + \sum_{j=1}^k l_j$. For a subset D of $\{1, \dots, n\} \cup \{n + l + 1, \dots, N + 2n + l\}$, we describe the elements of D as $D = \{k_1, \dots, k_{|D|}\}$. Hence, for a given subset D , we introduce additional notations as $\mathcal{H}_D := \bigotimes_{j \in D} \mathcal{H}_j$. For a \mathbb{F}_q -valued vector $\mathbf{y} = (y_1, \dots, y_n, y_{n+l+1}, \dots, y_{N+2n+l}) \in \mathbb{F}_q^{N+2n}$, we define the vector $\mathbf{y}(D) :=$

$(y_{k_1}, \dots, y_{k_{|D|}}) \in \mathbb{F}_q^{|D|}$ and the state $|\mathbf{y}(D)\rangle_D (\in \mathcal{H}_D) := |y_{k_1}\rangle_{k_1} \otimes \dots \otimes |y_{k_{|D|}}\rangle_{k_{|D|}}$, where $|y_{k_j}\rangle_{k_j}$ is a state on \mathcal{H}_{k_j} . To distinguish a classical system from a quantum one easily, we introduce sets

$$\begin{aligned} \mathbf{QI}(j) &:= \{k \in \mathbf{I}(j) \mid 1 \leq k \leq n, \text{ or } n+l+1 \leq k\} \\ \mathbf{CI}(j) &:= \{k \in \mathbf{I}(j) \mid n+1 \leq k \leq n+l\}, \end{aligned}$$

where $\mathbf{I}(j)$ is defined by Eq.(3). Using these notations, depending on the matrix $\theta = \{\theta_{jk}\}$, we define the controlled unitary $U_{j_1 \dots j_k | i_1 \dots i_m}(\theta)$ acting on the Hilbert space $\mathcal{H}_{j_1 \dots j_k} \otimes \mathcal{H}_{i_1 \dots i_m}$ as

$$\begin{aligned} &U_{j_1 \dots j_k | i_1 \dots i_m}(\theta) \\ &= \sum_{y_{j_1}, \dots, y_{j_k} \in \mathbb{F}_q} \sum_{x_{i_1}, \dots, x_{i_m} \in \mathbb{F}_q} \left| y_{j_1} + \sum_{t=1}^l \theta_{j_1 i_t} x_{i_t}, \dots, y_{j_k} + \sum_{t=1}^l \theta_{j_k i_t} x_{i_t}, x_{i_1}, \dots, x_{i_m} \right\rangle \left\langle y_{j_1}, \dots, y_{j_k}, x_{i_1}, \dots, x_{i_m} \right|. \end{aligned}$$

Now, we define the Fourier basis $\{|\tilde{z}\rangle_j \in \mathcal{H}_j\}_{z \in \mathbb{F}_q}$ of the computational basis $\{|x\rangle_j\}_{x \in \mathbb{F}_q} \subset \mathcal{H}_j$ as

$$|\tilde{z}\rangle_j := \sum_{x \in \mathbb{F}_q} \omega^{\text{tr} xz} |x\rangle_j,$$

where $\omega := \exp\left(-\frac{2\pi i}{p}\right)$. Here, $\text{tr} z$ expresses the element $\text{Tr} M_z \in \mathbb{F}_p$, where M_z denotes the matrix representation of the multiplication map $x \mapsto zx$ with identifying the finite field \mathbb{F}_q with the vector space \mathbb{F}_p^t and t is the degree of algebraic extension of \mathbb{F}_q . For the details, see [32, Sect. 8.1.2]. We also define the generalized Pauli operators $\mathbf{X}(s)$ and $\mathbf{Z}(t)$ as $\mathbf{X}(s) := \sum_{x \in \mathbb{F}_q} |x+s\rangle\langle x|$ and $\mathbf{Z}(t) := \sum_{x \in \mathbb{F}_q} \omega^{\text{tr} xt} |x\rangle\langle x|$. Now, based on the coefficients $\{\theta_{jk}\}$, we present the quantum network code by using shared random variable for a general network as Protocol 1 (See the next page.).

To discuss how well Protocol 1 works, we introduce another set of q -dimensional Hilbert spaces \mathcal{H}'_j for $1 \leq j \leq n$ on the source node $\mathbf{v}_O(\mathbf{e}(j))$. We prepare the Hilbert spaces \mathcal{H}'_j as a reference space and never perform any operations on it. Remember that the transmission of quantum states is mathematically equivalent to sharing the maximally entangled state between the input and output systems. We consider the following virtual protocol. Initially, the maximally entangled state $|\Phi\rangle_j \in \mathcal{H}_j \otimes \mathcal{H}'_j$ is on a source node $\mathbf{v}_O(\mathbf{e}(j))$ for j satisfying $1 \leq j \leq n$. Then, we check whether the final state is the maximally entangled state on $\mathcal{H}'_j \otimes \mathcal{H}_{N+n+l+j}$ between a source node $\mathbf{v}_O(\mathbf{e}(j))$ and a terminal node $\mathbf{v}_I(\mathbf{e}(N+n+l+j))$ for all j satisfying $1 \leq j \leq n$.

Therefore, as a generalization of [22, Theorem 1], we obtain the following theorem.

Theorem 1. *Any state in $\otimes_{j=1}^n \mathcal{H}_j$ can be transmitted to spaces $\otimes_{j=1}^n \mathcal{H}_{N+n+l+j}$ by a protocol given as Protocol 1 if the corresponding classical network coding identify by $\{\theta_{j,k}\}$ is a multiple-unicast network code. That is, when the maximally entangled state $|\Phi\rangle_j \in \mathcal{H}_j \otimes \mathcal{H}'_j$ is prepared as the initial state on a source node $\mathbf{v}_O(\mathbf{e}(j))$ for j satisfying $1 \leq j \leq n$, after finishing the above protocol, the resultant state is a maximally entangled state $|\Phi\rangle_j$ on $\mathcal{H}'_j \otimes \mathcal{H}_{N+n+l+j}$ for all j satisfying $1 \leq j \leq n$.*

Protocol 1. Quantum network coding protocol for general network code

Step 1: Initialization

First, the initial state is prepared on the input edges $\otimes_{j=1}^n \mathcal{H}_j$.

Step 2: Transmission

This step consists of $N+n$ substeps. Starting from $j = n+l+1$, we repeat substeps until $j = N+2n+l$. The j -th substeps start after the $j-1$ -th substep finished, and can be described as follows: The node $\mathbf{v}_I(\mathbf{e}(j))$ has the Hilbert space $\mathcal{H}_{\mathbf{QI}(j)}$ at this time. The node $\mathbf{v}_I(\mathbf{e}(j))$ prepares the Hilbert space \mathcal{H}_j in $|0\rangle_j$. Then, if the node $\mathbf{v}_I(\mathbf{e}(j))$ receives the shared random variable \mathbf{B}' via the edges $(\mathbf{CI}(j)) \neq \emptyset$ at this time, the node $\mathbf{v}_I(\mathbf{e}(j))$ operates the unitary

$$\mathcal{X}_j \left(\sum_{k \in \mathbf{CI}(j)} \theta_{j,n+k} b_k \right) U_{j|\mathbf{QI}(j)}(\theta) \tag{17}$$

on $\mathcal{H}_{\mathbf{QI}(j)} \otimes \mathcal{H}_j$. If the node $\mathbf{v}_I(\mathbf{e}(j))$ has no shared randomness at this time, it operates the controlled unitary $U_{j|\mathbf{QI}(j)}(\theta)$ on $\mathcal{H}_j \otimes \mathcal{H}_{\mathbf{QI}(j)}$. That is, \mathcal{H}_j is the controlled system and $\mathcal{H}_{\mathbf{QI}(j)}$ is the controlling system. The node $\mathbf{v}_I(\mathbf{e}(j))$ sends the Hilbert space \mathcal{H}_j to the node $\mathbf{v}_O(\mathbf{e}(j))$ through the quantum channel $\mathbf{e}(j)$. The node $\mathbf{v}_I(\mathbf{e}(j))$ classically announces that the j -th substep is finished.

Step 3: Measurement on Fourier-basis

For all j satisfying $1 \leq j \leq n$ or $n+l+1 \leq j \leq N+n+l$, the node $\mathbf{v}_O(\mathbf{e}(j))$ measures the Hilbert space \mathcal{H}_j in the Fourier basis, and sends the measurement outcome β_j to all the terminal nodes $\mathbf{v}_I(\mathbf{e}(N+n+l+k))$ satisfying $m_0(j,k) \neq 0$.

Step 4: Recovery

For all k satisfying $1 \leq k \leq n$, the terminal node $\mathbf{v}_I(\mathbf{e}(N+n+l+k))$ operates $Z\left(-\beta_k - \sum_{j=n+l+1}^{N+n+l} \beta_j m_0(j,k)\right)$ on the output Hilbert space $\mathcal{H}_{N+n+l+k}$, where a matrix M_0 is defined by Eq.(5).

Proof. We define the Hilbert spaces $\mathcal{H}_I, \mathcal{H}_G, \mathcal{H}_{G'}$ and \mathcal{H}_O as $\mathcal{H}_I := \otimes_{j=1}^n \mathcal{H}'_j$, $\mathcal{H}_G := \left(\otimes_{j=1}^n \mathcal{H}_j\right) \otimes \left(\otimes_{j=n+l+1}^{N+2n+l} \mathcal{H}_j\right)$, $\mathcal{H}_{G'} := \left(\otimes_{j=1}^n \mathcal{H}_j\right) \otimes \left(\otimes_{j=n+l+1}^{N+n+l} \mathcal{H}_j\right)$, $\mathcal{H}_O := \otimes_{j=N+n+l+1}^{N+2n+l} \mathcal{H}_j$. By straightforward calculation, we find that the state on the network after Step 2 is

$$\begin{aligned}
 & \frac{1}{q^{n+n'}} \sum_{\mathbf{b} \in \mathbb{F}_q^{n'}} \sum_{\mathbf{a}, \mathbf{a}' \in \mathbb{F}_q^n} |\mathbf{a}\rangle_I \langle \mathbf{a}'|_I \otimes |M_G \cdot \begin{pmatrix} \mathbf{a} \\ \mathbf{b} \end{pmatrix}\rangle_G \langle M_G \cdot \begin{pmatrix} \mathbf{a}' \\ \mathbf{b} \end{pmatrix}|_G \\
 &= \frac{1}{q^{n+n'}} \sum_{\mathbf{b} \in \mathbb{F}_q^{n'}} \sum_{\mathbf{a}, \mathbf{a}' \in \mathbb{F}_q^n} |\mathbf{a}\rangle_I \langle \mathbf{a}'|_I \otimes |M'_G \cdot \begin{pmatrix} \mathbf{a} \\ \mathbf{b} \end{pmatrix}\rangle_{G'} \langle M'_G \cdot \begin{pmatrix} \mathbf{a}' \\ \mathbf{b} \end{pmatrix}|_{G'} \otimes |\mathbf{a}\rangle_O \langle \mathbf{a}'|_O.
 \end{aligned}$$

In the above equation, matrices M_G and M'_G are defined as a submatrix of M_0 consisting of all j -th rows satisfying $j \in \{1, \dots, n\} \cup \{n+l+1, \dots, N+2n+l\}$ and a submatrix of M_0 consisting of all j -th rows satisfying $j \in \{1, \dots, n\} \cup \{n+l+1, \dots, N+n+l\}$, respectively, where a matrix M_0 is given by Eq.(5). Suppose that the measurement outcomes in Step 3 form a vector $\boldsymbol{\beta} \in \mathbb{F}_q^{N+n}$. Then, the state after Step 3 is

$$\frac{1}{q^n} \sum_{\mathbf{a}, \mathbf{a}' \in \mathbb{F}_q^n} \omega^{\text{tr} \boldsymbol{\beta}^T \cdot M'_G \cdot (\mathbf{a}' - \mathbf{a}, \mathbf{0})^T} |\mathbf{a}\rangle_I \langle \mathbf{a}'|_I \otimes |\mathbf{a}\rangle_O \langle \mathbf{a}'|_O,$$

where T is transposition. Finally, by applying $Z\left(-\beta_k - \sum_{j=n+l+1}^{N+n+l} \beta_j m_0(j, k)\right)$, where Z_j is the generalized Pauli Z on \mathcal{H}_j , the state after Step 4 can be written as

$$\frac{1}{q^n} \sum_{\mathbf{a}, \mathbf{a}' \in \mathbb{F}_q^n} |\mathbf{a}\rangle_I \langle \mathbf{a}'|_I \otimes |\mathbf{a}\rangle_O \langle \mathbf{a}'|_O,$$

which is the maximally entangled state to be constructed in this protocol.

3.2 Security Analysis

Next, we discuss the secrecy of the quantum state to be transmitted under the following two assumptions. We assume that Eve can eavesdrop and contaminate the information on all the edges in E_A , and also knows the network structure, i.e., the topology of the network and all the coefficients θ . The secrecy for the quantum state is related not only to the secrecy of the classical information but also to the recoverability of the classical information. To consider the relation with the recoverability of the classical information, we employ a set of protected edges E_P including a set of output edges; $E_O \subseteq E_P$. For a protected edge $\mathbf{e}(j) \in E_P \setminus E_O$, we also assume that the node $\mathbf{v}_O(\mathbf{e}(j))$ shares private classical randomness with the terminal nodes $\mathbf{v}_I(\mathbf{e}(N+n+l+k))$ when $m_0(j, k) \neq 0$ and $\mathbf{v}_O(\mathbf{e}(j)) \neq \mathbf{v}_O(\mathbf{e}(N+n+l+k))$. To express the second assumption clearly, we divide Step 3 into the following two steps:

Therefore, our protocol depends on the set of protected edges E_P . That is, our protocol is uniquely determined by the pair comparing of $\{\theta_{i,j}\}$ and E_P , and it is called the quantum network code $\{\theta_{ij}\}_{i \in \{1, \dots, |E|\}, j \in \mathbf{I}(i)}$ with the set of protected edges E_P .

Hence, in Step 3-1 of the protocol, Eve derives all measurement outcomes β_j as long as $\mathbf{e}(j)$ is not in E_P . We further define the notations: $E_A \subset \tilde{E}$ is the set of edges which Eve attacks, and h is the size of E_A , i.e. $h := |E_A|$. We define

Step 3-1: Measurement on non-protected edges

For all j satisfying $n + l + 1 \leq j \leq N + n + l$ and $\mathbf{e}(j) \notin E_P$, the node $\mathbf{v}_O(\mathbf{e}(j))$ measures the Hilbert space \mathcal{H}_j in the Fourier basis, and sends the measurement outcome β_j to the all terminal nodes $\mathbf{v}_I(\mathbf{e}(N + n + l + k))$ satisfying $m_0(j, k) \neq 0$ publicly.

Step 3-2: Measurement on protected edges

For all j satisfying $n + l + 1 \leq j \leq N + n + l$ and $\mathbf{e}(j) \in E_P$, the node $\mathbf{v}_O(\mathbf{e}(j))$ makes the same measurement and securely sends the measurement outcome β_j to the all terminal nodes $\mathbf{v}_I(\mathbf{e}(N + n + l + k))$ satisfying $m_0(j, k) \neq 0$ by use of the shared-randomness.

functions $\varsigma(j) \in \mathbb{N}$ and $\iota(j) \in \mathbb{N}$ so that E_A and E_P can be written as $E_A = \{\mathbf{e}(\varsigma(j))\}_{j=1}^h$ and $E_P = \{\mathbf{e}(\iota(j))\}_{j=1}^{h'}$ with $\varsigma(j) < \varsigma(j + 1)$ and $\iota(j) < \iota(j + 1)$; that is, Eve attacks the quantum system $\mathbf{e}(\varsigma(j))$ on the j -th attack. Note that we assume the condition that $E_O \subseteq E_P$, since this condition is natural from the viewpoint of the definition of protected edges. However, it is mathematically unnecessary, and we do not use it in the following part at all.

Now, in Step 2 of the protocol, the node $\mathbf{v}_I(\mathbf{e}(\varsigma(j) + 1))$ never sends the Hilbert space $\mathcal{H}_{\varsigma(j)+1}$ before the node $\mathbf{v}_O(\mathbf{e}(\varsigma(j)))$ receives the Hilbert space $\mathcal{H}_{\varsigma(j)}$. Hence, after Eve steals the Hilbert space $\mathcal{H}_{\varsigma(j)}$, she must return it to the edge $\mathbf{e}(\varsigma(j))$ before she steals the Hilbert space $\mathcal{H}_{\varsigma(j)+1}$. Further, since Step 3-1 never starts before Step 2 completes, Eve’s operations in Step 2 cannot depend on the measurement outcomes in Step 3-1. Hence, Eve’s attack in the Step 2 of the protocol can be described as follows:

Eve’s attack: Eve is assumed to attack only edges in E_A and not to attack any nodes. Eve first has her initial Hilbert space \mathcal{W} with state $|\phi_{ini}\rangle$, which is chosen to be sufficiently large so that Eve’s operations can be written as unitaries. For $j = 1$ to $j = h$, Eve repeats the following behavior: Eve applies the unitary V_j on $\mathcal{H}_{\varsigma(j)} \otimes \mathcal{W}$. Since Eve can hear the measurement outcomes on non-protected edges, the system of the the classical information is denoted by \mathcal{V} .

Hence, the security of the quantum network coding can be defined as follows:

Definition 3. *The quantum network code $\{\theta_{ij}\}_{i \in \{1, \dots, |E|\}, j \in \mathbf{I}(i)}$ with the set of protected edges E_P is called secure for Eve’s attack $\{V_j\}_{j=1}^h$ on the set of edges E_A if the following condition holds. When the initial state on the Hilbert space $\mathcal{H}_I \otimes \mathcal{W} \otimes \mathcal{V}$ is the same state as Theorem 1, the final state of the protocol is a product state with respect to the partition between \mathcal{H}_I and $\mathcal{W} \otimes \mathcal{V}$.*

In the above definition, $\rho \in \mathfrak{B}(\mathcal{H}_1 \otimes \mathcal{H}_2)$ is called a product state if there exist $\rho_1 \in \mathfrak{B}(\mathcal{H}_1)$ and $\rho_2 \in \mathfrak{B}(\mathcal{H}_2)$ such that $\rho = \rho_1 \otimes \rho_2$. Note that, if the quantum network coding is secure, even for any entangled initial state in $\otimes_{j=1}^n \mathcal{H}_j$ which has no correlation with Eve initially, there is no correlation between the entangled initial state and Eve’s final state after the protocol. Now, we can present the main result of this paper:

Theorem 2. *The quantum network code $\{\theta_{ij}\}_{i \in \{1, \dots, |E|\}, j \in \mathbf{I}(i)}$ with the set of protected edges E_P is secure for all Eve's attacks on the set of edges E_A if the following two conditions hold. (i) The classical network code $\{\theta_{ij}\}_{i \in \{1, \dots, |E|\}, j \in \mathbf{I}(i)}$ is secret for Eve's attacks on the set of edges E_A . (ii) The set of protected edges E_P is recoverable for Eve's attacks on E_A in the sense of the classical network coding.*

Notice that Theorem 2 does not assume the condition $E_P \cap E_A = \emptyset$. This theorem guarantees that the secrecy analysis of our quantum network coding is reduced to the analysis of the secrecy and the recoverability of the corresponding classical network coding.

4 Conclusion

Based on a secure classical network code, we have proposed a canonical way to make a secure quantum network code in the multiple-unicast setting. This protocol certainly transmits quantum states when there is no attack. We have also shown the secrecy of the quantum network code under the secrecy and the recoverability of the corresponding classical network code.

Acknowledgments. The authors are very grateful to Professor Ning Cai and Professor Vincent Y. F. Tan for helpful discussions and comments. The works reported here were supported in part by the JSPS Grant-in-Aid for Scientific Research (A) No. 23246071, (C) No. 16K00014, (B) No. 16KT0017, (C) No. 17K05591, the Okawa Research Grant, and Kayamori Foundation of Informational Science Advancement.

Appendix

A Security Proof Based on Computation Basis Security

To show the security theorem, we prepare an important result for the recovery of the maximally entangled state from evaluation of classical information. First, we consider a sufficient condition to approximately and locally generate the maximally entangled state $|\Phi\rangle := \sum_{x=1}^d \frac{1}{\sqrt{d}} |x\rangle_A \otimes |x\rangle_{A'} \in \mathcal{H}_A \otimes \mathcal{H}_{A'}$ on the composite system $\mathcal{H}_A \otimes \mathcal{H}_{A'}$, where $\{|x\rangle_A\}$ and $\{|x\rangle_{A'}\}$ are the CONSs of \mathcal{H}_A and $\mathcal{H}_{A'}$, respectively. For this purpose, we focus on the following two conditions for a pure state ρ on the composite system $\mathcal{H}_A \otimes \mathcal{H}_B \otimes \mathcal{H}_R$.

- ϵ_1 -**classical secrecy**: Let id_R be the identity operation, $\rho_{\text{mix},A}$ be the completely mixed state, κ_A be the pinching with respect to the computation basis of \mathcal{H}_A , i.e., $\kappa_A(\sigma) := \sum_{x=1}^d |x\rangle_A \langle x| \sigma |x\rangle_A \langle x|$. The relation $F(\kappa_A \otimes \text{id}_R(\rho_{AR}), \rho_{\text{mix},A} \otimes \rho_R) \geq 1 - \epsilon_1$ holds.
- ϵ_2 -**error classical recoverability**: There exists a POVM $\mathbf{M} = \{M_x\}_{x=1}^d$ on \mathcal{H}_B such that $\sum_{x=1}^d \text{Tr} \rho_{AB} |x\rangle_A \langle x| \otimes M_x \geq 1 - \epsilon_2$.

The following proposition is known.

Proposition 1 (Renes[30]¹). *Assume that a state $\rho = |\Psi\rangle\langle\Psi|$ on the composite system $\mathcal{H}_A \otimes \mathcal{H}_B \otimes \mathcal{H}_R$ satisfies both of the above conditions. Then, there is a TP-CP map $\kappa : \mathcal{S}(\mathcal{H}_B) \rightarrow \mathcal{S}(\mathcal{H}_A)$ such that*

$$F(\text{id}_A \otimes \kappa(\rho_{AB}), |\Phi\rangle\langle\Phi|) \geq 1 - (\sqrt{\epsilon_2} + \sqrt{\epsilon_1})^2. \quad (18)$$

Proposition 1 guarantees that we can generate the maximally entangled state between systems \mathcal{H}_A and \mathcal{H}_B only by an operation on the system \mathcal{H}_B if the bit information on the system \mathcal{H}_A is a uniform random number almost independent of the environment system \mathcal{H}_E and can be recovered in the system \mathcal{H}_B . In our situation, these two conditions can be checked by the secrecy and the recoverability.

Proof of Theorem 2: Part 1: To show the theorem, we prove that an entangled state can be shared by sending entanglement halves from sink nodes by applying Proposition 1 to the state after Step 3-1. We prepare notations, while we employ the same notation as in the proof of Theorem 1. We introduce the quantum systems $\mathcal{H}_{n+1}, \dots, \mathcal{H}_{n+l}$ to describe the shared-randomness edges $\mathbf{e}(n+1), \dots, \mathbf{e}(n+l)$. Given a shared-randomness vertex r_k , by using the notation $|b\rangle_{r_k} := |b, \dots, b\rangle_{n+\sum_{j=1}^{k-1} l_j+1, \dots, n+\sum_{j=1}^k l_j}$, the initial state on the composite system $\mathcal{H}_{n+\sum_{j=1}^{k-1} l_j+1} \otimes \dots \otimes \mathcal{H}_{n+\sum_{j=1}^k l_j}$ connected to the shared-randomness vertex r_k . can be regarded as the super position state $|\Phi\rangle_{r_k} := \frac{1}{\sqrt{q}} \sum_{x \in \mathbb{F}_q} |b\rangle_{r_k}$. Hence, we denote the system span by $|\mathbf{b}\rangle_{SR} := |b_1\rangle_{r_1} \dots |b_{n'}\rangle_{r_{n'}}$ by \mathcal{H}_{SR} .

In this protocol, it is important to consider the path, in which the sequence of the messages is $\mathbf{a} \in \mathbb{F}_q^n$, the sequence of the shared random numbers is $\mathbf{b} \in \mathbb{F}_q^{n'}$, the sequence of Eve's injections is $\mathbf{c} \in \mathbb{F}_q^h$, the sequence of inputs of attacked edges is $\mathbf{z} \in \mathbb{F}_q^h$, and the sequence of outputs of all edges $\tilde{E} \cup E_O$ is $\mathbf{y} = (y_{n+l+1}, \dots, y_{2n+l+N})$. Here, when $e(j)$ is attacked, y_j expresses the information after the attack.

