Math-for-industry
Education & Research Hub

ISSN2188-1200

IMI Workshop of the Joint Usage Research Projects

Exploring Mathematical and Practical Principles of Secure Computation and Secret Sharing

Editors: Hiroaki Anada, Yasuhiko Ikematsu, Koji Nuida, Satsuya Ohata, Yuntao Wang

九州大学マス·フォア·インダストリ研究所

MI Lecture Note Vol.85 : Kyushu University

IMI Workshop of the Joint Usage Research Projects

Exploring Mathematical and Practical Principles of Secure Computation and Secret Sharing

Editors:

Hiroaki Anada, Yasuhiko Ikematsu, Koji Nuida, Satsuya Ohata, Yuntao Wang

About MI Lecture Note Series

The Math-for-Industry (MI) Lecture Note Series is the successor to the COE Lecture Notes, which were published for the 21st COE Program “Development of Dynamic Mathematics with High Functionality,” sponsored by Japan’s Ministry of Education, Culture, Sports, Science and Technology (MEXT) from 2003 to 2007. The MI Lecture Note Series has published the notes of lectures organized under the following two programs: “Training Program for Ph.D. and New Master’s Degree in Mathematics as Required by Industry,” adopted as a Support Program for Improving Graduate School Education by MEXT from 2007 to 2009; and “Education-and-Research Hub for Mathematics-for-Industry,” adopted as a Global COE Program by MEXT from 2008 to 2012.

In accordance with the establishment of the Institute of Mathematics for Industry (IMI) in April 2011 and the authorization of IMI’s Joint Research Center for Advanced and Fundamental Mathematics-for-Industry as a MEXT Joint Usage / Research Center in April 2013, hereafter the MI Lecture Notes Series will publish lecture notes and proceedings by worldwide researchers of MI to contribute to the development of MI.

October 2018
Osamu Saeki
Director
Institute of Mathematics for Industry

IMI Workshop of the Joint Usage Research Projects

Exploring Mathematical and Practical Principles of Secure Computation and Secret Sharing

MI Lecture Note Vol.85, Institute of Mathematics for Industry, Kyushu University
ISSN 2188-1200
Date of issue: February 9, 2022
Editors: Hiroaki Anada, Yasuhiko Ikematsu, Koji Nuida, Satsuya Ohata, Yuntao Wang
Publisher:
Institute of Mathematics for Industry, Kyushu University
Graduate School of Mathematics, Kyushu University
Motooka 744, Nishi-ku, Fukuoka, 819-0395, JAPAN
Tel +81-(0)92-802-4402, Fax +81-(0)92-802-4405
URL https://www.imi.kyushu-u.ac.jp/

Preface

As operation of the ultra-high speed and ultra-low delay fifth generation communication service begins in countries of the world, the expectation to a cryptographic technology increases in our society. For example, there is demand of treating data with a guarantee that no leakage of private information arise in the analysis handling customer data across their organizations. To meet the nee, secure computation in cryptology is being developed by companies, aiming practical application of commercial level. As another example, secret sharing that can, in theory, attain confidentiality and reliability of cloud storage is being developed to obtain more availability and efficiency. However, these developments are at an intermediate point of the spiral intertwined with research activity.

For the techniques of secure computation and secret sharing to be taken in and to be used actually, mathematical investigation, rigorous security proofs and recapturing usage performance are indispensable. Especially, the following directions are important from the point of view of mathematics: (1) classifying mathematical approaches such as abstract algebra, information theory, coding theory, combinatorics and game theory; (2) mitigating assumptions of security, that is, semi-honest adversaries versus active adversaries, computational security versus information-theoretic security, etc.; (3) improving efficiency, that is, decreasing computational amount, communication cost, the number of rounds and complexity of randomness.

The purpose of this workshop was to gather researchers in industry and academia in order to share their experience on mathematical approaches and practical implementations of secure computation and secret sharing for securing distributed data processing and data storage. Then the participants discussed the actual problems which the industry was facing when implementing the cryptographic technologies. Also, they discussed the appropriate solutions. The workshop consisted of the invited lectures and tutorials on recent results of secure computation and secret sharing. We hope that this lecture note would help readers obtain some intuition in the technologies.

Hiroaki Anada, Representative of the Organizers

Acknowledgements

This work was supported by 2021 IMI Joint Use Research Program Workshop (I) "Exploring Mathematical and Practical Principles of Secure Computation and Secret Sharing".

IMI Joint Research Project in 2021

Exploring Mathematical and Practical Principles of Secure Computation and Secret Sharing

Date:

November 8(Mon)-10(Wed), 2021

Keynote speaker:

Johannes BUCHMANN, Technische Universität Darmstadt
"Cryptographic long-term security"

Invited speakers:

Reo ERIGUCHI, The University of Tokyo
Keitaro HIWATASHI, The University of Tokyo
Kosuke KANEKO, Robert T.Huang Entrepreneurship Center of Kyushu University
Yi LU, Tokyo Institute of Technology
Ibuki MISHINA, NTT Social Informatics Laboratories
Kirill MOROZOV, University of North Texas
Hikaru TSUCHIDA, NEC Corporation

Venue: Online

■Organizing Committee ‣ Hiroaki ANADA, University of Nagasaki
Yasuhiko IKEMATSU, Institute of Mathematics for Industry, Kyushu University
Koji NUIDA, Institute of Mathematics for Industry, Kyushu University
Satsuya OHATA
Yuntao WANG, Japan Advanced Institute of Science and Technology

■Sponsored by ‣ Institute of Mathematics for Industry, Kyushu University
■Registration fee ‣ Free

https://www.imi.kyushu-u.ac.jp/kyodo-riyo/research_meetings/view/30

Contact : imikyoten@jimu.kyushu-u.ac.jp (For general inquiries) Institute of Mathematics for Industry, Kyushu University

九州大学 IMI 共同利用・研究集会（I）

秘密計算・秘密分散の数理と実用の探求

Exploring Mathematical and Practical Principles of Secure Computation and Secret Sharing

日　　時： 2021 年 11 月 08 日（月）16:00 ～ 18:25
2021 年 11 月 09 日（火）09:00 ～ 11:25
2021 年 11 月 10 日（水）16:00 ～ 18:15
場　　所： Zoom によるオンライン開催
組織委員　：
- Hiroaki Anada (University of Nagasaki) （研究代表者）
- Yasuhiko Ikematsu (IMI, Kyushu University)
- Koji Nuida (IMI, Kyushu University)
- Satsuya Ohata
- Yuntao Wang (Japan Advanced Institute of Science and Technology)

プログラム

11 月 08 日（月）

16:00-16:05
オープニング

16:05-16:55
講演者：Johannes Buchmann (Technische Universität Darmstadt)
講演タイトル：“Cryptographic Long-Term Security”

17:05-17:40
講演者：Yi Lu (Tokyo Institute of Technology / National Institute of Advanced Industrial Science and Technology)
講演タイトル： “Efficient Two-party Exponentiation from Quotient Transfer”

17:50-18:25
講演者：Hikaru Tsuchida (NEC Corporation)
講演タイトル：“General-purpose Compiler for Secure Three-party Computation and Its Application to Prediction by Machine Learning Model”

11 月 09 日（火）

09:00-09:05
第 2 日オープニング

09:05-09:55
講演者：Kirill Morozov (University of North Texas)
講演タイトル：“Evolving Secret Sharing From Evolving Perfect Hash Families”

10:05-10:40
講演者：Ibuki Mishina (NTT Social Informatics Laboratories)
講演タイトル：“Secure-Computation AI : a Python Library for Machine Learning in Secure Computation”

10:50-11:25
講演者：Kosuke Kaneko (Robert T.Huang Entrepreneurship Center of Kyushu University)
講演タイトル：“Possibility of Secret Sharing using EtherCAT”

11月10日（水）

16:00-16:05
第3日オープニング

16:05-16:40
講演者：Yasuhiko Ikematsu (Institute of Mathematics for Industry)
講演タイトル：“An Indeterminate Equation Scheme having Homomorphic Property”

16:50-17:25
講演者：Reo Eriguchi (The University of Tokyo)
講演タイトル：“Homomorphic Secret Sharing for Multipartite and General Adversary Structures Supporting Parallel Evaluation of Low-Degree Polynomials”

17:35-18:10
講演者：Hiroaki Anada (University of Nagasaki)
講演タイトル：“A Comparison of How to Garble Arithmetic and Boolean Circuits”

18:10-18:15
クロージング

最新情報及び参加情報は下記URL(QRコード)のウェブサイトにて御確認下さい.
https://www.imi.kyushu-u.ac.jp/kyodo-riyo/research_meetings/view/30

Contents

Cryptographic Long-term Security

Johannes Buchmann
Technische Universität Darmstadt
johannes.buchmann@tu-darmstadt.de

Digitization is omnipresent and all important areas of our private, political, social, and economic lives depend on it. As a result, digitization must meet ever greater security requirements. In particular, security must be guaranteed for a very long period of time. One important technology that enables cybersecurity is cryptography. In the talk, I talk about how cryptography can enable long-term protection and the important role Secret Sharing plays in this.

Cryptographic Long-Term Security

Johannes Buchmann, TU Darmstadt

CROSSING

The challenge

Japan Agency for Medical Research and Development

AMED

Programs | Project for Genome and Health Related Data

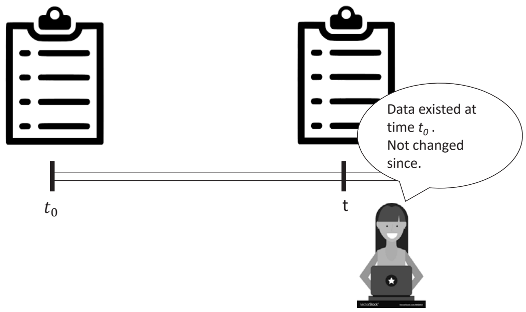
Requires long-term protection!

Overview

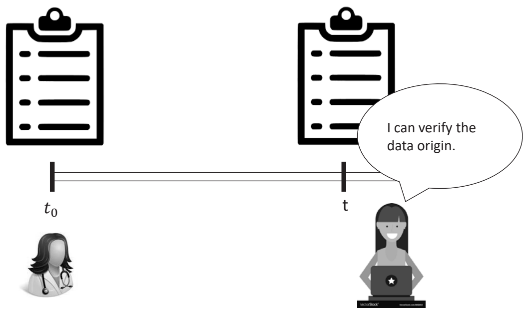
Toward realization of personalized medicine, this project promotes the development and utilization of genome data infrastructures and supports R&D that contributes to the prevention, diagnosis, and treatment of diseases based on the relationship between genetic mutations and polymorphism and the development of diseases, from a life-stage perspective.

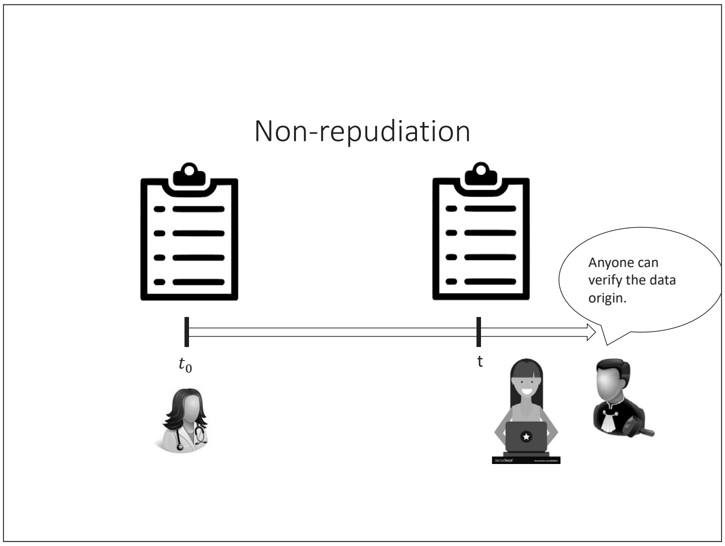
Protection goals

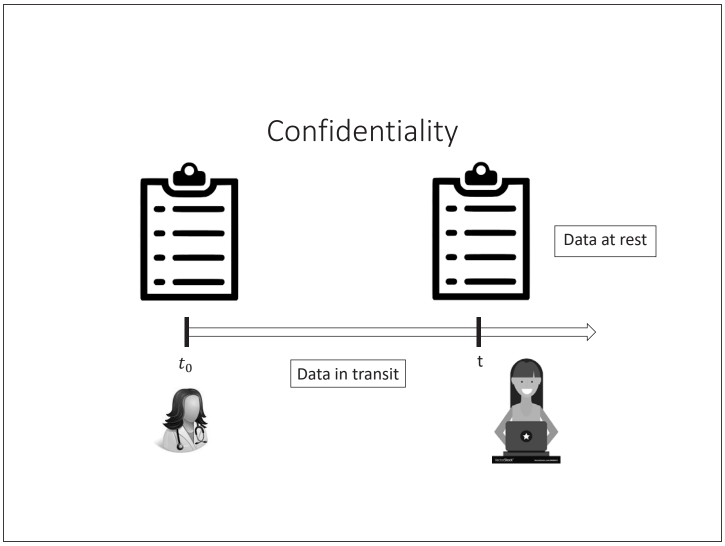
Integrity



Authenticity




Non-repudiation
Anyone can verify the data origin.
t_0
t


Confidentiality
Data at rest
t_0
Data in transit
t


Availability

Protection by cryptography

Protection goal	Cryptographic method
Confidentiality	Encryption + key exchange
Integrity	Hash, MAC, digital signature
Authenticity	MAC, digital signature
Non-repudiation	Digital signature
Availability	----------------------

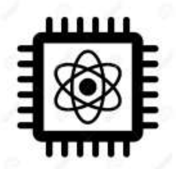
Today's cryptography is complexity-based

Cryptographic method	Algorithms	Hard problem
Key Exchange + Encryption	RSA/Diffie-Hellman AES	Factoring/Discrete Logarithm AES
Hash, MAC	SHA-3, HMAC	SHA-3
Digital signature	RSA, ECDSA	Factoring, EC-DL

Today's complexity-based cryptography is not sustainable

Algorithm	Standardized	Broken	By	Lifetime
DES	1977	1997	Brute force	20 years
Diffie-Hellman	1999	2030?	Quantum computer	31 years?
MD5	1992	1996/2004	Analysis of algorithm	4 years
RSA	1991	2030?	Quantum computer	39 years?
RSA-512		2000	Number Field Sieve	9 years
ECDSA	2005	2030?	Quantum computer	25 years?

Aspects of cryptographic long-term security



Cryptography that can resist new attacks

Long-term protection of data

Availability of Cryptography

Since 2017

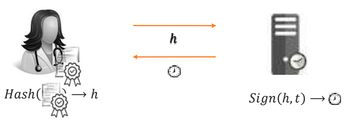
Long-term protection of data

Confidentiality | Integrity | Authenticity | Non-repudiation

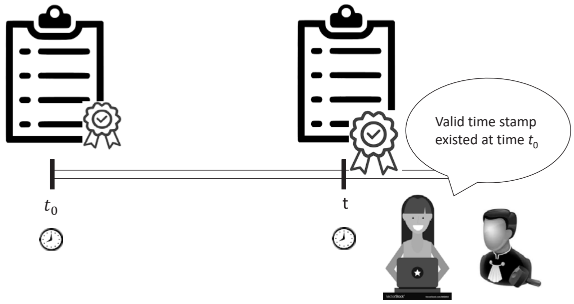
Protection time: human life expectancy – one century

Integrity, authenticity, non-repudiation

Time stamp



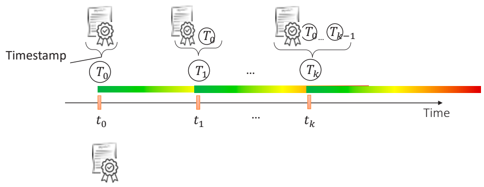
Integrity, authenticity, non-repudiation



Time stamps use signatures
which are not long-term

Time stamp security can be prolonged!

Timestamp chain – RFC 4998



Challenges

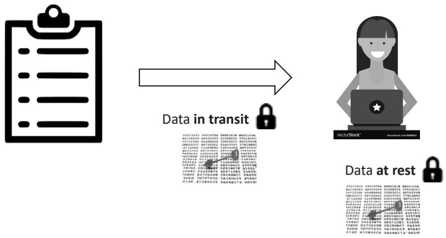
Hash value may reveal information about signed data

Sudden break of cryptography

Format change

Confidentiality

Encryption



Long-term secure?

CYBERCRIME MAGAZINE

The World Will Store 200 Zettabytes Of Data By 2025

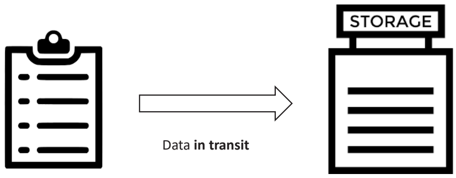
200 Zettabytes = $2*10^{23}$ Bytes ~ 200 TB/Person in the world

Cyphertexts may be stored now and can be decrypted later

Encryption security cannot be prolonged

Information theoretic confidentiality protection is required

Data in transit



Quantum Key Distribution

+

One-Time Pad encryption

IEEE Spectrum

NEWS | TELECOMMUNICATIONS

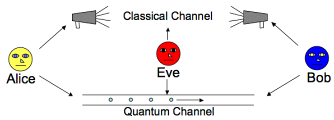
Quantum Crypto Crams Into System-on-a-Chip ›

Toshiba reports the first photonic chip to deliver full-stack quantum key distribution

BY CHARLES Q. CHOI | 27 OCT 2021 | 2 MIN READ

Challenges

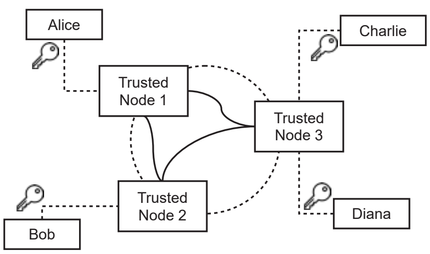
QKD is point-to-point



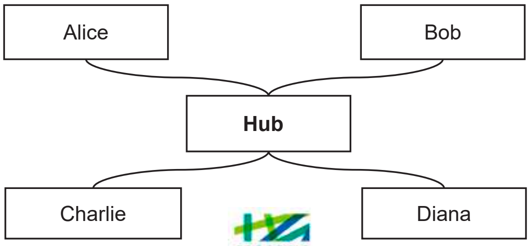
Trusted nodes required

Solutions to point-to-point challenge

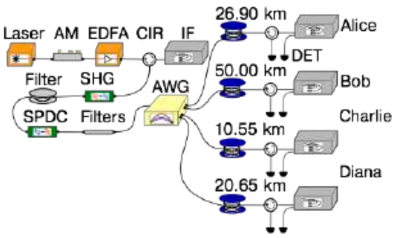
Backbone with pre-shared keys



Quantum key hub

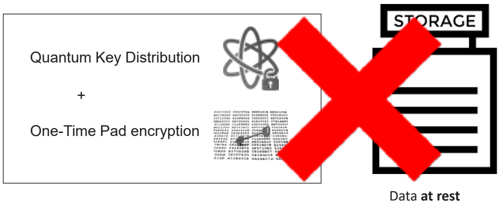


Quantum Key Hub Darmstadt



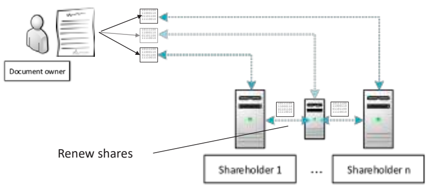
Walther et. al. 2021

Data at rest



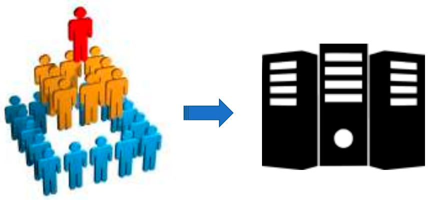
Solution: secret sharing

Proactive Secret Sharing



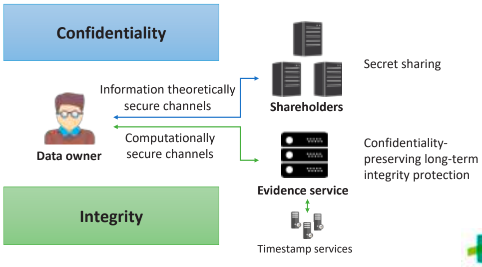
Herzberg 1995

Hierarchical secret sharing



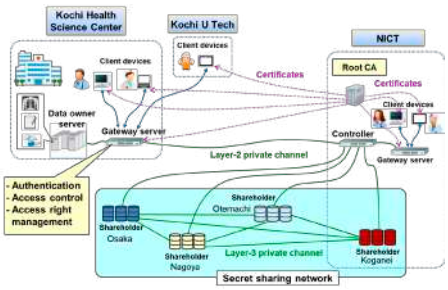
Traverso et al. 2016

LINCOS architecture



Geihs et al. 2016

H-LINCOS with NICT



Availability

How can historians access encrypted data?

Conclusion

Security models

Provably and efficient secure LT private channels

LT integrity protection preserving confidentiality

Complex and private access structures

An architecture for LT confidentiality and integrity

Verifiable and private outsourced computation

Format change

?

Sudden break of cryptography

?

High performance LT key exchange

?

Long-term availability

?

Thank you very much for your attention!

Efficient Two-party Exponentiation from Quotient Transfer

Yi Lu (Joint work with Keisuke Hara, Kazuma Ohara, Jacob Schuldt, and Keisuke Tanaka)

Tokyo Institute of Technology / National Institute of Advanced Industrial Science and Technology

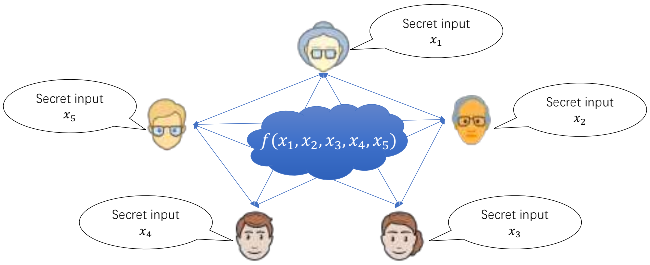
lu.y.ai@m.titech.ac.jp

Secure multi-party computation (MPC) allows participating parties to jointly compute a function over their inputs while keeping them private. In particular, MPC based on additive secret sharing has been widely studied as a tool to obtain efficient protocols secure against a dishonest majority, including the important two-party case. In this paper, we propose a two-party protocol for an exponentiation functionality based on an additive secret sharing scheme. Our proposed protocol is based on a new simple but efficient approach involving quotient transfer that allows the parties to perform the most expensive part of the computation locally. Our protocol requires 6 rounds and 4 invocations of multiplication. This is the first two-party protocol for an exponentiation functionality with constant-round efficiency based on an additive secret sharing scheme. As an intermediate primitive for our efficient two-party exponentiation protocol, we propose an efficient modulus conversion protocol, which may be of independent interest.

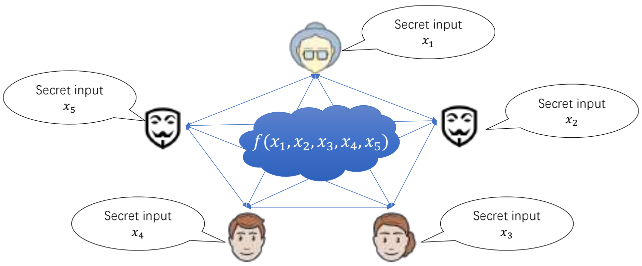
Efficient Two-party Exponentiation from Quotient Transfer

LU YI (Tokyo Tech/AIST)
Hara Keisuke (Tokyo Tech/AIST)
Ohara Kazuma (AIST)
Jacob Schuldt (AIST)
Tanaka Keisuke (Tokyo Tech)

Multi Party Computation (MPC)



Multi Party Computation (MPC)



Basic Properties for MPC

- Correctness : the function is computed correctly
- Security : Only the output is revealed

Modeling Adversaries

- Adversarial Behavior
  - Semi-honest : follows the protocol specification
  - Malicious : follows any arbitrary strategy

- Adversarial power
  - Polynomial-time : computational security
  - Computationally unbounded :
    information-theoretic security

Modeling Adversaries

- Adversarial Number
  - Honest-Majority: The number of honest party is over half of all participants
  - Dishonest-Majority: The number of adversary is over half of all participants

Semi-honest and Computationally unbounded

The performance of MPC

- Rounds: The number of communications, which can be done simultaneously will be counted as one round
- Communication Complexity: The number of bits which are transferred in communication
  The number of invocations of Multiplication.
- Computation Complexity: The number of computations which is done in protocol

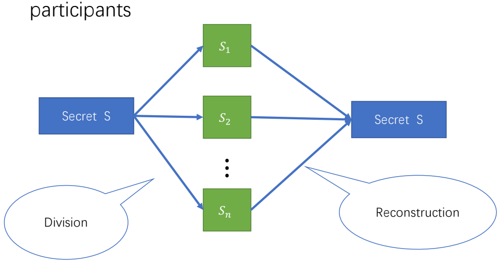
Multi Party Computation (MPC)

Method	Communication Complexity	Rounds	Computation Complexity
Secret Sharing	small	big	small
FHE	big	small	big
Garbled Circuit(GC)	big	small	normal-big

Table 2. Three Method to Implement MPC

Secret Sharing

A method for distributing a secret among a group of participants



(k,n) threshold Secret Sharing Scheme

- Divide secret data (D) into pieces (n)
- Knowledge of some pieces (k) enables to derive secret data (D)
- Knowledge of any pieces (k-1) makes secret data (D) completely undetermined.

