Math-for-industry
Education & Research Hub

ISSN2188-1200

IMI Workshop of the Joint Research Projects

Secret Sharing for Dependability, Usability and Security of Network Storage and Its Mathematical Modeling

Editors: Hiroaki Anada, Kirill Morozov, Yuji Suga, Shinya Okumura, Kouichi Sakurai

九州大学マス·フォア·インダストリ研究所

MI Lecture Note Vol.73 : Kyushu University

IMI Workshop of the Joint Research Projects

Secret Sharing for Dependability, Usability and Security of Network Storage and Its Mathematical Modeling

Editors: Hiroaki Anada, Kirill Morozov, Yuji Suga, Shinya Okumura, Kouichi Sakurai

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 2014
Yasuhide Fukumoto
Director
Institute of Mathematics for Industry

Secret Sharing for Dependability, Usability and Security of Network Storage and Its Mathematical Modeling

MI Lecture Note Vol.73, Institute of Mathematics for Industry, Kyushu University
ISSN 2188-1200
Date of issue: March 15, 2017
Editors: Hiroaki Anada, Kirill Morozov, Yuji Suga, Shinya Okumura, Kouichi Sakurai
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 http://www.imi.kyushu-u.ac.jp/

Printed by
Kijima Printing, Inc.
Shirogane 2-9-6, Chuo-ku, Fukuoka, 810-0012, Japan
TEL +81-(0)92-531-7102 FAX +81-(0)92-524-4411

IMI Workshop of the Joint Research Projects

Secret Sharing for Dependability, Usability and Security of Network Storage and Its Mathematical Modeling

September 5th – 7th, 2016

Industry-University-Government Collaboration Innovation Plaza

3-8-34 Momochihama Sawara-ku Fukuoka 814-0001, Japan

Sponsored by

Institute of Mathematics for Industry (IMI),

Kyushu University

Organized by

Hiroaki Anada, Kirill Morozov, Yuji Suga,

Shinya Okumura and Kouichi Sakurai

Acknowledgements

Prof. Yvo Desmedt was partially supported by a *kakenhi* Grant-in-Aid for Scientific Research (C) 15K00186 from Japan Society for the Promotion of Science concerning his visit and participation to this workshop.

Prof. Jon-Lark Kim was partially supported by a *kakenhi* Grant-in-Aid for Scientific Research (C) 15K00186 from Japan Society for the Promotion of Science concerning his visit and participation to this workshop.

One of organizers, Kirill Morozov, was partially supported by a *kakenhi* Grant-in-Aid for Scientific Research (C) 15K00186 from Japan Society for the Promotion of Science concerning his visit and participation to this workshop.

Prof. Patrick P. C. LEE was partially supported by a *kakenhi* Grant-in-Aid for Scientific Research (C) 15H02711 from Japan Society for the Promotion of Science concerning his visit and participation to this workshop.

One of organizers, Hiroaki Anada, was partially supported by a *kakenhi* Grant-in-Aid for Scientific Research (C) 15H02711 from Japan Society for the Promotion of Science concerning his visit and participation to this workshop.

Preface

Confidentiality and reliability had been two basic requirements for outsourced storage including the clouds, and these had been pursued using encryption and error-correction, respectively and independently. In the recent years, the secret sharing technology has been increasing getting attention as an alternative method for achieving both these requirements at once. At present, there even exist commercial-level systems released by vendor companies. However, theoretical and practical aspects such as communication cost vs. computational cost and computational security vs. information-theoretic security still need to be rigorously evaluated with respect to their impact on dependability, usability and security.

The purpose of this workshop was to discuss those aspects. There were held 15 distinguished lectures as well as one panel discussion gathering more than 40 attendees. The goal of these lecture notes is to raise awareness in the topics and results discussed at this workshop, among both researchers in mathematics, and developers in cloud computing and information security.

Hiroaki Anada, Representative of the Organizers

Table 1. List of attendees.