Depending on this path, the matrix component on Eve's memory \mathcal{W} is determined. For $(\mathbf{a}, \mathbf{b}, \mathbf{c}) \in \mathbb{F}_q^{n+n'+h}$ and $\mathbf{y} \in \mathbb{F}_q^{N+n}$, the matrix component $V(\mathbf{a}, \mathbf{b}, \mathbf{c}, \mathbf{y})$ is given as

$$V(\mathbf{a}, \mathbf{b}, \mathbf{c}, \mathbf{y}) := \left(\prod_{j=n+l+1}^{2n+l+N} \delta(y_j, (M'(\mathbf{a}, \mathbf{b}, \mathbf{c})^T)_j) \right) \left(\prod_{i=1}^h \langle c_i | V_i | (M_\zeta(\mathbf{a}, \mathbf{b}, \mathbf{c})^T)_i \rangle \right).$$

Since the information $\mathbf{c} \in \mathbb{F}_q^h$ does not appear in the final state, we define the vector;

$$|\Phi[\mathbf{a}, \mathbf{b}, \mathbf{y}]\rangle := \sum_{\mathbf{c}} V(\mathbf{a}, \mathbf{b}, \mathbf{c}, \mathbf{y}) |\phi_{ini}\rangle_{\mathcal{W}} |\mathbf{a}\rangle_{\mathcal{I}} |\mathbf{b}\rangle_{SR} |\mathbf{a}, \mathbf{y}\rangle_{\mathcal{G}}, \quad (19)$$

Therefore, the state after Step 2 on the whole system is given as

$$q^{-(n+n')/2} \sum_{\mathbf{a}, \mathbf{b}, \mathbf{y}} |\Phi[\mathbf{a}, \mathbf{b}, \mathbf{y}]\rangle_{\mathcal{I}, SR, \mathcal{G}, \mathcal{W}}. \quad (20)$$

¹ This theorem is also reviewed in [31, Section 8.15.1].

Now, for a sequence $\mathbf{y} \in \mathbb{F}_q^{N+n}$, we introduce the sequence $\mathbf{y}^c := (y_j)_{j \in E_O \cup \tilde{E} \setminus E_P} \in \mathbb{F}_q^{N+n-h'}$, which expresses the information on the non-protected edges $E_O \cup \tilde{E} \setminus E_P$. When we observe the measurement outcome $\beta = (\beta_j)_{j \in E \setminus E_P} \in \mathbb{F}_q^{N+n-h'}$ in Step 3-1, the resultant state is

$$|\Psi_\beta\rangle := \sum_{\mathbf{a}, \mathbf{b}, \mathbf{y}} \omega^{-\text{tr} \beta \cdot (\mathbf{a}, \mathbf{y}^c)} q^{(N+n-h'-n-n')/2} {}_{NP} \langle \mathbf{a}, \mathbf{y}^c | \Phi[\mathbf{a}, \mathbf{b}, \mathbf{y}] \rangle_{I, SR, G, W}, \quad (21)$$

where $\mathcal{H}_{NP} := (\otimes_{j=1}^n \mathcal{H}_j) \otimes (\otimes_{j \in E_O \cup \tilde{E} \setminus E_P} \mathcal{H}_j)$.

Now we set $\mathcal{H}_A := \mathcal{H}_I$, $\mathcal{H}_B := \mathcal{H}_P \otimes \mathcal{H}_{SR} \otimes \mathcal{V}'$, and $\mathcal{H}_R := \mathcal{W} \otimes \mathcal{V}$, where $\mathcal{H}_P := \otimes_{j: \mathbf{e}(\iota(j)) \in E_P} \mathcal{H}_{\iota(j)}$, and \mathcal{V}' expresses the Hilbert space of the measurement outcome possessed by the system \mathcal{H}_B . Then, the final state is pure on the composite system $\mathcal{H}_A \otimes \mathcal{H}_B \otimes \mathcal{H}_R$. Then, due to Proposition 1, it is enough to show the 0-bit secrecy and the 0-bit recoverability separately for the state $|\Psi_\beta\rangle$ with any measurement outcome β .

Part 2: Next, we discuss the 0-classical secrecy. Since \mathcal{H}_B does not belong to Eve's system, it is sufficient to prove the secrecy when we apply the Fourier basis measurement for shared-randomness edges and protected edges and send the measurement outcome α to Eve. That is, it is sufficient to show that the unnormalized state $\sum_{\mathbf{b}, \mathbf{y}_\iota} {}_{SR} \langle \mathbf{b} | {}_P \langle \mathbf{y}_\iota | \omega^{-\text{tr} \alpha \cdot (\mathbf{b}, \mathbf{y}_\iota)} q^{n/2} {}_I \langle \mathbf{a} | \Psi_\beta \rangle$ does not depend on \mathbf{a} for each $(\alpha, \beta) \in \mathbb{F}_q^{N+2n+n'}$. Based on Lemma 2, we choose $\mathbf{b}(\mathbf{a}) \in \mathbb{F}_q^{n'}$ for $\mathbf{a} \in \mathbb{F}_q^n$. Since the vector $\mathbf{y}(\mathbf{a}) := M'(-\mathbf{a}, \mathbf{b}(\mathbf{a}), 0)^T$ satisfies $M'(\mathbf{a}, \mathbf{b}, \mathbf{c})^T + \mathbf{y}(\mathbf{a}) = M'(0, \mathbf{b} + \mathbf{b}(\mathbf{a}), \mathbf{c})^T$ and $M_\zeta(\mathbf{a}, \mathbf{b}, \mathbf{c}) = M_\zeta(0, \mathbf{b} + \mathbf{b}(\mathbf{a}), \mathbf{c})$, we have $V(\mathbf{a}, \mathbf{b}, \mathbf{c}, \mathbf{y}) = V(0, \mathbf{b} + \mathbf{b}(\mathbf{a}), \mathbf{c}, \mathbf{y} + \mathbf{y}(\mathbf{a}))$. Hence, we have

$$\begin{aligned} & q^{-(N-h'-n')/2} \sum_{\mathbf{b}, \mathbf{y}_\iota} {}_{SR} \langle \mathbf{b} | {}_P \langle \mathbf{y}_\iota | \omega^{-\text{tr} \alpha \cdot (\mathbf{b}, \mathbf{y}_\iota)} q^{n/2} {}_I \langle \mathbf{a} | \Psi_\beta \rangle \\ &= \sum_{\mathbf{b}, \mathbf{y}} \omega^{-\text{tr}(\alpha, \beta) \cdot (\mathbf{a}, \mathbf{b}, \mathbf{y})} {}_{SR} \langle \mathbf{b} | {}_I \langle \mathbf{a} | {}_G \langle \mathbf{a}, \mathbf{y} | \Phi[\mathbf{a}, \mathbf{b}, \mathbf{y}] \rangle_{I, SR, G, W} \\ &= \sum_{\mathbf{b}, \mathbf{y}} \omega^{-\text{tr}(\alpha, \beta) \cdot (\mathbf{a}, \mathbf{b}, \mathbf{y})} \sum_{\mathbf{c}} V(\mathbf{a}, \mathbf{b}, \mathbf{c}, \mathbf{y}) |\phi_{ini}\rangle_W \\ &= \omega^{\text{tr}(\alpha, \beta) \cdot (\mathbf{a}, \mathbf{b}(\mathbf{a}), \mathbf{y}(\mathbf{a}))} \sum_{\mathbf{b}, \mathbf{y}} \omega^{-\text{tr}(\alpha, \beta) \cdot (0, \mathbf{b} + \mathbf{b}(\mathbf{a}), \mathbf{y} + \mathbf{y}(\mathbf{a}))} \\ & \quad \sum_{\mathbf{c}} V(0, \mathbf{b} + \mathbf{b}(\mathbf{a}), \mathbf{c}, \mathbf{y} + \mathbf{y}(\mathbf{a})) |\phi_{ini}\rangle_W \\ &= \omega^{\text{tr}(\alpha, \beta) \cdot (\mathbf{a}, \mathbf{b}(\mathbf{a}), \mathbf{y}(\mathbf{a}))} \sum_{\mathbf{b}', \mathbf{y}'} \omega^{-\text{tr}(\alpha, \beta) \cdot (0, \mathbf{b}', \mathbf{y}')} \sum_{\mathbf{c}} V(0, \mathbf{b}', \mathbf{c}, \mathbf{y}') |\phi_{ini}\rangle_W, \end{aligned}$$

where $\mathbf{b}' := \mathbf{b} + \mathbf{b}(\mathbf{a})$ and $\mathbf{y}' := \mathbf{y} + \mathbf{y}(\mathbf{a})$. Since $\omega^{\text{tr}(\alpha, \beta) \cdot (\mathbf{a}, \mathbf{b}(\mathbf{a}), \mathbf{y}(\mathbf{a}))}$ is the global phase factor, Eve's information on $\mathcal{W} \otimes \mathcal{V}$ is independent of \mathbf{a} . Thus, we obtain the 0-classical security.

Part 3: To show the 0-error classical recoverability, we give a POVM $\{M_{\mathbf{a}'}\}_{\mathbf{a}' \in \mathbb{F}_q^n}$ on \mathcal{H}_P to recover Alice's message \mathbf{a} , which does not use the outcome β of the Fourier basis measurement on $E \setminus E_P$. The condition is given as

$$\text{Tr}(M_{\mathbf{a}'} \otimes |\mathbf{a}\rangle_I \langle \mathbf{a}| \otimes I_R) |\Psi_\beta\rangle \langle \Psi_\beta| = \frac{\delta(\mathbf{a}, \mathbf{a}')}{q^n}. \quad (22)$$

Now, using the function $f_{\mathbf{b}}$ given in Definition 2, we define the POVM $M_{\mathbf{a}'} := \sum_{\mathbf{b} \in \mathbb{F}_q^{n'}, \mathbf{y}_\ell \in \mathbb{F}_q^{h'}: f_{\mathbf{b}}(\mathbf{y}_\ell) = \mathbf{a}'} |\mathbf{b}\rangle_{SR} \langle \mathbf{b}| \otimes |\mathbf{y}_\ell\rangle_P \langle \mathbf{y}_\ell|$. When we make the measurement $\{|\mathbf{b}\rangle_{SR} \langle \mathbf{b}| \otimes |\mathbf{y}_\ell\rangle_P \langle \mathbf{y}_\ell| \otimes |\mathbf{a}\rangle_I \langle \mathbf{a}| \otimes I_R\}_{\mathbf{b}, \mathbf{y}_\ell, \mathbf{a}}$, for observed outcomes $\mathbf{y}_\ell, \mathbf{a}, \mathbf{b}$, there exists a sequence \mathbf{c} such that $\mathbf{y}_\ell = M'_\ell(\mathbf{a}, \mathbf{b}, \mathbf{c})$. Since the relation (16) guarantees the relation $f_{\mathbf{b}}(\mathbf{y}_\ell) = \mathbf{a}$, we obtain the desired condition (22). \blacksquare

In summary, the above proof shows Theorem 2 via the 0-classical secrecy and the 0-error classical recoverability.

Remark 1. Here, we remark on the relation between our security proof and our protocol. Using Proposition 1, the security proof gives a protocol to transmit a quantum state to \mathcal{H}_B . Hence, one might consider that this protocol can be used for our purpose. However, this protocol cannot be used for three reasons. (i) Whereas our real setting is multiple-unicast, the protocol in the security proof assumes one receiver. (ii) The protocol in the security proof requires a measuring operation on the shared-randomness as a coherent superposition state across the sharing edges. In the real situation, each receiver possesses only a part of edges. (iii) To realize the protocol given in the security proof, we need to identify the edges attacked by Eve. However, the legitimate users know only the range of Eve’s possible attack. Hence, they cannot perform the decoding protocol. Due to three problems, we cannot apply the protocol given in the security proof in our multiple unicast setting.

Remark 2. One might consider that it is sufficient to apply Proposition 1 to the case when $\mathcal{H}_A := \mathcal{H}_I$, $\mathcal{H}_B := \mathcal{H}_P \otimes \mathcal{H}_{SR}$, and $\mathcal{H}_R := \mathcal{W}$ for the respective measurement outcome β . However, this application only shows that the whole density on $\mathcal{H}_I \otimes \mathcal{W} \otimes \mathcal{V}$ is written as

$$\sum_{\beta} p_{\beta} \rho_{I, \beta} \otimes \rho_{E, \beta} \otimes |\beta\rangle \langle \beta|. \quad (23)$$

Hence, we need to apply Proposition 1 to the case when $\mathcal{H}_A := \mathcal{H}_I$, $\mathcal{H}_B := \mathcal{H}_P \otimes \mathcal{H}_{SR} \otimes \mathcal{V}'$, and $\mathcal{H}_R := \mathcal{W} \otimes \mathcal{V}$.

Indeed, one might consider that the combination of form (23) and Part 3 of our proof shows the desired statement because Part 3 of our proof shows the independence of Eve’s state from \mathcal{H}_I when \mathcal{H}_I is measured in a computational basis. This fact only shows that the state $\langle \mathbf{a} | \rho_{I, \beta} | \mathbf{a} \rangle \rho_{E, \beta}$ is independent of \mathbf{a} . That is, the form (23) and Part 3 of our proof do not deny the possibility of the correlation between the resultant state on Eve’s system and \mathcal{H}_I when the state on \mathcal{H}_I is measured in another basis.

B Constructions of Matrices Describing Network

In this appendix, we concretely construct the matrices describing the network structure.

B.1 Construction of M_0

The definition of input edges and shared-randomness edges determine the coefficients $\{m_0(i, k)\}_{i, k}$ for $1 \leq i \leq n + l$ as follows: For $1 \leq i \leq n$, $\mathbf{e}(i)$ is an input edge, that is, $\mathbf{e}(i) \in E_I$. Thus, the definition of input edges determines $\{m_0(i, k)\}_{k=1}^{n+n'}$ as

$$\{m_0(i, k)\}_{k=1}^{n+n'} = (\overbrace{0, \dots, 0}^{i-1}, 1, \overbrace{0, \dots, 0}^{n'+n-i}) \quad \text{for } 1 \leq i \leq n. \quad (24)$$

For $n + 1 \leq i \leq n + l$, $\mathbf{e}(i)$ is a shared-randomness edge, that is, $\mathbf{e}(i) \in E_R$. Hence, there uniquely exists an integer $i' \in [1, m]$ such that $n + \sum_{j=1}^{i'-1} l_j + 1 \leq i \leq n + \sum_{j=1}^{i'} l_j$. Thus, the definition of shared-randomness edges determines $\{m_0(i, k)\}_{k=1}^{n+n'}$ as

$$\{m_0(i, k)\}_{k=1}^{n+n'} = (\overbrace{0, \dots, 0}^{n+i'-1}, 1, \overbrace{0, \dots, 0}^{m-i'}). \quad (25)$$

Using Eqs. (4) and (5), we derive the recurrence relation of $m_0(i, k)$ as

$$m_0(i, k) = \sum_{j=1}^{i-1} \theta_{ij} m_0(j, k). \quad (26)$$

Thus, the coefficients $\{\theta_{ij}\}_{i \in \{1, \dots, |E|\}, j < i}$ completely determine all the coefficients $\{m_0(i, k)\}_{i, k}$ through Eqs. (24), (25), and (26).

B.2 Construction of M

Since

$$Y_i = \sum_{j \in \mathbf{I}(i)} \theta_{ij} Y'_j,$$

Eqs. (10) and (7) lead

$$m(i, k) = \sum_{j \in \mathbf{I}(i)} \theta_{ij} m'(j, k). \quad (27)$$

Thus, from Eqs. (27) and (9), we derive the following recurrence relations for $m(i, k)$:

$$m(i, k) = \sum_{j \in \mathbf{I}(i) \setminus E_A} \theta_{ij} m(j, k) + \sum_{i'=1}^h \theta_{i\varsigma(i')} \delta_{k, n+n'+i'}, \quad (28)$$

where we define $\theta_{i\varsigma(i')} = 0$ for i' such that $\varsigma(i') \notin \mathbf{I}(i)$.

References

1. Buhrman, H.R., Cleve, R., Wigderson, A.: Quantum vs. classical communication and computation. In: Proceedings of the 30th Annual ACM Symposium on Theory of Computing, pp. 63–68. ACM, New York (1999)
2. Raz, R.: Exponential separation of quantum and classical communication complexity. In: Proceedings of the Thirty-First Annual ACM Symposium on Theory of Computing, STOC 1999, pp. 358–367. ACM, New York (1999)
3. Cai, N., Yeung, R.: Secure network coding. In: Proceedings of 2002 IEEE International Symposium on Information Theory (ISIT), p. 323 (2002)
4. Cai, N., Yeung, R.W.: Network error correction, Part 2: lower bounds. *Commun. Inf. and Syst.* **6**(1), 37–54 (2006)
5. Bhattad, K., Member, S., Narayanan, K.R.: Weakly secure network coding. In: First Workshop on Network Coding, Theory, and Applications, Riva del Garda (2005)
6. Liu, R.L.R., Liang, Y.L.Y., Poor, H., Spasojevic, P.: Secure nested codes for type II wiretap channels. In: 2007 IEEE Information Theory Workshop, pp. 337–342 (2007)
7. Rouayheb, S.Y.E., Soljanin, E.: On wiretap networks II. In: Proceedings of 2007 IEEE International Symposium on Information Theory (ISIT), pp. 551–555 (2007)
8. Harada, K., Yamamoto, H.: Strongly secure linear network coding. *IEICE Trans. Fundam. Electron. Commun. Comput. Sci.* **E91-A**(10), 2720–2728 (2008)
9. Ho, T.H.T., Leong, B.L.B., Koetter, R., Medard, M., Effros, M., Karger, D.: Byzantine modification detection in multicast networks with random network coding. *IEEE Trans. Inf. Theory* **54**(6), 2798–2803 (2008)
10. Jaggi, S., Langberg, M., Katti, S., Ho, T., Katabi, D., Medard, M., Effros, M.: Resilient network coding in the presence of byzantine adversaries. *IEEE Trans. Inf. Theory* **54**(6), 2596–2603 (2008)
11. Nutman, L., Langberg, M.: Adversarial models and resilient schemes for network coding. In: Proceedings of 2008 IEEE International Symposium on Information Theory (ISIT), pp. 171–175 (2008)
12. Yu, Z.Y.Z., Wei, Y.W.Y., Ramkumar, B., Guan, Y.G.Y.: An efficient signature-based scheme for securing network coding against pollution attacks. In: IEEE INFOCOM 2008 - The 27th Conference on Computer Communications (2008)
13. Cai, N., Chan, T.: Theory of secure network coding. *Proc. IEEE* **99**, 421–437 (2011)
14. Cai, N., Yeung, R.W.: Secure network coding on a wiretap network. *IEEE Trans. Inf. Theory* **57**(1), 424–435 (2011)
15. Matsumoto, R., Hayashi, M.: Secure multiplex network coding. In: 2011 International Symposium on Networking Coding (2011). <https://doi.org/10.1109/ISNETCOD.2011.5979076>
16. Matsumoto, R., Hayashi, M.: Universal secure multiplex network coding with dependent and non-uniform messages. *IEEE Trans. Inform. Theory*; Arxiv preprint [arXiv: 1111.4174](https://arxiv.org/abs/1111.4174) (2011) (Accepted)
17. Agarwal, G.K., Cardone, M., Fragouli, C.: On (secure) information flow for multiple-unicast sessions: analysis with butterfly network. [arXiv: 1606.07561](https://arxiv.org/abs/1606.07561) (2016)
18. Hayashi, M., Iwama, K., Nishimura, H., Raymond, R., Yamashita, S.: Quantum network coding. In: Thomas, W., Weil, P. (eds.) STACS 2007. LNCS, vol. 4393, pp. 610–621. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-70918-3_52

19. Hayashi, M.: Prior entanglement between senders enables perfect quantum network coding with modification. *Phys. Rev. A* **76**(4), 40301 (2007)
20. Kobayashi, H., Le Gall, F., Nishimura, H., Rötteler, M.: General scheme for perfect quantum network coding with free classical communication. In: Albers, S., Marchetti-Spaccamela, A., Matias, Y., Nikolettseas, S., Thomas, W. (eds.) ICALP 2009. LNCS, vol. 5555, pp. 622–633. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-02927-1_52
21. Leung, D., Oppenheim, J., Winter, A.: Quantum network communication; the butterfly and beyond. *IEEE Trans. Inf. Theory* **56**(7), 3478–3490 (2010)
22. Kobayashi, H., Le Gall, F., Nishimura, H., Rötteler, M.: Perfect quantum network communication protocol based on classical network coding. In: Proceedings of 2010 IEEE International Symposium on Information Theory (ISIT), pp. 2686–2690 (2010)
23. Kobayashi, H., Le Gall, F., Nishimura, H., Rötteler, M.: Constructing quantum network coding schemes from classical nonlinear protocols. In: Proceedings of 2011 IEEE International Symposium on Information Theory (ISIT), pp. 109–113 (2011)
24. Chiribella, G., D’Ariano, G.M., Perinotti, P.: Quantum circuit architecture. *Phys. Rev. Lett.* **101**, 060401 (2008)
25. Chiribella, G., D’Ariano, G.M., Perinotti, P.: Theoretical framework for quantum networks. *Phys. Rev. A* **80**, 022339 (2009)
26. Gottesman, D., Chuang, I.L.: Demonstrating the viability of universal quantum computation using teleportation and single-qubit operations. *Nature* **402**, 390–393 (1999)
27. Bennett, C.H., Brassard, G., Crépeau, C., Jozsa, R., Peres, A., Wootters, W.K.: Teleporting an unknown quantum state via dual classical and einstein-podolsky-rosen channels. *Phys. Rev. Lett.* **70**, 1895–1899 (1993)
28. Cheng, F., Tan, V.Y.F.: A numerical study on the wiretap network with a simple network topology. *IEEE Trans. Inf. Theory* **62**(5), 2481–2492 (2016)
29. Ahlswede, R., Cai, N., Li, S.-Y.R., Yeung, R.W.: Network information flow. *IEEE Trans. Inf. Theory* **46**(4), 1204–1216 (2000)
30. Renes, J.M.: Duality of privacy amplification against quantum adversaries and data compression with quantum side information. *Proc. Roy. Soc. A* **467**(2130), 1604–1623 (2011)
31. Hayashi, M.: *Quantum Information Theory: Mathematical Foundation*, Graduate Texts in Physics. Springer, Heidelberg (2017). Second Edition of *Quantum Information, An Introduction*. Springer (2017)
32. Hayashi, M.: *Group Representation for Quantum Theory*. Springer, Cham (2017)
33. Hayashi, M., Owari, M., Kato, G., Cai, N.: [arXiv: 1703.00723](https://arxiv.org/abs/1703.00723) (2017). 2017 IEEE International Symposium on Information Theory (ISIT), Aachen, Germany, 25–30 June (2017, Accepted)
34. Owari, M., Kato, G., Hayashi, M.: Single-shot secure quantum network coding on butterfly network with free public communication. *Quantum Sci. Technol.* Retrieved from [arXiv:1705.01474](https://arxiv.org/abs/1705.01474) (2017, in Press)

Secure Network Coding for Multiple Unicast: On the Case of Single Source

Gaurav Kumar Agarwal^(✉), Martina Cardone, and Christina Fragouli

Department of Electrical and Computer Engineering,
University of California Los Angeles, Los Angeles, CA 90095, USA
{gauravagarwal,martina.cardone,christina.fragouli}@ucla.edu

Abstract. This paper considers multiple unicast wireline noiseless networks where a single source wishes to transmit independent messages to a set of legitimate destinations. The primal goal is to characterize the secure capacity region, where the exchanged messages have to be secured from a passive external eavesdropper that has unbounded computational capabilities, but limited network presence. The secure capacity region for the case of two destinations is characterized and it is shown to be a function of only the min-cut capacities and the number of edges the eavesdropper wiretaps. A polynomial-time two-phase scheme is then designed for a general number of destinations and its achievable secure rate region is derived. It is shown that the secure capacity result for the two destinations case is not reversible, that is, by switching the role of the source and destinations and by reversing the directions of the edges, the secure capacity region changes.

1 Introduction

Information theoretical network security is increasingly gaining importance, as we are moving towards a quantum computing era. On the one hand, the computational power at our disposal is continuously increasing and on the other, terabytes of data per seconds are exchanged over communication networks, a large portion of which needs to be secure (e.g., banking, professional, health). However, we still have very limited understanding of information theoretical security bounds and schemes over arbitrary networks.

In this paper, we consider an arbitrary wireline noiseless network with unit capacity edges where a source needs to securely transmit information to one or more receivers. A passive external eavesdropper, Eve, wishes to learn some information about the data exchanged between the legitimate nodes. Eve has unbounded computational capabilities (e.g., a quantum computer), but has limited network presence, namely, she can wiretap at most k edges of her choice. Over such a network, information theoretical network security seeks to design transmission schemes that are unconditionally/perfectly secure.

The work of the authors was partially funded by NSF under awards 1321120 and 1740047. G. K. Agarwal is also supported by the Guru Krupa Fellowship.

Our first main result is to extend the secure network coding capacity, from the single unicast and multicast cases [1], to the case of two unicast sessions. In particular, in a unicast session, if the min-cut capacity between the source and the receiver is M , then we can securely transmit information at rate $M - k$, where k is the number of edges Eve wiretaps (and the same result extends to the case of multicasting [1]). We prove in this paper that, if the source needs to send two independent messages to two receivers, a surprising direct extension of the single unicast case applies, where again the secure capacity region is uniquely determined by the min-cut capacities $M_{\{1\}}$, $M_{\{2\}}$ and $M_{\{1,2\}}$ (towards the first, second and the union of the two receivers), reduced by the number of the eavesdropped edges k , and thus the network structure plays no role. This is enabled by the observation that the source can establish secure keys with the two receivers that need not to be independent, i.e., they may share common randomness that can be efficiently multicast using network coding techniques. To the best of our knowledge, this is the first result that provides the secure capacity region characterization for a general network where multiple unicast sessions take place simultaneously.