Such a scheme is called a (k,n) threshold scheme

Variants of Secret Sharing Scheme

Shamir' s Secret Sharing

s *is secret*

$f(x) = a_{t-1}x^{t-1} + \cdots + a_1 x + a_0$

$a_0 = s,$

select $a_i (i = 1, \ldots, t)$ *randomly*

share $s_i = f(i)\ (i = 1, \ldots, n)$

Additive Secret Sharing

$x \in \mathbb{G}$ *is secret*

$x_1 + \cdots + x_n \equiv x \bmod p$

share $x_i \in \mathbb{G}, (i = 1, \ldots, n)$

	Threshold	Set	Core Technique
Shamir' s Secret Sharing	k<n/2	$\mathbb{F}_p$ (p: prime)	Lagrange interporation
Additive Secret Sharing	n-1	$\mathbb{G}$ (Finite Additive Group)	

Merits of Additive Secret Sharing

$$x \in \mathbb{G} \qquad [x]_1 + [x]_2 + \cdots + [x]_n \equiv x \bmod p$$

$[x] \leftarrow share(x),\ [x] = ([x]_1, [x]_2, \ldots, [x]_n) \qquad x \leftarrow Reconstruction([x])$

- **Merits : The compatibility with well-known dishonest-majority frameworks.**
  - [Bea92] **"Beaver triple" is a well-known and easy way to introduce multiplication in the dishonest-majority setting**
  - **To the best of our knowledge, all known efficient offline protocols generating Beaver triples are designed for additive secret sharing,** [DPSZ12, DKL+13, KPR18], [ALSZ15,KOS16]

Addition and Multiplication in MPC

Party 1 generates $[a]$

Party 2 generates $[b]$

	formulation	Communication
Addition	$[a]+[b]=[a+b]$	Local (0 round)
Multiplication	$[a]\cdot[b]=[a\cdot b]$	1 round

In general, all of the computations are implemented by using addition and multiplication.

Research Background

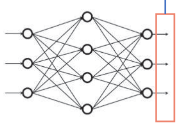
Background:
The meaning of Exponentiation MPC

2 party	【RSC+19】	【MLS+20】	【CVA18】
3 party	【KRC+20】	【AA20】	【CCPS19】
4 party	【BCP+20】	【CRS20】	

Table 3. Researches on Deep learning MPC Protocol

Softmax function

$$\sigma(z)_j = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}}, for\ j = 1, \ldots K$$



[RSC+19] M. Sadegh Riazi, Mohammad Samragh, Hao Chen, Kim Laine, KristinLauter, Farinaz Koushanfar. XONN: XNOR-based Oblivious Deep Neural Network Inference. USENIX Conference on Security Symposium 2019.
[KRC+20] Nishant Kumar, Mayank Rathee, Nishanth Chandran, Divya Gupta, Aseem Rastogi, Rahul Sharma. CrypTFlow: Secure TensorFlow Inference. IEEE S&P 2020.
[BCP+20] Megha Byali, Harsh Chaudhari*, Arpita Patra, and Ajith Suresh. FLASH:Fast and Robust Framework for Privacy-preserving Machine Learning. PoPETs 2020.
...

Background:
3 types of Exponentiation MPC

- Public Base: $a^{[x]}$
- Public Exponent: $[a]^{x}$
- Private Exponentiation: $[a]^{[x]}$

Our work consider the setting public base

$$\sigma(z)_j = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}}, for\ j = 1, \ldots K$$

Normal method to Compute EXP MPC:
Bit-decomposition (BD)

Convert the input of arithmetic circuits into ones of Boolean circuits

$$[x]_p \longrightarrow \boxed{\text{BD}} \longrightarrow [x]_B = [x_0]_p \ldots [x_{l-1}]_p$$

such as $x \in \mathbb{F}_p$

such as $x = \sum_{i=0}^{l-1} 2^i x_i, x_i \in \{0,1\}$

[DFK+06]

Round complexity: $O(1)$
Invocations of multiplication: $O(l \log l)$

Normal method to Compute EXP MPC:
Bit-decomposition (BD)

For example: [DFK+06]

$$[x]_B = [x_0]_p \ldots [x_{l-1}]_p, \sum_{i=0}^{l-1} 2^i x_i, x_i \in \{0,1\}$$

$$a^x = a^{\sum_{i=0}^{l-1} 2^i x_i} = \prod_{i=0}^{l-1} a^{2^i x_i} = \prod_{i=0}^{l-1} (x_i a^{2^i} + 1 - x_i)$$

Round complexity: $O(1)$
Invocations of multiplication: $O(l \log l)$

Existing Exponentiation protocol without BD

Protocol	Bit-Decomposition	Rounds	Multiplication**		Tool
[DFK+06]	Yes	119	$\mathcal{O}(\ell \log \ell)$	50176	Linear secret sharing
[NX11]*	No	20	$\mathcal{O}(\ell)$	10508	Linear secret sharing
[AAN18]	No	3	$\mathcal{O}(1)$	6	Shamir's secret sharing

Table 1. Comparison between two-party exponentiation protocols.

* The proposed protocol is a private exponent type protocol, not a public base type protocol. As the former implies the later, in our comparison, we use their private exponent type protocol as a public base type.
** We consider the case $\ell = 64$ when estimating the number of multiplications.

Motivation:

Without Bit-decomposition Additive secret sharing EXP MPC

New Framework for Modular Exponentiation Protocol

Naive Approach $a^x < p$

$[a^x]_p \leftarrow \mathcal{F}_{EXP}(a, [x]_p)$

Party A

Party B

$[x] \leftarrow share(x)$
$[x]_p^1 + [x]_p^2 \equiv x \bmod p$

Party A	Messages	Party B
Input : $a, [x]_p^1$	← $[x] \leftarrow share(x)$, $[x]_p^1+[x]_p^2 \equiv x \bmod p$ →	Input : $a, [x]_p^2$
Locol: $y_1 = a^{[x]_p^1}$		Locol: $y_2 = a^{[x]_p^2}$
Share: $[y_1] \leftarrow share(y_1)$, $[y_1] = \{[y_1]_p^1, [y_1]_p^2\}$	$[y_1]_p^2$ → ; ← $[y_2]_p^1$	Share: $[y_2] \leftarrow share(y_2)$, $[y_2] = \{[y_2]_p^1, [y_2]_p^2\}$
Mult: $[a^x] = [y_1] \cdot [y_2]$	↔	Mult: $[a^x] = [y_1] \cdot [y_2]$

Problem of Naive Approach

$$[x]_p^1 + [x]_p^2 \equiv x \; mod \; p \qquad a^x < p$$

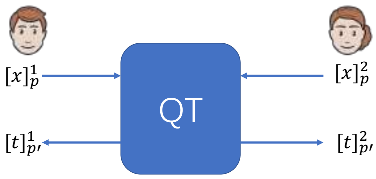
For example: $4 + 4 \equiv 3 \; mod \; 5$
$e^4 \cdot e^4 = e^8 \neq e^3$

$$e^{[x]_p^1 + [x]_p^2} = \begin{cases} e^x & [x]_p^1 + [x]_p^2 \leq p \\ e^{[x]_p^1 + [x]_p^2 - p} = e^{x+p} = e^{x+1} & else \end{cases}$$

$$= e^{x+tp} = e^{x+t} \quad (t \in \{1,0\})$$

Quotient Transfer Functionality

When $[x]_p^1 + [x]_p^2 \equiv x \; mod \; p$, which means
$[x]_p^1 + [x]_p^2 = x + tp$, $(t \in \{1,0\})$



When $[x]_p^1 + [x]_p^2 \leq p, t = 0$, else $t = 1$

Constrained Quotient transfer Protocol

Constrain means x is even.

$$x = 4, p = 7, p' = 11$$

$$[x]_7^1 + [x]_7^2 = 4 mod \; 7$$

$[x]_7^1 = 5$ $\qquad$ $[x]_7^2 = 6$

$\because [x]_7^1 + [x]_7^2 = 11 > 7$ $\qquad$ $\therefore t = 1$

$[LSB([x]_7^1)]_{11} = 1$ $\qquad$ $[LSB([x]_7^2)]_{11} = 0$

$$[LSB([x]_7^1)]_{11} + [LSB([x]_7^2)]_{11} - 2 \times [LSB([x]_7^2)]_{11} \times [LSB([x]_7^1)]_{11} = 1$$

Constrained Quotient transfer Protocol

Constrain means x is even.

P_1 | P_2

$[LSB([x]_p^1)]_{p'} = ([LSB([x]_p^1)]_{p'}^1, [LSB([x]_p^1)]_{p'}^2)$

$[LSB([x]_p^2)]_{p'} = ([LSB([x]_p^2)]_{p'}^1, [LSB([x]_p^2)]_{p'}^2)$

$$\xrightarrow{[LSB([x]_p^1)]_{p'}^2}$$

$$\xleftarrow{[LSB([x]_p^2)]_{p'}^1}$$

$[t]_{p'}^1 = [LSB([x]_p^1)]_{p'}^1 \oplus [LSB([x]_p^2)]_{p'}^1$

$[t]_{p'}^2 = [LSB([x]_p^1)]_{p'}^2 \oplus [LSB([x]_p^2)]_{p'}^2$

$$\because [x]_p^1 + [x]_p^2 = \begin{cases} x & [x]_p^1 + [x]_p^2 \le p \\ x+p & else \end{cases} = x + tp$$

$$\therefore t = LSB([x]_p^1) \oplus LSB([x]_p^2) = \begin{cases} 0 & [x]_p^1 + [x]_p^2 \le p \\ 1 & else \end{cases}$$

Constrained Quotient transfer Protocol

Algorithm 3 Our Constrained Quotient Transfer Protocol Π_{QT}

Input: $[x]_p, p'$

Output: $[t]_{p'}$

Each $P_i (i \in \{0,1\})$ computes $b_i = \text{LSB}([x]_p^i)$.

Each $P_i (i \in \{0,1\})$ computes $[b_i]_{p'} \leftarrow \textbf{Share}(b_i, p')$.

Each $P_i (i \in \{0,1\})$ sends $[b_i]_{p'}^{1-i}$ to P_{1-i}.

$[t]_{p'} = [b_0]_{p'} + [b_1]_{p'} - 2 \cdot [b_0]_{p'} \cdot [b_1]_{p'}$

Output $[t]_{p'} = ([t]_{p'}^0, [t]_{p'}^1)$

Exponentiation protocol $a^x < p$

$[x]_p^1 + [x]_p^2 \equiv x \bmod p$, x is even $\quad a^{[x]_p^1 + [x]_p^2} = a^{x+tp} = a^{x+t} \quad (t \in \{1,0\})$

Algorithm 1 Our framework for exponentiation protocol Π_{EXP}

Input: $a, [x]_p$

Output: $[o]_p$

Each $P_i (i \in \{0,1\})$ locally computes $y_i = a^{[x]_p^i}$

Each $P_i (i \in \{0,1\})$ generates $[y_i]_p \leftarrow \textbf{Share}(y_i)$

Each $P_i (i \in \{0,1\})$ sends $[y_i]_p^{1-i}$ to P_{1-i}.

$[d]_p \leftarrow [y_0]_p \cdot [y_1]_p$

$[t]_p \leftarrow \mathcal{F}_{\text{QT}}([x]_p, p)$

$[o_1]_p \leftarrow (1 - [t]_p)[d]_p$,

$[o_2]_p \leftarrow [t]_p [d]_p$

$[o]_p \leftarrow [o_1]_p + [o_2]_p (a)^{-1}$

If $t = 1$, $[x]_p^1 + [x]_p^2 > p$, $d = a^{x+1}$, $o_1 = 0$, $o_2 = d = a^{x+1}$
$o = o_2 a^{-1} = a^x$

If $t = 0$, $[x]_p^1 + [x]_p^2 \le p$, $d = a^x$, $o_1 = d = a^x$, $o_2 = 0$
$o = o_1 = a^x$

Problem 2： assumption

$$a^x = \sqrt{a}^{2x}$$

$\sqrt{a}$ dose not always exist in $\mathbb{Z}_p$

We need b and p, which satisfy

$$a = b^2 \bmod p' \quad a^x < p'$$

Assume we can find such b and p'

Conversion protocol

$$[x]_{p'} \leftarrow [x]_p$$

Algorithm 2 Our modulus conversion protocol Π_{Conv}

Input: $[x]_p, p'$
Output: $[x]_{p'}$
1: $[t']_{p'} \leftarrow \mathcal{F}_{\mathrm{QT}}([x]_p, p')$
2: Each $P_i (i \in \{0,1\})$ sets $[x]^i_{p'} = [x]^i_p - [t']^i_{p'} \cdot p$
3: Output $[x]_{p'}$

Correctness

$$[x']^i_{p'} = [x]^i_p - [t]^i_{p'} \times p$$

$$[x']^1_{p'} = [x]^1_p - [t]^1_{p'} \times p \qquad [x']^2_{p'} = [x]^2_p - [t]^2_{p'} \times p$$

$$[x']^1_{p'} + [x']^2_{p'} \bmod p' = [x]^1_p - [t]^1_{p'} \times p + [x]^2_p - [t]^2_{p'} \times p \quad \bmod p'$$

$$= [x]^1_p + [x]^2_p - \left([t]^1_{p'} + [t]^2_{p'}\right) \times p \quad \bmod p'$$

$$= x + t \times p - \left([t]^1_{p'} + [t]^2_{p'}\right) \times p \quad \bmod p'$$

Correctness

$$[x']^1_{p'} + [x']^2_{p'} \mod p' = x + t \times p - \left([t]^1_{p'} + [t]^2_{p'}\right) \times p \mod p'$$

$$\begin{cases} = x + t \times p - t \times p \mod p' = x & [t]^1_{p'} + [t]^2_{p'} < p' \\ = x + t \times p - (t + p') \times p \mod p' & \text{else} \end{cases}$$

$$= x$$

Our Exponentiation Protocol

Algorithm 4 Our concrete exponentiation protocol Π'_{EXP}

Input: $a, [x]_p, p'$

Output: $[o]_{p'}$

1: $b := \sqrt{a}$, where $b \in \mathbb{Z}_{p'}$
2: $[2x]_p \leftarrow 2[x]_p$
3: **if** $p \neq p'$ **then**
4: $\quad [2x]_{p'} \leftarrow \Pi_{\mathrm{Conv}}([2x]_p, p')$
5: $\quad v := [2x]_{p'}$
6: **else**
7: $\quad v := [2x]_p$
8: **end if**
9: Output $[o]_{p'} \leftarrow \Pi_{\mathrm{EXP}}(b, v)$

Our Result

Protocol	BD	Rounds	Multiplication†		Tool	Dishonest-Majority¶
[DFK+06]	Yes	119	$O(\ell \log \ell)$	50176	Linear secret sharing	No
[NX11]*	No	20	$O(\ell)$	10508	Linear secret sharing	No
[AAN18]	No	3	$O(1)$	6	Shamir's secret sharing	No
This work (with conversion)§	No	6	$O(1)$	4	Additive secret sharing	Yes
This work (w/o conversion)§	No	4	$O(1)$	3	Additive secret sharing	Yes

Table 1: Comparison between two-party exponentiation protocols.

* The proposed protocol is a private exponent type protocol, not a public base type protocol. As the former implies the later, in our comparison, we use their private exponent type protocol as a public base type.

† We consider the case $\ell = 64$ when estimating the number of multiplications.

§ Here, we consider two cases whether we need a modulus conversion. As mentioned in Section 1.3, in our protocol, if the public base does not have quadratic residue, we require an additional modulus conversion process. In this case, when our modulus conversion is used, we need additional 2 rounds and 1 invocation of multiplication.

¶ In this column, we present that each protocol is compatible with dishonest-majority frameworks.

Open Problem

2 party EXP MPC $\xrightarrow{\text{extend}}$ n party EXP MPC

2 party QT protocol $\longrightarrow$ n party QT protocol

No Efficient

$\boxed{\text{? Efficient n party EXP MPC}}$

Thank you for your listening

General-purpose Compiler for Secure Three-party Computation and Its Application to Prediction by Machine Learning Model

Hikaru Tsuchida
NEC Corporation
h_tsuchida@nec.com

Multiparty computation (MPC) based on a secret sharing scheme (SS-MPC) enables multiple parties to compute an arbitrary function represented as a circuit without revealing parties' inputs. In SS-MPC, each party distributes its inputs as *shares* that look like random numbers among several parties, and the computation proceeds by using shares locally and communicating among the parties. In particular, the secure three-party computation protocol based on a replicated secret sharing scheme (SS-3PC) over the ring [1] has gained attention in recent years because it can perform high throughput even when SS-3PC computes a complex function (e.g., machine learning applications) represented as mixed circuits (which are composed of Boolean and arithmetic circuits). When SS-MPC computes a complex function represented as mixed circuits, efficient share conversion protocols can improve performance. In particular, SS-3PC over $\mathbb{Z}_{2^k}$ can achieve faster share conversions than that over the prime-order field because $\mathbb{Z}_{2^k}$ preserves the structure of the individual bits more than the prime-order field.

While research on protocol design is ongoing, there is still a significant obstacle to implement the applications via MPC due to the high level of expertise required to design a specific MPC execution considering a trade-off between communication and round complexities. Research and development of general-purpose compilers have been actively conducted to mitigate this problem. It can compile the high-level codes to the mixed circuits that MPC computes. Hence, by using the general-purpose compilers, even non-experts of MPC can implement applications based on MPC.

In this talk, we explain one of the general-purpose compilers for SS-3PC, *NEC-SPDZ* and the share conversion protocols over $\mathbb{Z}_2$ and $\mathbb{Z}_{2^k}$ to compute a complex function via SS-3PC by referring to [2]. We also explain the implementation based on NEC-SPDZ and evaluation of the prediction by typical machine learning models, e.g., the decision tree and the hierarchical mixture of experts models via SS-3PC by referring to [2, 3].

References

[1] Toshinori Araki, Jun Furukawa, Yehuda Lindell, Ariel Nof, and Kazuma Ohara. High-Throughput Semi-Honest Secure Three-Party Computation with an Honest Majority. ACM-CCS 2016, pp.805-817.

[2] Toshinori Araki, Assi Barak, Jun Furukawa, Marcel Keller, Yehuda Lindell, Kazuma Ohara, and Hikaru Tsuchida. Generalizing the SPDZ Compiler For Other Protocols. ACM-CCS 2018, pp.880-895.

[3] Yusaku Maeda, Hikaru Tsuchida, Kazuma Ohara, Ryo Furukawa, Isamu Teranishi, and Koji Nuida. Implementation and Evaluation of Prediction by Heterogeous Mixture Models based on Three-Party Secure Computation. SCIS 2020, 3C3-5.

General-purpose Compiler for Secure Three-party Computation and Its Application to Prediction by Machine Learning Model

Hikaru Tsuchida (NEC Corporation)

Agenda

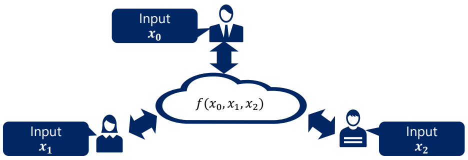
1. Introduction
2. General-purpose compiler
3. Private decision tree evaluation via three-party computation
4. Private hierarchical mixture of experts evaluation via three-party computation
5. Conclusion

Introduction

[A+16] Toshinori Araki, Jun Furukawa, Yehuda Lindell, Ariel Nof, and Kazuma Ohara. High-Throughput Semi-Honest Secure Three-Party Computation with an Honest Majority. ACM-CCS 2016, pp.805-817.
[A+18] Toshinori Araki, Assi Barak, Jun Furukawa, Marcel Keller, Yehuda Lindell, Kazuma Ohara, and Hikaru Tsuchida. Generalizing the SPDZ Compiler For Other Protocols. ACM-CCS 2018, pp.880-895.

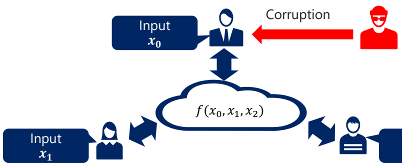
Multiparty computation (MPC)

◆ MPC provides only the outputs of the function to parties **without revealing parties' inputs.**



Multiparty computation (MPC)

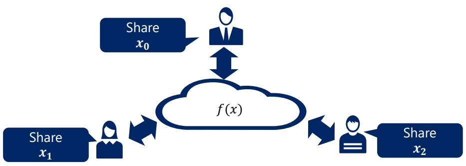
◆ **Even if there are some parties that are corrupted by an adversary** , MPC enables parties to compute the function securely.



MPC based on secret sharing (SS-MPC)

◆ Each party distributes its inputs as **shares** that look like random numbers among several parties.

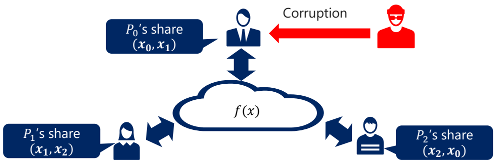
- Example of shares : $x = x_0 + x_1 + x_2 \bmod 2^k \ (k \in \mathbb{N})$



The computation proceeds by using shares locally and communicating among the parties.

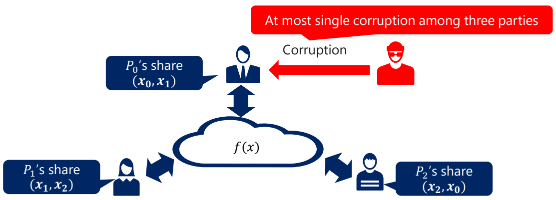
Three-party computation based on replicated secret sharing (SS-3PC)

- SS-3PC [A+16] over the ring has gained attention in recent years because it can achieve high throughput.
  - $x = x_0 + x_1 + x_2 \bmod 2^k \quad (k \in \mathbb{N})$
  - P_i's share over $\mathbb{Z}_{2^k}$: $[x]_{2^k,i} = (x_i, x_{(i+1) \bmod 3})$



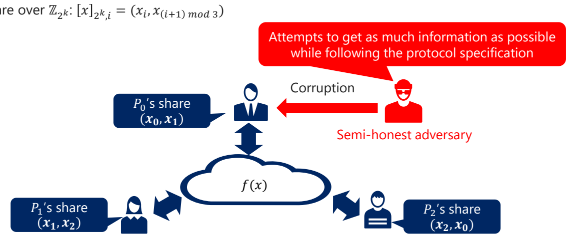
Three-party computation based on replicated secret sharing (SS-3PC)

- SS-3PC [A+16] over the ring has gained attention in recent years because it can achieve high throughput.
  - $x = x_0 + x_1 + x_2 \bmod 2^k \quad (k \in \mathbb{N})$
  - P_i's share over $\mathbb{Z}_{2^k}$: $[x]_{2^k,i} = (x_i, x_{(i+1) \bmod 3})$



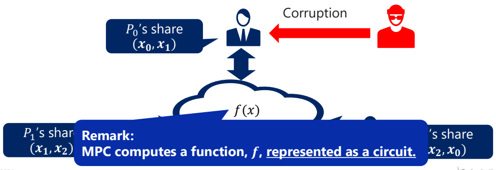
Three-party computation based on replicated secret sharing (SS-3PC)

- SS-3PC [A+16] over the ring has gained attention in recent years because it can achieve high throughput.
  - $x = x_0 + x_1 + x_2 \bmod 2^k \quad (k \in \mathbb{N})$
  - P_i's share over $\mathbb{Z}_{2^k}$: $[x]_{2^k,i} = (x_i, x_{(i+1) \bmod 3})$



Three-party computation based on replicated secret sharing (SS-3PC)

- SS-3PC [A+16] over the ring has gained attention in recent years because it can achieve high throughput.
  - $x = x_0 + x_1 + x_2 \bmod 2^k \ \ (k \in \mathbb{N})$
  - P_i's share over $\mathbb{Z}_{2^k}$: $[x]_{2^k,i} = (x_i, x_{(i+1) \bmod 3})$



Remark:
MPC computes a function, f, represented as a circuit.