Takuro Abe	Tsuyoshi Kanamaru	Satoshi Obana	Clyde Vassallo
Koichiro Akiyama	Ryo Kikuchi	Atsuya Otani	Rui Xu
Toshinori Araki	Jon-Lark Kim	Rocki H. Ozaki	Naoto Yanai
Chi Cheng	Yuichi Komano	Partha Sarathi Roy	Gen Yoneda
Yvo Desmedt	Hirotake Kurihara	Masao Sakai	Takayuki Nozaki
Hiroshi Doi	Nari Lee	Yoshihisa Sato	Tsuyoshi Takagi
Duong Hoang Dung	Patrick P. C. Lee	Masaaki Shirase	Kirill Morozov
Goichiro Hanaoka	Toshiaki Maeno	Arkadii Slinko	Shinya Okumura
Jo Hyungrok	Shinichi Matsumoto	Keisuke Tanaka	Kouichi Sakurai
Keiichi Iwamura	Yasuyuki Murakami	Kouya Tochikubo	Yuji Suga
Mosarrat Jahan	Koji Nuida	Tadaaki Tsuchiya	Hiroaki Anada
Shizuo Kaji	Yasuhide Numata	Danilo V. Vargas	–

Photograph 1. Group photo in front of the venue.

Photograph 2. Photos of the workshop lecturers.

Photograph 3. More snapshots.

KYUSHU UNIVERSITY

Institute of Mathematics for Industry Kyushu University

IMI Joint Research Project in 2016

ISIT

Secret Sharing for Dependability, Usability and Security of Network Storage and Its Mathematical Modeling

Date:

September 5(Mon) - 7(Wed), 2016

http://www.imi.kyushu-u.ac.jp/eng/events/view/1070

Invited Speakers:

Yvo Desmedt, *The University of Texas at Dallas*
"Applications of Secret Sharing: Beyond Storage Service"

Arkadii Slinko, *The University of Auckland*
"Classification of Ideal Secret Sharing Schemes with Weighted Access Structures"

Patrick P. C. Lee, *The Chinese University of Hong Kong*
"Unifying Reliability, Security, and Deduplication in Cloud Storage"

Rui Xu, *KDDI R&D Laboratories, Inc.*
Ryo Kikuchi, *NTT CORPORATION*
Toshinori Araki, *NEC Corporation*
Yuichi Komano, *TOSHIBA CORPORATION*
Keiichi Iwamura, *Tokyo University of Science*
Jon-Lark Kim, *Sogang University*
Hiroshi Doi, *Institute of Information Security*
Satoshi Obana, *Hosei University*
Rocki H. Ozaki, *Real Technology Inc.*
Partha Sarathi Roy, *Kyushu University*
Chi Cheng, *Kyushu University*
Yuji Suga, *Internet Initiative Japan Inc.*

Speakers' Affiliations

UT DALLAS · THE UNIVERSITY OF AUCKLAND · The Chinese University of Hong Kong · Sogang University · KDDI R&D LABS · NTT · TOSHIBA · Tokyo University of Science · INSTITUTE of INFORMATION SECURITY · HOSEI · IIJ Internet Initiative Japan

---

Venue: AirIMaQ, Momochi :
Seminar Room, 2F, Industry-University-Government Collaboration Innovation Plaza

3-8-34 Momochihama Sawara-ku Fukuoka 814-0001, JAPAN
http://airimaq.kyushu-u.ac.jp/en/airimaq/access.php

■Organizing Committee ▸
Hiroaki Anada (University of Nagasaki)
Kirill Morozov (Tokyo Institute of Technology)
Yuji Suga (Internet Initiative Japan Inc.)
Shinya Okumura (Institute of Systems, Information Technologies and Nanotechnologies)
Kouichi Sakurai (Institute of Systems, Information Technologies and Nanotechnologies)

■Sponsored by ▸ Institute of Mathematics for Industry, Kyushu University
■Co-sponsored by ▸ Institute of Systems, Information Technologies and Nanotechnologies
■Registration fee ▸ Free

Contact : anada@sun.ac.jp

(For general inquiries) Institute of Mathematics for Industry, Kyushu University
TEL: 092-802-4402 E-mail: kyodo_riyou@imi.kyushu-u.ac.jp

Program

Sep 5 (Monday) Afternoon Session

13:50-14:00 (Opening)

[1] 14:00-14:40 Yvo Desmedt, The University of Texas at Dallas

"Applications of Secret Sharing: Beyond Storage Service"

[2] 15:00-15:30 Satoshi Obana, Hosei University

"Cheating Detectable Secret Sharing Scheme Supporting Finite Fields of Characteristic Two"

[3] 15:30-16:00 Hiroshi Doi, Institute of Information Security

"Fast ({1,k},n) Hierarchical Secret Sharing Schemes"

[4] 16:20-16:50 Ryo Kikuchi, NTT CORPORATION

"SHSS: "Super High-speed (or, Sugoku Hayai) Secret Sharing" library for object storage systems"

Sep 6 (Tuesday) Morning Session

[5] 9:40-10:10 Rocki H. Ozaki, Real Technology Inc.