Our second main result focuses on the case where we have an arbitrary number m of unicast sessions. We first derive an outer bound on the secure capacity region and then design a polynomial-time transmission scheme and derive its achievable secure rate region. In particular, our achievable scheme consists of two phases, where first secure keys are exchanged between the source and the destinations, then messages are encoded with these keys and finally transmitted. Although this scheme is not optimal, it is computationally efficient and it provides a performance guarantee on the secure achievable rate region as a function of any rate m -tuple that is achievable in the absence of the eavesdropper Eve.

Finally, we also show that the secure capacity result is irreversible, i.e., the capacity region of the reverse network (obtained by switching the role of the source and the destinations and by reversing the directions of the edges) is not the same as the one of the original network. This is a surprising result since it implies that – different from the unsecure case where irreversible networks must necessary have non-linear network coding solutions [2, 3] – under security constraints even networks with linear network coding solutions can be irreversible if the traffic is multiple unicast.

Related Work. The benefits of network coding were first shown in the seminal paper by Ahlswede et al. [4], where the authors proved that, in a noiseless network (represented by a directed acyclic graph) with single source and multiple destinations, the source can multicast at a rate equal to the minimum among all the min-cut capacities. Later, Li et al. [5] showed that it suffices to use random linear coding operations to achieve the multicast capacity and, more recently Jaggi et al. [6] designed polynomial-time deterministic algorithms to achieve it. While for the case of single unicast and multicast traffic the capacity is well-known, the same is not true for the case of networks where multiple unicast sessions take place simultaneously and share some of the network resources. For instance, even though the cut-set bound was proved to be tight for some

special cases, such as single source with non-overlapping demands and single source with non-overlapping demands and a multicast demand [7], in general it is not tight [8]. It was also recently showed by Kamath et al. [9] that characterizing the capacity of a general network where two unicast sessions take place simultaneously is as hard as characterizing the capacity of a network with general number of unicast sessions. For the case of single source and two destinations with a non-overlapping demand and a multicast demand, Ramamoorthy et al. [10] proposed a nice graph theory based approach to characterize the capacity region.

Cai et al. [1] characterized the secure capacity of a network with multicast traffic, where a passive external eavesdropper wiretaps any k edges of her choice. In particular, the authors showed that a secure multicast communication rate of $M - k$ can always be achieved, where M is the minimum among all the min-cut capacities. Also, for a multicast scenario, Cui et al. [11] designed a secure achievable scheme when Eve can wiretap only some of the edges (i.e., among all possible subsets of k edges, the eavesdropper can wiretap only some of them) and when the edge capacities are non-uniform. Since, even in the absence of the eavesdropper, the capacity of a multiple unicast network is not known in general, very few results are available for security. For instance, recently Agarwal et al. characterized the secure capacity region for some variations of the butterfly network both for noiseless [12] and erasure channels [13]. Although the results in [12] and [13] were the first that provided secure capacity results in multiple unicast scenarios, they are tailored to some specific network topologies. We here extend these results to a general multiple unicast network with single source (for which the capacity in absence of Eve is given by the cut-set bound [7]) and we characterize the secure capacity region for the case of two destinations.

Paper Organization. This paper is organized as follows. In Sect. 2, we define the setup (i.e., the multiple unicast network with single source and general number of destinations) and we formulate the problem. In Sect. 3, we focus on the secure capacity region characterization for our setup. In particular, we first derive an outer bound that holds for general number of destinations, we then show that this outer bound is tight for the case of two destinations and we finally design a two-phase secure transmission scheme for general number of destinations and compute its achievable rate region. In Sect. 4, we analyze and compare our designed schemes in terms of performance and complexity. In Sect. 4, we also show that the secure capacity result is irreversible and we finally conclude the paper.

2 Setup and Problem Formulation

Notation. Calligraphic letters indicate sets; \emptyset is the empty set and $|\mathcal{A}|$ is the cardinality of \mathcal{A} ; for two sets $\mathcal{A}_1, \mathcal{A}_2$, $\mathcal{A}_1 \subseteq \mathcal{A}_2$ indicates that \mathcal{A}_1 is a subset of \mathcal{A}_2 , $\mathcal{A}_1 \cup \mathcal{A}_2$ indicates the union of \mathcal{A}_1 and \mathcal{A}_2 , $\mathcal{A}_1 \sqcup \mathcal{A}_2$ indicates the disjoint union of \mathcal{A}_1 and \mathcal{A}_2 , $\mathcal{A}_1 \cap \mathcal{A}_2$ is the intersection of \mathcal{A}_1 and \mathcal{A}_2 and $\mathcal{A}_1 \setminus \mathcal{A}_2$ is the

set of elements that belong to \mathcal{A}_1 but not to \mathcal{A}_2 ; $[n_1 : n_2]$ is the set of integers from n_1 to $n_2 \geq n_1$; $[x]^+ := \max\{0, x\}$ for $x \in \mathbb{R}$.

We represent a wireline noiseless network with a directed acyclic graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where \mathcal{V} is the set of nodes and \mathcal{E} is the set of directed edges. We further assume that each edge $e \in \mathcal{E}$ is of unit capacity. If an edge $e \in \mathcal{E}$ connects a node i to a node j , we refer to node i as the tail and to node j as the head of e , i.e., $\text{tail}(e) = i$ and $\text{head}(e) = j$. For each node $v \in \mathcal{V}$, we define $\mathcal{I}(v)$ as the set of all incoming edges of node v and $\mathcal{O}(v)$ as the set of all outgoing edges of node v .

In this network (graph), there is one source node S and m destination nodes $D_i, i \in [1 : m]$. The source node does not have incoming edges, i.e., $\mathcal{I}(S) = \emptyset$, and each destination node does not have outgoing edges, i.e., $\mathcal{O}(D_i) = \emptyset, \forall i \in [1 : m]$. The source S has a message W_i for each destination $D_i, i \in [1 : m]$. The m messages are assumed to be independent. Thus, this network consists of multiple unicast traffic where m unicast sessions take place simultaneously and share some of the network resources. A passive eavesdropper Eve is also present in the network and can wiretap any k edges of her choice. We highlight that Eve is an external eavesdropper, i.e., it is not one of the destinations.

Each message $W_i, i \in [1 : m]$, is of q -ary entropy rate R_i and each channel is a discrete noiseless channel accepting alphabets over \mathbb{F}_q . Over this network, we are interested in finding all possible feasible m -tuples (R_1, R_2, \dots, R_m) such that each destination $D_i, i \in [1 : m]$, reliably decodes the message W_i and Eve receives no information about the messages. In particular, we are interested in information theoretic secure communication, i.e., we consider ‘‘perfect secrecy’’.

The symbol transmitted (respectively, received) over n channel uses on edge $e \in \mathcal{E}$ is denoted as X_e^n (respectively, Y_e^n). Similarly, $Z_e^n, e \in \mathcal{E}$, is the symbol received by Eve on edge $e \in \mathcal{E}$ over n channel uses. Clearly, since the channels are noiseless, $Y_{ei} = Z_{ei} = X_{ei}, \forall i \in [1 : n]$; throughout the paper, we use these symbols interchangeably. In addition, for $\mathcal{E}_t \subseteq \mathcal{E}$ we define $X_{\mathcal{E}_t}^n = \{X_e^n : e \in \mathcal{E}_t\}$, $Y_{\mathcal{E}_t}^n = \{Y_e^n : e \in \mathcal{E}_t\}$ and $Z_{\mathcal{E}_t}^n = \{Z_e^n : e \in \mathcal{E}_t\}$. We assume that the source node S has an independent and infinite source of randomness Θ , while the other nodes in the network do not have any randomness.

Definition 1. A rate m -tuple (R_1, R_2, \dots, R_m) is said to be securely achievable if there exist a block length n , a set of encoding functions $f_e, \forall e \in \mathcal{E}$, such that

$$X_e^n = \begin{cases} f_e(W_1, W_2, \dots, W_m, \Theta) & \text{if } \text{tail}(e) = \{S\} \\ f_e(\{Y_\ell^n : \ell \in \mathcal{I}(\text{tail}(e))\}) & \text{otherwise} \end{cases},$$

and destination D_i can reliably (with zero error) decode the message W_i i.e., $H(W_i | \{Y_e^n : e \in \mathcal{I}(D_i)\}) = 0$. Moreover, $\forall \mathcal{E}_Z \subseteq \mathcal{E}, |\mathcal{E}_Z| \leq k, I(W_{[1:m]} | Z_{\mathcal{E}_Z}^n) = 0$ (strong secrecy requirement). The closure of all such feasible rate m -tuples is the secure capacity region.

3 Secure Capacity

In this section we focus on the secure capacity region characterization for the network described in Sect. 2, when an eavesdropper Eve wiretaps any k edges of

her choice. In particular, we first derive an outer bound for a general number m of destinations and then design a secure transmission scheme that achieves the outer bound for $m = 2$. This result leads to the secure capacity region characterization for $m = 2$. Finally, we provide the design of a two-phase secure achievable scheme for a general number m of destinations and compute its achievable rate.

3.1 Outer Bound

We here derive an outer bound on the secure capacity region of a multiple unicast network with a single source and m destinations. In particular, the outer bound is provided in the next theorem.

Theorem 1. *An outer bound on the secure capacity region for a multiple unicast network with single source and m destinations is given by*

$$R_{\mathcal{A}} \leq [M_{\mathcal{A}} - k]^+, \quad \forall \mathcal{A} \subseteq [1 : m], \quad (1)$$

where $R_{\mathcal{A}} := \sum_{i \in \mathcal{A}} R_i$ and $M_{\mathcal{A}}$ is the min-cut capacity between the source S and the set of destinations $D_{\mathcal{A}} := \{D_i : i \in \mathcal{A}\}$.

Proof. Let $\mathcal{E}_{\mathcal{A}}$ be a min-cut between the source S and $D_{\mathcal{A}}$ and $\mathcal{E}_{\mathcal{Z}} \subseteq \mathcal{E}_{\mathcal{A}}$ be the set of k edges wiretapped by Eve and define $\mathcal{I}(D_{\mathcal{A}}) := \bigcup_{i \in \mathcal{A}} \mathcal{I}(D_i)$. If $|\mathcal{E}_{\mathcal{A}}| < k$, let $\mathcal{E}_{\mathcal{Z}} = \mathcal{E}_{\mathcal{A}}$. We have,

$$\begin{aligned} nR_{\mathcal{A}} &= H(W_{\mathcal{A}}) \stackrel{(a)}{=} H(W_{\mathcal{A}}) - H(W_{\mathcal{A}}|X_{\mathcal{I}(D_{\mathcal{A}})}^n) \\ &\stackrel{(b)}{=} H(W_{\mathcal{A}}) - H(W_{\mathcal{A}}|X_{\mathcal{E}_{\mathcal{A}}}^n) \\ &\stackrel{(c)}{=} I(W_{\mathcal{A}}; X_{\mathcal{E}_{\mathcal{Z}}}^n, X_{\mathcal{E}_{\mathcal{A}} \setminus \mathcal{E}_{\mathcal{Z}}}^n) \\ &= I(W_{\mathcal{A}}; X_{\mathcal{E}_{\mathcal{Z}}}^n) + I(W_{\mathcal{A}}; X_{\mathcal{E}_{\mathcal{A}} \setminus \mathcal{E}_{\mathcal{Z}}}^n | X_{\mathcal{E}_{\mathcal{Z}}}^n) \\ &\stackrel{(d)}{=} I(W_{\mathcal{A}}; X_{\mathcal{E}_{\mathcal{A}} \setminus \mathcal{E}_{\mathcal{Z}}}^n | X_{\mathcal{E}_{\mathcal{Z}}}^n) \\ &\stackrel{(e)}{\leq} H(X_{\mathcal{E}_{\mathcal{A}} \setminus \mathcal{E}_{\mathcal{Z}}}^n) \\ &\stackrel{(f)}{\leq} n[M_{\mathcal{A}} - k]^+, \end{aligned}$$

where $W_{\mathcal{A}} = \{W_i, i \in \mathcal{A}\}$ and where: (i) the equality in (a) follows because of the decodability constraint; (ii) the equality in (b) follows because $X_{\mathcal{I}(D_{\mathcal{A}})}^n$ is a deterministic function of $X_{\mathcal{E}_{\mathcal{A}}}^n$; (iii) the equality in (c) follows from the definition of mutual information and since $\mathcal{E}_{\mathcal{A}} = \mathcal{E}_{\mathcal{Z}} \cup \mathcal{E}_{\mathcal{A} \setminus \mathcal{Z}}$; (iv) the equality in (d) follows because of the perfect secrecy requirement; (v) the inequality in (e) follows since the entropy of a discrete random variable is a non-negative quantity and because of the ‘conditioning reduces the entropy’ principle; (vi) finally, the inequality in (f) follows since each link has unit capacity and since $|\mathcal{E}_{\mathcal{A}} \setminus \mathcal{E}_{\mathcal{Z}}| = [M_{\mathcal{A}} - k]^+$. By dividing both sides of the above inequality by n we obtain that $R_{\mathcal{A}}$ in (1) is an outer bound on the capacity region of the multiple unicast network with single source and m destinations. This concludes the proof of Theorem 1.

Remark 1. Since the eavesdropper Eve wiretaps any k edges of her choice, intuitively Theorem 1 states that if she wiretaps k edges of a cut with capacity M , we can at most hope to reliably transmit at rate $M - k$. However, this holds only for the case of single source; indeed, as we will see in Sect. 4.2 through an example, higher rates can be achieved for the case of single destination and multiple sources.

3.2 Secure Capacity Region for $m = 2$

We here prove that the outer bound in Theorem 1 is indeed tight for the case $m = 2$. In particular, our main result is stated in the following theorem.

Theorem 2. *The outer bound in (1) is tight for the case $m = 2$, i.e., the secure capacity region of a multiple unicast network with single source and $m = 2$ destinations is given by*

$$R_1 \leq [M_{\{1\}} - k]^+ , \quad (2a)$$

$$R_2 \leq [M_{\{2\}} - k]^+ , \quad (2b)$$

$$R_1 + R_2 \leq [M_{\{1,2\}} - k]^+ . \quad (2c)$$

Proof. Clearly, from the result in Theorem 1, the rate region in (2) is an outer bound on the capacity region of a multiple unicast network with single source and $m = 2$ destinations. Hence, we now need to prove that the rate region in (2) is also achievable. Towards this end, we start by providing the following definition of *separable* graphs.

Definition 2. *A graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with a single source and m destinations is said to be **separable** if its edge set \mathcal{E} can be partitioned as $\mathcal{E} = \sqcup_{\ell=1}^{2^m-1} \mathcal{E}'_{\ell}$ such that $\mathcal{G}'_{\ell} = (\mathcal{V}, \mathcal{E}'_{\ell})$ and*

$$M_{\mathcal{A}} = \sum_{\substack{\mathcal{J} \subseteq [1:m] \\ \mathcal{J} \cap \mathcal{A} \neq \emptyset}} M_{\mathcal{J}}^* , \quad \forall \mathcal{A} \subseteq [1 : m] ,$$

where $M_{\mathcal{A}}$ is the min-cut capacity between the source S and the set of destinations $D_{\mathcal{A}} := \{D_i : i \in \mathcal{A}\}$ in \mathcal{G} and $M_{\mathcal{J}}^*$ is the min-cut capacity between the source S and the set of destinations $D_{\mathcal{B}} := \{D_b : b \in \mathcal{B}\}$, $\forall \mathcal{B} \subseteq \mathcal{J}$ for the graph \mathcal{G}'_{ℓ} with $\ell \in [1 : 2^m - 1]$ being the decimal representation of the binary vector of length m that has a one in all the positions indexed by $j \in \mathcal{J}$ and zero otherwise, with the least significant bit in the first position.

To better understand the above definition, consider a graph \mathcal{G} with $m = 2$ destinations. Then, the graph \mathcal{G} is separable if it can be partitioned into 3 graphs such that:

- \mathcal{G}'_1 has the following min-cut capacities: $M_{\{1\}}^*$ from S to D_1 and zero from S to D_2 ,

- \mathcal{G}'_2 has the following min-cut capacities: zero from S to D_1 and $M_{\{2\}}^*$ from S to D_2 ,
- \mathcal{G}'_3 has the following min-cut capacities: $M_{\{1,2\}}^*$ from S to D_1 , $M_{\{1,2\}}^*$ from S to D_2 and $M_{\{1,2\}}^*$ from S to $\{D_1, D_2\}$,

where the quantities $M_{\{1\}}^*$, $M_{\{2\}}^*$ and $M_{\{1,2\}}^*$ can be computed using the following set of equations:

$$M_{\{i\}} = M_{\{i\}}^* + M_{\{1,2\}}^*, \forall i \in [1 : 2] , \tag{3a}$$

$$M_{\{1,2\}} = M_{\{1\}}^* + M_{\{2\}}^* + M_{\{1,2\}}^* . \tag{3b}$$

For example, consider the network \mathcal{G}_0 in Fig. 1(a), which has min-cut capacities $M_{\{1\}} = M_{\{2\}} = 3$ and $M_{\{1,2\}} = 4$. It is not difficult to see that \mathcal{G}_0 in Fig. 1(a) can be partitioned in three graphs $\mathcal{G}'_i, i \in [1 : 3]$ as shown in Figs. 1(b)–(d), with min-cut capacities equal to (see (3)) $M_{\{1\}}^* = M_{\{2\}}^* = 1$ and $M_{\{1,2\}}^* = 2$.

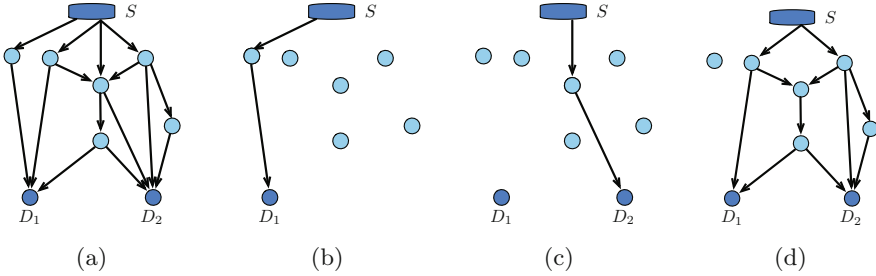


Fig. 1. A 2-destination separable network \mathcal{G}_0 in Fig. 1(a) and its partition graphs $\mathcal{G}'_i, i \in [1 : 3]$ in Figs. 1(b)–(d).

We now state the following lemma, which is a direct consequence of [10, Theorem 1] and we will use to prove the achievability of the rate region in (2).

Lemma 1. *Any graph with a single source and $m = 2$ destinations is separable.*

For completeness we report the proof of Lemma 1 in Appendix A. By leveraging the result in Lemma 1, we are now ready to prove Theorem 2. In particular, we consider two cases depending on the value of k (i.e., the number of edges the eavesdropper wiretaps). Without loss of generality, we assume that $k < \min_{i \in [1:2]} M_i$, as otherwise secure communication to the set of destinations $\{D_i : k \geq M_i\}$ is not possible at any rate, and hence we can just remove this set of destinations from the network.

1. **Case 1:** $k \geq M_{\{1,2\}}^*$. In this case, by substituting the quantities in (3) into (2), we obtain that the constraint in (2c) is redundant. Thus, we will now prove that the rate pair $(R_1, R_2) = (M_{\{1\}} - k, M_{\{2\}} - k)$ is securely achievable,

which along with the time-sharing argument proves the achievability of the entire region in (2).

We denote with y_1, y_2, \dots, y_k the k key packets and with $m_i^{(1)}, m_i^{(2)}, \dots, m_i^{(R_i)}$ (with $i \in [1 : 2]$) the R_i message packets for D_i . With this, our scheme is as follows:

- We multicast $y_i, \forall i \in [1 : M_{\{1,2\}}^*]$, to both D_1 and D_2 using \mathcal{G}'_3 , which has edges denoted by \mathcal{E}'_3 . This is possible as \mathcal{G}'_3 has a min-cut capacity $M_{\{1,2\}}^*$ to both D_1 and D_2 (see Definition 2).
- We unicast $y_\ell, \forall \ell \in [M_{\{1,2\}}^* + 1 : k]$, to $D_i, \forall i \in [1 : 2]$, using $k - M_{\{1,2\}}^*$ paths out of the $M_{\{i\}}^*$ disjoint paths in \mathcal{G}'_i . We denote by $\hat{\mathcal{E}}_i$ the set that contains all the first edges of these paths. Clearly, $|\hat{\mathcal{E}}_i| = k - M_{\{1,2\}}^*, \forall i \in [1 : 2]$. Notice that $\hat{\mathcal{E}}_i \subseteq \mathcal{E}'_i, \forall i \in [1 : 2]$ (see Definition 2).
- We send the $R_i, \forall i \in [1 : 2]$, encrypted message packets (i.e., encoded with the keys) of D_i on the remaining $M_{\{i\}}^* - k + M_{\{1,2\}}^*$ disjoint paths in \mathcal{G}'_i . We denote by $\bar{\mathcal{E}}_i$ the set that contains all the first edges of these paths in \mathcal{G}'_i . Clearly, $|\bar{\mathcal{E}}_i| = R_i, \forall i \in [1 : 2]$, $\bar{\mathcal{E}}_i \subseteq \mathcal{E}'_i$ and $\bar{\mathcal{E}}_i \cap \hat{\mathcal{E}}_i = \emptyset$ (see Definition 2).

This scheme achieves $R_i = M_{\{i\}}^* - k + M_{\{1,2\}}^* = M_{\{i\}} - k, \forall i \in [1 : 2]$, where the second equality follows by using the definitions in (3). Now we prove that this scheme is also secure. We start by noticing that, thanks to Definition 2, the edges $\mathcal{E}'_3, \hat{\mathcal{E}}_i$ and $\bar{\mathcal{E}}_i$, with $i \in [1 : 2]$, do not overlap. We write these transmissions in a matrix form (with G and U being the encoding matrices) and we obtain

$$\begin{bmatrix} X_{\mathcal{E}'_3} \\ X_{\hat{\mathcal{E}}_1} \\ X_{\hat{\mathcal{E}}_2} \end{bmatrix} = \underbrace{\begin{bmatrix} g_{11} & g_{12} & \dots & g_{1k} \\ g_{21} & g_{22} & \dots & g_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ g_{\ell 1} & g_{\ell 2} & \dots & g_{\ell k} \end{bmatrix}}_G \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_k \end{bmatrix}, \quad \ell = |\mathcal{E}'_3| + 2 \left(k - M_{\{1,2\}}^* \right),$$

$$\begin{bmatrix} X_{\bar{\mathcal{E}}_1} \\ X_{\bar{\mathcal{E}}_2} \end{bmatrix} = \underbrace{\begin{bmatrix} u_{11} & u_{12} & \dots & u_{1k} \\ u_{21} & u_{22} & \dots & u_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ u_{r1} & u_{r2} & \dots & u_{rk} \end{bmatrix}}_U \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_k \end{bmatrix} \oplus \begin{bmatrix} m_1^{(1)} \\ \vdots \\ m_1^{(R_1)} \\ m_2^{(1)} \\ \vdots \\ m_2^{(R_2)} \end{bmatrix}, \quad r = R_1 + R_2.$$

The eavesdropper Eve wiretaps $k_1 \leq k$ edges from the collection of edges $\{\mathcal{E}'_3, \hat{\mathcal{E}}_1, \hat{\mathcal{E}}_2\}$, over which the linear combinations $X_{\mathcal{E}'_3}, X_{\hat{\mathcal{E}}_1}$ and $X_{\hat{\mathcal{E}}_2}$ of keys are transmitted, and $k_2 = k - k_1$ edges from the collection of edges $\{\bar{\mathcal{E}}_1, \bar{\mathcal{E}}_2\}$ over which the messages encoded with the keys $X_{\bar{\mathcal{E}}_1}$ and $X_{\bar{\mathcal{E}}_2}$ are transmitted. We here note that on the other edges $\mathcal{E} \setminus \{\mathcal{E}'_3 \cup \hat{\mathcal{E}}_1 \cup \bar{\mathcal{E}}_1 \cup \hat{\mathcal{E}}_2 \cup \bar{\mathcal{E}}_2\}$, of the network, we either do not transmit any symbol or simply route the symbols from

$\{X_{\mathcal{E}_1}, X_{\mathcal{E}_2}, X_{\mathcal{E}_1}, X_{\mathcal{E}_2}\}$ (corresponding to the symbols transmitted on disjoint paths). Thus, without loss of generality, we can assume that Eve does not wiretap any of these edges. Since the first $|\mathcal{E}'_3|$ rows of G (i.e., those that correspond to multicasting of they keys) are determined by the network coding scheme for multicasting [4], we assume that we do not have any control over the construction of G .

Thus, we would like to construct the code matrix U such that all the linear combinations of the keys used to encrypt the messages are mutually independent and are independent from the linear combinations of the keys wiretapped on the k_1 edges (notice that this makes the symbols wiretapped by the eavesdropper completely independent from the messages). In particular, since in the worst case Eve wiretaps k_1 edges which are independent linear combinations, we would like that any matrix formed by k_1 independent rows of the matrix G and k_2 rows of the matrix U is full rank. Since there is a finite number of such choices and the determinant of each of these possible matrices can be written in a polynomial form – which is not identically zero – as a function of the entries of U , then we can choose the entries of U such that all these matrices are invertible. Thus, we can always construct the code matrix U such that the edges wiretapped by Eve have an independent key and hence Eve does not get any information about the message packets, i.e., the scheme is secure. This implies that the rate pair $(R_1, R_2) = (M_{\{1\}} - k, M_{\{2\}} - k)$ is securely achievable.