Types of circuits

1. Boolean circuit
2. Arithmetic circuit
3. **Mixed circuit (= Boolean & arithmetic circuits)**
   - SS-3PC over the ring computes a complex function (e.g., machine learning application) represented as a mixed circuit **using the share conversion protocols.**

Remark:
We denote the share of x over $\mathbb{Z}_{2^k}$ as $[x]_{2^k} = ([x]_{2^k,0}, [x]_{2^k,1}, [x]_{2^k,2})$.

Toy example requiring share conversion protocols

- ex. Less-than operation

Share over $\mathbb{Z}_{2^k}$: $[a-b]_{2^k} = [a]_{2^k} - [b]_{2^k}$
※ P_i sets $[a-b]_{2^k,i} = (a_i - b_i \bmod 2^k,\ a_{i+1} - b_{i+1} \bmod 2^k)$.

Share conversion **(bit decomposition)**

Array of shares over $\mathbb{Z}_2$: $\left\{[(a-b)|_j]_2\right\}_{j=0}^{k-1}$
※ $a - b = \sum_{j=0}^{k-1} 2^j \cdot (a-b)|_j$ s.t. $(a-b)|_j \in \mathbb{Z}_2$.

MSB extraction

Share over $\mathbb{Z}_2$: $[(a<b)]_2 = [(a-b)|_{k-1}]_2$

Remark:
Since $\mathbb{Z}_{2^k}$ preserves the structure of the individual bits, SS-3PC over $\mathbb{Z}_{2^k}$ can achieve the fast share conversion protocols [A+18].

Obstacle to implement applications via MPC

- While research on protocol design is ongoing, there is still a significant obstacle to implement the applications via MPC.
- It is too hard to describe the complex function as a mixed circuit and implement it **even for an expert.**

Solution to overcome this obstacle is **a general-purpose compiler.**

  \Orchestrating a brighter world NEC

General-purpose compiler

[A+18] Toshinori Araki, Assi Barak, Jun Furukawa, Marcel Keller, Yehuda Lindell, Kazuma Ohara, and Hikaru Tsuchida. Generalizing the SPDZ Compiler For Other Protocols. ACM-CCS 2018, pp.880-895.

\Orchestrating a brighter world NEC

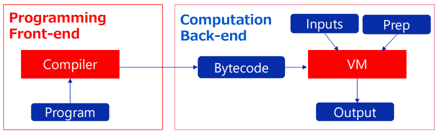
General-purpose compiler

- General-purpose MPC compiler can compile the high-level descriptions into the MPC operations based on various MPC protocols.
  - Ex) **SPDZ** (also known as SCALE-MAMBA), MP-SPDZ, Obliv-C, …
- Researchers and developers around the world are interested in the general-purpose compiler.
  - **SoK paper in IEEE S&P'19**
    - HASTINGS, Marcella, et al. SoK: General purpose compilers for secure multi-party computation. In: *2019 IEEE symposium on security and privacy (SP)*. IEEE, 2019. p. 1220-1237.
  - **Contributed talk in RWC'20**
    - https://rwc.iacr.org/2020/slides/Hastings.pdf

  \Orchestrating a brighter world NEC

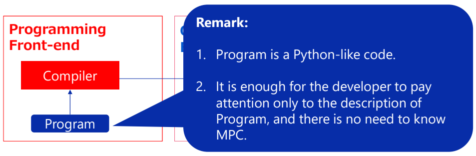
SPDZ (@CRYPTO'14, Eurocrypt'18, etc.)

- SPDZ compiler
  - It takes the high-level description **(Program)** as input and outputs the intermediate code **(Bytecode)** with optimization.
- SPDZ VM
  - It takes the pre-computed values **(Prep)** as inputs, interprets Bytecode as MPC operations based on SPDZ protocols, and run it.



SPDZ (@CRYPTO'14, Eurocrypt'18, etc.)

- SPDZ compiler
  - It takes the high-level description **(Program)** as input and outputs the intermediate code **(Bytecode)** with optimization.
- SPDZ VM
  - It takes the pre-computed values **(Prep)** as inputs, interprets Bytecode as MPC operations based on SPDZ protocols, and run it.



NEC-SPDZ

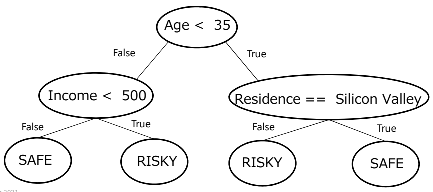
- In [A+18], we extended the SPDZ compiler to work with not only SPDZ protocols but various MPC protocols including SS-3PC.
- **NEC-SPDZ** is the variants of extended SPDZ compiler and VM working with SS-3PC.
  - https://github.com/nec-mpc
- The following applications can run on NEC-SPDZ.
  1. Decision tree evaluation
  2. Hierarchical mixture of experts evaluation

Private decision tree evaluation via three-party computation

[A+18] Toshinori Araki, Assi Barak, Jun Furukawa, Marcel Keller, Yehuda Lindell, Kazuma Ohara, and Hikaru Tsuchida. Generalizing the SPDZ Compiler For Other Protocols. ACM-CCS 2018, pp.880-895.

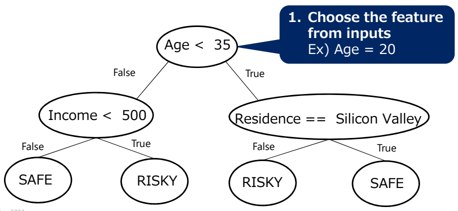
Decision tree

- It is a commonly-used tool for decision support and widely studied in machine learning.
  - ex. credit decision
  - If it takes $(Age, Income, Residence) = (20, 600,\ Tokyo)$ as inputs, then it outputs RISKY.



Decision tree

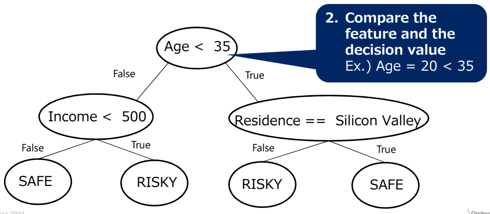
- It is a commonly-used tool for decision support and widely studied in machine learning.
  - ex. credit decision
  - If it takes $(Age, Income, Residence) = (20, 600,\ Tokyo)$ as inputs, then it outputs RISKY.



Decision tree

◆ It is a commonly-used tool for decision support and widely studied in machine learning.

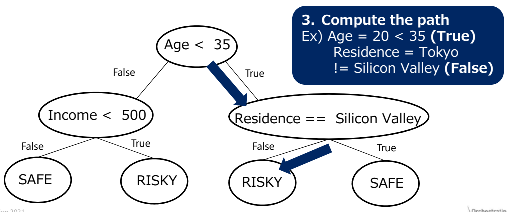
- ex. credit decision
- If it takes $(Age, Income, Residence) = (20, 600,\ Tokyo)$ as inputs, then it outputs RISKY.



Decision tree

◆ It is a commonly-used tool for decision support and widely studied in machine learning.

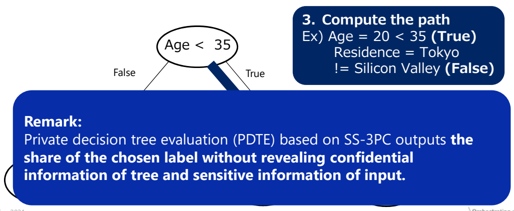
- ex. credit decision
- If it takes $(Age, Income, Residence) = (20, 600,\ Tokyo)$ as inputs, then it outputs RISKY.



Decision tree

◆ It is a commonly-used tool for decision support and widely studied in machine learning.

- ex. credit decision
- If it takes $(Age, Income, Residence) = (20, 600,\ Tokyo)$ as inputs, then it outputs RISKY.



Remark:
Private decision tree evaluation (PDTE) based on SS-3PC outputs **the share of the chosen label without revealing confidential information of tree and sensitive information of input.**

Experimental setting of PDTE

- Task
  - Credit decision
- Structure of tree
  - It is built from real data published for the following paper.
    - Vivek Kumar Singh, Burcin Bozkaya, Alex Pentland, Money Walks: Implicit Mobility Behavior and Financial Well-Being, PLoS ONE 10(8): e0136628. https://doi.org/10.1371/journal.pone.0136628
  - It has 1,256 leaves at depths from 4 to 30.
- Environments
  - AWS m5.12xlarge instances in a single region providing 10Gbps network communication

Single execution of PDTE

- The execution time of PDTE via SS-3PC is shorter than that via the other MPC protocols.

	Resource	SS-3PC ($\mathbb{Z}_{2^k}$)	SPDZ ($\mathbb{F}_q$)	[LN17] ($\mathbb{F}_q$)	BMR (※1)
Security (n: # parties, t: # corruptions)		**Semi-honest** $t < n/2$	Malicious (※2) $t < n$	Malicious (※2) $t < n/2$	Malicious (※2) $t < n$
Online time [sec]	1 core	**0.4641**	0.3005	3.0416	0.5353
# rounds		**2746**	783	584	28
Pre-computation time [sec]	48 cores	**(Not required)**	5.2204 (※3)	(Not required)	1041.8

※1… BMR ran on i3.2xlarge instances. ※2… The malicious security is stronger than the semi-honest security. ※3… It ran on r4.8xlarge instances.
[LN17] Lindell, Yehuda, and Ariel Nof. "A framework for constructing fast MPC over arithmetic circuits with malicious adversaries and an honest-majority." *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*. 2017.

Batch vectorization of PDTE

- We have implemented the batch vectorization of PDTE at a single machine.

	Non-batch [sec]	Batch × 1 [sec]	Batch × 64 [sec]
PDTE via SS-3PC	0.464	2.949	5.945

Non-batch: 0.464 [sec]
Batch × 1 : 2.949 [sec]
⇒ It is slowed down by the overhead of parallel processing.

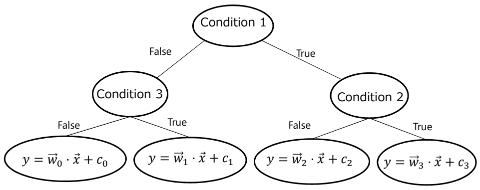
Batch × 1 : 2.949 [sec]
Batch × 64 : 5.945 [sec]
⇒ Even if the parallelism is multiplied by 64, the execution time is **only about 2x.**
⇒ Inference of decision trees can be done in **less than 0.1 seconds on average.**

Private hierarchical mixture of experts evaluation via three-party computation

[M+20] Yusaku Maeda, Hikaru Tsuchida, Kazuma Ohara, Ryo Furukawa, Isamu Teranishi, and Koji Nuida. Implementation and Evaluation of Prediction by Heterogeneous Mixture Models based on Three-Party Secure Computation. SCIS 2020, 3C3-5.

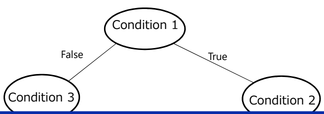
Hierarchical mixture of experts (HME)

- Loosely speaking, HME is almost the same as the decision tree, but differs in that it assigns experts (not labels) to leaves.



Hierarchical mixture of experts (HME)

- Loosely speaking, HME is almost the same as the decision tree, but differs in that it assigns experts (not labels) to leaves.



Remark:
1. An application of HME is electricity demand forecasting.
2. Private HME evaluation (PHMEE) based on SS-3PC outputs **the share of the result of the chosen expert without revealing confidential information of tree and sensitive information of input.**

Experimental setting of PHMEE

- Tree is pre-trained by synthetic data.
- Environments
  - Each server: Intel (R) Xenon (R) CPU E5-2697 v4 @ 2.30 GHz
  - We use three servers providing 10Gbps network communication.

Execution time change of PHMEE by change in # dimension of input vector

- Height of tree : 4
- Number of input data : 10

# Dimension of input vector	Execution time [sec]		
	plaintext	MPC (fixed-point number)	MPC (floating-point number)
1	4.23×10^{-5}	1.68×10^{-1}	1.61
2	4.27×10^{-5}	2.22×10^{-1}	2.36
3	4.54×10^{-5}	2.97×10^{-1}	3.51
4	4.88×10^{-5}	3.43×10^{-1}	4.10
5	5.11×10^{-5}	3.88×10^{-1}	5.06

Remark:
As # dimension increases, the execution time over MPC of fixed-point/floating-point number increases by about 0.056/0.86 seconds.

Execution time change of PHMEE by change in height of tree

- # dimension of input vector : 3
- Number of input data : 10

Height of tree	Execution time [sec]		
	plaintext	MPC (fixed-point number)	MPC (floating-point number)
2	3.64×10^{-5}	6.91×10^{-2}	8.56×10^{-1}
3	3.88×10^{-5}	1.32×10^{-1}	1.56
4	4.55×10^{-5}	2.97×10^{-1}	3.51
5	5.12×10^{-5}	4.88×10^{-1}	5.25
6	5.86×10^{-5}	9.52×10^{-1}	11.8

Remark:
As the height of tree increases, the execution time over MPC increases exponentially.

Execution time change of PHMEE by change in number of input data

- # dimension of input vector : 3
- Height of tree : 3

Number of input data	Execution time [sec]		
	plaintext	MPC (fixed-point number)	MPC (floating-point number)
1	8.87×10^{-6}	3.47×10^{-2}	4.65×10^{-1}
10	4.55×10^{-5}	2.97×10^{-1}	3.51
20	1.10×10^{-5}	4.57×10^{-1}	5.51
30	1.67×10^{-5}	5.05×10^{-1}	7.36
40	1.79×10^{-5}	8.47×10^{-1}	11.6

Remark:
As the number of input data increases by one, the execution time over MPC of fixed-point/floating-point number increases by about 0.019/0.265 seconds.

Conclusion

Conclusion

- SS-3PC [A+16] over the ring has gained attention in recent years because it can achieve high throughput.

- By using the general-purpose compiler, MPC applications (e.g., machine learning applications) can be implemented and evaluated by non-experts.
  - Ex) PDTE [A+18] and PHMEE [M+20]

\Orchestrating a brighter world

NEC creates the social values of safety, security,
fairness and efficiency to promote a more sustainable world
where everyone has the chance to reach their full potential.

\Orchestrating a brighter world

NEC

Evolving Secret Sharing From Evolving Perfect Hash Families

Kirill Morozov

University of North Texas

Kirill.Morozov@unt.edu

The concept of Evolving Secret Sharing introduced by Komargodski, Naor and Yogev [4] puts forward an idea of maintaining secret sharing schemes with potentially infitine number of participants. Specifically, in this framework, new shares are generated for new participants on demand, and no communication with old participants is required.

Armed with the relation between perfect hashing families (PHF) and secret sharing schemes [2, 1, 5], we introduce an evolving (non-abelian) multiplicative secret sharing scheme. An importance of secret sharing over non-abelain groups is that it encompasses, e.g., permutation groups—a basis for MIX operations used, in particular, in electronic voting.

To achieve our goal, we introduce a novel concept of Evolving PHF. In these families, a domain of the hash function is not known in advance, but may be increased in the future—according to a particular application. The framework of Evolving PHF may be of independent interest, and it may encompass other combinatorial objects.

This talk is based on a joint work with Yvo Desmedt and Sabyasachi Dutta [3].

References

[1] Blackburn, S. R., Burmester, M., Desmedt, Y. and Wild, P. R. , “Efficient multiplicative sharing schemes”, Eurocrypt ’96, LNCS 1070 , 107-118 (1996)

[2] Desmedt, Y., Di Crescenzo, G. and Burmester, M., “Multiplicative Non-abelian Sharing Schemes and their Application to Threshold Cryptography”, Asiacrypt ’94: 21-32 (1994)

[3] Desmedt, Y., Dutta, S., Morozov, K. “Evolving Perfect Hash Families: A Combinatorial Viewpoint of Evolving Secret Sharing”, CANS 2019: 291-307 (2019)

[4] Komargodski, I., Naor, M. and Yogev, E., “How to Share a Secret, Infinitely”, TCC (B2) 2016: 485-514 (2016)

[5] Safavi-Naini R., Wang H., “Robust Additive Secret Sharing Schemes over Z_m. Cryptography and Computational Number Theory. Progress in Computer Science and Applied Logic, vol 20: 357-368, Birkhauser (2001)

UNT
UNIVERSITY OF NORTH TEXAS
EST. 1890

Evolving Secret Sharing From Evolving Perfect Hash Families

Kirill Morozov (諸蔵 霧流)
University of North Texas

IMI Workshop of the Joint Research Projects

Exploring Mathematical and Practical Principles of Secure Computation and Secret Sharing

November 9, 2021

Credits

- This presentation is based on a joint work with Yvo Desmedt (University of Texas at Dallas) and Sabyasachi Dutta (University of Calgary), published at CANS 2019

2

Plan of this talk

- Short introduction of UNT
- Secret sharing and its applications
- Evolving secret sharing (Komargodski, Naor, Yogev, TCC 2016)
- Secret sharing from perfect hashing (multiplicative scheme)
- Evolving perfect hash families: definition and construction
    - Implication for secret sharing
- Conclusion and future works

3

University of North Texas (UNT)

A green light to greatness.

- Celebrating 130 years in 2020
  - Tier One research university by the Carnegie Classification

- 40,000 students enrolled
- Location: Denton, Texas
  - Part of the Dallas-Fort Worth metroplex
  - 25 miles to DFW International Airport

4

Cybersecurity research and education at UNT

- **Center for Information and Cyber Security (CICS)** https://cics.unt.edu
  - Research areas: blockchain applications, network security, cloud security, cryptographic protocols, privacy-preserving computation, …
  - NSA/DHS National Center of Academic Excellence (CAE)
- **BSc and MSc in Cybersecurity** offered by the Department of Computer Science and Engineering (CSE), UNT
- **Growing graduate enrollment**, CSE department is now hiring

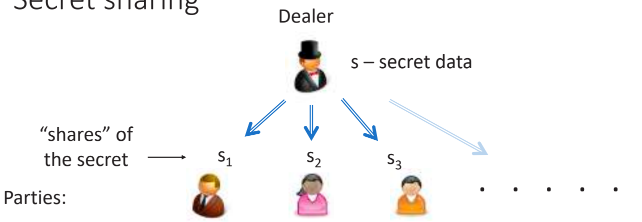
A green light to greatness. UNT 5

Secret sharing and its applications

6

Secret sharing

[Blakley, AFIPS'79]
[Shamir, Commun. ACM'79]



- "Forbidden sets" of shares provide **no information** about the secret
- "Access sets" of shares allow for **efficient reconstruction**

7

Information-theoretic vs. computational security

- Computationally secure cryptographic systems (overwhelming majority of practical protocols, e.g., TLS):
  - Rely on unproven complexity assumptions
  - Threatened by advances in algorithm theory and in computing technologies (e.g., quantum)
  - Require continuous security evaluation and extension of key sizes
- Information-theoretically secure systems rely on **physical assumptions** (part of communication model)

8

Applications of secret sharing

Building block for:

- Threshold cryptography
- Multi-party computation
- Perfectly secure message transmission
- …

"Multi-signatures" used in cryptocurrencies

E-voting
E-auctions
E-commerce
E-government

Stand-alone protocol for:

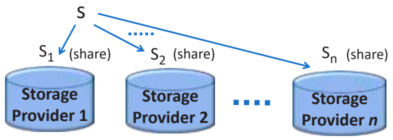
- Secure and reliable storage (e.g., in the cloud setting)
- …

9

Application of secret sharing for cloud storage

User data

- Goal: Security + reliability



- Example: SecureSlice, a component of IBM Cloud Object Storage

10

Shamir (k,n) Threshold Secret Sharing

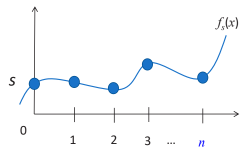
[Shamir, Commun. ACM 22(11) '79]

- **Share Generation:** Dealer chooses

  $f_s(x) \in_R \mathrm{F}_p[\mathrm{X}]$: $\deg(f_s) \leq k\text{-}1$,

  $f_s(x) = s + b_1 x + b_2 x^2 + \ldots + b_{k-1} x^{k-1}$

  - s is the secret, $b_j \leftarrow_R \mathrm{F}_p$, $1 \leq j \leq k\text{-}1$, $p > n$
  - shares are $s_i = f_s(i)$, $1 \leq i \leq n$



- **Reconstruction:** Using Lagrange interpolation, any subset of k parties can compute the secret s

11

Example application: Long-term storage

- As new storage providers emerge and old ones go out of business, it would be convenient for the data owner (dealer) to keep adding providers (parties) on the rolling basis
- Problem: Scalability, as we need p > # parties for Shamir's scheme
- What if the number of parties is not known in advance?
  - E.g., potentially infinite
- May choose a large "p" but still cannot support infinitely many parties

12

Solutions

- [Cachin '95], [Csirmaz and Tardos, '12]: On-line secret sharing
  - # of authorized sets a party can join is bounded
- [Komargodski, Naor, Yogev '16]: Evolving secret sharing
  - More efficient, no limitation as above
  - Many follow-up works

13

Comparison of secret sharing protocols

- Secret redistribution (Wong, Wing, Wang '02):
  The parties change access structure
  without involving the dealer (e.g., enroll new members)
  - Note: No secret reconstruction is done

- Evolving secret sharing: *The dealer* adds parties on demand
  - Note: The dealer knows the secret
    (or can reconstruct it using existing parties)

14

Evolving secret sharing (KNY16)

- General evolving access structures:
  The size of T-th participant's share is 2^{T-1}
- Evolving k-threshold's share size:
  $\sigma'(T) = \log T + (k-1)\cdot\max\{ \log T + k , \sigma(\log T +k) \}$,
  where the share size of the base scheme is $\sigma(t)$
  - When Shamir secret sharing is the base scheme
- Let us consider this construction for k=2
  - (Any two parties can reconstruct the secret)

15

KNY16 (2,∞)-threshold scheme

- Secret $s \in GF(p)$; $b_i \leftarrow_R GF(p)$
- Divide parties in "generations"; gen. g has Size(g) = 2^g
  - Party T belongs to generation $g = \lfloor \log T \rfloor$
- (k,n)-threshold Shamir sharing of $s \in GF(p)$ is denoted by Sh(k,n)(s)
- Gen 0 : P_1 [b_1]
- Gen 1 : P_2 [$s + b_1$, b_2 , $Sh_1(2,2)(s)$] ; P_3 [$s + b_1$, b_2 , $Sh_1(2,2)(s)$]
- Gen 2 : P_4 [$s + b_1$, $s + b_2$, b_3 , $Sh_2(2,4)(s)$] ; P_5 [$s + b_1$, $s + b_2$, b_3 , $Sh_2(2,4)(s)$]
  P_6 [$s + b_1$, $s + b_2$, b_3 , $Sh_2(2,4)(s)$] ; P_7 [$s + b_1$, $s + b_2$, b_3 , $Sh_2(2,4)(s)$]

(2,∞)-threshold scheme: Correctness and security (sketch)

Gen 0 : P_1 [b_1]
Gen 1 : P_2 [$s + b_1$, b_2 , $Sh_1(2,2)(s)$] ; P_3 [$s + b_1$, b_2 , $Sh_1(2,2)(s)$]
Gen 2 : P_4 [$s + b_1$, $s + b_2$, b_3 , $Sh_2(2,4)(s)$] ; P_5 [$s + b_1$, $s + b_2$, b_3 , $Sh_2(2,4)(s)$]
P_6 [$s + b_1$, $s + b_2$, b_3 , $Sh_2(2,4)(s)$] ; P_7 [$s + b_1$, $s + b_2$, b_3 , $Sh_2(2,4)(s)$]

- Reconstruction: Gen 0 (P_1) uses b_1 with any party down the generations
- Security: One-time pad (p-ary)
- Reconstruction: Gen 1: Within the same generation, use Shamir's share,
  down the generations, use b_2
- Security: Within the same generation, Shamir's scheme
  down the generations, one-time pad

17

(2,∞)-threshold scheme: Correctness and security (sketch), cont.