"Unequal Secret Sharing Scheme - a proposal"

[6] 10:10-10:40 Keiichi Iwamura, Tokyo University of Science

"Integration of IoT and big data security by using asymmetric secret sharing scheme"

[7] 11:00-11:30 Jon-Lark Kim , Sogang University

"Secret sharing schemes based on additive codes"

[8] 11:30-12:10 Arkadii Slinko, The University of Auckland

"Classification of Ideal Secret Sharing Schemes with Weighted Access Structures"

Sep 6 (Tuesday) Afternoon Session

[9] 14:30-15:00 Yuichi Komano, TOSHIBA CORPORATION

"Toward Highly Secure Metering Data Management in the Smart Grid"

[10] 15:20-15:50 Chi Cheng, Kyushu University

"Homomorphic authentication schemes for network coding"

[11] 15:50-16:30 Patrick P. C. Lee, The Chinese University of Hong Kong

"Unifying Reliability, Security, and Deduplication in Cloud Storage"

[12] 16:30-17:15 (Panel Discussion) Panelists: Yvo Desmedt, Jon-Lark Kim, Patrick P. C. Lee, Rocki H. Ozaki, Satoshi Obana, Moderator: Kirill Morozov

"Secret Sharing in Real-Life Distributed Systems: Perspectives and Challenges"

Sep 7 (Wednesday) Morning Session

[13] 9:40-10:10 Partha Sarathi Roy, Kyushu University

"On The Robustness of Secret Sharing Schemes"

[14] 10:10-10:40 Rui Xu, KDDI R&D Laboratories, Inc.

"Secret Sharing against Cheaters"

[15] 10:40-11:10 Toshinori Araki, NEC Corporation

"High-Throughput Secure Computation using bit slicing"

[16] 11:30-12:00 Yuji SUGA, Internet Initiative Japan Inc.

"XOR-based (2, 2^m) threshold schemes"

12:00-12:10 (Closing)

Table of Contents

Applications of Secret Sharing: Beyond Storage Service

Yvo Desmedt

The University of Texas at Dallas and University College London
Yvo.Desmedt@utdallas.edu

Secure Multiparty Computation is likely the most known application of secret sharing beyond storage. However, this is only one application in which one computes with shares. Other examples that will be explained are Function Secret Sharing and Threshold Cryptography, a technique used in e-voting. Moreover, recently, secret sharing has been used to improve Chaum code (internet) voting approach. A proper application of these techniques can protect against, e.g., state-sponsored malware.

Besides its applications in secure distributed computations, secret sharing is the foundation of private and reliable communication, which we briefly explain.

We also systematically analyze the concepts used in the context of secret sharing. We explain why the concept of Access Structure is a Trust concept and explain its potential applications in such areas as Access Control, Critical Infrastructures and Disaster Prevention.

We discuss how two of these techniques may have prevented the Fukushima disaster.

Applications of Secret Sharing: Beyond Storage

Yvo Desmedt

Dept. of Computer Science
Univ. of Texas at Dallas
USA

and

University College London
UK

September 5, 2016

Some of the ideas presented here have not been published yet.

Overview

Part I. The building blocks of secret sharing

Part II. Access Structures as Trust Structures

II.1. Color Based Access Structures

II.2. Application: Critical Infrastructures

II.3. Application: Communication Systems

II.4. Application: Access Control

II.5. Application: Reliable Computation and Untrusted Hardware/Software

Part III. Secret Sharing as building block

III.1. Communication Systems

III.2. Computations: Secure Multiparty Computation

III.3. Computations: Threshold Cryptography

III.4. Applications of Threshold Cryptography

Part IV. What many missed

IV.1. Solution: using humans

IV.2. Internet-voting as an application

IV.3. Solution: using physics

Part V. Lessons & Challenges

Part I. The building blocks of secret sharing

A typical way to describe secret sharing is to state:

A secret sharing scheme contains two algorithms:

1. one which creates shares of a secret $k \in \mathcal{K}$ for the n parties in $\mathcal{P}$, so that
2. any $\mathcal{B} \in \Gamma_{\mathcal{P}}$ can regenerate the secret using the second algorithm, however any $\mathcal{B} \notin \Gamma_{\mathcal{P}}$ can not. (In the perfect case, $\mathcal{B} \notin \Gamma_{\mathcal{P}}$ has no knowledge of the secret).
   One calls $\Lambda_{\mathcal{P}} \subset 2^{\mathcal{P}}$ an adversary structure on $\mathcal{P}$ if its complement, i.e., $\Lambda^c_{\mathcal{P}} = 2^{\mathcal{P}} \setminus \Lambda_{\mathcal{P}}$ is a monotone access structure.