2. **Case 2:** $k < M_{\{1,2\}}^*$. By substituting the quantities in (3), the rate region in (2) becomes

$$R_i \leq M_{\{i\}} - k = M_{\{i\}}^* + M_{\{1,2\}}^* - k, \forall i \in [1 : 2] , \quad (4a)$$

$$R_1 + R_2 \leq M_{\{1,2\}} - k = M_{\{1\}}^* + M_{\{2\}}^* + M_{\{1,2\}}^* - k . \quad (4b)$$

We now show that we can achieve the following two corner points i.e., the rate pair

$$\begin{aligned} (R_1, R_2) &= (\alpha(M_{\{1,2\}} - M_{\{2\}}) + (1 - \alpha)(M_{\{1\}} - k), \\ &\quad \alpha(M_{\{2\}} - k) + (1 - \alpha)(M_{\{1,2\}} - M_{\{1\}})) \\ &\stackrel{(a)}{=} (M_{\{1\}}^* + \alpha(M_{\{1,2\}}^* - k), M_{\{2\}}^* + (1 - \alpha)(M_{\{1,2\}}^* - k)) , \end{aligned} \quad (5)$$

for $\alpha \in \{0, 1\}$, where the equality in (a) follows by using the definitions in (3). This along with the time-sharing argument proves the achievability of the entire region in (4). We recall that we denote with y_1, y_2, \dots, y_k the k key packets and with $m_i^{(1)}, m_i^{(2)}, \dots, m_i^{(R_i)}$ (with $i \in [1 : 2]$) the R_i message packets for D_i . With this, our scheme is as follows:

- Using the graph \mathcal{G}'_3 we multicast to both destinations D_1 and D_2 :
 - (i) $y_i, \forall i \in [1 : k]$, (ii) $\alpha(M_{\{1,2\}}^* - k)$ encrypted message packets (i.e., encoded with the keys) for D_1 and (iii) $(1 - \alpha)(M_{\{1,2\}}^* - k)$ encrypted message packets for D_2 . Recall that the edges of the graph \mathcal{G}'_3 are denoted

by \mathcal{E}'_3 (see Definition 2). We also highlight that the message packets multicast to the two destinations are encrypted using the key packets, where the encryption is based on the secure network coding result on multicasting [1], which ensures perfect security from an adversary wiretapping any k edges.

- We send $M_{\{i\}}^*$ encrypted message packets of D_i on the $M_{\{i\}}^*$ disjoint paths to D_i in the graph \mathcal{G}'_i , and denote by $\hat{\mathcal{E}}_i$ the set that contains all the first edges of these paths for $i \in [1 : 2]$.

This scheme achieves the rate pair in (5). Now we prove that this scheme is also secure. For ease of representation, in what follows we let $R_1^* = \alpha(M_{\{1,2\}}^* - k)$ and $R_2^* = (1 - \alpha)(M_{\{1,2\}}^* - k)$. We again notice that, thanks to Definition 2, the edges \mathcal{E}'_3 , $\hat{\mathcal{E}}_1$ and $\hat{\mathcal{E}}_2$ do not overlap. We write these transmissions in a matrix form (with G , U and S being encoding matrices) and we obtain,

$$X_{\mathcal{E}'_3} = \underbrace{\begin{bmatrix} g_{11} & g_{12} & \dots & g_{1k} \\ g_{21} & g_{22} & \dots & g_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ g_{\ell 1} & g_{\ell 2} & \dots & g_{\ell k} \end{bmatrix}}_G \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_k \end{bmatrix} \oplus \underbrace{\begin{bmatrix} s_{11} & s_{12} & \dots & s_{1k} \\ s_{21} & s_{22} & \dots & s_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ s_{\ell 1} & s_{\ell 2} & \dots & s_{\ell k} \end{bmatrix}}_S \begin{bmatrix} m_1^{(1)} \\ \vdots \\ m_1^{(R_1^*)} \\ m_2^{(1)} \\ \vdots \\ m_2^{(R_2^*)} \end{bmatrix}, \quad \ell = |\mathcal{E}'_3|,$$

$$\begin{bmatrix} X_{\hat{\mathcal{E}}_1} \\ X_{\hat{\mathcal{E}}_2} \end{bmatrix} = \underbrace{\begin{bmatrix} u_{11} & u_{12} & \dots & u_{1k} \\ u_{21} & u_{22} & \dots & u_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ u_{r1} & u_{r2} & \dots & u_{rk} \end{bmatrix}}_U \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_k \end{bmatrix} \oplus \begin{bmatrix} m_1^{(R_1^*+1)} \\ \vdots \\ m_1^{(R_1)} \\ m_2^{(R_2^*+1)} \\ \vdots \\ m_2^{(R_2)} \end{bmatrix}, \quad r = R_1 + R_2 - (M_{\{1,2\}}^* - k).$$

The eavesdropper Eve wiretaps $k_1 \leq k$ edges from \mathcal{E}'_3 , over which the linear combinations $X_{\mathcal{E}'_3}$ of key packets and message packets are sent, and $k_2 = k - k_1$ edges from the collection of edges $\{\hat{\mathcal{E}}_1, \hat{\mathcal{E}}_2\}$ over which the messages encoded with the keys $X_{\hat{\mathcal{E}}_1}$ and $X_{\hat{\mathcal{E}}_2}$ are transmitted. Similar to Case 1, on the other edges $\mathcal{E} \setminus \{\mathcal{E}'_3 \cup \hat{\mathcal{E}}_1 \cup \hat{\mathcal{E}}_2\}$ of the network, we either do not transmit any symbol or simply route the symbols from $\{X_{\hat{\mathcal{E}}_1}, X_{\hat{\mathcal{E}}_2}\}$ (corresponding to the symbols transmitted on disjoint paths). Thus, without loss of generality, we can assume that the eavesdropper does not wiretap any of these edges. Since the matrices G and S are determined by the secure network coding scheme for multicasting [1], we do not have any control over their construction. Thus, we would like to construct the code matrix U in order to ensure security. Again, similar to the argument used in Case 1, we can create U such that any subset of k_2 rows of U are linearly independent and not in the span of

any subset of k_1 rows of G . With this, the keys used to encrypt the messages over any k_2 edges of $\{\hat{\mathcal{E}}_1, \hat{\mathcal{E}}_2\}$ are mutually independent and independent from the keys used over any k_1 edges of \mathcal{E}'_3 . This, together with the fact that the messages transmitted using \mathcal{G}'_3 are already secure, makes our scheme secure. This implies that the rate pair (R_1, R_2) in (5) is securely achievable.

This concludes the proof of Theorem 2.

3.3 A Two-Phase Scheme

We now propose the design of a secure transmission scheme that consists of two phases, namely the key generation phase (in which secret keys are generated between the source and the m destinations) and message sending phase (in which the message packets are first encoded using the secret keys and then transmitted to the m destinations). The corresponding achievable rate region is presented in Theorem 3.

Theorem 3. *Let $(\hat{R}_1, \hat{R}_2, \dots, \hat{R}_m)$ be an achievable rate m -tuple in absence of the eavesdropper Eve. Then, the rate m -tuple (R_1, R_2, \dots, R_m) with*

$$R_i = \hat{R}_i \left(1 - \frac{k}{M}\right), \forall i \in [1 : m], \quad (6)$$

where M is the minimum min-cut between the source and any destination, is securely achievable in the presence of an eavesdropper Eve who wiretaps any k edges of her choice.

Proof. Let M_i be the min-cut capacity between the source and the destination D_i with $i \in [1 : m]$. We define M as the minimum among all these individual min-cut capacities, i.e., $M = \min_{i \in [1:m]} M_i$. Let $(\hat{R}_1, \hat{R}_2, \dots, \hat{R}_m) \in \mathbb{R}^m$ be the unsecure rate m -tuple achieved in the absence of the eavesdropper. We start by approximating this rate m -tuple with rational numbers; notice that this is always possible since the set of rationals \mathbb{Q} is dense in \mathbb{R} . Moreover, an information flow through the network (from the source S to an artificial destination D' connected to all the destinations $D_i, i \in [1 : m]$ – see also Appendix B) that achieves this rate m -tuple might involve fractional flows over the edges since the rate m -tuple may be fractional. To make the rate m -tuple integral and thereby also the flow over each edge, we multiply the capacity of each edge by a common factor T . This implies that to achieve $(\hat{R}_1, \hat{R}_2, \dots, \hat{R}_m)$, then $(T\hat{R}_1, T\hat{R}_2, \dots, T\hat{R}_m)$ is achieved over T instances of the network after which the flow over each edge is an integer. In what follows, we describe our coding scheme and we show that

$$(R_1, R_2, \dots, R_m) = \left(1 - \frac{k}{M}\right) (\hat{R}_1, \hat{R}_2, \dots, \hat{R}_m) \quad (7)$$

is achievable. This particular scheme consists of the two following phases.

- *Key generation.* This first phase – in which secure keys are generated between the source and the destinations – consists of k subphases. In each subphase, the source multicasts $M - k$ random packets securely to all destinations. This is possible thanks to the secure network coding result of [1], since the minimum min-cut capacity is M and Eve has access to k edges. Thus, at the end of this phase, a total of $Tk(M - k)$ secure keys are generated, since in each phase we use the network T times.
- *Message sending.* We choose Tk packets out of the $Tk(M - k)$ securely shared (in the key generation phase) random packets. For each choice of Tk packets, we convert the unsecure scheme achieving $(TR_1, TR_2, \dots, TR_m)$ to a secure scheme achieving the same rate m -tuple. Towards this end, we expand the Tk shared packets into $\sum_{j=1}^m T\hat{R}_j$ packets using an MDS code matrix. With this, we have the same number of random packets as the message packets. We then encode the message packets with the random packets and transmit them as it was done in the corresponding unsecure scheme. We repeat this process until we run out of the shared random packets, i.e., we repeat this process $M - k$ times by using T instances of the network each time.

Proof of security. We know that, in absence of security considerations, a time-sharing based scheme is optimal (i.e., capacity achieving) for a multiple unicast network with single source, i.e., network coding is not beneficial [7] (see also Appendix B) Given that we are not using network coding operations and since each edge carries an integer information flow, then the eavesdropper will be able to wiretap at most Tk different messages each encrypted with an independent key. Hence, the eavesdropper will not be able to obtain any information about any of the m messages.

Analysis of the achieved rate m -tuple. The secure scheme described above requires a total of M phases. In particular, in the first k phases we generate the secure keys and in the remaining $M - k$ phases we securely transmit at rates of $(TR_1, TR_2, \dots, TR_m)$, over T network instances. Thus, the achieved secure message rate (R_1, R_2, \dots, R_m) is

$$R_j = \frac{M - k}{M} \hat{R}_j = \left(1 - \frac{k}{M}\right) \hat{R}_j, \forall j \in [1 : m] . \quad (8)$$

This concludes the proof of Theorem 3.

4 Discussion and Conclusions

In this section, we analyze, discuss and compare the results that we have derived in the paper. In particular, we first compare the secure capacity region in (2) with the capacity region of the same network in the absence of the eavesdropper. We then show that the secure capacity result in (2), different from the unsecure counterpart, is irreversible. We also analyze the complexity of the capacity achieving scheme and of the two-phase scheme. Finally, we summarize our main contributions and conclude the paper.

4.1 Secure Capacity Versus Unsecure Capacity

For the network with single source and multiple destinations described in Sect. 2, the unsecure capacity (i.e., in the absence of the eavesdropper) is well known [7, Theorem 9] and given by the following lemma. For completeness we report the proof of the following lemma in Appendix B.

Lemma 2. *The unsecure capacity region for a multiple unicast network with single source node and m destination nodes is given by*

$$R_{\mathcal{A}} \leq M_{\mathcal{A}}, \quad \forall \mathcal{A} \subseteq [1 : m], \quad (9)$$

where $R_{\mathcal{A}} := \sum_{i \in \mathcal{A}} R_i$ and $M_{\mathcal{A}}$ is the min-cut capacity between the source S and the set of destinations $D_{\mathcal{A}} := \{D_i : i \in \mathcal{A}\}$.

We now focus on the case of $m = 2$ destinations and compare the secure capacity region in Theorem 2 and the unsecure capacity region in Lemma 2. By comparing (2) with (9) (evaluated for the case $m = 2$), we observe that in the presence of the eavesdropper we lose at most a rate k in each dimension compared to the unsecure case. We notice that the same result holds for the case of $m = 1$ destination and for the case of multicasting the same message to all receivers [1] (i.e., we have a rate loss of k with respect to the min-cut capacity M). However, here it is more surprising since the messages to the $m = 2$ receivers (and potentially the keys) are different.

4.2 Reversibility and Non-reversibility

In order to characterize the unsecure capacity region in (9) network coding is not necessary and routing is sufficient (see also Appendix B). Thus, from the result in [3], it directly follows that the capacity result in (9) is reversible. In particular, let \mathcal{G} be a network with single source and m destinations with a certain capacity region (that can be computed from Lemma 2). Then, the reverse graph \mathcal{G}' is constructed by switching the role of the source and destinations and by reversing the directions of the edges. Thus, \mathcal{G}' will have m sources and one single destination. The result in [3] ensures that \mathcal{G} and \mathcal{G}' will have the same capacity region, i.e., the result in Lemma 2 characterizes also the unsecure capacity region for a multiple unicast network with m sources and single destination.

We now focus on the secure case. In Sect. 3, we have characterized the secure capacity region for a multiple unicast network with single source and $m = 2$ destinations. In particular, Theorem 2 implies that the secure capacity region does not depend on the specific topology of the network and it can be fully characterized by the min-cut capacities $M_{\{1\}}$, $M_{\{2\}}$ and $M_{\{1,2\}}$ and by the number k of edges wiretapped by Eve. We now show that this result is irreversible, i.e., the secure capacity region of the reverse network is not the same as the one of the original network. Moreover, we also show that the secure capacity region with 2 sources and single destination cannot anymore be characterized by only the min-cut capacities, i.e., it depends on the specific network topology.

Consider the three networks in Fig. 2 and assume $k = 1$, i.e., Eve wiretaps one edge of her choice. For the network in Fig. 2(a) we have min-cut capacities $(M_{\{1\}}, M_{\{2\}}, M_{\{1,2\}}) = (1, 2, 2)$ and hence from Theorem 2 it follows that the secure capacity for this network is given by $(R_1, R_2) = (0, 1)$. This point can be achieved by simply using the scheme shown in Fig. 2(a), where y represents the key and W_2 the message for D_2 . Now, consider the network in Fig. 2(b) that is obtained from Fig. 2(a) by switching the role of the source and destinations and by reversing the directions of the edges. For this network, which has the same min-cut capacities as the network in Fig. 2(a), the rate pair $(R_1, R_2) = (1, 0)$ is securely achievable using the scheme shown in Fig. 2(a) where W_1 is the message of S_1 and y_1 and y_2 are the keys generated by S_1 and S_2 , respectively. The rate pair $(R_1, R_2) = (1, 0)$, which is securely achieved by the network in Fig. 2(b), cannot be securely achieved by the network in Fig. 2(a). This result implies that a secure rate pair that is feasible for one network might not be feasible for the reverse network, i.e., the secure capacity regions can be different and hence cannot be derived from one another. The achievability of the pair $(R_1, R_2) = (1, 0)$ in Fig. 2(b) also shows that the outer bound in (1) does not hold for the case of single destination and multiple sources, in which case it is possible to achieve rates outside this region.

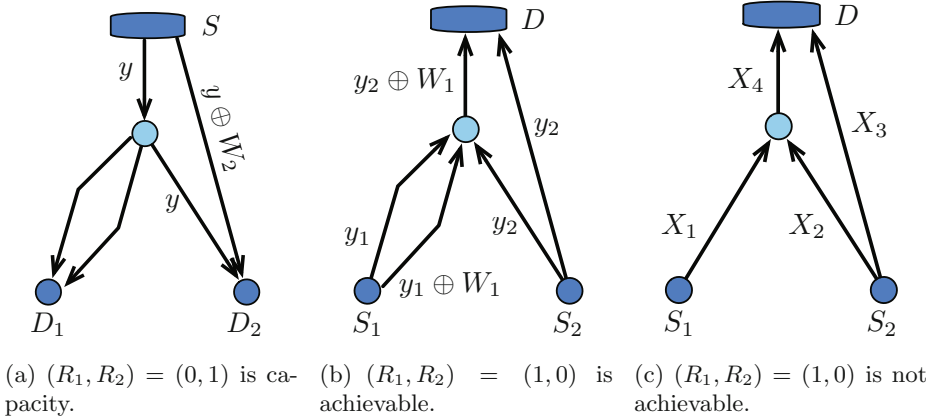


Fig. 2. Network examples.

Consider now the network in Fig. 2(c), which has the same min-cut capacities $(M_{\{1\}}, M_{\{2\}}, M_{\{1,2\}}) = (1, 2, 2)$ as the network in Fig. 2(b). We now show that the rate pair $(R_1, R_2) = (1, 0)$, which can be securely achieved in the network in Fig. 2(b), cannot be securely achieved in the network in Fig. 2(c). Let $X_i, i \in [1 : 4]$, be the transmitted symbols as shown in Fig. 2(c). With this, we have

$$\begin{aligned}
R_1 = H(W_1) &\stackrel{(a)}{=} H(W_1) - H(W_1|X_3, X_4) \stackrel{(b)}{\leq} H(W_1) - H(W_1|X_1, X_2, X_3) \\
&= I(W_1; X_1, X_2, X_3) \\
&= I(W_1; X_1) + I(W_1; X_2, X_3|X_1) \\
&\stackrel{(c)}{=} I(W_1; X_2, X_3|X_1) \\
&= H(X_2, X_3|X_1) - H(X_2, X_3|W_1, X_1) \\
&\stackrel{(d)}{=} H(X_2, X_3) - H(X_2, X_3) = 0 \text{ ,}
\end{aligned}$$

where: (i) the equality in (a) follows because of the decodability constraint; (ii) the inequality in (b) follows because of the ‘conditioning reduces the entropy’ principle and since X_4 is a deterministic function of (X_1, X_2) ; (iii) the equality in (c) follows because of the perfect secrecy requirement; (iv) finally, the equality in (d) follows since (X_2, X_3) is independent of (W_1, X_1) . This result shows that the rate pair $(R_1, R_2) = (1, 0)$ is not securely achievable in the network in Fig. 2(c). This implies that, for a network with single destination and multiple sources, we cannot characterize the secure capacity region based only on the min-cut capacities $(M_{\{1\}}, M_{\{2\}}, M_{\{1,2\}})$, i.e., the result would depend on the specific network topology.

4.3 Complexity Analysis

The capacity achieving scheme for $m = 2$ destinations that we have proposed (see Sect. 3.2) first requires that we edge-partition the original graph \mathcal{G} into three graphs (i.e., an edge in \mathcal{G} appears in only one of these three graphs). At this stage, this step requires an exhaustive search over all possible paths in the network, which requires an exponential number of operations in the number of nodes. It therefore follows that the scheme proposed in Sect. 3.2, even though it allows to characterize the secure capacity region, is of exponential complexity.

Differently, the two-phase scheme proposed in Sect. 3.3 runs in polynomial time. This is because all the operations that it requires (i.e., find a T such that over T instances all flows are integer, multicast the keys in the key generation phase, encrypt messages at the source (i.e., encode the messages with the keys) and route the encrypted messages) can be performed in polynomial time in the number of edges. However, the two-phase scheme described in Sect. 3.3 is sub-optimal and does not achieve the outer bound in (1). However, this scheme offers a guarantee on the secure rate region that can always be achieved as a function of any rate m -tuple that is achievable in the absence of the eavesdropper Eve (see (6) in Theorem 3).

One reason behind this is that in the key generation phase some edges in the network are not used. Indeed, when we multicast the M random packets to generate the keys (where M is the minimum of the min-cut capacities and k is the number of edges wiretapped by the eavesdropper) – out of which $M - k$ linear combinations are secure keys – it might have been possible to use the other edges (i.e., those through which the random packets do not flow) to transmit

some encrypted message packets. For instance, consider the network example in Fig. 3(a), where the eavesdropper wiretaps $k = 1$ edge of her choice. Our two-phase scheme would multicast $M = \min_{i \in [1:2]} M_{\{i\}} = 2$ random packets y_1 and y_2 (y_1 is transmitted over the solid edges and y_2 over the dashed edges in Fig. 3(a)), out of which $M - k = 1$ is securely received by D_1 and D_2 . Hence, the combination $y_1 \oplus y_2$ can be used to securely transmit the message packets. However, we see that in the first phase the dotted edge (i.e., the one that connects S directly to D_2) is not used. This brings to a reduction in the achievable rate region since this edge could have been used to securely transmit a message packet to D_2 by using $W_2 \oplus y_1$ as shown in Fig. 3(a). Given this, we believe that what makes the two-phase scheme suboptimal is the fact that it does not fully leverage all the network resources. In Fig. 3(b), we plotted different rate regions for the network in Fig. 3(a), which has min-cut capacities $M_{\{1\}} = 2$, $M_{\{2\}} = 3$ and $M_{\{1,2\}} = 3$. In particular, the region contained in the solid curve is the unsecure capacity region (given by (9) in Lemma 2), the region inside the dashed curve is the secure capacity region (given by (2) in Theorem 2) and the region contained inside the dotted line is the secure rate region that can be achieved by the two-phase scheme (given by (6) in Theorem 3). From Fig. 3(b), we indeed observe that the rate region achieved by the two-phase scheme is contained inside the secure capacity region.

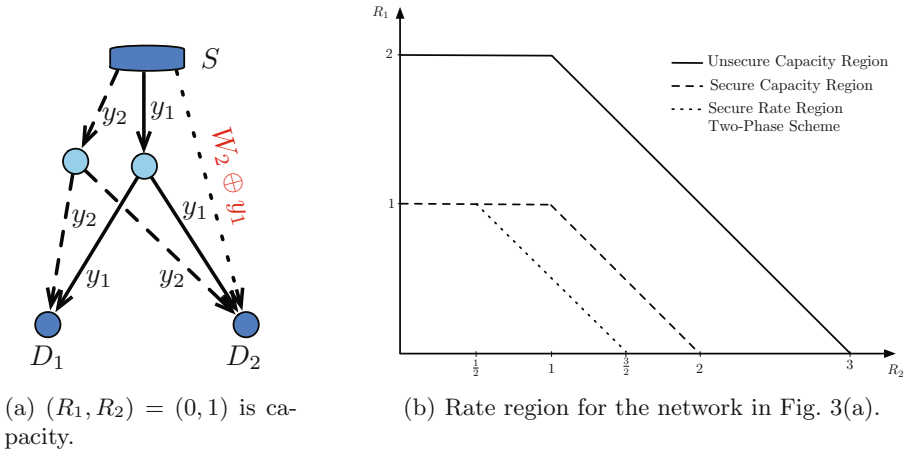


Fig. 3. Network example for which the two-phase scheme is not optimal.

4.4 Summary

In this paper, we analyzed wireline noiseless networks where a single source would like to convey independent messages to different destinations in the presence of a passive external eavesdropper, who can wiretap any k edges of her choice.

We first derived an outer bound on the capacity region that holds for any number of destinations and then showed that this bound is indeed tight for the case of two destinations. To the best of our knowledge, this is the first secure capacity result for a general network where multiple unicast sessions take place simultaneously (i.e., single source and two destinations). We also showed that this secure capacity result, different from the insecure counterpart, is irreversible. Finally, we have proposed a secure two-phase transmission scheme for general number of destinations and computed its achievable rate region. An appealing feature of this scheme is that, even though it does not achieve the secure capacity region, it can be implemented in polynomial time and it provides a performance guarantee on the secure achievable rate region as a function of any rate tuple that is achievable in the absence of the eavesdropper Eve.

Appendix A

For completeness, we here report the proof of the result in Lemma 1, which is a direct consequence of [10, Theorem 1]. In particular, this result shows that any graph \mathcal{G} with single source and $m = 2$ destinations is separable. The graph \mathcal{G} has min-cut capacity $M_{\{i\}}, i \in [1 : 2]$, towards destination D_i and min-cut capacity $M_{\{1,2\}}$ towards $\{D_1, D_2\}$, from which $M_{\{i\}}, i \in [1 : 2]$, and $M_{\{1,2\}}^*$ can be computed by using the expressions in (3). We represent these min-cut capacities by the triple

$$(M_{\{1\}}, M_{\{2\}}, M_{\{1,2\}}) = (M_{\{1\}}^* + M_{\{1,2\}}^*, M_{\{2\}}^* + M_{\{1,2\}}^*, M_{\{1\}}^* + M_{\{2\}}^* + M_{\{1,2\}}^*) ,$$

where the equality follows by using (3). We now prove Lemma 1 in two steps. We first show that the graph \mathcal{G} can be separated into two graphs: \mathcal{G}_a with min-cut capacities $(M_{\{1\}}^*, 0, M_{\{1\}}^*)$ and \mathcal{G}_b with min-cut capacities

$$(M_{\{1,2\}}^*, M_{\{2\}}^* + M_{\{1,2\}}^*, M_{\{2\}}^* + M_{\{1,2\}}^*) .$$

Then, by applying the same principle we further separate the graph \mathcal{G}_b into two graphs: \mathcal{G}_c with min-cut capacities $(0, M_{\{2\}}^*, M_{\{2\}}^*)$ and \mathcal{G}_d with min-cut capacities $(M_{\{1,2\}}^*, M_{\{1,2\}}^*, M_{\{1,2\}}^*)$. This would complete the proof of Lemma 1.