Gen 0 : P_1 [b_1]
Gen 1 : P_2 [$s + b_1$, b_2 , $Sh_1(2,2)(s)$] ; P_3 [$s + b_1$, b_2 , $Sh_1(2,2)(s)$]
Gen 2 : P_4 [$s + b_1$, $s + b_2$, b_3 , $Sh_2(2,4)(s)$] ; P_5 [$s + b_1$, $s + b_2$, b_3 , $Sh_2(2,4)(s)$]
P_6 [$s + b_1$, $s + b_2$, b_3 , $Sh_2(2,4)(s)$] ; P_7 [$s + b_1$, $s + b_2$, b_3 , $Sh_2(2,4)(s)$]

- To continue, apply the same reasoning as in the previous slide, recursively
- Reconstruction: Gen 2: Within the same generation, use Shamir's share,
  down the generations, use b_3
- Security: Within the same generation, Shamir's scheme
  down the generations, one-time pad

18

Discussion

- A (k,∞)-threshold scheme was also presented in [KNY16]
  - Out of scope today
- For the above $(2,\infty)$-threshold scheme, we need to work over a field (because Shamir's scheme is used)

19

Our goals

- Understand combinatorial interpretation of evolving secret sharing schemes
- Avoid the use of finite fields
  - To accommodate the most general case, e.g., a permutation group (which is non-abelian)

20

Perfect hash family – Definition

- A family of functions F is called an (N;n,m,w)-Perfect Hash Family (PHF), if:
  - Each $f \in F : [n] \to [m]$ with $|F| = N$, and
  - For any w-subset $X \subset [n]$, $\exists\, g \in F \;:\; g|_X$ is one-to-one

21

PHF – Example

- A family of functions F is called an (N;n,m,w)-Perfect Hash Family (PHF), if:
  - Each $f \in F : [n] \rightarrow [m]$ with $|F| = N$, and
  - For any w-subset $X \subset [n]$, $\exists f_i \in F : f_i|_X$ is one-to-one

$$M = \begin{array}{c} \begin{array}{cccccccc} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \end{array} & \\ \begin{pmatrix} 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{pmatrix} & \begin{array}{c} f_1 \\ f_2 \\ f_3 \end{array} \end{array}$$

- i-th row represents a function $f_i : \{0,1\}^3 \rightarrow \{0,1\}$
- For any 2-subset $S \subset \{0,1\}^3$ $\exists$ i s.t. f_i restricted to S is one-to-one
- E.g., restrict to {2,3}, then f_3 is one-to-one,
  restrict to {1,2}, then f_2 and f_3 are one-to-one

22

Remark on PHF

- Any binary matrix with pairwise distinct columns represents a PHF
- Fact: For any $t \geq 2$, there exists a (t ; 2^t, 2, 2)-PHF

$$M = \begin{array}{c} \begin{array}{cccccccc} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \end{array} & \\ \begin{pmatrix} 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{pmatrix} & \begin{array}{c} f_1 \\ f_2 \\ f_3 \end{array} \end{array}$$

M represents a (3,8,2,2)-PHF

23

(2,n)-threshold secret sharing from PHF

$$M = \begin{array}{c} \begin{array}{cccccccc} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \end{array} & \\ \begin{pmatrix} 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{pmatrix} & \begin{array}{c} f_1 \\ f_2 \\ f_3 \end{array} \end{array}$$

[Desmedt, DiCrescenzo, Burmester: Asiacrypt '94]
[Safavi-Naini, Wang: Progress in CS and Applied Logic '02]

- Secret $s \in G(*)$; parties P_j , j = 0..7
- ShareGen: For i = 1..3 : $b_i \leftarrow_R G(*)$, share of P_j : $s^{M[i,j]} * b_i$ for i = 1..3
- Example: Share of P_0 : (b_1 , b_2, b_3), Share of P_1 : (b_1 , b_2, $s*b_3$)
- Reconstruction: Solve the system $x*b_3 = s*b_3$ to obtain the secret s
- Security: One-time pad (q-ary, multiplicative)

24

(2,n)-threshold secret sharing from PHF (cont.)

$$M = \begin{array}{c} \begin{array}{cccccccc} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \end{array} \\ \begin{pmatrix} 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{pmatrix} \end{array} \begin{array}{c} \\ f_1 \\ f_2 \\ f_3 \end{array}$$

- Secret $s \in G(*)$; parties P_j , j = 0..7
- ShareGen: For i = 1..3 : $b_i \leftarrow_R G(*)$
  - Share of P_j : $s^{M[i,j]} * b_i$ for i = 1..3

- In general: Reconstruction follows by the w-subset (last) property of PHF
- Question: Can we construct an evolving scheme?
- Natural approach: “Evolving” PHF

25

Preliminary definitions

- Def.: Evolving family of sets: A sequence of sets $\{X_n\}_{n\geq 0}$ is an evolving family of sets if $X_i \subset X_{i+1}$ for all $i \geq 0$, i.e., the family is strictly monotone increasing
- Def. (Partial function): A rule $X \rightarrow Y$ is called a partial function, if there exists a subset $X' \subset X$ s.t. when restricted to X', $f|_{X'} : X' \rightarrow Y$ is a (total) function

26

Evolving Perfect Hash Functions – Definition

- Def.: Let $\{X_r\}$ be an evolving family of sets, $\{Y_r\}$ be a sequence of sets (which may or may not be evolving) and $\{w_r\}$ be a non-decreasing sequence of positive integers

  A sequence of families of partial and total functions $\{\mathcal{F}_r\}$ is called an ($\{X_r\}$, $\{Y_r\}$, $\{w_r\}$)-Evolving PHF, if:
  - Each $f \in \mathcal{F}_r$ is a partial/total function from $X_r \rightarrow Y_r$ and
  - For any w_r-subset $X' \subset X_r$, there exists $g \in \mathcal{F}_r$ such that the restriction of g of X’ is one-to-one

27

Remarks

- In the paper, we make a distinction between "evolving" PHF family, which is finite and "perpetually evolving", which is infinite
- In such the families, only the sequence of domains $\{X_r\}$ needs to be an evolving family of sets
- The sequence of co-domains $\{Y_r\}$ need not be evolving, in fact, it can be constant, i.e., $Y_r = Y$ for all r
- In addition, the non-decreasing sequence of $\{w_r\}$ can be a constant sequence

28

Our proposal of Perpetually Evolving PHF

- Focus on the binary case, i.e., co-domain $Y_r = Y = \{0,1\}$ and $w_r = w = 2$, for all r
- Notation: An m-dimensional vector of zeroes as 0_m and that of ones as 1_m
  - After introduction of r-th partial row, the evolved matrix is denoted as M(r)
  - Denote the non-zero columns of M as $C_1,\ldots,C_{2^t-1}$
    - Each of them is a t-bit column vector

29

Our construction

- Consists of the following three procedures:
- **Init**: Assign 0_t as the first column of M(0)
- **1st Partial Row:**
  1. Place the remaining 2^t-1 columns $C_1,\ldots,C_{2^t-1}$ to the right of 0_t
  2. Append a partial row 0_{2^t-1} just below them
  3. Copy $C_1,\ldots,C_{2^t-1}$ to the right as columns 2^t+1 to $2^{t+1}-1$
  4. Append a partial row 1_{2^t-1} just below the columns copied above

30

Our construction (cont.)

- **r-th Partial Row to M(r-1):**
  1. Choose the last a = $\lceil \alpha/2 \rceil$ columns B[1], B[2], …, B[a] of M(r-1), where α denotes # columns in M(r-1)
  2. Append a partial row 0_a just below B[1], B[2], …, B[a]
  3. Copy B[1], B[2], …, B[a] to the right of M(r-1)
  4. Append a partial row 1_a just below the columns copied above

31

Example: Evolving PHF

$$M(0) := M = \begin{array}{c} \begin{matrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \end{matrix} \\ \begin{pmatrix} 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{pmatrix} \end{array}$$

- Let us with a PHF defined by M
- Denote it as M(0)
- Next, let us compute the next generation M(1)

32

Example: Evolving PHF (cont.)

$$M(1) = \begin{array}{c} \begin{matrix} & C_1 & C_2 & C_3 & C_4 & C_5 & C_6 & C_7 & & & & & & & \end{matrix} \\ \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \end{array}$$

- **Init**: 1st column is all-zero.
- **1st Partial Row:** 1. The remaining columns C_1,…,C_7 are to the right of 0_3
  2. Append a partial row 0_7.
  3. Copy C_1,…,C_7 to the right.
  4. Append a partial row 1_7.

33

Example (cont.)

$$
\begin{array}{c}
\begin{array}{ccccccccccccccc} C_0 & C_1 & C_2 & C_3 & C_4 & C_5 & C_6 & C_7 & C_8 & C_9 & C_{10} & C_{11} & C_{12} & C_{13} & C_{14} \end{array} \\
M(1) = \left[\begin{array}{c|ccccccc|ccccccc}
0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\
0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\
0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\
 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1
\end{array}\right]
\end{array}
$$

- This is an empty entry (partial function)
- Claim: M(1) defines a (partial) PHF
- Proof sketch: C_0 and each block individually is M(0) => PHF; across the blocks, use the last row

34

Example (cont.)

$$
\begin{array}{c}
\begin{array}{ccc} \text{Block } \mathcal{A} & \text{Block } \mathcal{B} & \text{Block } \mathcal{C} \end{array} \\
M(2) = \left[\begin{array}{cccccccc|cccccccc|cccccccc}
0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\
0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\
0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\
 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
 & & & & & & & & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1
\end{array}\right]
\end{array}
$$

- Claim: M(2) defines a (partial) PHF
- Consider the blocks of columns $\mathcal{A}$, $\mathcal{B}$ and $\mathcal{C}$ as marked above
- Any pair of columns in Blocks $\mathcal{B}$ and $\mathcal{C}$ differ in at least position due to the last row
- Any pair of columns in Block $\mathcal{A}$ and Block $\mathcal{B}$ (resp. Block $\mathcal{C}$) differ in at least one position because M(1) is PHF

35

Main result

- Thm.: Our construction implements a perpetually evolving PHF.
- Proof (sketch): By induction.
  - The base case intuition: two slides back.
  - The induction step intuition: the previous slide.
- Corollary: There exists an evolving $(2,\infty)$-threshold multiplicative secret sharing scheme
- Note: The underlying group may be non-abelian

36

Parameters and share size

- If the 1st column of M(0) is t-dimensional, r-th partial row adds $\lceil (3/2)^{r-1}2^t \rceil$ new columns, i.e., increases the domain by exponentially many elements
- Share size for T-th participant: $(\ t + O(\log(r,T))\ )\cdot\log|G|$

37

Conclusion

- Studied a combinatorial interpretation of evolving secret sharing
- Proposed a recursive construction of perpetually evolving PHF
  - It implies an evolving $(2,\infty)$-threshold multiplicative secret sharing scheme

38

Future work

- Further study of evolving combinatorial objects
  - Blueprint: Start with a recursive construction for such, and develop it into an evolving scheme
- Extension to (k,∞)-threshold case

39

References

1. Blakley, G.R.: Safeguarding cryptographic keys. AFIPS 1979.
2. Cachin, C.: On-line secret sharing. Cryptography and Coding, 1995.
3. Csirmaz, L., Tardos, G.: On-line secret sharing. Des. Codes Crypt. 63(1), 2012.
4. Desmedt, Y., Di Crescenzo, G., Burmester, M.: Multiplicative non-abelian sharing schemes and their application to threshold cryptography. ASIACRYPT 1994.
5. Desmedt, Y., Dutta, S., Morozov, K.: Evolving Perfect Hash Families: A Combinatorial Viewpoint of Evolving Secret Sharing, CANS 2019.
6. Komargodski, I., Naor, M., Yogev, E.: How to share a secret, infinitely. TCC 2016.
7. Safavi-Naini, R., Wang, H.: Robust additive secret sharing schemes over Zm. Cryptography and Computational Number Theory. Progress in Computer Science and Applied Logic (20), 2001.
8. Shamir, A.: How to share a secret. Commun. ACM 22(11), 1979.
9. Wong, T. M., Wang, C., and Wing, J. M.: Verifiable secret redistribution for archive system," IEEE Security in Storage Workshop, 2002.

UNT
UNIVERSITY OF NORTH TEXAS
EST. 1890

Thank you very much
for your attention,
and questions, please!

(Kirill.Morozov@unt.edu)

Secure-Computation AI : a Python Library for Machine Learning in Secure Computation

Ibuki Mishina (Joint work with Dai Ikarashi, Koki Hamada and Ryo Kikuchi)
NTT Corporation
ibuki.mishina.br@hco.ntt.co.jp

Big data analysis using machine learning (AI) is expected to be a technology that enables complex analysis and inference, but because it requires a large amount of data, including personal information, it often faces issues related to privacy. Therefore, as a solution to this problem, a technology has been attracting attention in recent years, in which learning and inference is calculated while keeping data encrypted using secure computation.

Research on secure computation for machine learning, especially deep learning has been very active in the past few years, and faster methods have been proposed one after another[1]. In addition, there has been research in the area of proposing and implementing easy-to-understand software framework for machine learning researchers and engineers[2]. Thus, various researches on secure computation for machine learning are being conducted, not only on performance but also usability and so on.

In our research, we have implemented various machine learning methods such as logistic regression and deep learning in secure computation with high speed and high accuracy[3, 4]. Furthermore, we have implemented an software framework for machine learning in secure computation as a Python library[5], with an application programming interface similar to general machine learning libraries. Our secure-computation AI is characterized by high performance in terms of accuracy and processing speed, a rich lineup of analyses, and ease of use, all of which are necessary for an AI library. In this paper, we introduce the performance, the lineup of analysis methods, and application programming an interface of our secure-computation AI library.

References

[1] Wagh, Sameer and Tople, Shruti and Benhamouda, Fabrice and Kushilevitz, Eyal and Mittal, Prateek and Rabin, Tal. Falcon: Honest-majority maliciously secure framework for private deep learning. arXiv preprint arXiv:2004.02229, 2020.

[2] Knott, Brian and Venkataraman, Shobha and Hannun, Awni and Sengupta, Shubho and Ibrahim, Mark and van der Maaten, Laurens. CrypTen: Secure multi-party computation meets machine learning. arXiv preprint arXiv:2109.00984, 2021.

[3] Ibuki Mishina, Koki Hamada, Dai Ikarashi and Ryo Kikuchi. Fast and Accurate Logistic Regression and Data Standardization Using Secure Real Number Operations. In CSS(in Japanese), 2020.

[4] Ibuki Mishina, Koki Hamada and Dai Ikarashi. Realization of Practical Secure Deep Learning. In CSS(in Japanese), 2019.

[5] Ibuki Mishina, Koki Hamada, Dai Ikarashi and Ryo Kikuchi. A Design and an Implementation of a Python Library for Secure Deep Learning. In SCIS(in Japanese), 2021.

Secure-Computation AI:

a Python Library for Machine Learning in Secure Computation

NTT Social Informatics Laboratories
Ibuki Mishina

Secure Computation × Machine Learning(AI)

Outline

1. What's Secure-computation AI?
2. Our Research
    1. Algorithm Impementation
    2. API Implementation

3

Outline

NTT

1. What's Secure-computation AI?
2. Our Research
   1. Algorithm Impementation
   2. API Implementation

4

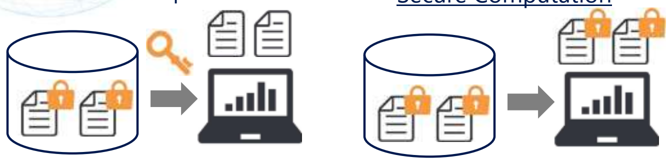
Secure Computation

NTT

Computation while keeping data encrypted

Usual computation

Secure Computation



Encrypt and store

Decrypt when in use

Encrypt and store

Use while encrypted

5

Machine Learning(AI)

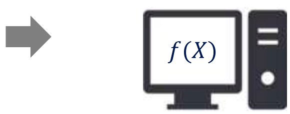
NTT

Build a model based on training data
in order to make inferences

- Structural data
- Images
- Time-series data

etc...

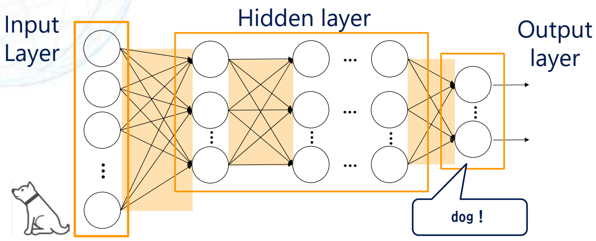
Train data and build models



6

Deep Learning

Used for image classification, etc.



Secure-Computation AI

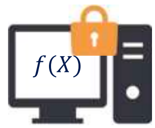
AI training and inference

while keeping data encrypted

- Structural data
- Images
- Time-series data

etc...

Train data and build models



8

Research Topics in Secure-Computation AI

- Algorithm Implementation
  - There are many methods for machine learning
  - Often the plaintext algorithm cannot be used as is
  - e.g. secureML, secureNN, ABY3...
- API Implementation
  - Implement APIs that are easy <u>for data scientists</u> to use
  - e.g. Crypten(Facebook), tf-encrypted(google)

9

Outline

NTT

1. What's Secure-computation AI?
2. Our Research
   1. Algorithm Impementation
   2. API Implementation

10

Our Research Goals

NTT

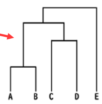
- Practical Secure-Computation AI
  - Many analytical methods available
  - Fast & high accuracy
  - Easy-to-use interface for data analysts

11

What we've implemented

NTT

- Linear Regression(CSS2020) $f(x) = ax_1 + bx_2 + cx_3 \ldots$
- Logistic Regression(CSS2018/SCIS2019/CSS2019/CSS2020) $\frac{1}{1+e^{-x}}$
- Deep Learning(FIT2019/CSS2019/SCIS2020/SCIS2021) $\frac{e^{x_k}}{\sum e^{x_i}}$
- Decision Trees & Gradient Booting(CSS2021)
- Hierarchical clustering(CSS2021)
- Medical Statistics(CSS2019/CSS2020)



12

Secure Logistic Regression/Linear Regression

NTT

Previous Research

- Optimizer→SGD

Easy, but slow convergence.

Our Research

- Optimizer → Newton-CG

Fast convergence,
no need for inverse matrix calculation

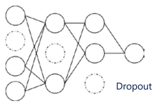
13

Secure Deep Learning(1/2)

NTT

Previous Research

- Optimizer→SGD only
- Dropout → ✕
- Scaling → ✕



Our Research

- Optimizer → SGD, Adam, momentum etc...
- Dropout → O
- Scaling → O

14

Secure Deep Learning(2/2)

NTT

Dataset：MNIST(784×60,000)

Model : 784→128→128→10(2 hidden layers)

	time[sec]	accuracy[%]
ABY3[MR18]	2700	94
NTT(SCIS2021)*	95	96

*CPU：3.50GHz(8core)×2 / RAM：768GB / NW：10G

15

Outline

1. What's Secure-computation AI?
2. Our Research
   1. Algorithm Impementation
   2. API Implementation

16

Famous machine learning libraries(plaintext)

- Keras:
  - deep learning library with a simple interface
- Scikit-learn：
  - Machine learning library for various types of analysis
- Pandas:
  - Libraries to support data analysis

These are all
Python libraries

17

Our Goals

Python Library for Secure-Computation AI

Keras + scikit-learn + pandas

18

MEVAL : Secure computation Library

NTT

- Secret sharing based secret computation library being developed by NTT
- Can be freely programmed to combine more than 100 operations, including arithmetic, logic, comparison, and real number operations.
- Optimized for secure computation, enabling fast and accurate processing.

19

Background

NTT

Optimized interface for secure computation

is different from plain text

↓

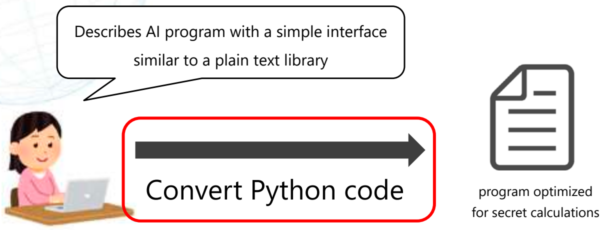
For data analysts unfamiliar with secure computation,

an interface optimized for secure computation is difficult

20

Our Contribution

NTT



21

Idea

A secure-computation AI program written in meval

is several thousand lines long,

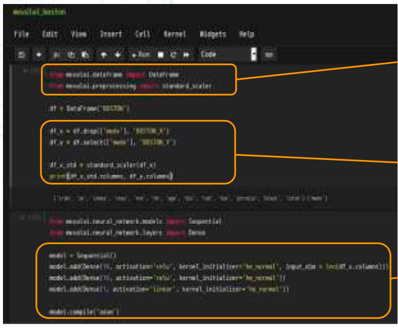
but there are not many parameters for the data analyst to rewrite.

⬇

1. Prepare a secure-computation AI program in which all but the parameters are described in advance
2. Embed parameters specified by the analyst from the plaintext IF into the secure computation program

22

Our Secure-Computation AI Library



import as
a Python library

preprocessing

Keras like
NN model design

23

Comparison of Others

- Crypten(Facebook), tf-encrypted(google)
  - Model : Deep Learning only
  - Processing performance of secure computation is not high
  - Interfaces compatible with Pytorch or tensorflow
- NTT
  - Model : Deep Learning, Decision Trees, Clustering, etc...
  - Fast & high accuracy

24

Future work

- Further expansion of AI methods
- Expansion of functions other than AI methods
  - Preprocessing, Model evaluation, etc…
- Publishing Secure-Computation AI Library

25

Conclusion

- Secure-Computation AI:
  - AI training and inference while keeping data encrypted
- Research Topics in Secure-Computation AI:
  - Algorithm Implementation
    - Deep Learning, Logistic Regression, Decision Trees, etc…
  - API Impementation
    - Python Library for Secure-Computation AI

26

Possibility of Secret Sharing using EtherCAT

Kosuke Kaneko

Robert T.Huang Entrepreneurship Center of Kyushu University
kaneko.kosuke.437@m.kyushu-u.ac.jp

In this presentation, we explain the possibility of Secret Sharing using EtherCAT(Ethernet for Control Automation Technology) which is an industrial network technology. We implemented an algorithm of Secret Sharing using EtherCAT and evaluated their time performance of encryption/decription. We discuss the possibility of Secret Sharing using EtehrCAT based on results of the evaluation.

Possibility of Secret Sharing using EtherCAT

Kosuke Kaneko

Robert T.Huang Entrepreneurship Center of Kyushu University

This research was supported by Skydisc, Inc.

About me

- 2014:
  - Ph.D of Information Science at Kyushu University
- 2014 – 2016:
  - Assistant Professor at Innovation Center for Educational Resource, Kyushu University Library
- 2016 – 2021:
  - Associate Professor at Cybersecurity Center in Information Infrastructure Iniciative, Kyushu University
- 2021 - :
  - Associate Professor at Robert T.Huang Entrepreneurship Center, Kyushu University

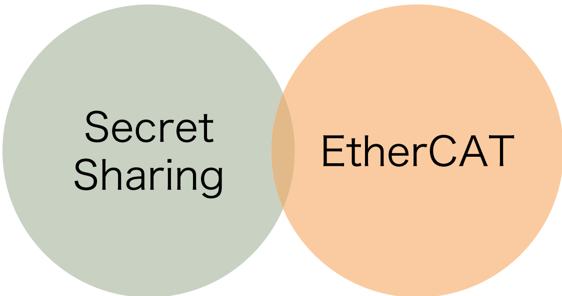
Outline

1. Research Background and Purpose
   - Smart Factory, EtherCAT
   - Our IDEA of Secret Sharing X EtherCAT
2. Method and Implementation
   - Our proposed protocol
3. Experiment and Result
   - Calculation time for encryption/decription
4. Conclusion and Discussion

Reserch Background

Exploring Mathematical and Practical Principles of Secure Computation and Secret Sharing @ IMI, Kyushu University. Nov. 9, 2021. 4

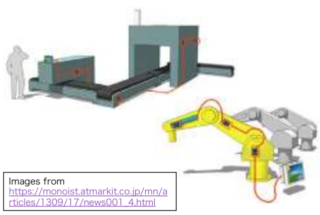
What is today’s topic?