This definition only make sense when the adversary is passive.

Generalizing the approach used by Dolev-Dwork-Waarts-Yung, we should define:

- An adversary structure attacking privacy, $\Lambda_{\mathcal{P},\text{privacy}}$
- An adversary structure attacking reliability, i.e., in which subset of parties may deviate from the protocol, $\Lambda_{\mathcal{P},\text{reliability}}$

The case usually studied in the active case is the one in which

$$\Lambda_{\mathcal{P},\text{privacy}} = \Lambda_{\mathcal{P},\text{reliability}}.$$

However, as we will see soon, such a restriction dramatically reduces the applications!

So, we distinguish between the main building blocks:

I.1. The concepts of adversary and access structures,

I.2. The SS and VSS schemes that realize this.

Notes:

- SS and VSS schemes rely on combinatorics, algebra, etc.
- the concept of secret sharing predates Blakley and Shamir (Shamir cites Liu's 1968 book). We will call old SS schemes mechanical ones.
- There are secondary building blocks, such as:
  - Homomorphic secret sharing
  - Proactive secret sharing
  - Redistribution of shares

Part II. Access Structures as Trust Structures

A lot of research has been done by the computer security community related to trust (see e.g., at ESORICS, Beth-Borcherding-Klein 1994, Maurer 1996 and several papers by Jøsang). However, they are quite different from the trust expressed by Access Structures.

Are probabilities better?

- They are often difficult to measure,
- When probabilities are independent, then when assuming the threshold t is big enough, the remaining probability will vanish exponentially fast.
- Conditional probabilities seem a better measure.

II.1. Color Based Access Structures

Definition 1. An access structure $\Gamma_{\mathcal{P}}$ is called color based if there exist a function f from $\mathcal{P}$ to $\mathcal{C}$, called the set of colors, such that, for some constant t:

$$\Gamma_{\mathcal{P}} = \{\mathcal{B} \mid |f(\mathcal{B})| \geq t\}.$$

Why are these access structures important?

As we will see, they can be used to describe trust failures that are "correlated." So, they might be the solution to deal with conditional probability.

We will also see that in many circumstances, a color based access structure models modern problems we have to deal with in (information) security, well.

II.2. Application: Critical Infrastructures

The idea of color based access structures was first introduced informally when:

- modeling the computers used in a PKI (Public Key Infrastructure) system (see Burmester-Desmedt, Comm. ACM 2004).
  Today we have very few operating systems and CAs (Certifying Authorities) and RCAs (Root Certifying Authorities) use computers. Often a weakness in one platform can be exploited to attack many computers running the same platform. To model this dependency, computers running the same platform were given the same color.
- The topic was generalized to model "failures" in critical infrastructures (Burmester-Desmedt-Wang, IASTED 2003).

The importance of this model has been made clear with, e.g.,

- the Hengchun earthquake that on Tuesday December 26, 2006 which caused several underwater internet cables to fail in Asia,

We see the same technology being used in circumstances that have the same vulnerability.

II.3. Application: Communication Systems

Classical results

This goes back to World War I, after the cable ship Telconia lifted from the bed of the North Sea the German overseas telegraph cables:

If an adversary can destroy t nodes, then $t + 1$ vertex disjoint paths are needed and sufficient to communicate from sender (node A) to receiver (node B). If any two non-destroyed nodes want to communicate, it is necessary and sufficient that the directed graph must be strongly $t + 1$ connected.

Illustration: node disjoint paths: a closed station

Dolev-Dwork-Waarts-Yung generalized the Byzantine general problem to also include private communication. They used secret sharing to achieve private and reliable (secure) communication when the adversary can take over t nodes in a point-to-point communication network.

In practice routers are used in communication systems. Few companies are making routers. A formal study of the color adversary setting in the context of communication systems was done by Desmedt-Wang-Burmester (ISAAC 2005). The first author wanted to take this correlated vulnerability into account. It turned out that:

- Addressing the general case (i.e., model adversary in nodes by using a General Adversary Structure) was conceptual easier.