We now prove that we can separate the graph \mathcal{G} into the two graphs \mathcal{G}_a and \mathcal{G}_b . Towards this end, from the original graph \mathcal{G} , we create a new directed acyclic graph \mathcal{G}' where a new node D' is connected to D_1 through an edge of capacity $M_{\{1\}}^* + M_{\{1,2\}}^*$ and to D_2 through an edge of capacity $M_{\{2\}}^*$. By following similar steps as in the proof of the direct part (achievability) of Lemma 2 (see Appendix B), it is not difficult to see that in \mathcal{G}' the min-cut capacity between S and D' is $M_{\{1\}}^* + M_{\{1,2\}}^* + M_{\{2\}}^* = M_{\{1,2\}}^*$, where the equality follows from (3b). From the max-flow min-cut theorem, we can find $M_{\{1,2\}}^*$ edge-disjoint paths from S to D' ; we color the edges in these paths *green*. We can also find $M_{\{2\}}^*$ edge-disjoint

paths from S to D_2 ; we color the edges in these paths *red*. Notice that, at the end of this process, some of the edges can have both *green* and *red* colors. We also highlight that:

- Out of the $M_{\{1,2\}}$ *green* paths from S to D' , $M_{\{1\}}^* + M_{\{1,2\}}^*$ paths flow through D_1 and $M_{\{2\}}^*$ flow through D_2 .
- If a path is exclusively *green*, it flows through D_1 since otherwise, in addition to the $M_{\{2\}}$ *red* edge-disjoint paths from S to D_2 , we would have also this path and thereby violate the min-cut capacity constraint to D_2 .

The second observation above implies that, if there are $M_{\{1\}}^*$ exclusively *green* paths, then we can separate the graph \mathcal{G} into two graphs: \mathcal{G}'_a that contains all these $M_{\{1\}}^*$ exclusively *green* paths and \mathcal{G}'_b that contains all the edges of \mathcal{G} that are not in \mathcal{G}'_a . Given this, by simply removing the node D' and its incoming edges, we get \mathcal{G}_a and \mathcal{G}_b . We now show how we can obtain these $M_{\{1\}}^*$ exclusively *green* paths. Towards this end, we denote with \mathcal{P} the set of all *green* paths from S to D' (notice that these paths might have also some *red* edges). Then, until there exists a path $p \in \mathcal{P}$ such that either it is not exclusively *green* or it does not start with an edge that is both *red* and *green*, we apply the two following steps:

1. Let e be the first edge in p , which is both *green* and *red* and denote with g the *red* path from S to D_2 that contains the edge e . Recall that, since the $M_{\{2\}}$ *red* paths are edge-disjoint, there is only one *red* path g passing through e . We split the path p into two parts as $p_1 - e - p_2$ and similarly we split the path g into $g_1 - e - g_2$.
2. We add the *red* color to p_1 (that before was all *green*) and we remove the *red* color from g_1 , i.e., now each edge in g_1 is either *green* or it does not have any color. Note that in this way we replace the *red* path $g_1 - e - g_2$ with $p_1 - e - g_2$ from source S to D_2 , which is also disjoint from the rest of $M_{\{2\}} - 1$ *red* paths.

We note that this process will stop only when all the $M_{\{1,2\}}$ paths from S to D' are either exclusively *green* or start with an edge that is both *red* and *green*. We also note that, since we did not remove any edge, clearly we also did not change any min-cut capacity during this process. Since initially there were $M_{\{2\}}$ *red* edges coming out of S and, in the process of the algorithm, we replaced one *red* by another *red*, then the number of *red* edges outgoing from S still remains the same. Thus, among the $M_{\{1,2\}}$ paths from S to D' , only at most $M_{\{2\}}$ paths start with an edge that is both *green* and *red* and therefore, by using (3), at least $M_{\{1\}}^*$ are exclusively *green* paths. This proves that the original graph \mathcal{G} can be separated into the two graphs \mathcal{G}_a and \mathcal{G}_b . By using similar arguments, one can then show that the graph \mathcal{G}_b can be separated into the two graphs \mathcal{G}_c and \mathcal{G}_d . This concludes the proof of Lemma 1.

Appendix B

We here give the proof of Lemma 2 (originally proved in [7, Theorem 9]). In particular, we first prove the converse (i.e., the rate region in (9) is an outer

bound) and then the direct part (i.e., the rate region in (9) is achievable) of Lemma 2.

Outer Bound: Let $\mathcal{E}_{\mathcal{A}} \subseteq \mathcal{E}$ be a min-cut between the source S and $D_{\mathcal{A}}$ and define $\mathcal{I}(D_{\mathcal{A}}) := \bigcup_{i \in \mathcal{A}} \mathcal{I}(D_i)$. Then, for any $\mathcal{A} \subseteq [1 : m]$ we have,

$$\begin{aligned} nR_{\mathcal{A}} &= H(W_{\mathcal{A}}) \stackrel{(a)}{=} H(W_{\mathcal{A}}) - H(W_{\mathcal{A}}|X_{\mathcal{I}(D_{\mathcal{A}})}^n) \\ &\stackrel{(b)}{=} H(W_{\mathcal{A}}^n) - H(W_{\mathcal{A}}^n|X_{\mathcal{E}_{\mathcal{A}}}^n) \\ &= I(W_{\mathcal{A}}^n; X_{\mathcal{E}_{\mathcal{A}}}^n) \\ &\stackrel{(c)}{\leq} H(X_{\mathcal{E}_{\mathcal{A}}}^n) \\ &\stackrel{(d)}{\leq} nM_{\mathcal{A}} \text{ ,} \end{aligned}$$

where $W_{\mathcal{A}} = \{W_i, i \in \mathcal{A}\}$ and where: (i) the equality in (a) follows because of the decodability constraint; (ii) the equality in (b) follows because $X_{\mathcal{I}(D_{\mathcal{A}})}^n$ is a deterministic function of $X_{\mathcal{E}_{\mathcal{A}}}^n$; (iii) the inequality in (c) follows since the entropy of a discrete random variable is a non-negative quantity; (iv) finally, the inequality in (d) follows since each link has unit capacity and since $|\mathcal{E}_{\mathcal{A}}| = M_{\mathcal{A}}$. By dividing both sides of the above inequality by n we obtain that $R_{\mathcal{A}}$ in (9) is an outer bound on the unsecure capacity region of the multiple unicast network with single source and m destinations.

Achievability: Assume that a rate m -tuple (R_1, R_2, \dots, R_m) satisfies the constraint in (9). We now prove that this m -tuple is achievable. Towards this end, from the original graph \mathcal{G} , we create a new directed acyclic graph \mathcal{G}' where a new node D' is connected to each $D_i, i \in [1 : m]$, through an edge \mathcal{E}'_i of capacity R_i . It is not difficult to see that in \mathcal{G}' , the min-cut capacity between S and D' is $\sum_{i=1}^m R_i$. This can be explained as follows. Suppose that the min-cut from S to D' , in addition to a subset of \mathcal{E} (i.e., the set of edges in the original \mathcal{G}), also contains some edges $\mathcal{E}'_{\mathcal{J}}$, with $\mathcal{J} \subseteq [1 : m]$. This clearly implies that the subset of edges from \mathcal{E} should form a cut between source S and $D_{[1:m] \setminus \mathcal{J}}$, otherwise we would not have a cut between S and D' . Thus, the min-cut has a capacity of at least $\sum_{i \in \mathcal{J}} R_i + M_{\{D_{[1:m] \setminus \mathcal{J}}\}}$ and, since $\sum_{i \in [1:m] \setminus \mathcal{J}} R_i \leq M_{\{D_{[1:m] \setminus \mathcal{J}}\}}$ (this follows from the outer bound proved above), the min-cut has a capacity of at least $\sum_{i=1}^m R_i$. Then, since the set $\mathcal{E}'_{[1:m]}$ is a cut of capacity $\sum_i^m R_i$, it follows that the min-cut has a capacity of at most $\sum_i^m R_i$. This implies that the min-cut capacity between S and D' in \mathcal{G}' is $\sum_{i=1}^m R_i$. With this, the achievability of the rate m -tuple (R_1, R_2, \dots, R_m) that satisfies the constraint in (9) directly follows from the max-flow min-cut theorem. Indeed, since one can communicate a total information of $\sum_i^m R_i$ from S to D' in \mathcal{G}' , then this is possible only if an amount

R_i of information flows through $D_i, i \in [1 : m]$, in \mathcal{G} . This concludes the proof of Lemma 2. Notice that in order to transmit $\sum_{i=1}^m R_i$ message packets from S to D' (single unicast session) network coding is not needed. Thus, there is no need of coding operations to characterize the capacity region of a network with single source and multiple destinations.

References

1. Cai, N., Yeung, R.W.: Secure network coding. In: Proceedings IEEE International Symposium on Information Theory (ISIT), p. 323, July 2002
2. Koetter, R., Effros, M., Ho, T.: Network codes as codes on graphs. In: Conference on Information Sciences and Systems (CISS) (2004)
3. Riis, S.: Reversible and irreversible information networks. *IEEE Trans. Inf. Theor.* **53**(11), 4339–4349 (2007)
4. Ahlswede, R., Cai, N., Li, S.Y.R., Yeung, R.W.: Network information flow. *IEEE Trans. Inf. Theor.* **46**(4), 1204–1216 (2000)
5. Li, S.Y.R., Yeung, R.W., Cai, N.: Linear network coding. *IEEE Trans. Inf. Theor.* **49**(2), 371–381 (2003)
6. Jaggi, S., Sanders, P., Chou, P.A., Effros, M., Egner, S., Jain, K., Tolhuizen, L.M.G.M.: Polynomial time algorithms for multicast network code construction. *IEEE Trans. Inf. Theor.* **51**(6), 1973–1982 (2005)
7. Koetter, R., Medard, M.: An algebraic approach to network coding. *IEEE/ACM Trans. Netw.* **11**(5), 782–795 (2003)
8. Kamath, S.U., Tse, D.N.C., Anantharam, V.: Generalized network sharing outer bound and the two-unicast problem. In: International Symposium on Networking Coding (NetCod), pp. 1–6, July 2011
9. Kamath, S., Tse, D.N.C., Wang, C.C.: Two-unicast is hard. In: IEEE International Symposium on Information Theory (ISIT), pp. 2147–2151, June 2014
10. Ramamoorthy, A., Wesel, R.D.: The single source two terminal network with network coding. [arXiv:0908.2847](https://arxiv.org/abs/0908.2847), August 2009
11. Cui, T., Ho, T., Kliever, J.: On secure network coding with nonuniform or restricted wiretap sets. *IEEE Trans. Inf. Theor.* **59**(1), 166–176 (2013)
12. Agarwal, G.K., Cardone, M., Fragouli, C.: On secure network coding for two unicast sessions: studying butterflies. In: IEEE Globecom Workshops (GC Wkshps), pp. 1–6, December 2016
13. Agarwal, G.K., Cardone, M., Fragouli, C.: Coding across unicast sessions can increase the secure message capacity. In: IEEE International Symposium on Information Theory (ISIT), pp. 2134–2138, July 2016

Rényi Resolvability and Its Applications to the Wiretap Channel

Lei Yu^{1(✉)} and Vincent Y. F. Tan^{1,2}

¹ Department of Electrical and Computer Engineering,
National University of Singapore, Singapore, Singapore
{lei.yu,vtan}@nus.edu.sg

² Department of Mathematics, National University of Singapore,
Singapore, Singapore

Abstract. The conventional channel resolvability problem refers to the determination of the minimum rate needed for an input process to approximate the output distribution of a channel in either the total variation distance or the relative entropy. In this paper, we use the (normalized or unnormalized) Rényi divergence (with the Rényi parameter in $[0,2]$) to measure the level of approximation. We also provide asymptotic expressions for normalized Rényi divergence when the Rényi parameter is larger than or equal to 1 as well as (lower and upper) bounds for the case when the same parameter is smaller than 1. We characterize the minimum rate needed to ensure that the Rényi resolvability vanishes asymptotically. The optimal rates are the same for both the normalized and unnormalized cases. In addition, the minimum rate when the Rényi parameter no larger than 1 equals the minimum mutual information over all input distributions that induce the target output distribution similarly to the traditional case. When the Rényi parameter is larger than 1 the minimum rate is, in general, larger than the mutual information. We apply these results to the wiretap channel, and completely characterize the optimal tradeoff between the rates of the secret and non-secret messages when the leakage measure is given by the (unnormalized) Rényi divergence (which is a generalization of *effective secrecy*). This tradeoff differs from the conventional setting when the leakage is measured by the traditional mutual information.

1 Introduction

How much information is needed to simulate a random process through a given channel so that it mimics a target output distribution? This is so-called “channel resolvability problem”, first studied by Han and Verdú [1]. In [1], the total variation (TV) distance and the normalized Kullback-Leibler (KL) divergence were used to measure the level of approximation. The resolvability problem with *unnormalized* KL divergence measure was studied by Hayashi [2,3]. In [1–3] it was shown that in memoryless case the minimum rates of randomness needed for simulating a channel output under measure of TV, normalized KL divergence, or unnormalized KL divergence are the same, and all equal to the minimum mutual

information over all input distributions that induce the target output distribution. Recently, Liu, Cuff, and Verdú [4] extended the theory of resolvability by considering E_γ metric with $\gamma \geq 1$ to measure the level of approximation. The E_γ metric reduces to the TV distance when $\gamma = 1$, but it is weaker than the TV distance when $\gamma > 1$. Hence, the E_γ metric generalizes the TV distance by *weakening* the approximation measure. In contrast, we generalize the channel resolvability problem by *strengthening* the unnormalized KL divergence metric and considering a continuum of secrecy measures indexed by the Rényi parameter.

The channel resolvability problem is closely related to the common information (or distributed sources simulation) problem, which was first studied by Wyner [5]. For the achievability part, both problems rely on the so-called soft-covering lemmas [6]. The channel resolvability or common information problems have several interesting applications—including secrecy, channel synthesis, and source coding. For example, in [7] it was used to study the performance of a wiretap channel system under different secrecy measures. In [8] it was used to study the reliability and secrecy exponents of a wiretap channel with cost constraints. In [9] it was used to study the exact secrecy and reliability exponents for a wiretap channel. In [10], Hou and Kramer used ideas from channel resolvability to study the *effective secrecy capacity* of wiretap channels. This work is contrasted to the present work in greater detail in Sect. 3.

In contrast to the aforementioned works, in this paper, we use the (normalized or unnormalized) Rényi divergence to measure the level of approximation between the simulated and target output distributions. Our work is partly motivated by Shikata [11] who quantified lengths of secret keys in terms of Rényi entropies of general orders and Bai *et al.* [12] who showed that the Rényi divergence is particularly suited for simplifying some security proofs. Our contributions are threefold:

1. We provide asymptotic expressions for the Rényi divergence between the simulated and target output distributions—we term this the *Rényi resolvability*. We distinguish between the case when the Rényi parameter is ≥ 1 —in which case we have a tight expression—and the case when the same parameter is < 1 —in which case we only have bounds (which are tight in some regime).
2. We characterize the minimum rate needed to guarantee that the (normalized or unnormalized) Rényi resolvability vanishes asymptotically. Interestingly, these rates are the same regardless of whether we employ the normalized or unnormalized Rényi divergences. The optimal rate when the Rényi parameter is ≤ 1 is just equal to the minimum mutual information over all input distribution that induce target output distribution. This is similar to the traditional case [1–3]. In contrast if the Rényi parameter is > 1 , the optimal rate is, in general, larger than the minimum mutual information.
3. As a concrete application of the above mathematical results, we consider the wiretap channel and completely characterize the optimal tradeoff between the rates of the secret and non-secret messages when the leakage is measured by the unnormalized Rényi divergence. This part of work can be seen

as a generalization of the *effective secrecy capacity* studied by Hou and Kramer [10]. Note that different from Csiszár and Körner's work (with secrecy measured by the mutual information) [13], the optimal rates tradeoff provided by us are achieved by a single-layered code. Hence, it has a different expression from the one given by in [13]. See Remark 7.

It is also worth noting that our work is partly motivated by the work of Hayashi and Tan [14, 15]. In their work, the Rényi divergence was used to measure the level of approximation of a distribution induced by a *hash function*, typically used for source compression; in our work, it is used to measure the level of approximation of an input process that is sent through a *channel*. Hence our work can be considered as a counterpart of theirs, just as the *channel coding* is a counterpart of the *source hashing*.

1.1 Notation

In this paper, we use $P_X(x)$ to denote the probability distribution of a random variable X , which is also shortly denoted as $P_X(x)$ or $P(x)$ (when the random variable X is clear from the context). We also use P_X, \tilde{P}_X and Q_X to denote various probability distributions with alphabet \mathcal{X} . All alphabets considered in the sequel are finite.

Fix distributions $P_X, Q_X \in \mathcal{P}(\mathcal{X})$ (the set of probability mass functions on \mathcal{X}). Then the *relative entropy* and the *Rényi divergence of order $1+s$* are respectively defined as

$$D(P_X \| Q_X) := \sum_{x \in \mathcal{X}} P_X(x) \log \frac{P_X(x)}{Q_X(x)} \quad (1)$$

$$D_{1+s}(P_X \| Q_X) := \frac{1}{s} \log \sum_{x \in \mathcal{X}} P_X(x)^{1+s} Q_X(x)^{-s}, \quad (2)$$

where throughout, \log is to the natural base e and $s \geq -1$. It is known that $\lim_{s \rightarrow 0} D_{1+s}(P_X \| Q_X) = D(P_X \| Q_X)$ so a special case of the Rényi divergence is the usual relative entropy.

Given P_X and $P_{Y|X}$, we write $[P_{Y|X} \circ P_X](y) := \sum_x P_{Y|X}(y|x) P_X(x)$.

1.2 Problem Formulation

In this paper, we consider the channel resolvability problem illustrated in Fig. 1. Given a random transformation $P_{Y|X}$ and a target distribution Q_Y , we wish to minimize the alphabet size of a message M_n that is uniformly distributed over¹ $\mathcal{M}_n := \{1, \dots, e^{nR}\}$ (R is a positive number known as the *rate*), such that given common randomness U_n , the output distribution

$$P_{Y^n|U_n}(y^n|u_n) := \frac{1}{|\mathcal{M}_n|} \sum_{m \in \mathcal{M}_n} \prod_{i=1}^n P_{Y|X}(y_i | f_{u_n, i}(m)) \quad (3)$$

¹ For simplicity, we assume that e^{nR} and similar expressions are integers.

forms a good approximation to the product distribution $Q_{Y^n} := Q_Y^n$. Here U_n is a random variable independent of the message M_n . If we set $U_n = \{X^n(m)\}_{m \in \mathcal{M}_n}$ with $X^n(m) \sim P_{X^n}$, $m \in \mathcal{M}_n$, and set $f_{U_n}(m) = X^n(m)$, then the random mapping is known as a *conventional random code*. If the input distribution is i.i.d., i.e., $P_{X^n} = P_X^n$, then it is known as an *i.i.d. random code*.

In contrast to previous works on the channel resolvability problem [1], here we employ the Rényi divergence

$$D_{1+s}(P_{Y^n U_n} \| Q_{Y^n} P_{U_n}) \quad (4)$$

to measure the discrepancy between P_{Y^n} and Q_{Y^n} .

Observe that

$$\begin{aligned} & e^{sD_{1+s}(P_{Y^n U_n} \| Q_{Y^n} P_{U_n})} \\ &= \mathbb{E}_{U_n} \left[\sum_{y^n} \sum_m P(m) P(y^n | f_{U_n}(m)) \left(\frac{\sum_m P(m) P(y^n | f_{U_n}(m))}{Q(y^n)} \right)^s \right]. \quad (5) \end{aligned}$$

Hence to guarantee that $D_{1+s}(P_{Y^n U_n} \| Q_{Y^n} P_{U_n})$ finite for $s \geq 0$, we assume $P_{Y|X=x} \ll Q_Y$ for all $x \in \mathcal{X}$; otherwise, we can remove all the values x such that $P_{Y|X=x} \not\ll Q_Y$ from \mathcal{X} . However, it is worth noting that we do not need to do so for $-1 \leq s < 0$, since $D_{1+s}(P_{Y^n U_n} \| Q_{Y^n} P_{U_n})$ is always finite regardless of whether $P_{Y|X=x} \ll Q_Y$ for all $x \in \mathcal{X}$ or $P_{Y|X=x} \not\ll Q_Y$ for some $x \in \mathcal{X}$.

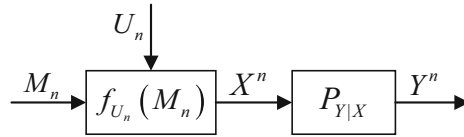


Fig. 1. Channel resolvability problem: U_n is independent of the message $M_n \in \mathcal{M}_n$, and f_{U_n} is a random function (induced by U_n).

2 Rényi Resolvability

2.1 Asymptotic Expressions

For the channel resolvability problem, the asymptotics of the Rényi divergence is characterized by single-letter expressions. We have an exact/tight result when the Rényi parameter $\in [1, 2]$ and upper and lower bounds when the Rényi parameter $\in (0, 1)$. This result is proved in Appendix B.

Theorem 1 (Asymptotics of Rényi Resolvability). *For any $s \in [0, 1]$, we have*

$$\begin{aligned} & \lim_{n \rightarrow \infty} \frac{1}{n} \inf_{f_{U_n}} D_{1+s}(P_{Y^n U_n} \| Q_{Y^n} P_{U_n}) \\ &= \min_{\tilde{P}_X} \max \left\{ \sum_x \tilde{P}_X(x) D_{1+s}(P_{Y|X}(\cdot|x) \| Q_Y) - R, \right. \\ & \qquad \qquad \qquad \left. \max_{\tilde{P}_{Y|X}} \eta_{1+s}(P_{Y|X}, Q_Y, \tilde{P}_X, \tilde{P}_{Y|X}) \right\}, \end{aligned} \tag{6}$$

where

$$\eta_{1+s}(P_{Y|X}, Q_Y, \tilde{P}_X, \tilde{P}_{Y|X}) := \left(-\frac{1}{s} - 1\right) D(\tilde{P}_{Y|X} \| P_{Y|X} | \tilde{P}_X) + D(\tilde{P}_Y \| Q_Y). \tag{7}$$

For any $s \in (0, 1)$, we have

$$\Gamma_{1-s}^{\text{LB}}(P_{Y|X}, Q_Y, R) \leq \liminf_{n \rightarrow \infty} \frac{1}{n} \inf_{f_{U_n}} D_{1-s}(P_{Y^n U_n} \| Q_{Y^n} P_{U_n}) \tag{8}$$

$$\leq \limsup_{n \rightarrow \infty} \frac{1}{n} \inf_{f_{U_n}} D_{1-s}(P_{Y^n U_n} \| Q_{Y^n} P_{U_n}) \tag{9}$$

$$\leq \Gamma_{1-s}^{\text{UB}}(P_{Y|X}, Q_Y, R), \tag{10}$$

where

$$\begin{aligned} & \Gamma_{1-s}^{\text{LB}}(P_{Y|X}, Q_Y, R) \\ &:= \min_{\tilde{P}_X, \tilde{P}_{Y|X}} \max \left\{ \left(\frac{1}{s} - 1\right) D(\tilde{P}_{Y|X} \| P_{Y|X} | \tilde{P}_X) + D(\tilde{P}_{Y|X} \| Q_Y | \tilde{P}_X) - R, \right. \\ & \qquad \qquad \qquad \left. \left(\frac{1}{s} - 1\right) D(\tilde{P}_{Y|X} \| P_{Y|X} | \tilde{P}_X) + D(\tilde{P}_Y \| Q_Y) \right\}, \end{aligned} \tag{11}$$

$$\begin{aligned} & \Gamma_{1-s}^{\text{UB}}(P_{Y|X}, Q_Y, R) \\ &:= \min_{\tilde{P}_X, \tilde{P}_{Y|X}} \max \left\{ \left(\frac{1}{s} - 1\right) D(\tilde{P}_{Y|X} \| P_{Y|X} | \tilde{P}_X) + D(\tilde{P}_{Y|X} \| Q_Y | \tilde{P}_X) - R, \right. \\ & \qquad \frac{1}{s} D(\tilde{P}_{Y|X} \| P_{Y|X} | \tilde{P}_X) + D(\tilde{P}_Y \| Q_Y) \\ & \qquad \qquad \qquad \left. - \min_{\hat{P}_{Y|X}: \hat{P}_{Y|X} \circ \tilde{P}_X = \tilde{P}_{Y|X} \circ \tilde{P}_X} D(\hat{P}_{Y|X} \| P_{Y|X} | \tilde{P}_X) \right\}. \end{aligned} \tag{12}$$

We also have

$$\lim_{n \rightarrow \infty} \frac{1}{n} \inf_{f_{U_n}} D_0(P_{Y^n U_n} \| Q_{Y^n} P_{U_n}) = 0. \tag{13}$$

Furthermore, the infima in (6) and $\Gamma_{1-s}^{\text{UB}}(P_{Y|X}, Q_Y, R)$ are achieved by a sequence of conventional random codes.