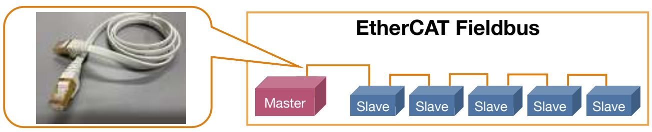
Exploring Mathematical and Practical Principles of Secure Computation and Secret Sharing @ IMI, Kyushu University. Nov. 9, 2021. 5

What is EtherCAT ? - Overview -

- **Ethernet for Control Automation Technology**
- **A network for industry**
  (factory automation netowrk, belt conveyor)
- **Device to device communication**
  (master and slaves)
- **IEEE 802.3 Ethernet/Ethernet Cable**
  (Common Open Technology)



Images from
https://monoist.atmarkit.co.jp/mn/a
rticles/1309/17/news001_4.html

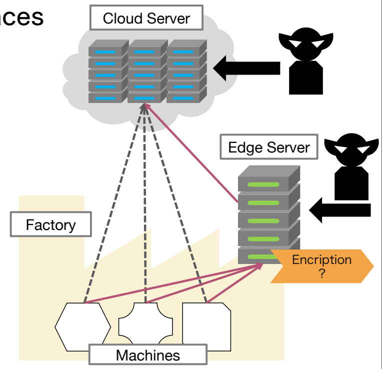


Exploring Mathematical and Practical Principles of Secure Computation and Secret Sharing @ IMI, Kyushu University. Nov. 9, 2021. 6

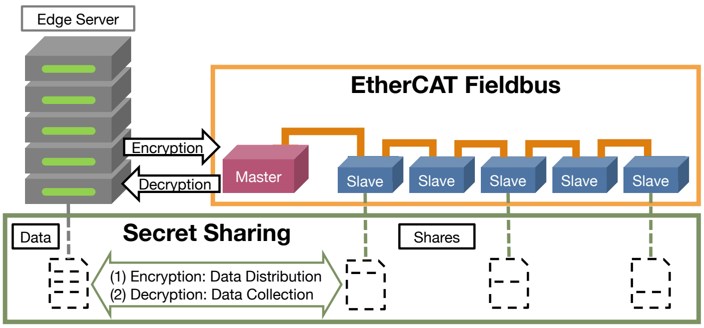
Smart Factory Circumstances

- Industry 4.0: Factory x AI

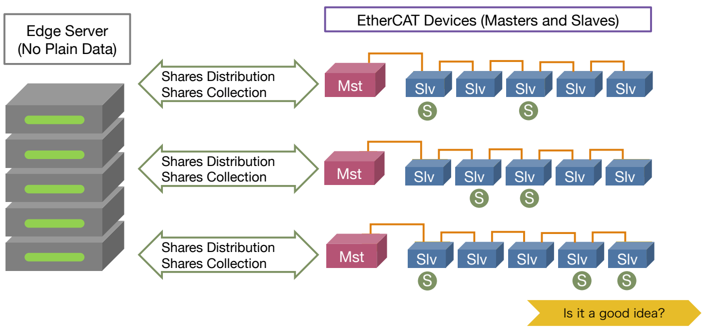
- They don't like to take data out of their factory so much



What is our IDEA? (1/2)



What is our IDEA? (2/2)



Research Purpose

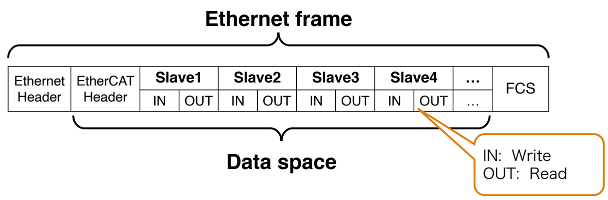
- Investigation for discussing possibility of Secret Sharing with EtherCAT
- Evaluation: Calculation time for encryption/decryption
  - To investigate calculation time by changing situations
    - Number of slaves in EtherCAT fieldbus
    - Number of shares for Secret Sharing
    - Number of required shares for Secret Sharing

Method and Implementation

What is EtherCAT ? - Data Frame -

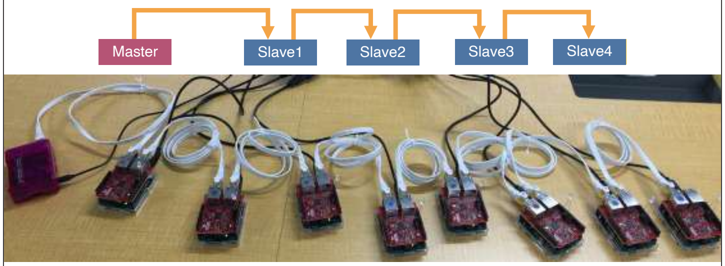
EtherCAT is based on Ethernet:

Data frame for communication is Ethernet frame.



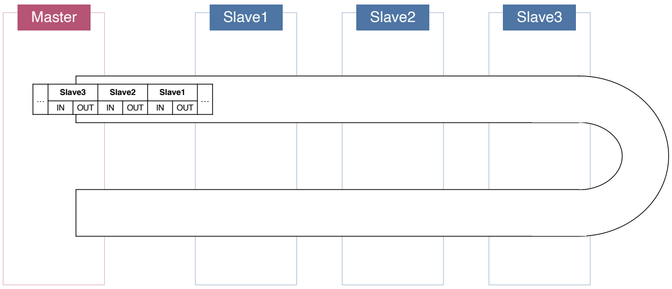
What is EtherCAT ? - Network Structure -

- EtherCAT adopts daisy chain network structure
- Each device has two ethernet ports



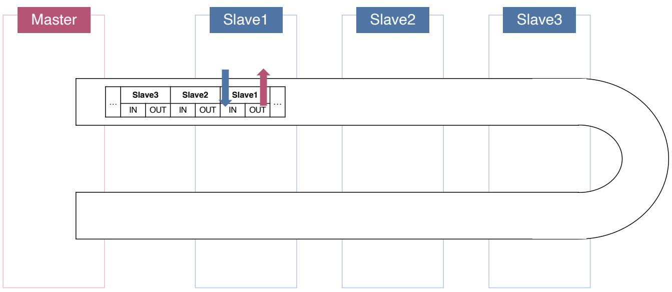
What is EtherCAT ? - EtherCAT Communication -

One Ethernet cable has two way paths



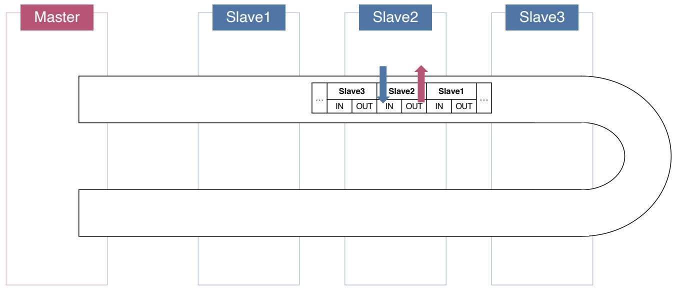
What is EtherCAT ? - EtherCAT Communication -

One Ethernet cable has two way paths



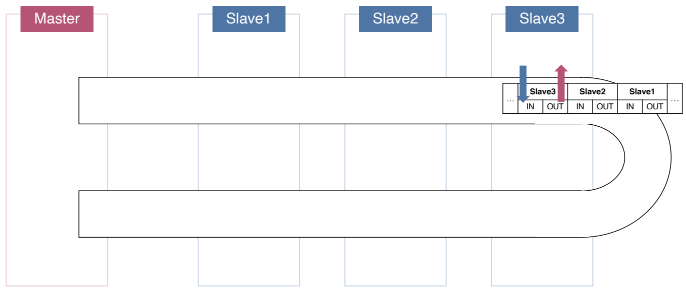
What is EtherCAT ? - EtherCAT Communication -

One Ethernet cable has two way paths



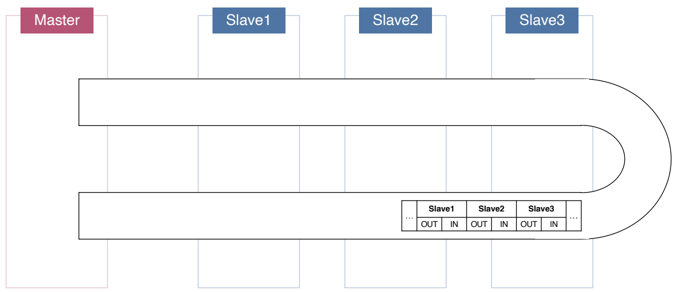
What is EtherCAT ? - EtherCAT Communication -

One Ethernet cable has two way paths



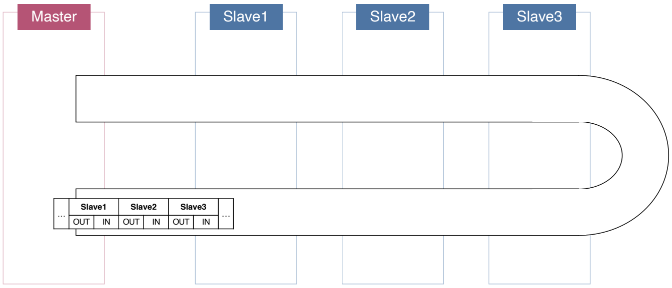
What is EtherCAT ? - EtherCAT Communication -

One Ethernet cable has two way paths



What is EtherCAT ? - EtherCAT Communication -

One Ethernet cable has two way paths

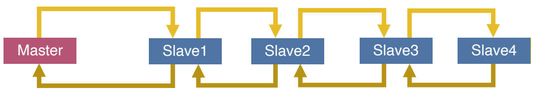


Exploring Mathematical and Practical Principles of Secure Computation and Secret Sharing @ IMI, Kyushu University. Nov. 9, 2021.

19

What is EtherCAT ? - Communication Protocol -

- No handshake protocol, not like TCP/IP
- Transmission frequency: 125μs (8,000 times/sec.)

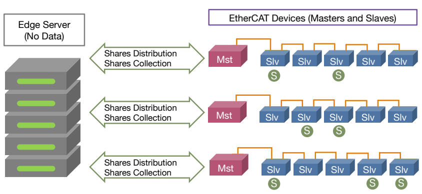


Exploring Mathematical and Practical Principles of Secure Computation and Secret Sharing @ IMI, Kyushu University. Nov. 9, 2021.

20

What dvantage does EtherCAT has for Secret Sharing?

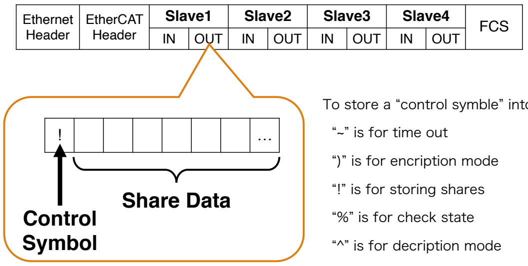
- Network structure is originally suitable to Secret Sharing
  - Devices are distributed in Factory and they communicate each other anytime
- Quick communication for shares distribution and collection
  - No handshake protocol, 8,000 times/sec.



Exploring Mathematical and Practical Principles of Secure Computation and Secret Sharing @ IMI, Kyushu University. Nov. 9, 2021.

21

Our Proposed Protocol (1/3)



To store a “control symble” into the first bit like:

“~” is for time out

“)” is for encription mode

“!” is for storing shares

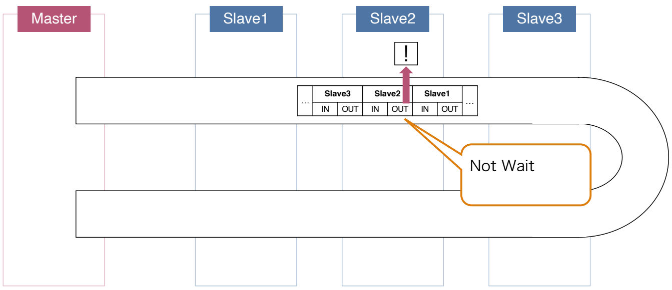
“%” is for check state

“^” is for decription mode

“&” is for done the process

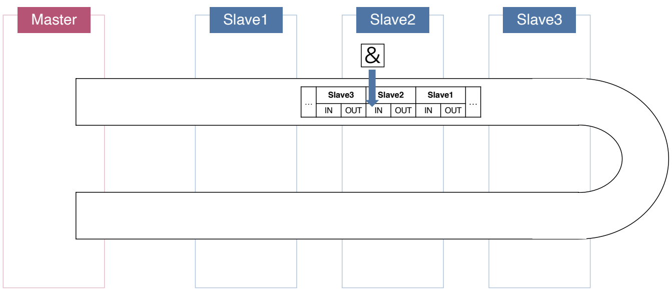
Our Proposed Protocol (2/3)

Transmission frequency: 125μs (8,000 times/sec.)



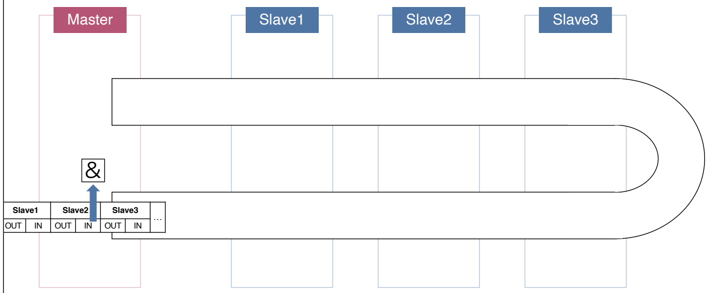
Our Proposed Protocol (2/3)

Transmission frequency: 125μs (8,000 times/sec.)



What is EtherCAT ? - EtherCAT Communication -

One Ethernet cable has two way paths



Experiment and Result

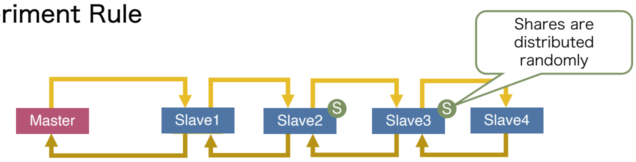
Experiment Environment

Master Device	Raspberry Pi 3 B+
Slave Devices	EasyCAT Shield for Arduino
EtherCAT	SOEM Library (https://github.com/OpenEtherCATsociety/SOEM)
Network Tool	Ethernet Cable (CAT7, 1m)
Protocol	Our Proposed Protocol
Secret Sharing	Shamir Secret Sharing (https://github.com/dsprenkels/sss)

Problems

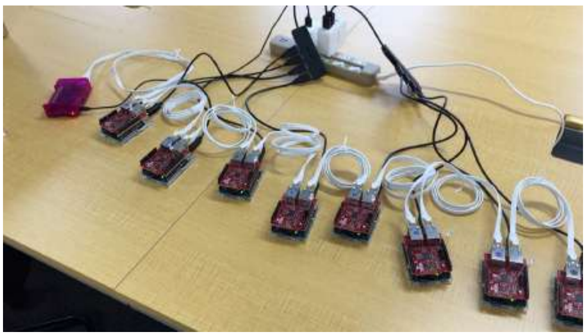
- We needed to divide a share into 4 data; it took 4 times long time to transmit shares.
  - Shamir Secret Sharing library (https://github.com/dsprenkels/sss) is generate 112 bytes size shares.
  - EasyCAT Shield for Arduino can transfor only 32 bite in one time.
- Timeout:
  - Some slave devices could not receive a share sometimes. We set timeout as 0.03 sec. If elapsed time was over the timeout, then transmission was restarted again.

Experiment Rule

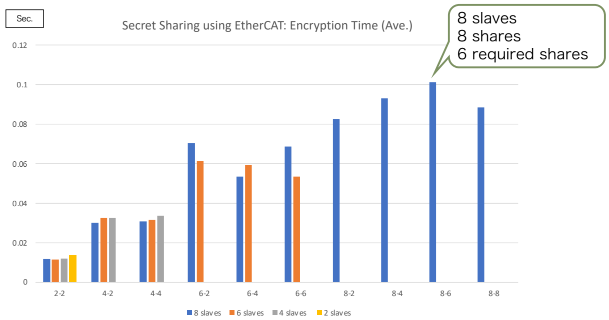


- Exp. 1: To change number of slaves for EtherCAT (2 - 8)
- Exp. 2: To change number of shares for Secret Sharing (2 – 8)
- Exp. 3: To change number of required shares for Secret Sharing (2 - 8)
- Investigation for calcluation times for ecription and decription in each case
  - To do encription/decription 100 times and calculate average time

Experiment Scene (8 slaves)

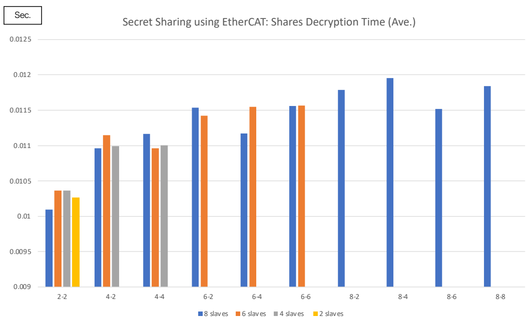


Result: Culculation times for encription by each case



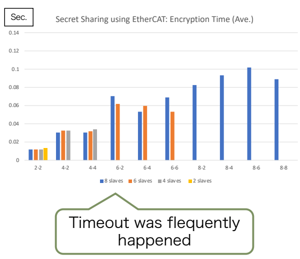
Exploring Mathematical and Practical Principles of Secure Computation and Secret Sharing @ IMI, Kyushu University. Nov. 9, 2021. 31

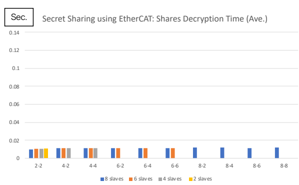
Result: Culculation times for decription by each case



Exploring Mathematical and Practical Principles of Secure Computation and Secret Sharing @ IMI, Kyushu University. Nov. 9, 2021. 32

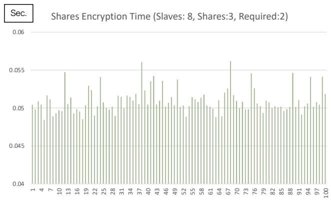
Result: Compare times by the same scale



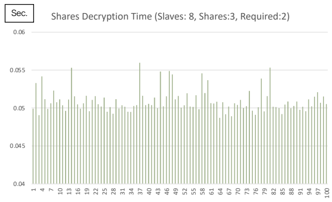


Exploring Mathematical and Practical Principles of Secure Computation and Secret Sharing @ IMI, Kyushu University. Nov. 9, 2021. 33

Slaves: 8, Shares: 3, Required: 2



AVE. 0.051 sec.



AVE. 0.051 sec.

Conclusion and Discussion

Conclusion

- Calculation times for encryption/decryption were significantly influenced by number of shares, but it were not so much influenced by number of slaves and number of required shares.

- If it was good condition (no timeout), it took about 0.05 sec. for encryption/decryption.

Discussion Time

An indeterminate equation scheme having homomorphic property

Yasuhiko Ikematsu

Institute of Mathematics for Industry, Kyushu University
ikematsu@imi.kyushu-u.ac.jp

Indeterminate encryption schemes are public key cryptosystems using indeterminate equations having a solution with small coefficients over a finite field. At Sac 2017, Akiyama et al. proposed an indeterminate encryption scheme "Giophantus(TM)" whose public key is a polynomial in two variables over a finite ring. In this talk, we introduce the construction of the Giophantus scheme and explain that it becomes a somewhat homomorphic encryption (SHE).

An Indeterminate Equation Scheme Having Homomorphic Property

*Yasuhiko Ikematsu (Kyushu University)
Koichiro Akiyama (Toshiba Corporation)

Institute of Mathematics for Industry
Kyushu University

10th November 2021

1

PQC

2/19

Post-Quantum Cryptography

- **Lattice base・・・SVP, CVP**
- **Code base・・・Syndrome decoding problem**
- **Isogeny base・・・Isogeny path finding problem**
- **Multivariate base・・・MQ problem**

NIST PQC standardization third round in 2020

NIST 3rd	Signature	Encryption/KEM
Lattice	2	5
Code	0	3
Isogeny	0	1
Multivariate	2	0
Else	2	0

Overview

3/19

➢ Indeterminate equation cryptosystem for PQC

Section Finding Problem

$X(x, y) = 0$ over $\mathbb{F}_q[t]$

➢ Giophantus™ proposed by Akiyama et al.

Small Section Problem

$X(x, y) = 0$ over $\mathbb{F}_q[t]/(t^n - 1)$

➢ Homomorphic property

Additive & Multiplicative

Contents

1.1 Section finding problem

$q \in \mathbb{N}$: prime, $n > 0$: integer

$\mathbb{F}_q[t, x, y]$: three-variable polynomial ring

Section Finding Problem

$X(t, u_x, u_y) = 0$

Given $X(t, x, y) \in \mathbb{F}_q[t, x, y]$

Find one solution (section) $(u_x(t), u_y(t)) \in \mathbb{F}_q[t]^2$ with degree n.

- If $\deg_{x,y} X(t, x, y) > 1$, the problem is difficult.
- From this problem, some cryptosystems are constructed.

1.2 Indeterminate equation cryptosystem

$q \in \mathbb{N}$: prime, $n, d > 0$: integer

Secret Key

$$u_x = a_0 + a_1 t + \cdots + a_{n-1} t^{n-1} \in \mathbb{F}_q[t]$$
$$u_y = b_0 + b_1 t + \cdots + b_{n-1} t^{n-1} \in \mathbb{F}_q[t]$$

Public Key

$X(t, x, y) \in \mathbb{F}_q[t, x, y]$ degree d s.t. $X(t, u_x, u_y) = 0$ in $\mathbb{F}_q[t]$

- Secret key is randomly chosen from $\mathbb{F}_q[t]$.
- Public key is obtained by the linear system in $c_{i,j}$

$$\sum_{i,j} c_{i,j}(t) u_x^i u_y^j = 0, \quad \text{where } X \coloneqq \textstyle\sum_{i,j} c_{i,j}(t) x^i y^j.$$

- It is difficult to compute (u_x, u_y) from the public key X.

1.2 Indeterminate equation cryptosystem

Message: $m(t) = m_0 + m_1 t + \cdots + m_{n-1} t^{n-1} \in \mathbb{F}_q[t]$

Encryption: 1. Randomly choose $r(t,x,y) \in \mathbb{F}_q[t,x,y]$

2. Compute $c \coloneqq m + X \cdot r$

Decryption: 1. Compute

$$c(u_x, u_y) = m + X(t, u_x, u_y) \cdot r(t, u_x, u_y) = m$$

1.3 Progression of IEC

$$c = m + X \cdot r$$

multiple structure ↓ Linear Algebra Attack, Reduction Attack

$$c = m(t) \cdot s + X \cdot r(t)$$

three variables ↓ Trace Attack by Voloch

$$c = m \cdot s + X \cdot r \quad \text{PKC2009[2]}$$

noise addition ↓ Ideal Decomposition Attack[3]

GiophantusTM

$$\boxed{c = m(t) + X \cdot r + \ell \cdot e \mod t^n - 1}$$

[2] K. Akiyama et al., An Algebraic Surfaces Cryptosytem, PKC2009, LNCS 5443, pp. 425–442
[3] J. Faug`ere et al., Algebraic Cryptanalysis of the PKC'09 Algebraic Surface Cryptosystem, PKC2010, LNCS 6056, pp. 35–52

Contents

2.1 Small section problem

$q \in \mathbb{N}$: prime, $n > 0$: integer, $l > 0$: small integer

$X(x, y) \in \mathbb{F}_q[t, x, y]/(t^n - 1)$

Definition

$(u_x, u_y) \in \mathbb{F}_q[t]^2$ is called a *small section* of X
if $X(u_x, u_y) = 0$ and $0 \leq$their coefficients$\leq l - 1$.

Small Section Problem

Given $X(x, y) \in \mathbb{F}_q[t, x, y]/(t^n - 1)$ with a small section,
Find one small section (u_x, u_y) of X.

Giophantus[4] is constructed based on this problem.

[4] K. Akiyama et al, A Public-key Encryption Scheme Based on Non-linear Indeterminate Equations (Giophantus), IACR ePrint2017/1241

2.2 Construction

$q \in \mathbb{N}$: large prime, $n, d > 0$: integer, $l > 0$: small integer

$R_{q,n} := \mathbb{F}_q[t]/(t^n - 1)$

Secret Key

$u_x = a_0 + a_1 t + \cdots + a_{n-1} t^{n-1}$
$u_y = b_0 + b_1 t + \cdots + b_{n-1} t^{n-1}$, where $0 \leq a_i, b_i \leq l - 1$.