- Kumar-Goundan-Srinathan-Rangan (2002) also looked at the problem in the case interaction is used.
- Deciding whether a network in which nodes are colored satisfies the color based access structure for a given t is co-**NP** complete (Desmedt-Wang-Burmester, CRITIS 2006).

The model was also used to design networks that are reliable (no privacy) when untrusted links are used, that have correlated failures (Wang-Desmedt 2011, IPL).

Note: for outdated survey articles on this huge topic see: IEEE Information Theory Workshop (2005, Japan) and BT Technology Journal (2006).

The importance of color based access structures has become clear in the following contexts:

- Cisco Faces Challenges As Chinese Media Urge Switching To Domestic Products For National Security Reasons In Wake Of NSA Surveillance Leaks
  http://www.ibtimes.com/cisco-faces-challenges-chinese-media-urge-switching-domestic-products-national-security-reasons-wake
- BT's use of Huawei's equipment:


hocked' over Huawei contract with BT - FT.com – Konqueror

ookmarks Tools Settings Window Help

Politics & Policy

Home UK World Companies Markets Global Economy Lex Commer

Business Economy UK Companies Politics & Policy UK Small Companies

Welcome to FT.com, the global source of business new analysis. Register now to receive 8 free articles per m

ast updated: June 6, 2013 8:04 pm

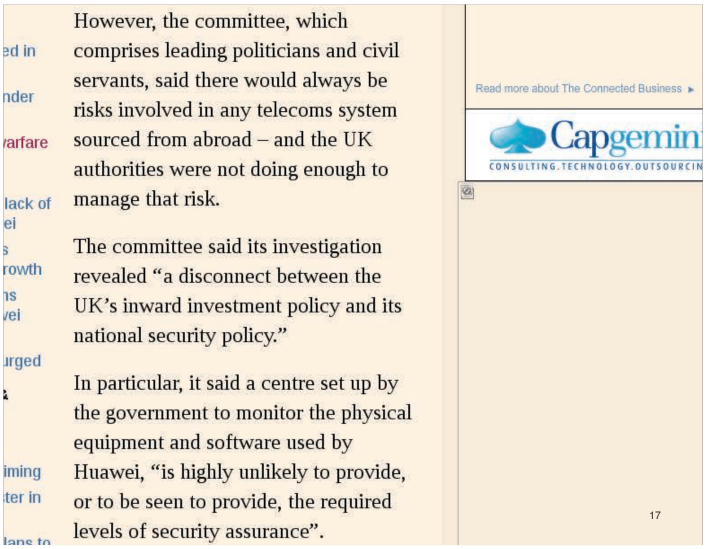
Share Clip

UK security committee 'shocked' over Huawei contract with BT

y James Blitz and Daniel Thomas

A parliamentary committee has attacked the British

EDITOR'S CHOIC

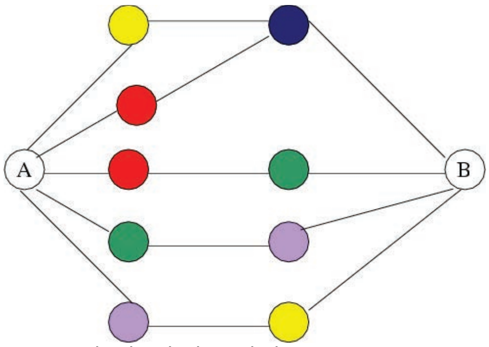

However, the committee, which comprises leading politicians and civil servants, said there would always be risks involved in any telecoms system sourced from abroad – and the UK authorities were not doing enough to manage that risk.

The committee said its investigation revealed "a disconnect between the UK's inward investment policy and its national security policy."

In particular, it said a centre set up by the government to monitor the physical equipment and software used by Huawei, "is highly unlikely to provide, or to be seen to provide, the required levels of security assurance".

Read more about The Connected Business ▸

Note that the results are very different than in the case of t-connected networks (when removing nodes).



Colors: not as in classical graph theory.

II.4. Application: Access Control

Classical access control gives access to subjects, which are usually single parties. So, elements of the access structure correspond to singletons (cardinality 1).
Desmedt-Shaghaghi (submitted) briefly considered using general access structures to specify what subsets of subjects have access to a certain object.
Access structures as we now know may not be the best way to describe access control. Indeed, there are many circumstances that need another approach, such as:

- a subject A is allowed access, after another subject B authorized it.
- a subject A is allowed access, after an entry has been made in a log file, readable by subject B.