Remark 1. The expression in (6) for $s \in [0, 1]$ and $\Gamma_{1-s}^{\text{LB}}(P_{Y|X}, Q_Y, R)$ or $\Gamma_{1-s}^{\text{UB}}(P_{Y|X}, Q_Y, R)$ for $s \in (-1, 0)$ may appear to be inconsistent; however, this is not true. It can be easily shown that

$$\begin{aligned} & \sum_x \tilde{P}_X(x) D_{1+s}(P_{Y|X}(\cdot|x) \| Q_Y) \\ &= \max_{\tilde{P}_{Y|X}} \left\{ \left(-\frac{1}{s} - 1\right) D(\tilde{P}_{Y|X} \| P_{Y|X} | \tilde{P}_X) + D(\tilde{P}_{Y|X} \| Q_Y | \tilde{P}_X) \right\}. \end{aligned} \quad (14)$$

Hence we can rewrite (6) as

$$\begin{aligned} & \lim_{n \rightarrow \infty} \frac{1}{n} \inf_{f_{U_n}} D_{1+s}(P_{Y^n U_n} \| Q_{Y^n} P_{U_n}) \\ &= \min_{\tilde{P}_X} \max_{\tilde{P}_{Y|X}} \max \left\{ \left(-\frac{1}{s} - 1\right) D(\tilde{P}_{Y|X} \| P_{Y|X} | \tilde{P}_X) + D(\tilde{P}_{Y|X} \| Q_Y | \tilde{P}_X) - R, \right. \\ & \quad \left. \left(-\frac{1}{s} - 1\right) D(\tilde{P}_{Y|X} \| P_{Y|X} | \tilde{P}_X) + D(\tilde{P}_Y \| Q_Y) \right\}. \end{aligned} \quad (15)$$

In other words, (6) for $s \in [0, 1]$ is consistent with $\Gamma_{1-s}^{\text{LB}}(P_{Y|X}, Q_Y, R)$ for $s \in (-1, 0)$.

Note that $\Gamma_{1-s}^{\text{UB}}(P_{Y|X}, Q_Y, R)$ and $\Gamma_{1-s}^{\text{LB}}(P_{Y|X}, Q_Y, R)$ differ only in the second term in the maximization. Moreover, when R is large enough, they are both equal to zero; see Theorem 2 in the next subsection.

We numerically calculate the asymptotics of the normalized Rényi resolvability for binary symmetric channel (BSC) $Y = X \oplus V, V \sim \text{Bern}(0.2)$ and $Q_Y = \text{Bern}(0.5)$, and display the result in Fig. 2. From this figure, we observe that the normalized Rényi resolvability decays as R increases, and finally vanishes for large enough R . Moreover, the rate at which the normalized Rényi resolvability transitions from a positive quantity to zero increases in R for the Rényi parameter $1 + s \in [1, 2]$, and remains the same when $1 + s \in (0, 1]$. A rigorous statement of this point will be provided in the next subsection.

2.2 Optimal Rates for Vanishing Rényi Resolvability

Normalized Rényi Resolvability. Now we compute the minimum rate R of the input process $\{X^n(m) : m \in \mathcal{M}_n\}$ to ensure that the Rényi resolvability $\frac{1}{n} D_{1+s}(P_{Y^n U_n} \| Q_{Y^n} P_{U_n})$ vanishes. We assume that

$$\mathcal{P}(P_{Y|X}, Q_Y) := \{P_X : P_{Y|X} \circ P_X = Q_Y\} \neq \emptyset. \quad (16)$$

Otherwise, there does not exist a code such that $\frac{1}{n} D_{1+s}(P_{Y^n U_n} \| Q_{Y^n} P_{U_n})$ vanishes. By Theorem 1 we easily obtain the following result.

Theorem 2 (Normalized Rényi Resolvability). *For $s \in [-1, 1]$, we have*

$$\inf \left\{ R : \frac{1}{n} \inf_{f_{U_n}} D_{1+s}(P_{Y^n U_n} \| Q_{Y^n} P_{U_n}) \rightarrow 0 \right\} = R_{1+s}(P_{Y|X}, Q_Y), \quad (17)$$

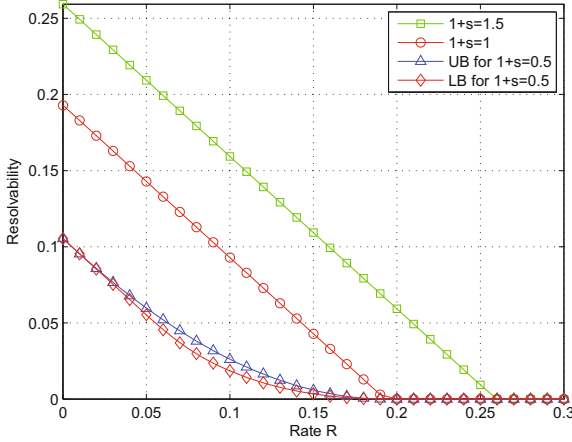


Fig. 2. Illustration of the Rényi resolvability measure $\frac{1}{n} \inf_{f_{U_n}} D_{1+s}(P_{Y^n U_n} \| Q_{Y^n} P_{U_n})$ for $s \in [0, 1]$ in (6) and the upper $\Gamma_{1-s}^{UB}(P_{Y|X}, Q_Y, R)$ and lower bounds $\Gamma_{1-s}^{LB}(P_{Y|X}, Q_Y, R)$ for $s \in (-1, 0)$ in (11) and (12), for the BSC $Y = X \oplus V, V \sim \text{Bern}(0.2)$ and the target distribution $Q_Y = \text{Bern}(0.5)$.

where

$$R_{1+s}(P_{Y|X}, Q_Y) := \begin{cases} \min_{P_X \in \mathcal{P}(P_{Y|X}, Q_Y)} \sum_x P_X(x) D_{1+s}(P_{Y|X}(\cdot|x) \| Q_Y) & s \in (0, 1] \\ \min_{P_X \in \mathcal{P}(P_{Y|X}, Q_Y)} I(X; Y) & s \in (-1, 0] \\ 0 & s = -1 \end{cases} \quad (18)$$

Remark 2. Since $\mathcal{P}(P_{Y|X}, Q_Y)$ is nonempty, $R_{1+s}(P_{Y|X}, Q_Y)$ is finite. Hence it can be shown $\lim_{s \downarrow 0} R_{1+s}(P_{Y|X}, Q_Y) = R_1(P_{Y|X}, Q_Y)$ (by using the continuity of Rényi divergence [16]). Hence $R_{1+s}(P_{Y|X}, Q_Y)$ is continuous in s for $s \in (-1, 1]$. See the bottom subfigure of Fig. 3.

Remark 3. This result for the case $s = 0$ (i.e., the KL divergence case) was first shown by Han and Verdú [1]. Hence our result is the generalization of theirs to the Rényi divergence of all orders in $[0, 2]$.

Remark 4. The first clause in (18) is the minimization of an expectation of Rényi divergences $\sum_x P_X(x) D_{1+s}(P_{Y|X}(\cdot|x) \| Q_Y)$ but it is *not* (and in general smaller than) the conventional conditional Rényi divergence $D_{1+s}(P_{XY} \| P_X Q_Y)$ (see Verdú [17] or Fong and Tan [18]). An optimal i.i.d. code can achieve a rate equal to the minimization of conventional conditional Rényi divergence $D_{1+s}(P_{XY} \| P_X Q_Y)$ [19, Thm. 14], while an optimal *constant composition code* (a code with channel input distributed according to the target distribution Q_{Y^n} but truncated to an appropriate typical set) can

achieve a better (smaller) rate equal to the first clause in (18). This shows that the expectation of Rényi divergences also admits an operational interpretation as the minimum rate needed to drive the Rényi divergence to zero when its parameter is ≥ 1 .

The result in Theorem 2 for the BSC $Y = X \oplus V, V \sim \text{Bern}(p)$ and $Q_Y = \text{Bern}(0.5)$ is illustrated in Fig. 3. For this case,

$$R_{1+s}(P_{Y|X}, Q_Y) = \begin{cases} \frac{1}{s} \log(p^{1+s}2^s + \bar{p}^{1+s}2^s) & s \in (0, 1] \\ 1 - H_2(p) & s = (-1, 0] \\ 0 & s = -1 \end{cases}. \quad (19)$$

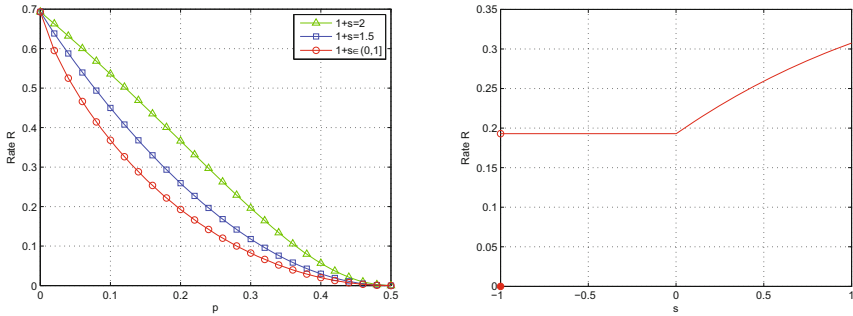


Fig. 3. Illustration of the optimal rates for vanishing resolvability in (17) for the BSC $Y = X \oplus V, V \sim \text{Bern}(p)$ and $Q_Y = \text{Bern}(0.5)$. For the right subfigure, $p = 0.2$.

Unnormalized Rényi Resolvability. For the unnormalized case, we can prove the following theorem. Due to space limitation, the proof is provided in [20].

Theorem 3 (*Unnormalized Rényi Resolvability*). For $s \in [-1, 1]$, we have

$$\inf \left\{ R : \inf_{f_{U_n}} D_{1+s}(P_{Y^n U_n} \| Q_{Y^n} P_{U_n}) \rightarrow 0 \right\} = R_{1+s}(P_{Y|X}, Q_Y), \quad (20)$$

where $R_{1+s}(P_{Y|X}, Q_Y)$ is defined in (18).

Remark 5. The minimum rates needed to guarantee that the normalized or unnormalized Rényi divergence vanish asymptotically are the same.

Remark 6. The case $s = 0$ (i.e., the unnormalized KL divergence case) has been shown in other works, such as Hayashi [2, 3] and Han, Endo, and Sasaki [8], which also imply the achievability result part for $s \in (-1, 0)$ (since the approximation measure D_α for $\alpha \in (0, 1)$ is weaker than D_1). Our results for other cases (converse for $s \in [-1, 1] \setminus \{0\}$ and achievability for $s \in (0, 1)$) are new.

3 Application to the Wiretap Channel

We apply the preceding results to the wiretap channel. In [10], Hou and Kramer proposed a new security measure, termed *effective secrecy*, for wiretap channels by exploiting the unnormalized KL divergence to quantify not only (the wiretapper’s) *confusion* but also *stealth*. In this section, we generalize Hou and Kramer’s result to a generalized divergence measure—the Rényi divergence. We provide a complete characterization of the secrecy capacity region under this new and generalized leakage measure.

Consider a discrete memoryless wiretap channel $P_{YZ|X}$, and two messages (M_0, M_1) that are uniformly distributed over $\mathcal{M}_0 := \{1, \dots, e^{nR_0}\}$ and $\mathcal{M}_1 := \{1, \dots, e^{nR_1}\}$ respectively. A sender wants to transmit the pair (M_0, M_1) to a legitimate user, and, at the same time, ensure that M_1 is almost independent from the wiretapper’s observation Z^n .

Definition 1. An (n, R_0, R_1) secrecy code is defined by two stochastic mappings $P_{\mathcal{X}^n|M_0M_1} : \mathcal{M}_0 \times \mathcal{M}_1 \mapsto \mathcal{X}^n$ and $P_{\widehat{M}_0\widehat{M}_1|Y^n} : \mathcal{Y}^n \mapsto \mathcal{M}_0 \times \mathcal{M}_1$.

Given a target distribution Q_Z , we wish to maximize the alphabet size (or rate) of M_1 such that the distribution $P_{M_1Z^n}$ induced by the code is approximately equal to the target distribution $P_{M_1}Q_{Z^n}$ (with $Q_{Z^n} = Q_Z^n$) and M_1 can be decoded correctly asymptotically.

Definition 2. The tuple (R_0, R_1) is $(Q_Z, 1 + s)$ -achievable if there exists a sequence of (n, R_0, R_1) secrecy codes with induced distribution P such that

1. Error constraint:

$$\lim_{n \rightarrow \infty} \mathbb{P} \left((M_0, M_1) \neq (\widehat{M}_0, \widehat{M}_1) \right) = 0; \tag{21}$$

2. Secrecy constraint (generalized effective secrecy):

$$\lim_{n \rightarrow \infty} D_{1+s}(P_{M_1Z^n} \| P_{M_1}Q_{Z^n}) = 0. \tag{22}$$

It is worth noting that (22) is a generalized version of the notion of effective secrecy considered in [10].

Here we assume Q_Z satisfies $\mathcal{P}(P_{Z|X}, Q_Z) \neq \emptyset$ ($\mathcal{P}(P_{Z|X}, Q_Z)$ is defined by (16)); otherwise, (22) cannot be satisfied by any secrecy code.

Definition 3. The $(Q_Z, 1 + s)$ -admissible region is defined as $\mathcal{R}_{1+s}(Q_Z) := \text{Closure} \{ (R_0, R_1) : (R_0, R_1) \text{ is } (Q_Z, 1 + s) \text{-achievable} \}$.

Our secrecy metric (even when $s = 0$) is stronger than the unnormalized KL divergence $D(P_{M_1Z^n} \| P_{M_1}P_{Z^n})$ (or $I(M_1; Z^n)$) considered in [7], since

$$D(P_{M_1Z^n} \| P_{M_1}Q_{Z^n}) = I(M_1; Z^n) + D(P_{Z^n} \| Q_{Z^n}) \geq I(M_1; Z^n). \tag{23}$$

For our secrecy metric, in addition to requiring that M_1 and Z^n are approximately independent, we also require that the wiretapper’s observation Z^n is

close to the product distribution Q_Z^n . This is similar to Csiszár and Narayan's work [21, Eq. (6)] but we consider a continuum of secrecy measures indexed by $s \in [-1, 1]$.

The interpretation of our secrecy measure with $s = 0$ can be found in [10], where the authors interpreted $I(M_1; Z^n)$ in (23) as a measure of “non-confusion” and $D(P_{Z^n} \| Q_{Z^n})$ in (23) as a measure of “non-stealth”. Under this interpretation, we set Q_{Z^n} to be the distribution that the wiretapper observes if the sender is not sending useful information. Hence if the secrecy constraint (22) is satisfied then we can say that *useful* information is being transmitted in a *stealthy* way.

3.1 Main Result for Deterministic Encoder

Before solving the problem, in this subsection we consider a simpler version of the problem—namely, a system with a deterministic encoder. That is, the encoder is restricted to a deterministic (non-stochastic) function $f : \mathcal{M}_0 \times \mathcal{M}_1 \mapsto \mathcal{X}^n$ (denote the $(Q_Z, 1 + s)$ -admissible region for this case as $\mathcal{R}_{1+s}^{\text{det}}(Q_Z)$). For this problem, we have the following theorem.

Theorem 4. *For $s \in [-1, 1]$, we have*

$$\mathcal{R}_{1+s}^{\text{det}}(Q_Z) = \bigcup_{P_X \in \mathcal{P}(P_{Z|X}, Q_Z)} \left\{ (R_0, R_1) : \begin{array}{l} R_0 + R_1 \leq I(X; Y) \\ R_0 \geq \tilde{R}_{1+s}(P_X, P_{Z|X}, Q_Z) \end{array} \right\}, \quad (24)$$

where $\tilde{R}_{1+s}(P_X, P_{Z|X}, Q_Z)$ is defined as

$$\tilde{R}_{1+s}(P_X, P_{Z|X}, Q_Z) := \begin{cases} \sum_x P_X(x) D_{1+s}(P_{Z|X}(\cdot|x) \| Q_Z) & s \in (0, 1] \\ I(X; Z) & s \in (-1, 0] \\ 0 & s = -1 \end{cases}. \quad (25)$$

From Theorem 4, we observe that for the problem with deterministic encoder, the achievability of a rate pair (R_0, R_1) does not necessarily imply the achievability of a rate pair (R'_0, R'_1) such that $R'_0 \leq R_0, R'_1 \leq R_1$. This is because to meet the resolvability constraint, a certain amount of local randomness (besides the secret message M_1) at the sender is needed; this local randomness only comes from the non-secret message M_0 (since the encoder is a deterministic function of M_0, M_1). Therefore, a rate less than R_0 may not satisfy the resolvability constraint.

3.2 Main Result for Stochastic Encoder

If a stochastic encoder is allowed, we can add a virtual memoryless channel $P_{X|W}^n$ between the deterministic encoder and the channel. Then we have the following achievability result.

Theorem 5. For $s \in [-1, 1]$, we have

$$\mathcal{R}_{1+s}(Q_Z) \supseteq \bigcup_{P_{X|W}, P_W \in \mathcal{P}(P_{Z|W}, Q_Z)} \left\{ \begin{array}{l} (R_0, R_1) : R_0 + R_1 \leq I(W; Y), \\ R_0 \geq \tilde{R}'_{1+s}(P_W, P_{Z|W}, Q_Z) \end{array} \right\}, \quad (26)$$

where $\tilde{R}'_{1+s}(P_W, P_{Z|W}, Q_Z)$ is given by (25).

However, adding a memoryless channel is not optimal in general. In the following theorem, we completely characterize the admissible region, and show that adding a channel with memory between the encoder and channel is optimal. The proof of this theorem is given in Appendix C.

Theorem 6. For $s \in [-1, 1]$, we have

$$\mathcal{R}_{1+s}(Q_Z) = \bigcup_{\tilde{P}_{W|X}, \tilde{P}_X \in \mathcal{P}(P_{Z|X}, Q_Z)} \left\{ \begin{array}{l} (R_0, R_1) : R_0 + R_1 \leq I_{\tilde{P}}(W; Y) \\ R_0 \geq \tilde{R}'_{1+s}(\tilde{P}_{W|X} \tilde{P}_X, P_{Z|X}, Q_Z) \end{array} \right\} \quad (27)$$

$$= \bigcup_{\tilde{P}_{W|X}, \tilde{P}_X \in \mathcal{P}(P_{Z|X}, Q_Z)} \left\{ \begin{array}{l} (R_0, R_1) : R_0 + R_1 \leq I_{\tilde{P}}(W; Y) \\ R_1 \leq I_{\tilde{P}}(W; Y) \\ -\tilde{R}'_{1+s}(\tilde{P}_{W|X} \tilde{P}_X, P_{Z|X}, Q_Z) \end{array} \right\}, \quad (28)$$

where $\tilde{R}'_{1+s}(\tilde{P}_{W|X} \tilde{P}_X, P_{Z|X}, Q_Z)$ is given by

$$\begin{aligned} & \tilde{R}'_{1+s}(\tilde{P}_{W|X} \tilde{P}_X, P_{Z|X}, Q_Z) \\ & := \begin{cases} \max_{\tilde{P}_{Z|WX}} \left\{ -\frac{1+s}{s} D(\tilde{P}_{Z|WX} \| P_{Z|X} | \tilde{P}_{XW}) \right. \\ \quad \left. + D(\tilde{P}_{Z|W} \| Q_Z | \tilde{P}_W) \right\}, & s \in (0, 1] \\ I_{\tilde{P}}(W; Z), & s \in (-1, 0] \\ 0, & s = -1 \end{cases}. \end{aligned} \quad (29)$$

Here $I_{\tilde{P}}(W; Y)$ in (27) and (28) and $I_{\tilde{P}}(W; Z)$ in (29) are the mutual informations evaluated under the distribution $\tilde{P}_{WX} P_{YZ|X}$. Furthermore, the ranges of W in (27) and (28) may be assumed to satisfy $|\mathcal{W}| \leq |\mathcal{X}| + 1$.

Remark 7. We can define the *effective secrecy capacity* with the leakage measured by the Rényi divergence with parameter $1 + s$ and with target output distribution Q_Z as $C_{1+s}(Q_Z) := \max_{(R_0, R_1) \in \mathcal{R}_{1+s}(Q_Z)} R_1$. The special case with $s = 0$ was defined by Hou and Kramer [10], and they showed

$$C_1(Q_Z) = \max_{\tilde{P}_{W|X}, \tilde{P}_X \in \mathcal{P}(P_{Z|X}, Q_Z)} \{I_{\tilde{P}}(W; Y) - I_{\tilde{P}}(W; Z)\}. \quad (30)$$

For the general case $s \in [-1, 1]$, by Theorem 6, we have

$$C_{1+s}(Q_Z) = \max_{\tilde{P}_{W|X}, \tilde{P}_X \in \mathcal{P}(P_{Z|X}, Q_Z)} \left\{ I_{\tilde{P}}(W; Y) - \tilde{R}'_{1+s}(\tilde{P}_{W|X} \tilde{P}_X, P_{Z|X}, Q_Z) \right\}, \quad (31)$$

which has a similar form as the conventional secrecy capacity (with secrecy measured by the normalized mutual information $\frac{1}{n}I(M; Z^n)$ or unnormalized mutual information $I(M; Z^n)$) given in [2, 3, 13],

$$C_{\text{MI}} = \max_{P_{W|X} P_X} \{I(W; Y) - I(W; Z)\}. \quad (32)$$

Note that $C_{\text{MI}} \geq \max_{Q_Z} C_{1+s}(Q_Z)$ for $s \in (0, 1]$ and $C_{\text{MI}} = \max_{Q_Z} C_{1+s}(Q_Z)$ for $s \in (-1, 0]$. This is because our secrecy measure is stronger than the conventional one. Furthermore, when considering the simultaneous transmission of secret and non-secret messages, the optimal rate region [13, Cor. 2]² is

$$\mathcal{R}_{\text{MI}} = \bigcup_{\substack{P_{U|W}, P_{W|X}, P_X: \\ I(U; Y) \leq I(U; Z)}} \left\{ \begin{array}{l} (R_0, R_1) : R_0 + R_1 \leq I(W; Y), \\ R_1 \leq I(W; Y|U) - I(W; Z|U) \end{array} \right\}, \quad (33)$$

which is different from the optimal region \mathcal{R}_{1+s} given by us. Obviously $\bigcup_{Q_Z} \mathcal{R}_{1+s}(Q_Z) \subseteq \mathcal{R}_{\text{MI}}$. Csiszár and Körner [13, Cor. 2] derived the optimal region \mathcal{R}_{MI} by using a two-layered code, but for our case, a single-layered code is sufficient to achieve the optimality; a similar conclusion for the $s = 0$ case can be drawn from the results in [22]. This is because our secrecy measure requires that M_1 and Z^n are approximately independent (similarly to the conventional setting) but also requires the wiretapper's observation Z^n to approximately follow a target memoryless distribution Q_Z^n (soft-covering the space according to the target distribution). We provide an intuitive interpretation for why a two-layered code is not necessary to achieve the optimal region for our problem. For simplicity, we consider the case with the Rényi parameter equal to 1; If we apply a two-layered code to our setting then to guarantee the soft-covering property (under the TV distance measure, which is weaker than the Rényi divergence), the non-secret message for each layer has to have rates that are appropriately lower bounded as follows: $R_0^{(1)} > I(U; Z)$, $R_0^{(1)} + R_0^{(2)} > I(UW; Z)$ for some $P_{UW|X}$ and $P_X \in \mathcal{P}(P_{Z|X}, Q_Z)$ [23], where $R_0^{(1)}$ and $R_0^{(2)}$ respectively denote the transmission rate of the non-secret message for the first and second layer. On the other hand, the total rate is still constrained by $I(W; Y)$, i.e., $R_0^{(1)} + R_0^{(2)} + R_1 \leq I(W; Y)$. Hence the achievable rate pair $(R_0^{(1)} + R_0^{(2)}, R_1)$ is still in $\mathcal{R}_1(P_Z)$. This is also true for the Rényi divergence measure.

Remark 8. The *semantic-security capacity* C_{SS} (with the secrecy measure³ $\max_{m_1} D(P_{Z^n|M_1=m_1} \| Q_{Z^n}) \rightarrow 0$), studied in [24], is proven to be equal to C_{MI} .

² Note that here we refer to Corollary 2 of [13], in which the common message rate is set to zero and the R_1 and R_e there respectively correspond to the $R_0 + R_1$ and R_1 of this paper. Although the setting in Corollary 2 of [13] does not implicitly indicate the secret and non-secret parts, it is easy to show that if divide the total rate into these two parts, the admissible region does not change.

³ This measure comes from [24, Thm. 2], but is different from and stronger than the original one $\max_{P_M \in \mathcal{P}(\mathcal{M})} I(M; Z^n)$, also considered in [24]. However, both measures result in the same secrecy capacity [24].

This secrecy measure is stronger than the one considered in this paper (when the Rényi divergence parameter is equal to 1). However, in [24] Goldfeld, Cuff, and Permuter focus only on the secrecy capacity C_{SS} , i.e., the maximum transmission rate of the secret message without a constraint on non-secret message required by the legitimate user. Here we consider a more general case: the simultaneous transmission of the secret and non-secret messages. Combining (the converse part of) our result with (the achievability part of) Goldfeld, Cuff, and Permuter’s result gives a complete characterization of the admissible region of (R_0, R_1) under the secrecy constraint $\max_{m_1} D(P_{Z^n|M_1=m_1} \| Q_{Z^n}) \rightarrow 0$, which turns out to be the same as $\mathcal{R}_1(Q_Z)$ (the admissible region under the constraint $D(P_{M_1 Z^n} \| P_{M_1} Q_{Z^n}) \rightarrow 0$). Furthermore, different from [24], we also consider the Rényi divergence measure for other cases $s \in [-1, 1] \setminus \{0\}$, in addition to the relative entropy.