Public Key

$X(x, y) \in R_{q,n}[x, y]$ degree d s.t. $X(u_x, u_y) = 0$ in $R_{q,n}$

- Secret key is randomly chosen from $\mathbb{F}_q[t]$.
- Public key is obtained by the linear system in $c_{i,j}$
- It is difficult to compute (u_x, u_y) from the public key X.

2.2 Construction

Message: $m(t) = m_0 + m_1 t + \cdots + m_{n-1} t^{n-1}$, $0 \leq m_i \leq l - 1$

Encryption:
1. Randomly choose $r(x, y) \in \mathbb{F}_q[t, x, y]$
2. Randomly choose small $e(x, y) \in \mathbb{F}_q[t, x, y]$
3. Compute $c := X \cdot r + l \cdot e + m \bmod (q, t^n - 1)$

Decryption:
1. Compute $c(u_x, u_y) = l \cdot e(u_x, u_y) + m$
2. Compute $m' = c(u_x, u_y) \bmod l$

If each coefficient of $c(u_x, u_y)$ is in $[0, q - 1]$, then $m = m'$.

Thus, we need to take q as follows:

$$q > l - 1 + l \sum_{k=0}^{\deg_{x,y} e} (k + 1) n^k (l - 1)^{k+1}$$

2.3 Attacks

1. Key Recovery Attack

Public key $X(t,x,y)$ → a small section (u_x, u_y)

SVP problem

2. Linear Algebraic Attack

Ciphertext $c \equiv X \cdot r + (l \cdot e + m)$ → Recover message m

known

small polynomial

CVP problem

Giophantus is IND-CPA under the IE-LWE assumption.

Contents

3.1 Additive homomorphic

$$c_1 := X \cdot r_1 + l \cdot e_1 + m_1$$

$$c_2 := X \cdot r_2 + l \cdot e_2 + m_2$$

$$\Rightarrow \quad c := c_1 + c_2 = X \cdot (r_1 + r_2) + l \cdot (e_1 + e_2) + m_1 + m_2$$

Decryption: 1. Compute $c(u_x, u_y) = l \cdot (e_1 + e_2)(u_x, u_y) + m_1 + m_2$

2. Compute $m' = c(u_x, u_y) \bmod l$

(i) Each coefficient of $c(u_x, u_y)$ is in $[0, q-1]$

(ii) Each coefficient of $m_1 + m_2$ is in $[0, l-1]$

Then $m' = m_1 + m_2$

3.1 Additive homomorphic 16/19

λ: max of coef of messages m_1, m_2

If the following holds, then $m' = m_1 + m_2$

(i) $l > 2\lambda$

(ii) $q > 2 \cdot \left(l - 1 + l \sum_{k=0}^{\deg e} (k+1) n^k (l-1)^{k+1}\right)$

■ <u>N_a-times additive homomorphic case</u> $c := c_1 + c_2 + \cdots + c_{N_a}$

If the following holds, then decryption succeeds

- $l > N_a \lambda$
- $q > N_a \left(l - 1 + l \sum_{k=0}^{\deg_{x,y} e} (k+1) n^k (l-1)^{k+1}\right)$

3.2 Multiplicative homomorphic 17/19

$$c_1 := X \cdot r_1 + l \cdot e_1 + m_1$$

$$c_2 := X \cdot r_2 + l \cdot e_2 + m_2$$

$$\begin{aligned} c &:= c_1 \cdot c_2 \\ &= X \cdot (X r_1 r_2 + \cdots) + l^2 e_1 e_2 + l e_1 m_2 + l e_2 m_1 + m_1 \cdot m_2 \end{aligned}$$

Decryption: 1. Compute $c(u_x, u_y)$

2. Compute $m' = c(u_x, u_y) \bmod l$

(i) Each coefficient of $c(u_x, u_y)$ is in $[0, q-1]$

(ii) Each coefficient of $m_1 m_2$ is in $[0, l-1]$

Then $m' = m_1 \cdot m_2$

Conclusion 18/19

- We introduced an indeterminate equation scheme called "Giophantus".
- Giophantus is considered to be a scheme for post-quantum cryptography.
- We explained some homomorphic property of Giophantus.

Future work

- Parameter selection
- Bootstrapping
- More efficient HE scheme based on IES

Thank you

Homomorphic Secret Sharing for Multipartite and General Adversary Structures Supporting Parallel Evaluation of Low-Degree Polynomials

Reo Eriguchi (Joint work with Koji Nuida)
The University of Tokyo
reo-eriguchi@g.ecc.u-tokyo.ac.jp

Homomorphic secret sharing (HSS) for a function f allows input parties to distribute shares for their private inputs and then locally compute output shares from which the value of f is recovered. HSS can be directly used to obtain a two-round multiparty computation protocol for possibly non-threshold adversary structures whose communication complexity is linear in its share size and independent of the size of f.

Although several constructions of HSS schemes have been proposed, they do not give a satisfactory solution to practical non-threshold adversary structures Δ. When many parties are involved, Δ is likely to be specified by a general adversary structure rather than by a single threshold. The scheme [2] needs to set a corruption threshold to the maximum size of $X \in \Delta$ and then are inapplicable if Δ contains at least one set of size exceeding their tolerable thresholds. The construction [3] is applicable to any adversary structure but results in exponentially large share size for a specific class of non-threshold adversary structures, e.g., multipartite ones. It is therefore important to construct HSS schemes tailored to given non-threshold adversary structures in order to tolerate corruptions in real-world situations.

In this talk, we introduce our constructions of HSS schemes tolerating multipartite and general adversary structures and supporting parallel evaluation of a single low-degree polynomial [1]. Our multipartite scheme tolerates a wider class of adversary structures than the previous multipartite one in the particular case of a single evaluation and has exponentially smaller share size than the general construction. While restricting the range of tolerable adversary structures (but still applicable to non-threshold ones), our schemes perform ℓ parallel evaluations with communication complexity approximately $\ell/\log\ell$ times smaller than simply using ℓ independent instances. We also formalize two classes of adversary structures taking into account real-world situations to which the previous threshold schemes are inapplicable. Our schemes then perform $O(m)$ parallel evaluations with almost the same communication cost as a single evaluation, where m is the number of parties.

References

[1] R. Eriguchi and K. Nuida. Homomorphic secret sharing for multipartite and general adversary structures supporting parallel evaluation of low-degree polynomials. ASIACRYPT 2021, accepted.

[2] Y. Ishai, R.W.F. Lai, and G. Malavolta. A geometric approach to homomorphic secret sharing. PKC 2021, pp. 92–119.

[3] K. Phalakarn, V. Suppakitpaisarn, N. Attrapadung, and K. Matsuura. Constructive t-secure homomorphic secret sharing for low degree polynomials. INDOCRYPT 2020, pp. 763–785.

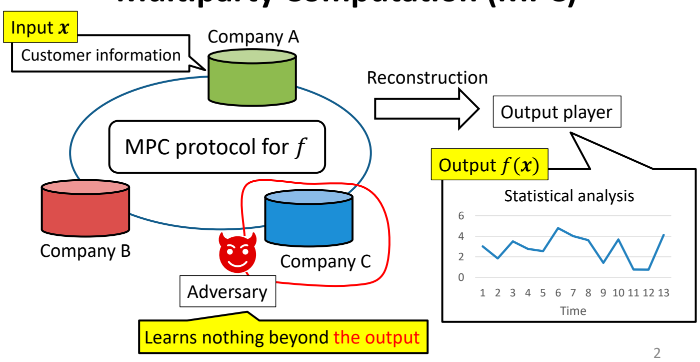
Homomorphic Secret Sharing for Multipartite and General Adversary Structures Supporting Parallel Evaluation of Low-Degree Polynomials

Exploring Mathematical and Practical Principles of
Secure Computation and Secret Sharing
November 10, 2021

Reo Eriguchi (The University of Tokyo, AIST)
Joint work with Koji Nuida

1

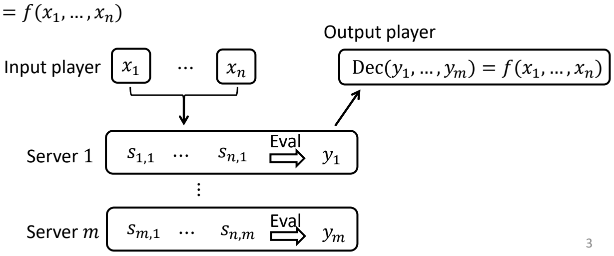
Multiparty Computation (MPC)



2

Homomorphic Secret Sharing (HSS) [BGI16]

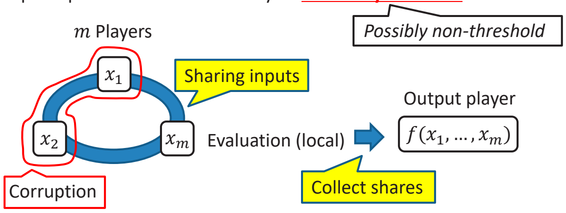
- Share generation: $(s_1, \dots, s_m) \leftarrow \mathrm{Share}(x)$
  - $\{s_j : j \in A\}$ reveals no information on x for a subset $A \subseteq [m]$.
- Evaluation: $y_j \leftarrow \mathrm{Eval}(f, s_{j,1}, \dots, s_{j,n})$, $z \leftarrow \mathrm{Dec}(y_1, \dots, y_m)$
  - $z = f(x_1, \dots, x_n)$



3

Application to MPC

- MPC based on HSS
  - *Two* rounds
  - *Succinctness*: communication cost ≈ share size (independent of the *size* of f)
  - Corruption power is characterized by its **adversary structure**



4

Adversary Structure

The collection Δ of subsets of players revealing no information on a secret input

- Threshold

$$\Delta = \{X : |X| \le t\}$$

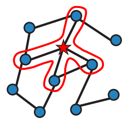
- Multipartite

  $\Pi = (P_1, \dots, P_L)$: partition of $[m]$

  Whether $X \in \Delta$ or not is uniquely determined by $(|X \cap P_1|, \dots, |X \cap P_L|)$

- General

  No assumption on Δ



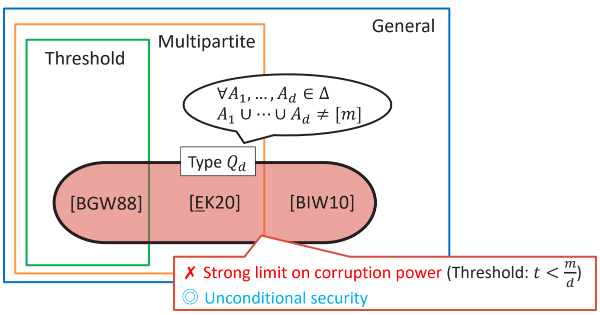
$\Pi = ([m]) \Rightarrow$ Threshold
$\Pi = (\{1\}, \dots, \{m\}) \Rightarrow$ General

Example: Δ induced by a **graph**
Adversary colludes with *adjacent* players

5

Previous HSS

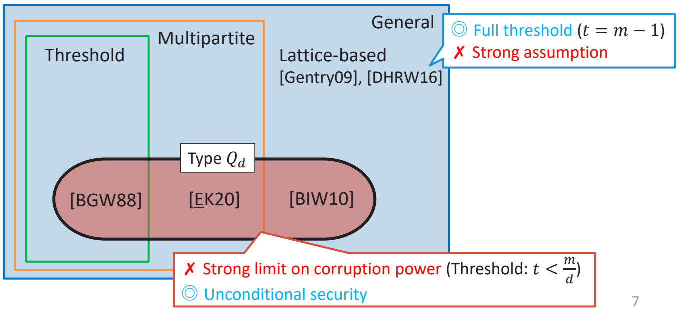
- HSS for degree-d polynomials



6

Previous HSS

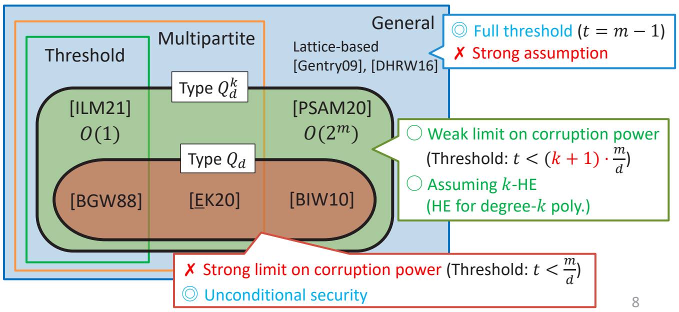
- HSS for degree-d polynomials



7

Previous HSS

- HSS for degree-d polynomials

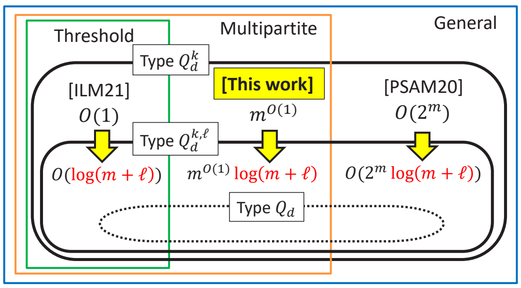


8

Our Results

✓ Multipartite HSS for degree-d polynomials from k-HE

✓ Extension to ℓ parallel evaluations of a single polynomial (SIMD operation)

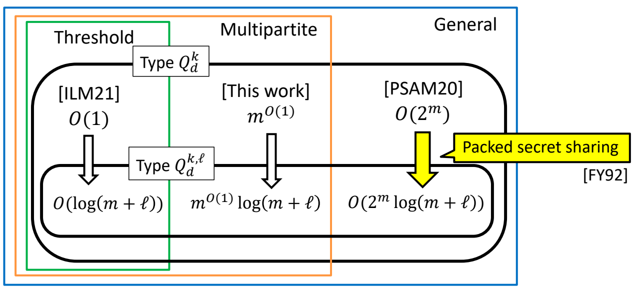


9

General HSS for Parallel Evaluation

10

Parallel Evaluation of Degree-d Poly.



11

Our Proposed HSS

HSS [PSAM20]

✓ General adversary structure

✗ NOT support parallel evaluation

Packed secret sharing [FY92]

✗ Threshold adversary structure

✓ Support parallel evaluation

Our HSS

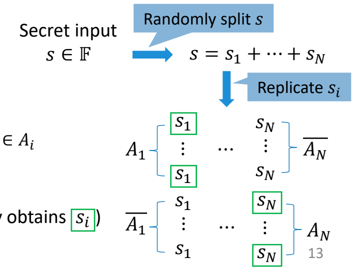
✓ General adversary structure

✓ Support parallel evaluation

12

Starting Point: HSS of [PSAM20]

- **General adversary structure** $\Delta \subseteq 2^{[m]}$
  - Monotonically decreasing ($A \subseteq B \subseteq [m]$ and $B \in \Delta \Rightarrow A \in \Delta$)
  - All maximal subsets $\Delta^+ = \{A_1, \dots, A_N\} = \{A : A \in \Delta \text{ and } B \notin \Delta \text{ for all } B \supsetneq A\}$
- **HSS scheme of [PSAM20]**
  - Assuming k-HE, $x \mapsto \boxed{x}$
  - Share generation:
    1. Randomly split s into $s_1, \dots, s_N$
    2. Give s_i to $j \in \overline{A_i}$ and give $\boxed{s_i}$ to $j \in A_i$
- **Privacy**
  - Coalition of $A_i \in \Delta^+$ misses s_i (only obtains $\boxed{s_i}$)



13

Packed Secret Sharing [FY92]

- Threshold adversary structure $\{X : |X| \leq t\}$
- Secret input: $(s_1, \dots, s_\ell) \in \mathbb{F}^\ell$
- **Share generation**:
  1. Choose a random polynomial $\varphi \in \mathbb{F}[X]$ such that
     $s_1 = \varphi(\alpha_1), \dots, s_\ell = \varphi(\alpha_\ell), \qquad \deg \varphi \leq t + \ell - 1.$
  2. Give $\varphi(j)$ to Server j
- **Privacy**
  $\varphi(j_1), \dots, \varphi(j_t)$ reveals no information on $(s_1, \dots, s_\ell)$
- **Reconstruction**
  φ can be recovered from shares of $A \subseteq [m]$ if $|A| \geq t + \ell$

14

Our Proposed HSS

HSS [PSAM20]

✓ General adversary structure

✗ NOT support parallel evaluation

Packed secret sharing [FY92]

✗ Threshold adversary structure

✓ Support parallel evaluation

Our HSS

✓ General adversary structure

✓ Support parallel evaluation

15

Our Proposed HSS

- **General adversary structure** $\Delta \subseteq 2^{[m]}$
  - All maximal subsets $\Delta^+ = \{A_1, \dots, A_N\} = \{A : A \in \Delta \text{ and } B \notin \Delta \text{ for all } B \supsetneq A\}$
- **Our HSS scheme**
  - Assuming k-HE, $x \mapsto \boxed{x}$
  - Share generation:
    1. Randomly split $\vec{s}$ into $\vec{s}_1, \dots, \vec{s}_N$
    2. Share $\vec{s}_u$ via packed secret sharing: Give $\varphi_u(j)$ to $j \in \overline{A_u}$ and give $\boxed{\varphi_u(j)}$ to $j \in A_u$ where $\deg \varphi_u \leq \ell - 1$
- **Privacy**
  - Coalition of $A_u \in \Delta^+$ misses $\vec{s}_u$

Secret input $\vec{s} \in \mathbb{F}^\ell$

Randomly split $\vec{s}$

$\vec{s} = \vec{s}_1 + \cdots + \vec{s}_N$

Share $\vec{s}_u$

$$\begin{array}{cccc} A_1 & \boxed{\varphi_1(j)} & \varphi_N(j) & \overline{A_N} \\ & \vdots \quad \cdots & \vdots & \\ \overline{A_1} & \varphi_1(j) & \boxed{\varphi_N(j)} & A_N \end{array}$$

16

Parallel Evaluation

- **Evaluation of a *single* degree-d polynomial**
  - Secret input: $\vec{x}^{(1)}, \dots, \vec{x}^{(d)} \in \mathbb{F}^\ell$
  - $\vec{x}^{(i)}$ is randomly split into $\vec{x}^{(i)} = \vec{y}_1^{(i)} + \cdots + \vec{y}_N^{(i)}$
  - $\vec{y}_u^{(i)} = \left(\varphi_u^{(i)}(\alpha_1), \dots, \varphi_u^{(i)}(\alpha_\ell)\right)$, $\deg \varphi_u^{(i)} \leq \ell - 1$

$$\Rightarrow \quad \vec{x}^{(1)} * \cdots * \vec{x}^{(d)} = \sum_{\boldsymbol{j}=(j_1,\dots,j_d)\in[N]^d} \vec{y}_{j_1}^{(1)} * \cdots * \vec{y}_{j_d}^{(d)}$$

More than $d(\ell - 1)$ points of $\varphi_{j_1}^{(1)} \cdots \varphi_{j_d}^{(d)}$ must be collected from servers

$*$ denotes the element-wise product 17

Parallel Evaluation

- **Evaluation of a *single* degree-d polynomial**
  - Server i has $\begin{cases} \varphi_u^{(1)}(i), \dots, \varphi_u^{(d)}(i) & \text{if } i \notin A_u \\ \boxed{\varphi_u^{(1)}(i)}, \dots, \boxed{\varphi_u^{(d)}(i)} & \text{if } i \in A_u \end{cases}$
  - Server i can compute

$$\boxed{(\varphi_{j_1}^{(1)} \cdots \varphi_{j_d}^{(d)})(i)} = \prod_{h:i\notin A_{j_h}} \varphi_{j_h}^{(h)}(i) \prod_{h:i\in A_{j_h}} \boxed{\varphi_{j_h}^{(h)}(i)}$$

Product of k ciphertexts

$$\vec{y}_u^{(1)} = \left(\varphi_u^{(1)}(\alpha_1), \dots, \varphi_u^{(1)}(\alpha_\ell)\right)$$

$$\begin{array}{cc} \overline{A_u} & \varphi_u^{(1)}(i) \\ & \vdots \\ A_u & \boxed{\varphi_u^{(1)}(i)} \end{array}$$

$$\forall (j_1, \dots, j_d) \in [N]^d, \quad \#\{i \in [m] : \#\{h : i \in A_{j_h}\} \leq k\} > d(\ell - 1)$$

18

Remaining Problem

- **Context hiding**
  - Output shares should reveal nothing beyond $f(x_1, \dots, x_n)$
  - Needs **re-randomization** with fresh shares of 0

$\Downarrow$

Type $Q_d^{k,\ell}$

$$\forall (j_1, \dots, j_d) \in [N]^d, \sum_{i \in [m]} \left(k - \#\{h : i \in A_{j_h}\}\right)_+ > (d+1)(\ell - 1)$$

Type $Q_d^{k,\ell}$ $\Rightarrow$ Type $Q_d^{k,1}$ $\Leftrightarrow$ Type Q_d^k

$(x)_+ := \max\{x, 0\}$

19

Efficiency

- **Share size**
  - Server i receives $O(N)$ field elements and $O(N)$ ciphertexts
  - $m + \ell$ different points are necessary

$\Rightarrow$ $O(2^m \log(m + \ell))$

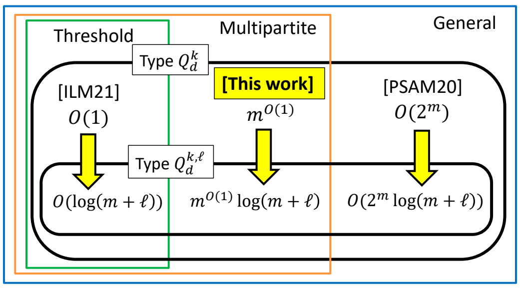
All maximal subsets $\Delta^+ = \{A_1, \dots, A_N\}$

20

Conclusion

✓ Multipartite HSS for degree-d polynomials from HE

✓ Extension to ℓ parallel evaluations of a single polynomial (SIMD operation)



21

References

[BGI16]: E. Boyle, N. Gilboa, and Y. Ishai. Breaking the circuit size barrier for secure computation under DDH. *CRYPTO.*

[BGW88]: M. Ben-Or, S. Goldwasser, and A. Wigderson. Completeness theorems for non-cyptographic fault-tolerant distributed computation. *STOC.*

[EK20]: R. Eriguchi and N. Kunihiro. d-Multiplicative secret sharing for multipartite adversary structures. *ITC.*

[BIW10]: O. Barkol, Y. Ishai, and E. Weinreb. On d-multiplicative secret sharing. *Journal of Cryptology.*

[Gentry09]: C. Gentry. Fully homomorphic encryption using ideal lattices. *STOC.*

[DHRW16]: Y. Dodis, S. Halevi, R.D. Rothblum, D. Wichs. Spooky encryption and its applications. *CRYPTO.*

22

References

[ILM21]: Y. Ishai, R.W.F. Lai, and G. Malavolta. A geometric approach to homomorphic secret sharing. *PKC.*

[PSAM20]: K. Phalakarn, V. Suppakitpaisarn, N. Attrapadung, and K. Matuura. Constructive t-secure homomorphic secret sharing for low degree polynomials. *INDOCRYPT.*

[FY92]: M. Franklin and M. Yung. Communication complexity of secure computation. *STOC.*

23

Thank you!

24

A Comparison of How to Garble Arithmetic and Boolean Circuits - Case of Functional Encryption -

Hiroaki Anada (Joint work with Kotaro Chinen)

University of Nagasaki
anada@sun.ac.jp

The technique of *garbling circuits* that was initiated by Yao [1] is currently one of the fundamental cryptographic primitives. It is generalized and enhanced as the *randomized encoding* of functions [2, 3], which can treat not only boolean circuits but also arithmetic circuits. In this talk, after warming up with examples of randomized encoding, we focus into garbling encryption circuits of functional encryption following the work of Goyal, Koppla and Waters [4]. We see that there is a gap between "boolean" and "arithmetic" in security proofs; in the case of boolean, we only have to select one of *two evaluated* keys, but in the case of arithmetic, we must evaluate a key-value for *any given* input.

References

[1] Andrew Chi-Chih Yao. How to generate and exchange secrets (extended abstract). In *27th Annual Symposium on Foundations of Computer Science, Toronto, Canada, 27-29 October 1986*, pages 162–167. IEEE Computer Society, 1986.

[2] Benny Applebaum, Yuval Ishai, and Eyal Kushilevitz. How to garble arithmetic circuits. In Rafail Ostrovsky, editor, *IEEE 52nd Annual Symposium on Foundations of Computer Science, FOCS 2011, Palm Springs, CA, USA, October 22-25, 2011*, pages 120–129. IEEE Computer Society, 2011.