- a subject A can only access an object when B is accessing the same object at the same time.

So, the definition of access structure should be generalized to have:

- A mix of unordered and ordered sets.
- To allow to specify the role of each subject in its order.

II.5. Application: Reliable Computation and Untrusted Hardware/Software

Assume we are not interested in privacy. Question: how can we achieve reliable computation.
Normal Model: replicate the computation and then use a majority vote.
General Access Structure: it is easy to see that we need that for any two sets $\mathcal{A} \in \Lambda_{\mathcal{P}}$ and $\mathcal{B} \in \Lambda_{\mathcal{P}}$ that $\mathcal{A} \cup \mathcal{B} \neq \mathcal{P}$.
One replicates the computation and one then "votes" in such a way, that if the same result is produced by each of the computers that belong to some set $\mathcal{A} \in \Gamma_{\mathcal{P}}$, then this result is considered correct.
When using color access structures, this might allow one to protect against state sponsored malware.
Important comment: see later, i.e., Part IV.

Part III. Secret Sharing as Building Block

In Part II we focused on how access structures can be used to describe trust and lack of it.

We now consider SS and VSS schemes as building blocks.

III.1. Communication Systems

When one desires privacy, secret sharing is the building block for PSMT (Private and "Secure" Message Transmission).
In the case of a threshold adversary, the non-interactive case corresponds with error-correcting codes. The interactive case also uses secret shares, but is much more complex (see e.g., Kurosawa-Suzuki 2008).
As stated before, there are many variants of these scenarios, e.g., using directed hypergraphs instead of point-to-point networks.
Implementations:

1. Erotokritou-Desmedt (unpublished) tried to implement the 1993 non-interactive solution of Dolev-Dwork-Waarts-Yung. The amazing problem we encountered is that:
   - the 1993 internet technology would had allowed a 1993

implementation,

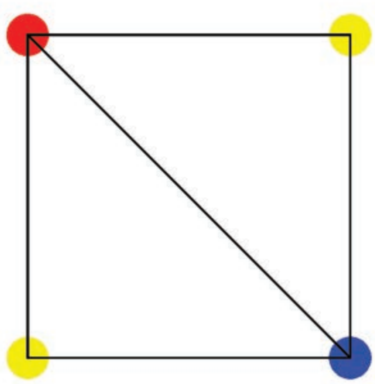
- the current internet technology no longer allows to implement this. Reasons:
  - to guarantee $3t+1$ vertex disjoint paths, we must specify the path a data packet has to follow. Today any packet that uses the standard TCP/IP option to specify the path is dropped by modern routers!!
  - companies want to keep the layout of the network private, which causes another difficulty!

2. Desmedt-Cheney (unpublished) designed and implemented a Thunderbird extension using mail servers, as gmail, hotmail, yahoo, etc. For example, gmail and hotmail are considered as intermediary nodes between the sender and receiver. So, we consider Google and Microsoft as potential adversaries, not working together.

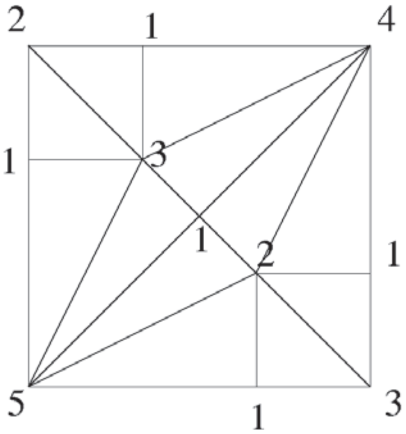
III.2. Computations: Secure Multiparty Computation

Secure Multiparty Computation (MPC) started as a theoretical problem. Today, many implementations have been programed and progress has been made in making it more practical, in both a conditional as unconditional setting. The May 30 - June 3, 2016 workshop on MPC in Aarhus clearly showed the progress in the area.
Note: a not so well known result is the link between color based access structures and MPC, which was made by Desmedt-Pieprzyk-Steinfeld-Wang at Crypto 2007 (see also the 2012 paper in Journal of Cryptology).
Following from an earlier result by Franklin-Yung (1995) follows that a reliability problem involving color based access structures implies

privacy-only MPC over non-Abelian groups.
Some examples: $t=1$ and $n=3$



$t = 2$ and $n = 5$



III.3. Computations: Threshold Cryptography

Threshold Cryptography: much faster than secure multiparty computation! Usually exploits homomorphic properties.