The result of Theorem 6 for the binary wiretap channel $Y = X \oplus V_1, V_1 \sim \text{Bern}(0.1)$ and $Z = X \oplus V_2, V_2 \sim \text{Bern}(0.3)$ with target distribution $Q_Z = \text{Bern}(0.5)$ and $s = 1$ is illustrated in Fig. 4. From the figure, we observe that different from the deterministic encoder case, for this case the achievability of a rate pair (R_0, R_1) indeed implies the achievability of a rate pair (R'_0, R'_1) such that $R'_0 \leq R_0, R'_1 \leq R_1$.

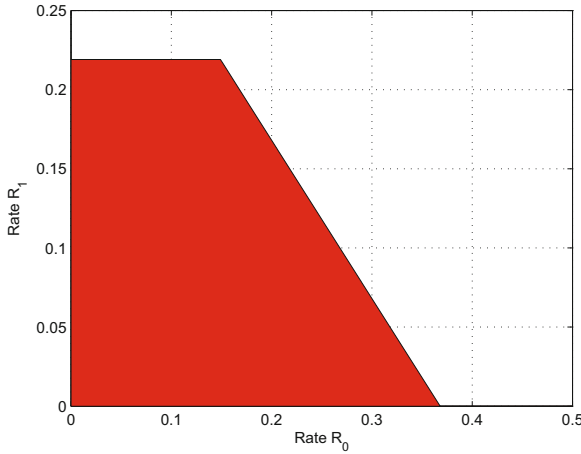


Fig. 4. Illustration of the admissible region for case of using a stochastic encoder and with Rényi parameter $1 + s = 2$ in (27) or (28) for the abovementioned channel.

4 Conclusion and Future Work

In this paper, we studied a generalized version of channel resolvability problem, in which the (normalized or unnormalized) Rényi divergence is used to measure the level of approximation. We also applied these results to the wiretap channel.

Our results generalize or extend several classical and recent results. Our resolvability results extend those by Han and Verdú [1] and by Hayashi [2, 3] as we consider Rényi divergences with orders in $[0, 2]$. Our results for the wiretap channel generalize those by Hou and Kramer [10], and extend those by Wyner [25] and Csiszár and Körner [13], as we measure the effective secrecy (or the leakage) using the Rényi divergence. As discussed in Remark 8, our result on the wiretap channel is also related to the semantic-security capacity studied by Golfeld, Cuff, and Permuter [24].

In the future, we plan to explore various closely related problems to the one contained herein.

1. *Rényi common information*: Wyner [5] defined the common information between two sources is the minimum rate of commonness needed to simulate these two source in a distributed fashion. In his original work, the normalized relative entropy was used to measure the level of approximation. We can generalize his problem by replacing the relative entropy with the Rényi divergence, and define the minimum rate for this case as *Rényi common information*. In fact, we have provided a complete solution for the Rényi common information [26].
2. *Distributed channel synthesis under the Rényi divergence*: The coordination problem or distributed channel synthesis problem was studied by Cuff, Permuter, and Cover [6, 27]. In this problem, an observer (encoder) of a source sequence describes the sequence to a distant random number generator (decoder) that produces another sequence. What is the minimum rate of description needed to achieve a joint distribution that is statistically indistinguishable, under the TV distance, from the distribution induced by a given channel? For this problem, Cuff [6] provided a complete characterization of the minimum rate. We can enhance the level of coordination by replacing the TV measure with the Rényi divergence. For this enhanced version of the problem, a natural question is whether the minimum rate remains the same as that given by Cuff.

Acknowledgements. The authors are supported by a Singapore National Research Foundation (NRF) National Cybersecurity R&D Grant (R-263-000-C74-281 and NRF2015NCR-NCR003-006).

Appendix

A Notation for The Proofs

The set of probability measures on \mathcal{X} is denoted as $\mathcal{P}(\mathcal{X})$, and the set of conditional probability measures on \mathcal{Y} given a variable in \mathcal{X} is denoted as $\mathcal{P}(\mathcal{Y}|\mathcal{X}) := \{P_{Y|X} : P_{Y|X}(\cdot|x) \in \mathcal{P}(\mathcal{Y}), x \in \mathcal{X}\}$.

We use $T_{x^n}(x) := \frac{1}{n} \sum_{i=1}^n 1\{x_i = x\}$ to denote the type (empirical distribution) of a sequence x^n , T_X and $V_{Y|X}$ to respectively denote a type of sequences in \mathcal{X}^n and a conditional type of sequences in \mathcal{Y}^n (given a sequence $x^n \in \mathcal{X}^n$).

For a type T_X , the type class (set of sequences having the same type T_X) is denoted by \mathcal{T}_{T_X} . For a conditional type $V_{Y|X}$ and a sequence x^n , the V -shell of x^n (the set of y^n sequences having the same conditional type $V_{Y|X}$ given x^n) is denoted by $\mathcal{T}_{V_{Y|X}}(x^n)$. The set of types of sequences in \mathcal{X}^n is denoted as

$$\mathcal{P}^{(n)}(\mathcal{X}) := \{T_{x^n} : x^n \in \mathcal{X}^n\}. \tag{34}$$

The set of conditional types of sequences in \mathcal{Y}^n given a sequence in \mathcal{X}^n with the type T_X is denoted as

$$\mathcal{P}^{(n)}(\mathcal{Y}|T_X) := \{V_{Y|X} \in \mathcal{P}(\mathcal{Y}|\mathcal{X}) : V_{Y|X} \times T_X \in \mathcal{P}^{(n)}(\mathcal{X} \times \mathcal{Y})\}. \tag{35}$$

For brevity, sometimes we use $T(x, y)$ to denote the joint distributions $T(x) V(y|x)$ or $T(y) V(x|y)$.

The ϵ -typical set of Q_X is denoted as

$$\mathcal{T}_\epsilon^n(Q_X) := \{x^n \in \mathcal{X}^n : |T_{x^n}(x) - Q_X(x)| \leq \epsilon Q_X(x), \forall x \in \mathcal{X}\}. \tag{36}$$

The conditionally ϵ -typical set of Q_{XY} is denoted as

$$\mathcal{T}_\epsilon^n(Q_{YX}|x^n) := \{y^n \in \mathcal{X}^n : (x^n, y^n) \in \mathcal{T}_\epsilon^n(Q_{XY})\}. \tag{37}$$

For brevity, sometimes we write $\mathcal{T}_\epsilon^n(Q_X)$ and $\mathcal{T}_\epsilon^n(Q_{YX}|x^n)$ as \mathcal{T}_ϵ^n and $\mathcal{T}_\epsilon^n(x^n)$ respectively.

The total variation distance between two probability mass functions P and Q with a common alphabet \mathcal{X} is defined by

$$|P - Q| := \frac{1}{2} \sum_{x \in \mathcal{X}} |P(x) - Q(x)|. \tag{38}$$

Finally, we use $\delta_n, \delta'_n, \delta''_n$ to denote three generic sequences tending to zero as $n \rightarrow \infty$.

B Proof of Theorem 1

B.1 One-Shot Bounds

To prove Theorem 1, we need the following one-shot (i.e., blocklength n equal to 1) bounds. Due to space limitations, the proofs are omitted but mostly follow from the proofs of the one-shot bounds in [14, 15].

Lemma 1 (One-Shot Bounds for Direct Part). *Consider a random mapping $f_U : \mathcal{M} = \{1, \dots, e^R\} \rightarrow \mathcal{X}$. We set $U = \{X(i)\}_{i \in \mathcal{M}}$ with $X(i) \sim P_X, i \in \mathcal{M}$, and set $f_U(m) = X(m)$. For this random code, we have for $s \in [0, 1]$,*

$$e^{sD_{1+s}(P_{YU} \| Q_Y P_U)} \leq e^{sD_{1+s}(P_{XY} \| P_X Q_Y) - sR} + e^{sD_{1+s}(P_Y \| Q_Y)} \tag{39}$$

$$\leq 2e^{s\Gamma_{1+s}(P_X, P_{Y|X}, Q_Y, R)}, \tag{40}$$

where

$$\Gamma_{1+s}(P_X, P_{Y|X}, Q_Y, R) := \max \{D_{1+s}(P_{XY} \| P_X Q_Y) - R, D_{1+s}(P_Y \| Q_Y)\}. \quad (41)$$

In the other direction with $s \in [0, 1)$, we have

$$\begin{aligned} e^{-sD_{1-s}(P_{YU} \| Q_Y P_U)} &\geq 2^{-s} \left[e^{sR} \sum_{x,y} P(x) P^{1-s}(y|x) Q^s(y) \mathbb{1} \left\{ \frac{P(y|x)}{P(y)} \geq e^R \right\} \right. \\ &\quad \left. + \sum_{x,y} P(x,y) P^{-s}(y) Q^s(y) \mathbb{1} \left\{ \frac{P(y|x)}{P(y)} < e^R \right\} \right]. \end{aligned} \quad (42)$$

Remark 9. A similar result to (39) was shown in [19, Thm. 14], but their result is a special case of ours with $P_Y = Q_Y$. We believe that [19, Thm. 14] cannot be directly applied to deriving the achievability results (including Theorems 1, 2, 3, and 6) in this paper for the case $s \in (0, 1]$, since we need set to P_{X^n} as either the uniform distribution over some type class or a truncated version of some product distribution and hence the resulting output P_{Y^n} is not equal to the target *product* distribution Q_Y^n . This is discussed in Appendices B.3 and C.

Lemma 2 (One-Shot Bounds for Converse Part). *For any random mapping $f_U : \mathcal{M} = \{1, \dots, e^R\} \rightarrow \mathcal{X}$, we have for $s \in [0, 1]$,*

$$e^{sD_{1+s}(P_{YU} \| Q_Y P_U)} \geq e^{s\Gamma_{1+s}(P_X, P_{Y|X}, Q_Y, R)} \quad (43)$$

for some P_X , where $\Gamma_{1+s}(P_X, P_{Y|X}, Q_Y, R)$ is given by (41). In the other direction with $s \in [0, 1)$, we have

$$\begin{aligned} e^{-sD_{1-s}(P_{YU} \| Q_Y \times P_U)} &\leq e^{sR} \sum_{x,y} P(x) P^{1-s}(y|x) Q^s(y) \mathbb{1} \left\{ \frac{P(y|x)}{P(y)} \geq \frac{e^R}{2} \right\} \\ &\quad + \sum_{x,y} P(x,y) P^{-s}(y) Q^s(y) \mathbb{1} \left\{ \frac{P(y|x)}{P(y)} < \frac{e^R}{2} \right\} \end{aligned} \quad (44)$$

for some P_X .

B.2 Multi-letter Characterization

We now assume that $\mathcal{M}_n = \{1, \dots, e^{nR}\}$ and the channel $P_{Y|X}$, used n times, is memoryless and stationary. Then the one-shot bounds can be used to prove the following result. Due to space limitations, the proofs are omitted.

Theorem 7 (Multi-letter Characterization). *For any $s \in [0, 1]$, we have*

$$\frac{1}{n} \inf_{f_{U_n}} D_{1+s}(P_{Y^n U_n} \| Q_{Y^n} P_{U_n}) = \Gamma_{1+s}^{(n)}(P_{Y|X}, Q_Y, R) + o(1), \quad (45)$$

where $o(1)$ denotes a term that tends to zero as $n \rightarrow \infty$, and

$$\Gamma_{1+s}^{(n)}(P_{Y|X}, Q_Y, R) := \inf_{P_{X^n}} \max \left\{ \frac{1}{n} D_{1+s}(P_{X^n Y^n} \| P_{X^n} Q_{Y^n}) - R, \frac{1}{n} D_{1+s}(P_{Y^n} \| Q_{Y^n}) \right\}, \quad (46)$$

Furthermore, for any $s \in (0, 1)$, and any positive integer k , we have

$$\Gamma_{1-s}^{(n)}(P_{Y|X}, Q_Y, R) + o(1) \leq \frac{1}{n} \inf_{f_{U_n}} D_{1-s}(P_{Y^n U_n} \| Q_{Y^n} P_{U_n}) \quad (47)$$

$$\leq \Gamma_{1-s}^{(k)}(P_{Y|X}, Q_Y, R) + o(1), \quad (48)$$

where $o(1)$ denotes a term tending to zero as $n \rightarrow \infty$, and

$$\Gamma_{1-s}^{(n)}(P_{Y|X}, Q_Y, R) := \inf_{P_{X^n}} \max_{t \in [0, s]} \left\{ -\frac{t}{s} R - \frac{1}{ns} \log \sum_{x^n, y^n} P(x^n, y^n) P^{-t}(y^n | x^n) P^{t-s}(y^n) Q^s(y^n) \right\}. \quad (49)$$

The infima in (45) and (48) are achieved by a sequence of conventional random codes.

Remark 10. Note that the lower bound and the upper bound differ only in the parameter of $\Gamma_{1-s}^{(\cdot)}$.

Remark 11. Theorem 7 holds even when the alphabets are not discrete.

Remark 12. From the definition of $\Gamma_{1+s}^{(n)}(P_{Y|X}, Q_Y, R)$, we have

$$\Gamma_{1+s}^{(n)}(P_{Y|X}, Q_Y, R) := \inf_{P_{X^n}} \min_{t \in [0, s]} \left\{ -\frac{t}{s} R + \frac{1}{ns} \log \sum_{x^n, y^n} P(x^n, y^n) P^t(y^n | x^n) P^{s-t}(y^n) Q^{-s}(y^n) \right\}. \quad (50)$$

Therefore, the notations $\Gamma_{1+s}^{(n)}$ and $\Gamma_{1-s}^{(n)}$ are consistent in the sense that if we set s to be $-s$ in $\Gamma_{1+s}^{(n)}$, we obtain $\Gamma_{1-s}^{(n)}$.

B.3 Proof of Theorem 1

To prove the achievability part of Theorem 1 for the case $1+s$ or $1-s$, we set

$$P(x^n) = \frac{1\{x^n \in \mathcal{T}_{\tilde{T}_X}\}}{|\mathcal{T}_{\tilde{T}_X}|} \quad (51)$$

where \tilde{T}_X is some type of n -length sequences. Substitute this into the multi-letter expressions (45) and (48), and single-letterize them by using the method of types. The achievability part of Theorem 1 then follows.

For the converse part of Theorem 1, we also use the method of types [28] to single-letterize (45).

Note that although we use the method of types for both the achievability and converse parts, we relax (45) and (48) in opposite directions for these two cases. Hence different bounding techniques are applied in these two cases. Furthermore, interested readers may refer to Appendix C, since it contains similar proof techniques to the one used herein.

C Proof of Theorem 6

We first prove (27).

Achievability: Considering the case $s \in (0, 1]$ suffices. The achievability result for $s \in (-1, 0]$ can be obtained from the result for the case $s \in (0, 1]$ by letting $s \downarrow 0$.

We use a similar random code as the one given in Lemma 1. That is, we set $U_n = \{W^n(m)\}_{m \in \mathcal{M}}$ with $W^n(m) \sim P_{W^n}, m \in \mathcal{M}$, and set the encoder as $f_{U_n}(m) = W^n(m)$. We insert a random mapping (virtual channel) between the encoder $f_{U_n}(m)$ and the channel, which is denoted as $P_{X^n|W^n}$. For this cascaded code, we set the distributions

$$P_{W^n}(w^n) = \frac{\tilde{P}_{W^n}(w^n) \mathbf{1}\{w^n \in \mathcal{T}_{\epsilon'}^n\}}{\tilde{P}_{W^n}(\mathcal{T}_{\epsilon'}^n)} \quad (52)$$

$$P_{X^n|W^n}(x^n|w^n) = \frac{\tilde{P}_{X^n|W^n}(x^n|w^n) \mathbf{1}\{(w^n, x^n) \in \mathcal{T}_{\epsilon'}^n\}}{\tilde{P}_{X^n|W^n}(\mathcal{T}_{\epsilon'}^n(w^n)|w^n)}, \quad (53)$$

where $\epsilon' < \epsilon$, and $\tilde{P}_{W^n X^n} := \tilde{P}_{W^n}^n$ for some $\tilde{P}_{W^n X}$ such that $\tilde{P}_X \in \mathcal{P}(P_{Z|X}, Q_Z)$.

Then by the method of types, we obtain

$$\begin{aligned} & \frac{1}{n} D_{1+s}(P_{W^n Z^n} \| P_{W^n} \times Q_{Z^n}) \\ &= \frac{1}{ns} \log \sum_{w^n, z^n} P(w^n) \left(\sum_{x^n} P(x^n|w^n) P(z^n|x^n) \right)^{1+s} Q^{-s}(z^n) \end{aligned} \quad (54)$$

$$\begin{aligned} &= \frac{1}{ns} \log \sum_{T_W} \sum_{w^n \in \mathcal{T}_{T_W}} \sum_{V_{Z|W}} \sum_{z^n \in \mathcal{T}_{V_{Z|W}}(w^n)} \frac{\tilde{P}(w^n) \mathbf{1}\{w^n \in \mathcal{T}_{\epsilon'}^n\}}{\tilde{P}_{W^n}(\mathcal{T}_{\epsilon'}^n)} \\ & \quad \left(\sum_{V_{X|WZ}} \sum_{x^n \in \mathcal{T}_{V_{X|WZ}}(w^n, z^n)} \frac{\tilde{P}(x^n|w^n) \mathbf{1}\{(w^n, x^n) \in \mathcal{T}_{\epsilon'}^n\}}{\tilde{P}_{X^n|W^n}(\mathcal{T}_{\epsilon'}^n(w^n)|w^n)} \right)^{1+s} \\ & \quad e^{n \sum_{x,z} T(x,z) \log P(z|x)} e^{-ns \sum_z T(z) \log Q(z)} \end{aligned} \quad (55)$$

$$\begin{aligned}
 &= \delta_n + \frac{1}{ns} \log \sum_{T_W} \sum_{w^n \in \mathcal{T}_{T_W}} \sum_{V_{Z|W}} \sum_{z^n \in \mathcal{T}_{V_{Z|W}}(w^n)} \tilde{P}(w^n) 1\{w^n \in \mathcal{T}_\epsilon^n\} \\
 &\quad \left(\sum_{V_{X|WZ}} \sum_{x^n \in \mathcal{T}_{V_{X|WZ}}(w^n, z^n)} \tilde{P}(x^n|w^n) 1\{(w^n, x^n) \in \mathcal{T}_\epsilon^n\} \right. \\
 &\quad \left. e^{n \sum_{x,z} T(x,z) \log P(z|x)} \right)^{1+s} e^{-ns \sum_z T(z) \log Q(z)} \tag{56}
 \end{aligned}$$

$$\begin{aligned}
 &\leq \delta_n + \delta'_n + \frac{1}{ns} \log \max_{T_W: |T_W - \tilde{P}_W| \leq \epsilon'} \max_{V_{Z|W}} \sum_{w^n \in \mathcal{T}_{T_W}} \sum_{z^n \in \mathcal{T}_{V_{Z|W}}(w^n)} \tilde{P}(w^n) \\
 &\quad \left(\max_{V_{X|WZ}: |(V_{X|WZ} \circ V_{Z|W})T_W - \tilde{P}_{WX}| \leq \epsilon} \sum_{x^n \in \mathcal{T}_{V_{X|WZ}}(w^n, z^n)} \tilde{P}(x^n|w^n) \right. \\
 &\quad \left. e^{n \sum_{x,z} T(x,z) \log P(z|x)} \right)^{1+s} e^{-ns \sum_z T(z) \log Q(z)} \tag{57}
 \end{aligned}$$

$$\begin{aligned}
 &= \max_{T_W, V_{Z|W}, V_{X|WZ}: |T_W - \tilde{P}_W| \leq \epsilon', |(V_{X|WZ} \circ V_{Z|W})T_W - \tilde{P}_{WX}| \leq \epsilon} \frac{1}{s} \left(H(V_{Z|W} \times T_W) + \sum_w T(w) \log \tilde{P}(w) \right) \\
 &\quad + \frac{1+s}{s} \left(H(V_{X|WZ}|T_W V_{Z|W}) + \sum_{w,x} T(w,x) \log \tilde{P}(x|w) \right. \\
 &\quad \left. + \sum_{x,z} T(x,z) \log P(z|x) \right) - \sum_z T(z) \log Q(z) + \delta_n + \delta'_n + \delta''_n \tag{58}
 \end{aligned}$$

$$\begin{aligned}
 &= \max_{T_{WX}, V_{Z|WX}: |T_W - \tilde{P}_W| \leq \epsilon', |T_{WX} - \tilde{P}_{WX}| \leq \epsilon} \left\{ \frac{1+s}{s} \sum_{w,x,z} T(w,x,z) \log \frac{\tilde{P}(w,x) P(z|x)}{T(w,x,z)} \right. \\
 &\quad \left. + \sum_{w,z} T(w,z) \log \frac{T(w,z)}{\tilde{P}(w) Q(z)} \right\} + \delta_n + \delta'_n + \delta''_n, \tag{59}
 \end{aligned}$$

where $[(V_{X|WZ} \circ V_{Z|W})T_W](x, w) := \sum_z V_{X|WZ}(x|w, z) V_{Z|W}(z|x) T_W(w)$, (56) follows since $\tilde{P}_{W^n}(\mathcal{T}_{\epsilon'}^n) \rightarrow 1$ and $\tilde{P}_{X^n|W^n}(\mathcal{T}_\epsilon^n(w^n)|w^n) \rightarrow 1$ for any $w^n \in \mathcal{T}_{\epsilon'}^n(\tilde{P}_W)$ by the law of large numbers, in (59) the arguments of maximization are replaced by $T_{WX}, V_{Z|WX}$ (this is feasible since both $(T_W, V_{Z|W}, V_{X|WZ})$ in (58) and $(T_{WX}, V_{Z|WX})$ in (59) run through all the types of sequences in $\mathcal{X}^n \times \mathcal{Y}^n \times \mathcal{Z}^n$).

Observe that in (59) T_{WX} is restricted to being close to \tilde{P}_{WX} but there is no restriction on $V_{Z|WX}$, hence it can be shown that as $n \rightarrow \infty$ and $\epsilon, \epsilon' \rightarrow 0$, (59) asymptotically equals

$$\max_{\tilde{P}_{Z|WX}} \left\{ -\frac{1+s}{s} \sum_{w,x,z} \tilde{P}(w,x,z) \log \frac{\tilde{P}(z|w,x)}{P(z|x)} + \sum_{w,z} \tilde{P}(w,z) \log \frac{\tilde{P}(z|w)}{Q(z)} \right\}, \tag{60}$$

in the sense that the difference between (59) and (60) vanishes as $n \rightarrow \infty$. Essentially, we can replace the (conditional) types with their corresponding (conditional) distributions. The detailed proof is omitted here. Hence $\frac{1}{n}D_{1+s}(P_{W^n Z^n} \| P_{W^n} \times Q_{Z^n}) \rightarrow (60)$ as $n \rightarrow \infty$. Comparing (60) to the definition of $\tilde{R}'_{1+s}(\tilde{P}_{W|X}\tilde{P}_X, P_{Z|X}, Q_Z)$ in (29), we can find that they are equal for the case of $s \in (0, 1]$. Hence

$$\lim_{n \rightarrow \infty} \frac{1}{n}D_{1+s}(P_{W^n Z^n} \| P_{W^n} \times Q_{Z^n}) = \tilde{R}'_{1+s}(\tilde{P}_{W|X}\tilde{P}_X, P_{Z|X}, Q_Z). \quad (61)$$

Furthermore, observe

$$P_{X^n}(x^n) = \sum_{w^n} \frac{\tilde{P}(w^n) 1\{w^n \in \mathcal{T}_{\epsilon'}^n\}}{\tilde{P}_{W^n}(\mathcal{T}_{\epsilon'}^n)} \frac{\tilde{P}(x^n|w^n) 1\{(w^n, x^n) \in \mathcal{T}_{\epsilon}^n\}}{\tilde{P}_{X^n|W^n}(\mathcal{T}_{\epsilon}^n(w^n)|w^n)} \quad (62)$$

$$\leq \sum_{w^n} \frac{\tilde{P}(w^n) 1\{w^n \in \mathcal{T}_{\epsilon'}^n\}}{1 - \delta_n} \tilde{P}(x^n|w^n) 1\{(w^n, x^n) \in \mathcal{T}_{\epsilon}^n\} \quad (63)$$

$$\leq \frac{\tilde{P}(x^n) 1\{x^n \in \mathcal{T}_{\epsilon}^n\}}{1 - \delta_n}, \quad (64)$$

where (63) follows since both $\tilde{P}(W^n \in \mathcal{T}_{\epsilon'}^n)$ and $\tilde{P}((w^n, X^n) \in \mathcal{T}_{\epsilon}^n | W^n = w^n)$ for $w^n \in \mathcal{T}_{\epsilon'}^n$ converge to 1 as $n \rightarrow \infty$. Therefore,

$$D_{1+s}(P_{X^n} \| \tilde{P}_{X^n}) \leq \frac{1}{s} \log \sum_{x^n} \left(\frac{\tilde{P}(x^n) 1\{x^n \in \mathcal{T}_{\epsilon}^n\}}{1 - \delta_n} \right)^{1+s} \tilde{P}^{-s}(x^n) \quad (65)$$

$$= \frac{1}{s} \log \frac{\tilde{P}_{X^n}(\mathcal{T}_{\epsilon}^n)}{(1 - \delta_n)^{1+s}} \rightarrow 0, \quad (66)$$

where (66) follows since $\tilde{P}_{X^n}(\mathcal{T}_{\epsilon}^n)$ converges to 1 as $n \rightarrow \infty$. Since P_{Z^n} and Q_{Z^n} are respectively the distributions of the channel output induced by the input P_{X^n} and \tilde{P}_{X^n} , by the data processing inequality [16], we have

$$D_{1+s}(P_{Z^n} \| Q_{Z^n}) \leq D_{1+s}(P_{X^n} \| \tilde{P}_{X^n}). \quad (67)$$

Hence $D_{1+s}(P_{Z^n} \| Q_{Z^n}) \rightarrow 0$ as well.