[3] Benny Applebaum. Garbled circuits as randomized encodings of functions: a primer. In Yehuda Lindell, editor, *Tutorials on the Foundations of Cryptography*, pages 1–44. Springer International Publishing, 2017.

[4] Rishab Goyal, Venkata Koppula, and Brent Waters. Semi-adaptive security and bundling functionalities made generic and easy. In Martin Hirt and Adam D. Smith, editors, *Theory of Cryptography - 14th International Conference, TCC 2016-B, Beijing, China, October 31 - November 3, 2016, Proceedings, Part II*, volume 9986 of *Lecture Notes in Computer Science*, pages 361–388, 2016.

Institute of Mathematics for Industry
Kyushu University

IMI Workshop of the Joint Research Projects
November 8 to 10, 2021

"Exploring Mathematical and Practical Principles of Secure Computation and Secret Sharing"

A Comparison of How to Garble Boolean & Arithmetic Circuits - Case of Functional Encryption -

Hiroaki ANADA
(Joint work with Kotaro CHINEN: 2nd grade of master course)
University of Nagasaki

Part I: Intuitive Introduction to

Randomized Encodings (RE)

2

Garbling techniques

[1] Yao: "How to generate and exchange secrets", SFCS 1986
[2] Applebaum, Ishai, Kushilevitz: "How to Garble Arithmetic Circuits", FOCS 2011

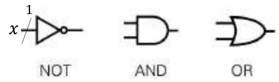
→ GC-based secure computation ☺

→ This talk: Garbling functional encryption circuit (☹?)

3

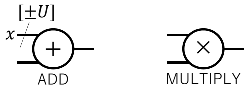
Gates: Boolean / Arithmetic

- Boolean gates



$x \in \{0,1\}$

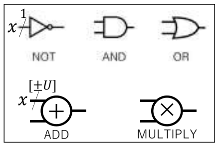
- Arithmetic gates



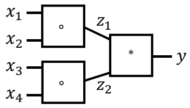
$x \in \mathbb{Z}$ (or finite ring R)

4

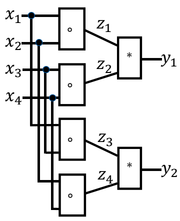
Formulars & Circuits



- Formula
  - $(x_1 \circ x_2) * (x_3 \circ x_4) \to y$



- Circuit
  - $(x_1 \circ x_2) * (x_3 \circ x_4) \to y_1$
  - $(x_1 \circ x_3) * (x_2 \circ x_4) \to y_2$



5

Yao's garbling [1]

$$C: \{0,1\}^n \to \{0,1\}^m$$
$$\Psi$$
$$x = x_1 \dots x_n$$

$$C \mapsto \left(\left(\hat{C}, \left((K_1^0, K_1^1), \dots, (K_n^0, K_n^1)\right)\right), \mathrm{Deco}, \mathrm{Simu}\right)$$

1. (Correctness)

$$\mathrm{Deco}\left(\hat{C}, K_1^{x_1}, \dots, K_1^{x_n}\right) \to C(x)$$

2. (Privacy)

$$\mathrm{Simu}\left(C(x)\right) \to S \approx \left(\hat{C}, K_1^{x_1}, \dots, K_1^{x_n}\right)$$

[1] Yao: "How to generate and exchange secrets", SFCS 1986

6

Garbling as randomized encoding (RE) of functions [2]

$$f: X \to Y$$
$$\Downarrow$$
$$x$$

$$f(\cdot) \mapsto \Big(\overbrace{\big(\hat{f}_{\mathrm{off}}(r), \hat{f}_{\mathrm{on}}(\,\cdot\,;\, r)\big)}^{(M,K)}, \mathrm{Deco}, \mathrm{Simu} \Big)$$

1. (Correctness)

$$\mathrm{Deco}\big(\hat{f}_{\mathrm{off}}(r), \hat{f}_{\mathrm{on}}(x;\, r)\big) \to f(x)$$

2. (Privacy)

$$\mathrm{Simu}\big(f(x)\big) \to S \approx \big(\hat{f}_{\mathrm{off}}(r), \hat{f}_{\mathrm{on}}(x;\, r)\big)$$

[3] Applebaum: “Garbled Circuits as Randomized Encodings of Functions: a Primer” in *Tutorials on the Foundations of Cryptography*, pp.1-44, 2017

DARE (: Decomposable Affine RE) [3]

1. Decomposable:
   - Each output-entry (y_j) contains at most one input-entry (x_i)
   - $y_1 = r_1 x_1^2 + r_2$

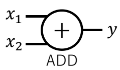
2. Affine:
   - Each output-entry (y_j) is affine function of input-entries $(x_1, \dots, x_n)$
   - $y_1 = r_1 x_1 + r_2 x_2$

[2] Applebaum, Ishai, Kushilevitz: "How to Garble Arithmetic Circuits", FOCS 2011

Example 1

$$f(\cdot) \mapsto \Big(\big(\hat{f}_{\mathrm{off}}(r), \hat{f}_{\mathrm{on}}(\,\cdot\,;\, r)\big), \mathrm{Deco}\big(\hat{f}_{\mathrm{off}}(r), \hat{f}_{\mathrm{on}}(x;\, r)\big), \mathrm{Simu}(f(x)) \Big)$$

$$f(x) = f(x_1, x_2) \coloneqq x_1 + x_2 = y$$



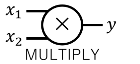
- $\hat{f}_{\mathrm{off}}(r) \coloneqq \emptyset$
- $\hat{f}_{\mathrm{on}}((\ _1, \ _2);\, r) \coloneqq (\ _1 + r, \ \ _2 - r)$

- $\mathrm{Deco}\big(\emptyset, (K_1, K_2)\big) \coloneqq K_1 + K_2$
  - $\mathrm{Deco}\Big(\emptyset, \big((x_1 + r), (x_2 - r)\big)\Big) \to (x_1 + r) + (x_2 - r) = x_1 + x_2 = y = f(x)$
- $\mathrm{Simu}(y; \tilde{r}) \coloneqq \big(\emptyset, (y + \tilde{r}, -\tilde{r})\big) = S$
  - $\mathrm{Simu}(y) \to S \approx \big(\emptyset, (x_1 + r, x_2 - r)\big)$

[2] Applebaum, Ishai, Kushilevitz: "How to Garble Arithmetic Circuits", FOCS 2011, pp.120-129
[3] Applebaum: “Garbled Circuits as Randomized Encodings of Functions: a Primer” in *Tutorials on the Foundations of Cryptography*, pp.1-44, 2017

Example 2

$f(\cdot) \mapsto \left(\left(\hat{f}_{\mathrm{off}}(r), \hat{f}_{\mathrm{on}}(\cdot\,;\, r)\right), \mathrm{Deco}\left(\hat{f}_{\mathrm{off}}(r), \hat{f}_{\mathrm{on}}(x;\, r)\right), \mathrm{Simu}(f(x))\right)$

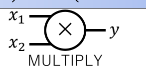


$f(x) = f(x_1, x_2) \coloneqq x_1 x_2 =: y$

- $\hat{f}_{\mathrm{off}}(r) \coloneqq \emptyset, \quad r = (r_1, r_2)$
- $\hat{f}_{\mathrm{on}}((\ {}_1,\ \ {}_2);\, r_1, r_2) \coloneqq (\ {}_1 + r_1,\ \ {}_2 + r_2,\ r_2\ \ {}_1 + r_1\ \ {}_2 + r_1 r_2)$
- $\mathrm{Deco}(\emptyset, (K_1, K_2, K_3)) \coloneqq K_1 K_2 - K_3$
  - $\mathrm{Deco}(\emptyset, (x_1 + r_1,\ x_2 + r_2,\ r_2 x_1 + r_1 x_2 + r_1 r_2)) \to (x_1 + r_1)(x_2 + r_2) - (r_2 x_1 + r_1 x_2 + r_1 r_2)$
    $= x_1 x_2 = y = f(x)$
- $\mathrm{Simu}(y; \tilde{r}_1, \tilde{r}_2) \coloneqq (\emptyset, (\tilde{r}_1, \tilde{r}_2, \tilde{r}_1 \tilde{r}_2 - y)) =: S$
  - $\mathrm{Simu}(y) \to S \approx (\emptyset, (x_1 + r_1,\ x_2 + r_2,\ r_2 x_1 + r_1 x_2 + r_1 r_2))$

Example 2’

$f(\cdot) \mapsto \left(\left(\hat{f}_{\mathrm{off}}(r), \hat{f}_{\mathrm{on}}(\cdot\,;\, r)\right), \mathrm{Deco}\left(\hat{f}_{\mathrm{off}}(r), \hat{f}_{\mathrm{on}}(x;\, r)\right), \mathrm{Simu}(f(x))\right)$

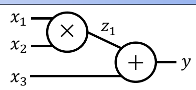


$f(x) = f(x_1, x_2) \coloneqq x_1 x_2 =: y$

- $\hat{f}_{\mathrm{off}}(r) \coloneqq \emptyset, \quad r = (r_1, r_2, r_3)$
- $\hat{f}_{\mathrm{on}}((\ {}_1,\ \ {}_2);\, r_1, r_2, r_3) \coloneqq (\ {}_1 + r_1,\ \ {}_2 + r_2,\ r_2\ \ {}_1 + r_3,\ r_1\ \ {}_2 + r_1 r_2 - r_3)$
  $= \left(\begin{bmatrix} {}_1 + r_1 \\ r_2\ \ {}_1 + r_3 \end{bmatrix}, \begin{bmatrix} {}_2 + r_2 \\ r_1\ \ {}_2 + r_1 r_2 - r_3 \end{bmatrix}\right)$
- $\mathrm{Deco}\left(\emptyset, \left(\begin{bmatrix} K_{11} \\ K_{12} \end{bmatrix}, \begin{bmatrix} K_{21} \\ K_{22} \end{bmatrix}\right)\right) \coloneqq K_{11} K_{21} - (K_{12} + K_{22})$
  - $\mathrm{Deco}\left(\emptyset, \left(\begin{bmatrix} x_1 + r_1 \\ r_2 x_1 + r_3 \end{bmatrix}, \begin{bmatrix} x_2 + r_2 \\ r_1 x_2 + r_1 r_2 - r_3 \end{bmatrix}\right)\right) \to (x_1 + r_1)(x_2 + r_2) - (r_2 x_1 + r_3 + r_1 x_2 + r_1 r_2 - r_3)$
    $= x_1 x_2 = y = f(x)$
- $\mathrm{Simu}(y; \tilde{r}_1, \tilde{r}_2, \tilde{r}_3) \coloneqq \left(\emptyset, \left(\begin{bmatrix} \tilde{r}_1 \\ \tilde{r}_1 \tilde{r}_2 - y + \tilde{r}_3 \end{bmatrix}, \begin{bmatrix} \tilde{r}_2 \\ -\tilde{r}_3 \end{bmatrix}\right)\right) =: S$
  - $\mathrm{Simu}(y) \to S \approx \left(\emptyset, \left(\begin{bmatrix} x_1 + r_1 \\ r_2 x_1 + r_3 \end{bmatrix}, \begin{bmatrix} x_2 + r_2 \\ r_1 x_2 + r_1 r_2 - r_3 \end{bmatrix}\right)\right)$

Example 3

$f(\cdot) \mapsto \left(\left(\hat{f}_{\mathrm{off}}(r), \hat{f}_{\mathrm{on}}(\cdot\,;\, r)\right), \mathrm{Deco}\left(\hat{f}_{\mathrm{off}}(r), \hat{f}_{\mathrm{on}}(x;\, r)\right), \mathrm{Simu}(f(x))\right)$

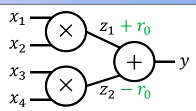


$f(x) = f(x_1, x_2, x_3) \coloneqq x_1 x_2 + x_3 =: y$

- $\hat{f}_{\mathrm{off}}(r) = \emptyset, r = (r_1, r_2, r_3, r_4)$
- $\hat{f}_{\mathrm{on}}((\ {}_1,\ \ {}_2,\ \ {}_3);\, r_1, r_2, r_3, r_4) \coloneqq \left(\begin{bmatrix} {}_1 + r_1 \\ r_2\ \ {}_1 + r_3 \end{bmatrix}, \begin{bmatrix} {}_2 + r_2 \\ r_1\ \ {}_2 + r_1 r_2 - r_3 - r_4 \end{bmatrix},\ \ {}_3 - r_4\right)$
- $\mathrm{Deco}\left(\emptyset, \left(\begin{bmatrix} K_{11} \\ K_{12} \end{bmatrix}, \begin{bmatrix} K_{21} \\ K_{22} \end{bmatrix}, K_3\right)\right) \coloneqq (K_{11} K_{21} - (K_{12} + K_{22})) + K_3$
  - $\mathrm{Deco}\left(\emptyset, \left(\begin{bmatrix} x_1 + r_1 \\ r_2 x_1 + r_3 \end{bmatrix}, \begin{bmatrix} x_2 + r_2 \\ r_1 x_2 + r_1 r_2 - r_3 - r_4 \end{bmatrix},\ x_3 - r_4\right)\right) \to x_1 x_2 + x_3 = y = f(x)$
- $\mathrm{Simu}(y; \tilde{r}_1, \tilde{r}_2, \tilde{r}_3, \tilde{r}_4) \coloneqq \left(\emptyset, \left(\begin{bmatrix} \tilde{r}_1 \\ \tilde{r}_1 \tilde{r}_2 - y + \tilde{r}_3 \end{bmatrix}, \begin{bmatrix} \tilde{r}_2 \\ -\tilde{r}_3 - \tilde{r}_4 \end{bmatrix}, -\tilde{r}_4\right)\right) =: S$
  - $\mathrm{Simu}(y) \to S \approx \left(\emptyset, \left(\begin{bmatrix} x_1 + r_1 \\ r_2 x_1 + r_3 \end{bmatrix}, \begin{bmatrix} x_2 + r_2 \\ r_1 x_2 + r_1 r_2 - r_3 - r_4 \end{bmatrix},\ x_3 - r_4\right)\right)$

Example 4

$f(\cdot) \mapsto \left(\left(\hat{f}_{\mathrm{off}}(r), \hat{f}_{\mathrm{on}}(\,\cdot\,;\, r)\right), \mathrm{Deco}\left(\hat{f}_{\mathrm{off}}(r), \hat{f}_{\mathrm{on}}(x;\, r)\right), \mathrm{Simu}(f(x))\right)$



$f(x) = f(x_1, x_2, x_3, x_4) \coloneqq x_1 x_2 + x_3 x_4$

- $\hat{f}_{\mathrm{off}}(r) = \hat{f}_{\mathrm{off}}(r_0, r_1, r_2, r_3, r_4, r_5, r_6, r_7, r_8) \coloneqq \begin{bmatrix} r_0 - r_4 \\ -r_0 - r_8 \end{bmatrix}$
- $\hat{f}_{\mathrm{on}}((\ {}_1,\ {}_2,\ {}_3,\ {}_4)\,; r_0, r_1, r_2, r_3, r_4, r_5, r_6, r_7, r_8)$

$$\coloneqq \left(\begin{bmatrix} {}_1 + r_1 \\ r_2 \quad {}_1 + r_3 \end{bmatrix}, \begin{bmatrix} {}_2 + r_2 \\ r_1 \quad {}_2 + r_1 r_2 - r_3 - r_4 \end{bmatrix}, \begin{bmatrix} {}_3 + r_5 \\ r_6 \quad {}_3 + r_7 \end{bmatrix}, \begin{bmatrix} {}_4 + r_6 \\ r_5 \quad {}_4 + r_5 r_6 - r_7 - r_8 \end{bmatrix}\right)$$

- $\mathrm{Deco}\left([M_{11}, M_{12}], \left(\begin{bmatrix} K_{11} \\ K_{12} \end{bmatrix}, \begin{bmatrix} K_{21} \\ K_{22} \end{bmatrix}, \begin{bmatrix} K_{31} \\ K_{32} \end{bmatrix}, \begin{bmatrix} K_{41} \\ K_{42} \end{bmatrix}\right)\right)$

$$\coloneqq \left(K_{11} K_{21} - (K_{12} + K_{22})\right) + W_{11} + \left(K_{31} K_{41} - (K_{32} + K_{42})\right) + W_{12}$$

- $\mathrm{Simu}(y; r_0, \tilde{r}_1, \tilde{r}_2, \tilde{r}_3, \tilde{r}_4, \tilde{r}_5, \tilde{r}_6, \tilde{r}_7, \tilde{r}_8)$

$$\coloneqq \left([\tilde{r}_0 - \tilde{r}_4, -\tilde{r}_0 - \tilde{r}_8], \left(\begin{bmatrix} \tilde{r}_1 \\ \tilde{r}_1 \tilde{r}_2 - y + \tilde{r}_3 \end{bmatrix}, \begin{bmatrix} \tilde{r}_2 \\ -\tilde{r}_3 - \tilde{r}_4 \end{bmatrix}, \begin{bmatrix} \tilde{r}_5 \\ \tilde{r}_5 \tilde{r}_6 - y + \tilde{r}_7 \end{bmatrix}, \begin{bmatrix} \tilde{r}_6 \\ -\tilde{r}_7 - \tilde{r}_8 \end{bmatrix}\right)\right) \eqqcolon S$$

13

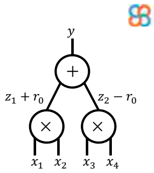
DARE (: Decomposable Affine RE) [3]

1. Decomposable:
   - Each output-entry (y_j) contains at most one input-entry (x_i)
   - $y_1 = r_1 x_1^2 + r_2$
2. Affine:
   - Each output-entry (y_j) is affine function of input-entries ($x_1, \ldots, x_n$)
   - $y_1 = r_1 x_1 + r_2 x_2$
3. Decomposable & Affine
   - $y_1 = r_1 x_1 + r_2$

[2] Applebaum, Ishai, Kushilevitz: "How to Garble Arithmetic Circuits", FOCS 2011

14

DARE:
Problem at Affinization



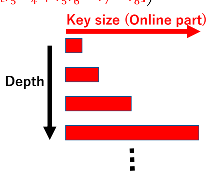
$f(x) = f(x_1, x_2, x_3, x_4) \coloneqq x_1 x_2 + x_3 x_4$

- $\hat{f}_{\mathrm{off}}(r) = \hat{f}_{\mathrm{off}}(r_0, r_1, r_2, r_3, r_4, r_5, r_6, r_7, r_8) \coloneqq \begin{bmatrix} r_0 - r_4 \\ -r_0 - r_8 \end{bmatrix}$ M
- $\hat{f}_{\mathrm{on}}((\ {}_1,\ {}_2,\ {}_3,\ {}_4)\,; r_0, r_1, r_2, r_3, r_4, r_5, r_6, r_7, r_8)$

$$\coloneqq \left(\begin{bmatrix} {}_1 + r_1 \\ r_2 \quad {}_1 + r_3 \end{bmatrix}, \begin{bmatrix} {}_2 + r_2 \\ r_1 \quad {}_2 + r_1 r_2 - r_3 - r_4 \end{bmatrix}, \begin{bmatrix} {}_3 + r_5 \\ r_6 \quad {}_3 + r_7 \end{bmatrix}, \begin{bmatrix} {}_4 + r_6 \\ r_5 \quad {}_4 + r_5 r_6 - r_7 - r_8 \end{bmatrix}\right)$$

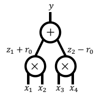
K

- **Exponential blowup** (of key size) in "**Depth of C**".. ☹

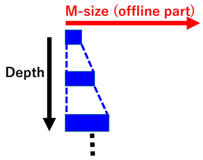


15

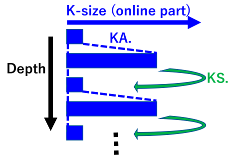
DARE:
Key-Affinization then Key-Shrinkage [2, 3]



- $\hat{f}_{\mathrm{off}}(r) = [\quad] = M$



- $\hat{f}_{\mathrm{on}}(x_1 \ldots x_n;\ r) = (K_1, \ldots, K_n) = K$



[2] Applebaum, Ishai, Kushilevitz: "How to Garble Arithmetic Circuits", FOCS 2011, pp.120-129
[3] Applebaum: "Garbled Circuits as Randomized Encodings of Functions: a Primer" in *Tutorials on the Foundations of Cryptography*, pp.1-44, 2017

16

Part II: Our Observation on

Garbling "Functional Encryption" Circuits - Boolean vs. Arithmetic -

17

Motivation (1/4)
Functional encryption (FE)

$\mathrm{FE} = (\mathrm{Setup},\ \mathrm{KG},\ \mathrm{Encr},\ \mathrm{Decr})$

- $\mathrm{Setup_{FE}}(1^\lambda, 1^n) \to (\mathrm{mpk}, \mathrm{msk})$
- $\mathrm{KG_{FE}}(\mathrm{mpk}, \mathrm{msk}, f) \to \mathrm{sk}_f$
- $\mathrm{Encr_{FE}}(\mathrm{mpk}, m; R) \to \mathrm{ct}$
- $\mathrm{Decr_{FE}}(\mathrm{mpk}, \mathrm{sk}_f, \mathrm{ct}) \to m$

- Identity-based Encr
- Attribute-based Encr
- Inner-product Encr

…

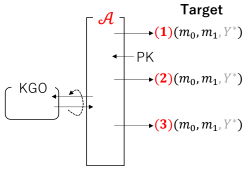
18

Motivation (2/4)

Security levels of FE

Table1: Available data before target-fixing

Data	(1)Selective	(2)Semi-adaptive	(3)Adaptive
PK	–	✓	✓
Oracle	–	–	✓



(1) < (2) < (3):

Advantageous for Adversary $\mathcal{A}$ ☹

But, when achieved, (3) or (2) is ideal ☺

19

Motivation (3/4)

(1)Selective-to-(2)Semi-adaptive Conversion[4]

- Circuit $C \coloneqq \mathrm{Encr_{sel}}(\cdot,\ m; R)$

$$C: \{0,1\}^n \to \{0,1\}^m$$
$$\mathrm{mpk} = x_1 \dots x_n$$

- Yao's garbling:

$$C \mapsto \left((\hat{C}, (K_1^0, K_1^1), \dots, (K_n^0, K_n^1)), \mathrm{Deco}, \mathrm{Simu} \right)$$

[4] Goyal, Koppla, Waters: "Semi-Adaptive Security and Bundling Functionalities Made Generic and Easy", TCC 2016

20

Motivation (4/4)

(1)Sel-to-(2)Sma. Conv.