Comments:

- Extending Shamir SS to deal with RSA (see Desmedt-Frankel, Siam Discr. Math. 1994) took two years.
- Often Shoup's scheme, which he called "Practical Threshold Signatures," is implemented, but as King (ACISP 2000 and Asiacrypt 2000) pointed out due to the use of $n!$, it is not so practical!

Recommendations:

- At the Eurocrypt 2014 Panel on Post-Snowden Cryptography Smart recommended one uses threshold cryptography with co-decryption (co-signature) units in different countries.

My recommendation: use software/hardware from different countries (color based adversary structures), e.g., from China (developing independent hardware and OS). (So far I know, Japan is not developing this).

- At Eurocrypt 2016 in his IACR Distinguish Lecture, Preneel recommended the use of Threshold Cryptography, but stated that there are few uses and few implementations of it!

III.4. Applications of Threshold Cryptography

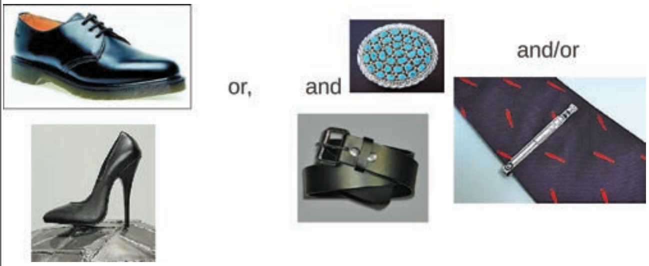
Just some example:

Threshold Things That Think (T^4)

Inspired by Things That Think:

> sensors and microcomputers in objects, in particular clothing e.g. in "sneakers, belt buckles, tie clasps, and wristwatches. These chips would communicate. They would for example allow a user to be identified when arriving in the lobby of an hotel, and the elevator will know which floor to take him to, and the door to his room will swing open as if by magic when he approaches."

Uses Threshold zero-knowledge. Store the shares as following:



Preneel's 2016 private comment:

> A PhD student of Preneel implemented T^4, but then when trying to convince companies to use this, they could not understand the concept of threshold or general adversary structure.

Part IV. What many missed

Today MPC is often promoted as a solution to state sponsored malware.

Now when using different cloud servers from different countries, it seems this problem is solved. However, the reliability community knows for decades that this is false!!.

Why? In reliable circuit design one teaches you that:

> The gates used for voting must be 100% reliable!

What does it mean in our context?

- When the servers you use are curious:

  The gates/computers to perform Lagrange interpolation must be 100% trustworthy. Means: you better build it yourself! (Yung recommendation at a panel at Intrust, Beijing.)

  However, Lagrange interpolation is too complex for many countries or corporations to build oneself.

- When the servers can be malicious (Byzantine):

  The gates/computers to perform a decoder of a Reed-Solomon code (e.g., Berlekamp-Massey or Berlekamp-Welch) must be 100% trustworthy. Means: you better build it yourself!

  However, these decoders are too complex for many countries or corporations to build oneself.

IV.1. Solution: using humans

One of our approaches (independent from Yung) uses a human brain.

Problems:

- Humans can not do Lagrange interpolation, moreover,
- they can not perform a Reed-Solomon decoder (e.g., Berlekamp-Massey or Berlekamp-Welch).

Our solution: we design special secret sharing schemes, which allows humans to recover the secret.

How realistic?

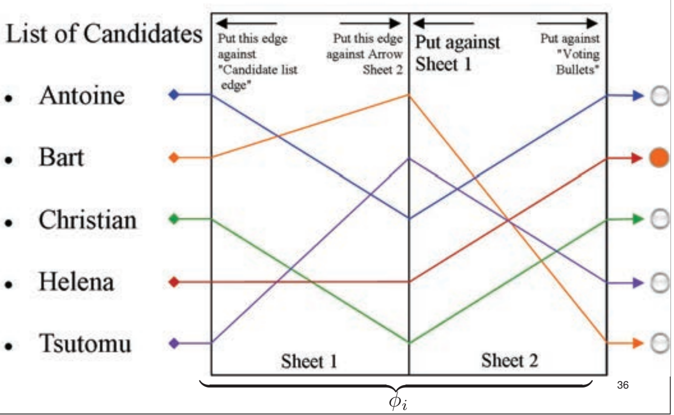
- we tested share reconstruction in the passive adversary case and got 99% accuracy.