Finally, by Lemma 1, we obtain

$$\begin{aligned} & e^{sD_{1+s}(P_{M_1 Z^n} \| P_{M_1} Q_{Z^n})} \\ & \leq e^{sD_{1+s}(P_{W^n Z^n} \| P_{W^n} Q_{Z^n}) - nsR_0} + e^{sD_{1+s}(P_{Z^n} \| Q_{Z^n})} \rightarrow 1, \end{aligned} \quad (68)$$

where (68) holds for $s \in (0, 1]$ if

$$R_0 > \tilde{R}'_{1+s}(\tilde{P}_{W|X}\tilde{P}_X, P_{Z|X}, Q_Z) \quad (69)$$

by (60) with a small enough ϵ . Hence the secrecy constraint is satisfied.

Moreover, using standard joint-typicality decoding, we have that error constraint

$$\mathbb{P} \left((M_0, M_1) \neq (\widehat{M}_0, \widehat{M}_1) \right) \rightarrow 0 \tag{70}$$

is satisfied as well if $R_0 + R_1 \leq I_{\widehat{P}}(W; Y)$.

Converse: Set $W = (M_0, M_1)$. By the data processing inequality,

$$R_0 + R_1 \leq \frac{1}{n} I(W; Y^n) \leq I(W; Y_J), \tag{71}$$

where $J \sim \text{Unif}[1 : n]$ denotes a time index variable, independent of (W, Y^n) . It is easy to verify that

$$P_{W X_J Y_J}(w, x, y) = P_W(w) \frac{1}{n} \sum_{j=1}^n \mathbb{P} \{ (X_j, Y_j) = (x, y) | W = w \} \tag{72}$$

$$= P_W(w) \mathbb{E}_{X^n Y^n | W=w} [T_{X^n Y^n}(x, y)], \tag{73}$$

and

$$P_{W X_J Y_J}(w, x, y) = P_{W X_J}(w, x) P(y|x) \tag{74}$$

$$= P_W(w) \mathbb{E}_{X^n | W=w} [T_{X^n}(x)] P(y|x), \tag{75}$$

where (75) is obtained similarly to (72)–(73).

We first consider the case $s \in (0, 1]$. Observe M_1 is independent of M_0 . Hence if we consider M_1 as U and M_0 as M , then the wiretap channel problem turns into the channel resolvability problem. By Lemma 2, we obtain

$$D_{1+s}(P_{M_1 Z^n} \| Q_{M_1 Z^n}) \geq \max \{ D_{1+s}(P_{M_0 M_1 Z^n} \| P_{M_0 M_1} \times Q_{Z^n}) - nR_0, D_{1+s}(P_{Z^n} \| Q_{Z^n}) \} \tag{76}$$

$$= \max \{ D_{1+s}(P_{W Z^n} \| P_W \times Q_{Z^n}) - nR_0, D_{1+s}(P_{Z^n} \| Q_{Z^n}) \}. \tag{77}$$

Define $\widetilde{P}_{Z|WX}$ as the maximizing distribution of

$$\begin{aligned} & \max_{\widetilde{P}_{Z|WX} \in \mathcal{P}(\mathcal{Z} | \mathcal{W} \times \mathcal{X})} \left\{ -\frac{1+s}{s} \sum_{w,x,z} P(w) P_{X_J|W}(x|w) \widetilde{P}(z|w, x) \log \frac{\widetilde{P}(z|w, x)}{P(z|x)} \right. \\ & \left. + \sum_{w,x,z} P(w) P_{X_J|W}(x|w) \widetilde{P}(z|w, x) \log \frac{\sum_x P_{X_J|W}(x|w) \widetilde{P}(z|w, x)}{Q(z)} \right\}, \tag{78} \end{aligned}$$

where $P_{W X_J Z_J}$ is the distribution of W, X_J, Z_J induced by the adopted code. Note that $\widetilde{P}_{Z|WX}$ is determined by the code, the channel $P_{Z|X}$, and the target distribution Q_Z .

For any $w \in \mathcal{W}$ and any $T_X \in \mathcal{P}^{(n)}(\mathcal{X})$, we can find a conditional type $V_{Z|X}^{(w)} \in \mathcal{P}^{(n)}(\mathcal{X} | T_X)$ such that

$$\left| T_X \times \widetilde{P}_{Z|XW}(\cdot | \cdot, w) - T_X \times V_{Z|X}^{(w)} \right| \leq \frac{|\mathcal{X}| |\mathcal{Z}|}{2n} = O\left(\frac{1}{n}\right). \tag{79}$$

The proof of this claim is similar to that of [29, Lem. 2.1.2], and hence omitted here.

Consider the first term of the maximization in (77), then we obtain

$$\begin{aligned} & \frac{1}{n} D_{1+s} (P_{WZ^n} \| P_W \times Q_{Z^n}) \\ &= \frac{1}{ns} \log \sum_{w \in \mathcal{W}} \sum_{T_Z} \sum_{z^n \in \mathcal{T}_{T_Z}} P(w) e^{-ns \sum_z T(z) \log Q(z)} \\ & \quad \left(\sum_{V_{X|Z}} P_{X^n|W} (\mathcal{T}_{V_{X|Z}}(z^n) | w) e^{n \sum_{x,z} T(x,z) \log P(z|x)} \right)^{1+s} \end{aligned} \quad (80)$$

$$\begin{aligned} & \geq \frac{1}{ns} \log \sum_{w \in \mathcal{W}} \sum_{T_Z} \sum_{z^n \in \mathcal{T}_{T_Z}} P(w) \sum_{V_{X|Z}} P_{X^n|W}^{1+s} (\mathcal{T}_{V_{X|Z}}(z^n) | w) \\ & \quad e^{n(1+s) \sum_{x,z} T(x,z) \log P(z|x) - ns \sum_z T(z) \log Q(z)} \end{aligned} \quad (81)$$

$$\begin{aligned} & \geq \frac{1}{ns} \log \sum_{w, T_Z, V_{X|Z}} |\mathcal{T}_{T_Z}| P(w) \left(\sum_{z^n \in \mathcal{T}_{T_Z}} \frac{1}{|\mathcal{T}_{T_Z}|} P_{X^n|W} (\mathcal{T}_{V_{X|Z}}(z^n) | w) \right)^{1+s} \\ & \quad e^{n(1+s) \sum_{x,z} T(x,z) \log P(z|x) - ns \sum_z T(z) \log Q(z)} \end{aligned} \quad (82)$$

$$\begin{aligned} & = \delta_n + \frac{1}{ns} \log \sum_{w, T_Z, V_{X|Z}} P(w) P_{X^n|W}^{1+s} (\mathcal{T}_{T_X} | w) e^{-ns H(T_Z) + n(1+s) H(V_{Z|X} | T_X)} \\ & \quad e^{n(1+s) \sum_{x,z} T(x,z) \log P(z|x) - ns \sum_z T(z) \log Q(z)} \end{aligned} \quad (83)$$

$$\begin{aligned} & \geq \delta_n + \frac{1}{ns} \log \sum_{w, T_X} P(w) P_{X^n|W}^{1+s} (\mathcal{T}_{T_X} | w) \\ & \quad e^{-ns H(V_{Z|X}^{(w)} \circ T_X) + n(1+s) (H(V_{Z|X}^{(w)} | T_X) + \sum_{x,z} T(x) V_{Z|X}^{(w)}(z|x) \log P(z|x))} \\ & \quad e^{-ns \sum_z (V_{Z|X}^{(w)} \circ T_X)(z) \log Q(z)} \end{aligned} \quad (84)$$

$$\begin{aligned} & = \delta_n + \frac{1}{ns} \log \sum_{w, T_X} P(w) P_{X^n|W}^{1+s} (\mathcal{T}_{T_X} | w) \\ & \quad e^{-ns H(\tilde{P}_{Z|W_X \circ T_X}) + n(1+s) (H(\tilde{P}_{Z|W_X} | T_X) + \sum_{x,z} T(x) \tilde{P}_{Z|W_X}(z|x) \log P(z|x))} \\ & \quad e^{-ns \sum_z (\tilde{P}_{Z|W_X \circ T_X})(z) \log Q(z) + n \cdot \delta'_n} \end{aligned} \quad (85)$$

$$\begin{aligned} & \geq \delta_n + \delta'_n + \frac{1}{ns} \log |\mathcal{P}^{(n)}(\mathcal{X})| \left(\sum_{w, T_X} \frac{1}{|\mathcal{P}^{(n)}(\mathcal{X})|} P(w) P_{X^n|W} (\mathcal{T}_{T_X} | w) \right. \\ & \quad \left. e^{-n \sum_{x,z} T(x) \tilde{P}(z|w,x) \log \frac{\tilde{P}(z|w,x)}{\tilde{P}(z|x)} + \frac{ns}{1+s} \sum_{x,z} T(x) \tilde{P}(z|w,x) \log \frac{\sum_x T(x) \tilde{P}(z|w,x)}{Q(z)}} \right)^{1+s} \end{aligned} \quad (86)$$

$$\begin{aligned} & \geq \delta_n + \delta'_n + \delta''_n + \frac{1+s}{ns} \log \sum_{w, T_X} P(w) P_{X^n|W} (\mathcal{T}_{T_X} | w) \\ & \quad e^{-n \sum_{x,z} T(x) \tilde{P}(z|w,x) \log \frac{\tilde{P}(z|w,x)}{\tilde{P}(z|x)} + \frac{ns}{1+s} \sum_{x,z} T(x) \tilde{P}(z|w,x) \log \frac{\sum_x T(x) \tilde{P}(z|w,x)}{Q(z)}} \end{aligned} \quad (87)$$

$$\begin{aligned} &\geq \delta_n + \delta'_n + \delta''_n + \sum_{w \in \mathcal{W}} P(w) \\ &\quad \left(-\frac{1+s}{s} \sum_{x,z} \mathbb{E}_{X^n|W=w} [T_{X^n}(x)] \tilde{P}(z|w,x) \log \frac{\tilde{P}(z|w,x)}{P(z|x)} \right. \\ &\quad \left. + \sum_{x,z} \mathbb{E}_{X^n|W=w} \left[T_{X^n}(x) \tilde{P}(z|w,x) \log \frac{\sum_x T_{X^n}(x) \tilde{P}(z|w,x)}{Q(z)} \right] \right) \end{aligned} \tag{88}$$

$$\begin{aligned} &\geq \delta_n + \delta'_n + \delta''_n + \sum_{w \in \mathcal{W}} P(w) \left(-\frac{1+s}{s} \sum_{x,z} P_{X_J|W}(x|w) \tilde{P}(z|w,x) \log \frac{\tilde{P}(z|w,x)}{P(z|x)} \right. \\ &\quad \left. + \sum_{x,z} P_{X_J|W}(x|w) \tilde{P}(z|w,x) \log \frac{\sum_x P_{X_J|W}(x|w) \tilde{P}(z|w,x)}{Q(z)} \right), \end{aligned} \tag{89}$$

where (81) follows since $\sum_i a_i^p \leq (\sum_i a_i)^p$ for non-negative real numbers $\{a_i\}$ and $p \geq 1$, (82) and (86) follow since $x \mapsto x^{1+s}$ is a convex function for nonnegative s , (83) follows since

$$\sum_{z^n \in \mathcal{T}_{T_Z}} P_{X^n|W}(\mathcal{T}_{V_{X|Z}}(z^n)|w) = \sum_{z^n \in \mathcal{T}_{T_Z}} \sum_{x^n \in \mathcal{T}_{V_{X|Z}}(z^n)} P(x^n|w) \tag{90}$$

$$= \sum_{x^n \in \mathcal{T}_{T_X}} \sum_{z^n \in \mathcal{T}_{V_{Z|X}}(x^n)} P(x^n|w) = \sum_{x^n \in \mathcal{T}_{T_X}} e^{nH(V_{Z|X}|T_X) + n\delta_n} P(x^n|w) \tag{91}$$

$$= e^{nH(V_{Z|X}|T_X) + n\delta_n} P_{X^n|W}(\mathcal{T}_{T_X}|w), \tag{92}$$

in (84) $V_{Z|X}^{(w)} : \mathcal{W} \mapsto \mathcal{P}^{(n)}(\mathcal{Z}|T_X)$ is an arbitrary conditional type chosen according to w ,⁴ (85) follows from (79) and [30, Lem. 8], (87) follows since the number of types in $\mathcal{P}^{(n)}(\mathcal{X})$ is polynomial in n , (88) follows since (1) $x \mapsto \log x$ is a concave function; (2) $P_{X^n|W}(\mathcal{T}_{T_X}|w) = \sum_{x^n \in \mathcal{T}_{T_X}} P_{X^n|W}(x^n|w)$; and (3) $\mathcal{T}_{T_X} \subseteq \mathcal{X}^n$ runs through all the sequences in \mathcal{X}^n , (89) follows since $x \mapsto x \log x$ is a convex function, and $\mathbb{E}_{X^n|W=w} [T_{X^n}(x)] = P_{X_J|W}(x|w)$; see (75).

By the choice of $\tilde{P}_{Z|WX}$, from (89) we have

$$\lim_{n \rightarrow \infty} \frac{1}{n} D_{1+s}(P_{WZ^n} \| P_W \times Q_{Z^n}) \geq (78). \tag{93}$$

Furthermore, it is easy to verify

$$\lim_{n \rightarrow \infty} \left| P_{X_J}^{(n)} \circ P_{Z|X} - Q_Z \right| = 0, \tag{94}$$

since $D_{1+s}(P_{Z^n} \| Q_{Z^n}) \rightarrow 0$ (see (77)).

⁴ Note that the choice of $V_{Z|X}^{(w)}$ is not the best one for the lower bound (84), since the best one should be independent of w . However, it is actually the best for the final lower bound (89).

Since $\mathcal{P}(\mathcal{X})$ is compact, for each w , there must exist some sequence of increasing integers $\{n_k\}_{k=1}^\infty$ such that $P_{X_J|W=w}^{(n_k)}$ converges to some distribution $\tilde{P}_{X|W=w}$. By (94), $\mathbb{E}_W[\tilde{P}_{X|W}(\cdot|W)] \in \mathcal{P}(P_{Z|X}, Q_Z)$ holds. Moreover, (71) and (93) respectively imply

$$R_0 + R_1 \leq I_{\tilde{P}}(W; Y), \quad (95)$$

and

$$R_0 \geq \max_{\tilde{P}_{Z|WX}} \left\{ -\frac{1+s}{s} \sum_{w,x,z} \tilde{P}(w,x,z) \log \frac{\tilde{P}(z|w,x)}{P(z|x)} + \sum_{w,z} \tilde{P}(w,z) \log \frac{\tilde{P}(z|w)}{Q(z)} \right\}. \quad (96)$$

Observe that the RHS of (96) is just $\tilde{R}'_{1+s}(\tilde{P}_{W|X}\tilde{P}_X, P_{Z|X}, Q_Z)$ with $s \in (0, 1]$. Hence $R_0 \geq \tilde{R}'_{1+s}(\tilde{P}_{W|X}\tilde{P}_X, P_{Z|X}, Q_Z)$.

Therefore, $P_W\tilde{P}_{X|W}$ is the desired distribution \tilde{P}_{WX} in (27). The proof for the case $s \in (0, 1]$ is complete.

Next we consider the case $s \in (-1, 0]$. This case can be proved by following similar steps as the proof of traditional channel resolvability problem (or distributed channel synthesis problem) [6]. Observe

$$R_0 \geq \frac{1}{n} I(M_0; Z^n | M_1) = \frac{1}{n} I(M_0 M_1; Z^n | M_1) \quad (97)$$

$$= \frac{1}{n} I(M_0 M_1; Z^n) - \frac{1}{n} I(M_1; Z^n) \quad (98)$$

$$= \frac{1}{n} H(Z^n) - \frac{1}{n} H(Z^n | M_0 M_1) - \delta_n \quad (99)$$

$$= H_Q(Z) - H(Z_J | W) - \delta_n + \delta'_n, \quad (100)$$

where (99) and (100) follow from the facts $|P_{M_1 Z^n} - P_{M_1} Q_{Z^n}| \rightarrow 0$ and $|P_{Z^n} - Q_{Z^n}| \rightarrow 0$, respectively.

Furthermore, for each w , there exist some increasing sequence of integers $\{n_k\}_{k=1}^\infty$ such that $P_{X_J|W=w}^{(n_k)}$ converges to some distribution $\tilde{P}_{X|W=w}$ that satisfies $\mathbb{E}_W[\tilde{P}_{X|W}(\cdot|W)] \in \mathcal{P}(P_{Z|X}, Q_Z)$. Hence letting $n = n_k$ and $k \rightarrow \infty$ in (100), we get

$$R_0 \geq H_{\tilde{P}}(Z) - H_{\tilde{P}}(Z_J | W) = I_{\tilde{P}}(W; Z). \quad (101)$$

On the other hand,

$$R_0 + R_1 \leq I_{\tilde{P}}(W; Y). \quad (102)$$

Combining (101) and (102) gives the converse part. Therefore, the proof of (27) is complete.

Next we prove (28). By adding an artificial non-secret message M'_0 (with rate R'_0) in the achievability scheme above, we have the following achievable region.

$$\bigcup_{\tilde{P}_{W|X}, \tilde{P}_X \in \mathcal{P}(P_{Z|X}, Q_Z)} \left\{ \begin{array}{l} (R_0, R_1) : R'_0 \geq 0, \\ R'_0 + R_0 + R_1 \leq I_{\tilde{P}}(W; Y), \\ R'_0 + R_0 \geq \tilde{R}'_{1+s}(\tilde{P}_{W|X}, \tilde{P}_X, P_{Z|X}, Q_Z) \end{array} \right\}. \quad (103)$$

Using Fourier–Motzkin Elimination (see [31, Appendix D]), it is easy to show the regions in (103) and (28) are the same. Hence (28) $\subseteq \mathcal{R}_{1+s}(Q_Z)$.

On the other hand, comparing the RHS (right-hand-side) of (27) and (28) yields that the RHS of (27) \subseteq (28). In addition, $\mathcal{R}_{1+s}(Q_Z)$ = the RHS of (27) as shown above. Hence $\mathcal{R}_{1+s}(Q_Z) \subseteq$ (28).

Therefore, $\mathcal{R}_{1+s}(Q_Z) =$ (28). Furthermore, by standard cardinality bounding techniques [31, Appendix C], the alphabet size of W can be limited to $|\mathcal{W}| \leq |\mathcal{X}| + 1$.

References

1. Han, T., Verdú, S.: Approximation theory of output statistics. *IEEE Trans. Inf. Theory* **39**(3), 752–772 (1993)
2. Hayashi, M.: General nonasymptotic and asymptotic formulas in channel resolvability and identification capacity and their application to the wiretap channel. *IEEE Trans. Inf. Theory* **52**(4), 1562–1575 (2006)
3. Hayashi, M.: Exponential decreasing rate of leaked information in universal random privacy amplification. *IEEE Trans. Inf. Theory* **57**(6), 3989–4001 (2011)
4. Liu, J., Cuff, P., Verdú, S.: E_γ -resolvability. *IEEE Trans. Inf. Theory* **63**(5), 2629–2658 (2017)
5. Wyner, A.: The common information of two dependent random variables. *IEEE Trans. Inf. Theory* **21**(2), 163–179 (1975)
6. Cuff, P.: Distributed channel synthesis. *IEEE Trans. Inf. Theory* **59**(11), 7071–7096 (2013)
7. Bloch, M.R., Laneman, J.N.: Strong secrecy from channel resolvability. *IEEE Trans. Inf. Theory* **59**(12), 8077–8098 (2013)
8. Han, T.S., Endo, H., Sasaki, M.: Reliability and secrecy functions of the wiretap channel under cost constraint. *IEEE Trans. Inf. Theory* **60**(11), 6819–6843 (2014)
9. Parizi, M.B., Telatar, E., Merhav, N.: Exact random coding secrecy exponents for the wiretap channel. *IEEE Trans. Inf. Theory* **63**(1), 509–531 (2017)
10. Hou, J., Kramer, G.: Effective secrecy: reliability, confusion and stealth. In: 2014 IEEE International Symposium on Information Theory (ISIT), pp. 601–605. IEEE (2014)
11. Shikata, J.: Design and analysis of information-theoretically secure authentication codes with non-uniformly random keys. IACR Cryptology ePrint Archive 2015: 250 (2015)
12. Bai, S., Langlois, A., Lepoint, T., Stehlé, D., Steinfeld, R.: Improved security proofs in lattice-based cryptography: using the Rényi divergence rather than the statistical distance. In: Iwata, T., Cheon, J.H. (eds.) ASIACRYPT 2015. LNCS, vol. 9452, pp. 3–24. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-662-48797-6_1

13. Csiszár, I., Körner, J.: Broadcast channels with confidential messages. *IEEE Trans. Inf. Theory* **24**(3), 339–348 (1978)
14. Hayashi, M., Tan, V.Y.F.: Equivocations, exponents, and second-order coding rates under various Rényi information measures. *IEEE Trans. Inf. Theory* **63**(2), 975–1005 (2017)
15. Tan, V.Y.F., Hayashi, M.: Analysis of remaining uncertainties and exponents under various conditional Rényi entropies. arXiv preprint [arXiv:1605.09551](https://arxiv.org/abs/1605.09551) (2016)
16. Van Erven, T., Harremos, P.: Rényi divergence and Kullback-Leibler divergence. *IEEE Trans. Inf. Theory* **60**(7), 3797–3820 (2014)
17. S. Verdú. α -mutual information. In: *Information Theory and Applications Workshop (ITA)*, pp. 1–6 (2015)
18. Fong, S.L., Tan, V.Y.F.: Strong converse theorems for classes of multmessage multicast networks: A Rényi divergence approach. *IEEE Trans. Inf. Theory* **62**(9), 4953–4967 (2016)
19. Hayashi, M., Matsumoto, R.: Secure multiplex coding with dependent and non-uniform multiple messages. *IEEE Trans. Inf. Theory* **62**(5), 2355–2409 (2016)
20. Yu, L., Tan, V.Y.F.: Rényi resolvability and its applications to the wiretap channel. arXiv preprint [arXiv:1707.00810](https://arxiv.org/abs/1707.00810) (2017)
21. Csiszár, I., Narayan, P.: Secrecy capacities for multiple terminals. *IEEE Trans. Inf. Theory* **50**(12), 3047–3061 (2004)
22. Kobayashi, D., Yamamoto, H., Ogawa, T.: Secure multiplex coding attaining channel capacity in wiretap channels. *IEEE Trans. Inf. Theory* **59**(12), 8131–8143 (2013)
23. Gohari, A., Anantharam, V.: Generating dependent random variables over networks. In: *2011 IEEE Information Theory Workshop (ITW)*, pp. 698–702 (2011)
24. Goldfeld, Z., Cuff, P., Permuter, H.H.: Semantic-security capacity for wiretap channels of type II. *IEEE Trans. Inf. Theory* **62**(7), 3863–3879 (2016)
25. Wyner, A.: The wire-tap channel. *Bell Labs Tech. J.* **54**(8), 1355–1387 (1975)
26. Yu, L., Tan, V.Y.F.: Wyner’s common information under Rényi divergence measures. arXiv preprint [arXiv:1709.02168](https://arxiv.org/abs/1709.02168) (2017)
27. Cuff, P., Permuter, H., Cover, T.: Coordination capacity. *IEEE Trans. Inf. Theory* **56**(9), 4181–4206 (2010)
28. Csiszár, I., Körner, J.: *Information Theory: Coding Theorems for Discrete Memoryless Systems*. Cambridge University Press, Cambridge (2011)
29. Dembo, A., Zeitouni, O.: *Large Deviations Techniques and Applications*, 2nd edn. Springer, Heidelberg (1998)
30. Yassaee, M., Gohari, A., Aref, M.: Channel simulation via interactive communications. *IEEE Trans. Inf. Theory* **61**(6), 2964–2982 (2015)
31. El Gamal, A., Kim, Y.-H.: *Network Information Theory*. Cambridge University Press, Cambridge (2011)

Author Index

- Adhikari, Avishek 58
Agarwal, Gaurav Kumar 188
- Cardone, Martina 188
Choudhury, Ashish 83
Cramer, Ronald 1
Cunningham, Robert 110
- Damgård, Ivan 1
Döttling, Nico 1
Dutta, Sabyasachi 58
- Fragouli, Christina 188
Fuller, Benjamin 110
- Giacomelli, Irene 1
- Hanaoka, Goichiro 135
Hashimoto, Yuji 135
Hayashi, Masahito 166
- Inamura, Masaki 135
Iwamoto, Mitsugu 153
- Kato, Go 166
- Malinowski, Daniel 26
- Nakai, Takeshi 153
Nuida, Koji 135
- Obana, Satoshi 73
Ohta, Kazuo 153
Owari, Masaki 166
- Patra, Arpita 83
- Ravi, Divya 83
- Shinagawa, Kazumasa 135
Shirouchi, Satoshi 153
- Tan, Vincent Y. F. 208
- Watanabe, Yohei 39
- Xing, Chaoping 1
- Yakoubov, Sophia 110
Yoshida, Maki 73
Yu, Lei 208
- Żebrowski, Karol 26