- $\mathrm{mpk_{sel}} = x_1 \dots x_n$

  $C \mapsto \left((\hat{C}, (K_1^0, K_1^1), \dots, (K_n^0, K_n^1)), \mathrm{Deco}, \mathrm{Simu} \right)$
- Use $\mathrm{PKE} = (\mathrm{KG_{PKE}}, \mathrm{Enc_{PKE}}, \mathrm{Dec_{PKE}})$

$\mathrm{FE_{sma}} = (\mathrm{Setup_{sma}}, \mathrm{KG_{sma}}, \mathrm{Encr_{sma}}, \mathrm{Decr_{sma}})$

- $\mathrm{Setup_{sma}}(1^\lambda, 1^n)$
  - $\mathrm{Setup_{sel}}(1^\lambda, 1^n) \to (\mathrm{mpk_{sel}}, \mathrm{msk_{sel}})$
  - $\mathrm{KG_{PKE}}(1^\lambda) \to$
  - $(\mathrm{pk}_{10}, \mathrm{sk}_{10}), (\mathrm{pk}_{10}, \mathrm{sk}_{11}), \dots, (\mathrm{pk}_{n0}, \mathrm{sk}_{n1}), (\mathrm{pk}_{n0}, \mathrm{sk}_{n1})$
  - $\mathrm{mpk_{smd}} \coloneqq (\mathrm{pk}_{ib})_i^b$
  - $\mathrm{msk_{smd}} \coloneqq ((\mathrm{sk}_{ib})_i^b, \mathrm{mpk_{sel}}, \mathrm{msk_{sel}})$
  - Return $(\mathrm{mpk_{smd}}, \mathrm{msk_{smd}})$
- $\mathrm{KG_{sma}}(\mathrm{mpk_{smd}}, \mathrm{msk_{smd}}, f)$
  - $\mathrm{KG_{sel}}(\mathrm{mpk_{sel}}, \mathrm{msk_{sel}}) \to \mathrm{sk}_{\mathrm{sel},f}$
  - $\mathrm{sk}_{\mathrm{sma},f} \coloneqq \left((\mathrm{sk}_{ix_i})_i, \mathrm{mpk_{sel}}, \mathrm{sk}_{\mathrm{sel},f} \right)$
  - Return $(\mathrm{sk}_{\mathrm{sma},f})$
- $\mathrm{Encr_{sma}}(\mathrm{mpk_{smd}}, m)$
  - $R \in \{0,1\}^{l(\lambda,n)}$
  - $C \coloneqq \mathrm{Encr_{sel}}(\cdot,\ m; R)$
  - $\mapsto \left((\hat{C}, (K_1^0, K_1^1), \dots, (K_n^0, K_n^1)), \mathrm{Deco}, \mathrm{Simu} \right)$
  - $\left(\mathrm{Enc_{PKE}}(\mathrm{pk}_{ib}, K_i^b) \to \mathrm{ct}_{ib} \right)_i^b$
  - Return $\mathrm{ct_{smd}} = \left(\hat{C}, (\mathrm{ct}_{ib})_i^b \right)$
- $\mathrm{Decr_{sma}}(\mathrm{mpk_{smd}}, \mathrm{sk}_{\mathrm{sma},f}, \mathrm{ct_{smd}})$
  - $(K_i^{x_i})_i, \mathrm{Deco}(\hat{C}, K_1^{x_1}, \dots, K_1^{x_n}) \to \mathrm{ct_{sel}}$
  - $\mathrm{Decr_{sel}}(\mathrm{mpk_{sel}}, \mathrm{sk}_{\mathrm{sel},f}, \mathrm{ct_{sel}}) \to m$
  - Return (m)

21

Our observation

A comparison of garbling Boolean & Arithmetic circuits

Boolean

- Only has to select one of two evaluated keys

$$\left(\hat{C}, K = (K_1^0, K_1^1), \ldots, (K_n^0, K_n^1)\right)$$

  - "Selection function" selects one of the two for each place

$$\mathrm{mpk}_{\mathrm{sel}} = x_1 \ldots x_n, \qquad \left(\hat{C}, K_1^{x_1}, \ldots, K_1^{x_n}\right)$$

Arithmetic

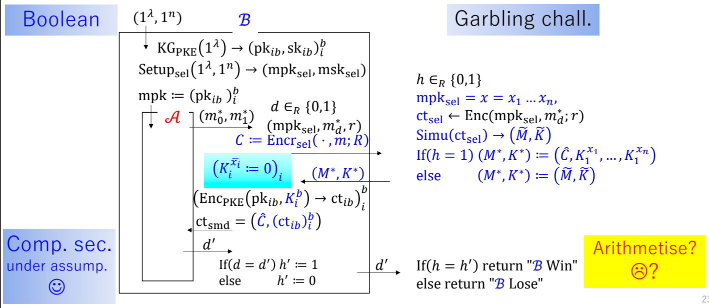
- Must evaluate a key-value for any given input in $[\pm U] \subset \mathbb{Z}$

$$\hat{f}(\cdot\,; r) = \left(\hat{f}_{\mathrm{off}}(r), \hat{f}_{\mathrm{on}}(\cdot\,; r)\right) = (M(r), K(\cdot\,; r))$$

22

What's problem?☹?

Game with garbling challenger for FE security



23

Problem to be solved..

- Establish a "arithmetic GKW" compiler..
  - Hopefully by using "arithmetic garbling" [3][4]
- For what?
  - More efficiecy☺

[2] Applebaum, Ishai, Kushilevitz: "How to Garble Arithmetic Circuits", FOCS 2011, pp.120-129
[3] Applebaum: "Garbled Circuits as Randomized Encodings of Functions: a Primer" in *Tutorials on the Foundations of Cryptography*, pp.1-44, 2017

24

Summary

- There is a gap between

Boolean & Arithmetic

Boolean

- Only has to select one of two evaluated keys

$$\left(\hat{C}, K = (K_1^0, K_1^1), \dots, (K_n^0, K_n^1)\right)$$

  - "Selection function" selects one of the two for each place

$$\mathrm{mpk}_{\mathrm{sel}} = x_1 \dots x_n, \qquad (\hat{C}, K_1^{x_1}, \dots, K_1^{x_n})$$

Arithmetic

- Must evaluate a key-value for any given input in $[\pm U] \subset \mathbb{Z}$

$$\hat{f}(\cdot\,; r) = \left(\hat{f}_{\mathrm{off}}(r), \hat{f}_{\mathrm{on}}(\cdot\,; r)\right) = (M(r), K(\cdot\,; r))$$

25

Thank you for your attention!☺

26

MI レクチャーノートシリーズ刊行にあたり

本レクチャーノートシリーズは、文部科学省 21 世紀 COE プログラム「機能数理学の構築と展開」(H.15-19 年度)において作成した COE Lecture Notes の続刊であり、文部科学省大学院教育改革支援プログラム「産業界が求める数学博士と新修士養成」(H19-21 年度)および、同グローバル COE プログラム「マス・フォア・インダストリ教育研究拠点」(H.20-24 年度)において行われた講義の講義録として出版されてきた。平成 23 年 4 月のマス・フォア・インダストリ研究所 (IMI) 設立と平成 25 年 4 月の IMI の文部科学省共同利用・共同研究拠点として「産業数学の先進的·基礎的共同研究拠点」の認定を受け、今後、レクチャーノートは、マス・フォア・インダストリに関わる国内外の研究者による講義の講義録、会議録等として出版し、マス・フォア・インダストリの本格的な展開に資するものとする。

平成 30 年 10 月
マス・フォア・インダストリ研究所
所長　佐伯修

2021年度　九州大学マス·フォア·インダストリ研究所　共同利用研究集会

秘密計算・秘密分散の数理と実用の探求

発　行　2022年２月９日
編　集　穴田啓晃，池松泰彦，縫田光司，大畑幸矢，Yuntao Wang
発　行　九州大学マス・フォア・インダストリ研究所
九州大学大学院数理学府
〒819-0395 福岡市西区元岡744
九州大学数理・IMI 事務室
TEL 092-802-4402　FAX 092-802-4405
URL https://www.imi.kyushu-u.ac.jp/

印　刷　城島印刷株式会社
〒810-0012 福岡市中央区白金２丁目９番６号
TEL 092-531-7102　FAX 092-524-4411

シリーズ既刊

Issue	Author／Editor	Title	Published
COE Lecture Note	Mitsuhiro T. NAKAO Kazuhiro YOKOYAMA	Computer Assisted Proofs - Numeric and Symbolic Approaches - 199pages	August 22, 2006
COE Lecture Note	M.J.Shai HARAN	Arithmetical Investigations - Representation theory, Orthogonal polynomials and Quantum interpolations- 174pages	August 22, 2006
COE Lecture Note Vol.3	Michal BENES Masato KIMURA Tatsuyuki NAKAKI	Proceedings of Czech-Japanese Seminar in Applied Mathematics 2005 155pages	October 13, 2006
COE Lecture Note Vol.4	宮田　健治	辺要素有限要素法による磁界解析 - 機能数理学特別講義 21pages	May 15, 2007
COE Lecture Note Vol.5	Francois APERY	Univariate Elimination Subresultants - Bezout formula, Laurent series and vanishing conditions - 89pages	September 25, 2007
COE Lecture Note Vol.6	Michal BENES Masato KIMURA Tatsuyuki NAKAKI	Proceedings of Czech-Japanese Seminar in Applied Mathematics 2006 209pages	October 12, 2007
COE Lecture Note Vol.7	若山　正人 中尾　充宏	九州大学産業技術数理研究センター キックオフミーティング 138pages	October 15, 2007
COE Lecture Note Vol.8	Alberto PARMEGGIANI	Introduction to the Spectral Theory of Non-Commutative Harmonic Oscillators 233pages	January 31, 2008
COE Lecture Note Vol.9	Michael I.TRIBELSKY	Introduction to Mathematical modeling 23pages	February 15, 2008
COE Lecture Note Vol.10	Jacques FARAUT	Infinite Dimensional Spherical Analysis 74pages	March 14, 2008
COE Lecture Note Vol.11	Gerrit van DIJK	Gelfand Pairs And Beyond 60pages	August 25, 2008
COE Lecture Note Vol.12	Faculty of Mathematics, Kyushu University	Consortium “MATH for INDUSTRY” First Forum 87pages	September 16, 2008
COE Lecture Note Vol.13	九州大学大学院 数理学研究院	プロシーディング「損保数理に現れる確率モデル」 — 日新火災・九州大学 共同研究2008年11月 研究会 — 82pages	February 6, 2009

シリーズ既刊

Issue	Author／Editor	Title	Published
COE Lecture Note Vol.14	Michal Beneš, Tohru Tsujikawa Shigetoshi Yazaki	Proceedings of Czech-Japanese Seminar in Applied Mathematics 2008 77pages	February 12, 2009
COE Lecture Note Vol.15	Faculty of Mathematics, Kyushu University	International Workshop on Verified Computations and Related Topics 129pages	February 23, 2009
COE Lecture Note Vol.16	Alexander Samokhin	Volume Integral Equation Method in Problems of Mathematical Physics 50pages	February 24, 2009
COE Lecture Note Vol.17	矢嶋　　徹 及川　正行 梶原　健司 辻　　英一 福本　康秀	非線形波動の数理と物理　66pages	February 27, 2009
COE Lecture Note Vol.18	Tim Hoffmann	Discrete Differential Geometry of Curves and Surfaces　75pages	April 21, 2009
COE Lecture Note Vol.19	Ichiro Suzuki	The Pattern Formation Problem for Autonomous Mobile Robots —Special Lecture in Functional Mathematics—　23pages	April 30, 2009
COE Lecture Note Vol.20	Yasuhide Fukumoto Yasunori Maekawa	Math-for-Industry Tutorial: Spectral theories of non-Hermitian operators and their application　184pages	June 19, 2009
COE Lecture Note Vol.21	Faculty of Mathematics, Kyushu University	Forum “Math-for-Industry” Casimir Force, Casimir Operators and the Riemann Hypothesis 95pages	November 9, 2009
COE Lecture Note Vol.22	Masakazu Suzuki Hoon Hong Hirokazu Anai Chee Yap Yousuke Sato Hiroshi Yoshida	The Joint Conference of ASCM 2009 and MACIS 2009: Asian Symposium on Computer Mathematics Mathematical Aspects of Computer and Information Sciences　436pages	December 14, 2009
COE Lecture Note Vol.23	荒川　恒男 金子　昌信	多重ゼータ値入門　111pages	February 15, 2010
COE Lecture Note Vol.24	Fulton B.Gonzalez	Notes on Integral Geometry and Harmonic Analysis　125pages	March 12, 2010
COE Lecture Note Vol.25	Wayne Rossman	Discrete Constant Mean Curvature Surfaces via Conserved Quantities 130pages	May 31, 2010
COE Lecture Note Vol.26	Mihai Ciucu	Perfect Matchings and Applications　66pages	July 2, 2010

シリーズ既刊

Issue	Author／Editor	Title	Published
COE Lecture Note Vol.27	九州大学大学院 数理学研究院	Forum "Math-for-Industry" and Study Group Workshop Information security, visualization, and inverse problems, on the basis of optimization techniques　100pages	October 21, 2010
COE Lecture Note Vol.28	ANDREAS LANGER	MODULAR FORMS, ELLIPTIC AND MODULAR CURVES LECTURES AT KYUSHU UNIVERSITY 2010　62pages	November 26, 2010
COE Lecture Note Vol.29	木田　雅成 原田　昌晃 横山　俊一	Magma で広がる数学の世界　157pages	December 27, 2010
COE Lecture Note Vol.30	原　　　隆 松井　　卓 廣島　文生	Mathematical Quantum Field Theory and Renormalization Theory 201pages	January 31, 2011
COE Lecture Note Vol.31	若山　正人 福本　康秀 高木　　剛 山本　昌宏	Study Group Workshop 2010 Lecture & Report　128pages	February 8, 2011
COE Lecture Note Vol.32	Institute of Mathematics for Industry, Kyushu University	Forum "Math-for-Industry" 2011 "TSUNAMI-Mathematical Modelling" Using Mathematics for Natural Disaster Prediction, Recovery and Provision for the Future　90pages	September 30, 2011
COE Lecture Note Vol.33	若山　正人 福本　康秀 高木　　剛 山本　昌宏	Study Group Workshop 2011 Lecture & Report　140pages	October 27, 2011
COE Lecture Note Vol.34	Adrian Muntean Vladimír Chalupecký	Homogenization Method and Multiscale Modeling　72pages	October 28, 2011
COE Lecture Note Vol.35	横山　俊一 夫　　紀恵 林　　卓也	計算機代数システムの進展　210pages	November 30, 2011
COE Lecture Note Vol.36	Michal Beneš Masato Kimura Shigetoshi Yazaki	Proceedings of Czech-Japanese Seminar in Applied Mathematics 2010 107pages	January 27, 2012
COE Lecture Note Vol.37	若山　正人 高木　　剛 Kirill Morozov 平岡　裕章 木村　正人 白井　朋之 西井　龍映 栄　伸一郎 穴井　宏和 福本　康秀	平成23年度 数学・数理科学と諸科学・産業との連携研究ワークショップ　拡がっていく数学　～期待される"見えない力"～ 154pages	February 20, 2012

シリーズ既刊

Issue	Author／Editor	Title	Published
COE Lecture Note Vol.38	Fumio Hiroshima Itaru Sasaki Herbert Spohn Akito Suzuki	Enhanced Binding in Quantum Field Theory　204pages	March 12, 2012
COE Lecture Note Vol.39	Institute of Mathematics for Industry, Kyushu University	Multiscale Mathematics; Hierarchy of collective phenomena and interrelations between hierarchical structures　180pages	March 13, 2012
COE Lecture Note Vol.40	井ノ口順一 太田　泰広 筧　　三郎 梶原　健司 松浦　　望	離散可積分系・離散微分幾何チュートリアル2012　152pages	March 15, 2012
COE Lecture Note Vol.41	Institute of Mathematics for Industry, Kyushu University	Forum “Math-for-Industry” 2012 “Information Recovery and Discovery”　91pages	October 22, 2012
COE Lecture Note Vol.42	佐伯　　修 若山　正人 山本　昌宏	Study Group Workshop 2012 Abstract, Lecture & Report　178pages	November 19, 2012
COE Lecture Note Vol.43	Institute of Mathematics for Industry, Kyushu University	Combinatorics and Numerical Analysis Joint Workshop　103pages	December 27, 2012
COE Lecture Note Vol.44	萩原　　学	モダン符号理論からポストモダン符号理論への展望　107pages	January 30, 2013
COE Lecture Note Vol.45	金山　　寛	Joint Research Workshop of Institute of Mathematics for Industry (IMI), Kyushu University “Propagation of Ultra-large-scale Computation by the Domain-decomposition-method for Industrial Problems (PUCDIP 2012)” 121pages	February 19, 2013
COE Lecture Note Vol.46	西井　龍映 栄　伸一郎 岡田　勘三 落合　啓之 小磯　深幸 斎藤　新悟 白井　朋之	科学・技術の研究課題への数学アプローチ ―数学モデリングの基礎と展開―　325pages	February 28, 2013
COE Lecture Note Vol.47	SOO TECK LEE	BRANCHING RULES AND BRANCHING ALGEBRAS FOR THE COMPLEX CLASSICAL GROUPS　40pages	March 8, 2013
COE Lecture Note Vol.48	溝口　佳寛 脇　　隼人 平坂　　貢 谷口　哲至 島袋　　修	博多ワークショップ「組み合わせとその応用」　124pages	March 28, 2013

シリーズ既刊

Issue	Author／Editor	Title	Published
COE Lecture Note Vol.49	照井　　章 小原　功任 濱田　龍義 横山　俊一 穴井　宏和 横田　博史	マス・フォア・インダストリ研究所　共同利用研究集会 II 数式処理研究と産学連携の新たな発展　137pages	August 9, 2013
MI Lecture Note Vol.50	Ken Anjyo Hiroyuki Ochiai Yoshinori Dobashi Yoshihiro Mizoguchi Shizuo Kaji	Symposium MEIS2013: Mathematical Progress in Expressive Image Synthesis　154pages	October 21, 2013
MI Lecture Note Vol.51	Institute of Mathematics for Industry, Kyushu University	Forum "Math-for-Industry" 2013 "The Impact of Applications on Mathematics"　97pages	October 30, 2013
MI Lecture Note Vol.52	佐伯　　修 岡田　勘三 髙木　　剛 若山　正人 山本　昌宏	Study Group Workshop 2013 Abstract, Lecture ＆ Report 142pages	November 15, 2013
MI Lecture Note Vol.53	四方　義啓 櫻井　幸一 安田　貴徳 Xavier Dahan	平成25年度　九州大学マス・フォア・インダストリ研究所 共同利用研究集会　安全・安心社会基盤構築のための代数構造 ～サイバー社会の信頼性確保のための数理学～　158pages	December 26, 2013
MI Lecture Note Vol.54	Takashi Takiguchi Hiroshi Fujiwara	Inverse problems for practice, the present and the future　93pages	January 30, 2014
MI Lecture Note Vol.55	栄　伸一郎 溝口　佳寛 脇　　隼人 渋田　敬史	Study Group Workshop 2013 数学協働プログラム Lecture & Report 98pages	February 10, 2014
MI Lecture Note Vol.56	Yoshihiro Mizoguchi Hayato Waki Takafumi Shibuta Tetsuji Taniguchi Osamu Shimabukuro Makoto Tagami Hirotake Kurihara Shuya Chiba	Hakata Workshop 2014 ~ Discrete Mathematics and its Applications ~　141pages	March 28, 2014
MI Lecture Note Vol.57	Institute of Mathematics for Industry, Kyushu University	Forum "Math-for-Industry" 2014: "Applications + Practical Conceptualization + Mathematics = fruitful Innovation"　93pages	October 23, 2014
MI Lecture Note Vol.58	安生健一 落合啓之	Symposium MEIS2014: Mathematical Progress in Expressive Image Synthesis　135pages	November 12, 2014

シリーズ既刊

Issue	Author／Editor	Title	Published
MI Lecture Note Vol.59	西井　龍映 岡田　勘三 梶原　健司 髙木　　剛 若山　正人 脇　　隼人 山本　昌宏	Study Group Workshop 2014 数学協働プログラム Abstract, Lecture & Report　196pages	November 14, 2014
MI Lecture Note Vol.60	西浦　　博	平成26年度九州大学 IMI 共同利用研究・研究集会（I） 感染症数理モデルの実用化と産業及び政策での活用のための新たな展開　120pages	November 28, 2014
MI Lecture Note Vol.61	溝口　佳寛 Jacques Garrigue 萩原　　学 Reynald Affeldt	研究集会 高信頼な理論と実装のための定理証明および定理証明器 Theorem proving and provers for reliable theory and implementations (TPP2014)　138pages	February 26, 2015
MI Lecture Note Vol.62	白井　朋之	Workshop on “β-transformation and related topics”　59pages	March 10, 2015
MI Lecture Note Vol.63	白井　朋之	Workshop on “Probabilistic models with determinantal structure” 107pages	August 20, 2015
MI Lecture Note Vol.64	落合　啓之 土橋　宜典	Symposium MEIS2015: Mathematical Progress in Expressive Image Synthesis　124pages	September 18, 2015
MI Lecture Note Vol.65	Institute of Mathematics for Industry, Kyushu University	Forum “Math-for-Industry” 2015 “The Role and Importance of Mathematics in Innovation”　74pages	October 23, 2015
MI Lecture Note Vol.66	岡田　勘三 藤澤　克己 白井　朋之 若山　正人 脇　　隼人 Philip Broadbridge 山本　昌宏	Study Group Workshop 2015 Abstract, Lecture & Report 156pages	November 5, 2015
MI Lecture Note Vol.67	Institute of Mathematics for Industry, Kyushu University	IMI-La Trobe Joint Conference “Mathematics for Materials Science and Processing” 66pages	February 5, 2016
MI Lecture Note Vol.68	古庄　英和 小谷　久寿 新甫　洋史	結び目と Grothendieck-Teichmüller 群 116pages	February 22, 2016
MI Lecture Note Vol.69	土橋　宜典 鍛治　静雄	Symposium MEIS2016: Mathematical Progress in Expressive Image Synthesis　82pages	October 24, 2016
MI Lecture Note Vol.70	Institute of Mathematics for Industry, Kyushu University	Forum “Math-for-Industry” 2016 “Agriculture as a metaphor for creativity in all human endeavors” 98pages	November 2, 2016
MI Lecture Note Vol.71	小磯　深幸 二宮　嘉行 山本　昌宏	Study Group Workshop 2016 Abstract, Lecture & Report　143pages	November 21, 2016

シリーズ既刊

Issue	Author／Editor	Title	Published
MI Lecture Note Vol.72	新井　朝雄 小嶋　　泉 廣島　文生	Mathematical quantum field theory and related topics　133pages	January 27, 2017
MI Lecture Note Vol.73	穴田　啓晃 Kirill Morozov 須賀　祐治 奥村　伸也 櫻井　幸一	Secret Sharing for Dependability, Usability and Security of Network Storage and Its Mathematical Modeling　211pages	March 15, 2017
MI Lecture Note Vol.74	QUISPEL, G. Reinout W. BADER, Philipp MCLAREN, David I. TAGAMI, Daisuke	IMI-La Trobe Joint Conference Geometric Numerical Integration and its Applications　71pages	March 31, 2017
MI Lecture Note Vol.75	手塚　　集 田上　大助 山本　昌宏	Study Group Workshop 2017 Abstract, Lecture & Report　118pages	October 20, 2017
MI Lecture Note Vol.76	宇田川誠一	Tzitzéica 方程式の有限間隙解に付随した極小曲面の構成理論 ―Tzitzéica 方程式の楕円関数解を出発点として―　68pages	August 4, 2017
MI Lecture Note Vol.77	松谷　茂樹 佐伯　　修 中川　淳一 田上　大助 上坂　正晃 Pierluigi Cesana 濵田　裕康	平成29年度　九州大学マス・フォア・インダストリ研究所 共同利用研究集会（I） 結晶の界面，転位，構造の数理　148pages	December 20, 2017
MI Lecture Note Vol.78	瀧澤　重志 小林　和博 佐藤憲一郎 斎藤　　努 清水　正明 間瀬　正啓 藤澤　克樹 神山　直之	平成29年度　九州大学マス・フォア・インダストリ研究所 プロジェクト研究　研究集会（I） 防災・避難計画の数理モデルの高度化と社会実装へ向けて 136pages	February 26, 2018
MI Lecture Note Vol.79	神山　直之 畔上　秀幸	平成29年度　AIMaP チュートリアル 最適化理論の基礎と応用　96pages	February 28, 2018
MI Lecture Note Vol.80	Kirill Morozov Hiroaki Anada Yuji Suga	IMI Workshop of the Joint Research Projects Cryptographic Technologies for Securing Network Storage and Their Mathematical Modeling　116pages	March 30, 2018
MI Lecture Note Vol.81	Tsuyoshi Takagi Masato Wakayama Keisuke Tanaka Noboru Kunihiro Kazufumi Kimoto Yasuhiko Ikematsu	IMI Workshop of the Joint Research Projects International Symposium on Mathematics, Quantum Theory, and Cryptography　246pages	September 25, 2019
MI Lecture Note Vol.82	池森　俊文	令和２年度　AIMaP チュートリアル 新型コロナウイルス感染症にかかわる諸問題の数理 145pages	March 22, 2021

シリーズ既刊

Issue	Author／Editor	Title	Published
MI Lecture Note Vol.83	早川健太郎 軸丸　芳揮 横須賀洋平 可香谷　隆 林　　和希 堺　　雄亮	シェル理論・膜理論への微分幾何学からのアプローチと その建築曲面設計への応用　49pages	July 28, 2021
MI Lecture Note Vol.84	Taketoshi Kawabe Yoshihiro Mizoguchi Junichi Kako Masakazu Mukai Yuji Yasui	SICE-JSAE-AIMaP Tutorial Advanced Automotive Control and Mathematics　110pages	December 27, 2021

Institute of Mathematics for Industry
Kyushu University

九州大学マス・フォア・インダストリ研究所
九州大学大学院 数理学府
〒819-0395 福岡市西区元岡744　TEL 092-802-4402　FAX 092-802-4405
URL http://www.imi.kyushu-u.ac.jp/