- for the active adversary case we use secret sharing schemes in which we can deal with errors using a variant of repeat codes. (Not tested.)

Erotokritou-Desmedt developed (SCN 2012) a solution in the context of communication with untrusted routers (PSMT). When combining this with the Desmedt-Pieprzyk-Steinfeld (SCN 2012) work, it is easy to achieve a theoretical solution for MPC in the active adversary case.

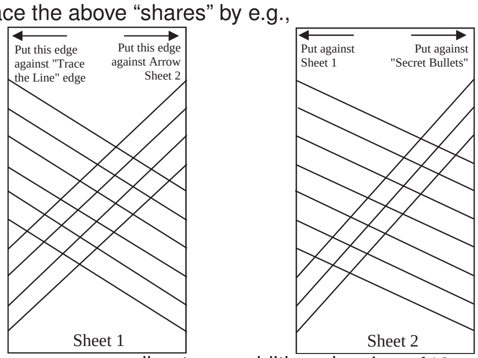
IV.2. INTERNET-VOTING AS AN APPLICATION

A user friendly approach: (multi-seat, not “code-voting”, $t = 1$)



In the single-seat election (mix friendly), we use code-voting ($t = 1$)

We regard the Abelian group $Z_{10}(+)$ as a subgroup of S_{10} and replace the above “shares” by e.g.,



These corresponding to an addition plus 4 $\mathrm{mod} 10$ and plus 3 $\mathrm{mod} 10$ respectively. We assume there are 10 candidates.

The secret sharing aspect was presented at SCN 2012 (Erotokritou-Desmedt).

The voting aspect, with a new unconditionally secure MIX server was presented at VoteID 2015 (Desmedt-Erotokritou).

IV.3. Solution: using physics

At ICITS 2016, De Prisco-D'Arco-Desmedt presented a solution to use visual cryptography to achieve MPC.

Problems with using Visual Cryptography:

- We want to avoid that all computations need to use visual cryptography (too slow)!
- But then, we seem to have an incompatibility of two secret sharing schemes!
- Shares are generated by a computer!!!

UCL ©Yvo Desmedt UT DALLAS 39

IV.4. Lessons & Challenges

We should start to use the concepts of secret sharing, in particular the one of Adversary Structure, in very different circumstances. Just two examples inspired by the Fukushima nuclear accident:

- **In the context of communication:** As required by regulations, two different phone providers were used at the plant to communicate with headquarters.
  Unfortunately, both phone providers were mobile ones and mobile phones usually fail in the case of earthquakes. So, communication between the plant and Tokyo Headquarters was impossible, resulting in not open safety valves, which lead to the explosion.
  Lesson: when using color based adversary structures one can color technology that has the same vulnerability with the same

UCL ©Yvo Desmedt UT DALLAS 40

color, showing the lack of proper redundancy.

- **In the context of the emergency cooling:** they had the same design, being at same location, they had the same vulnerabilities: 4 failures. The use of color based adversary structure might have helped.

Challenges: We have many, in particular:

- Lack of understanding by (non-)experts, e.g., in discussions with 2 full professors at University College London, both working in Information Security, it became clear that they have no trust in Secret Sharing (summer of 2016).
- Bringing the ideas towards deployment.

UCL ©Yvo Desmedt UT DALLAS 41

Cheating Detectable Secret Sharing Scheme Supporting Finite Fields of Characteristic Two

Satoshi OBANA
Hosei University
obana@hosei.ac.jp
(joint work with Hidetaka HOSHINO)

Cheating detectable secret sharing is a secret sharing scheme with an extra property to detect forged shares in reconstructing a secret. Such a property is indispensable when we have to store shares in possibly malicious environment (e.g., cloud storage.) Because of its importance in the real world applications, cheating detectable secret sharing is actively studied so far. When we can assume that cheaters do not know the secret, Ogata *et al.* derived the following lower bound on the size of shares [4]: $|\mathcal{V}_i| = (|\mathcal{S}|-1)/\epsilon+1$ where $\mathcal{V}_i$, $\mathcal{S}$, and ϵ denote a set of share of user P_i, a set of a secret, and successful cheating probability of cheaters, respectively. Cabello *et al.* presented an almost optimum cheating detectable scheme in which the size of share $|\mathcal{V}_i|$ satisfies $|\mathcal{V}_i| = |\mathcal{S}|/\epsilon$, only one bit larger than the lower bound [1]. However, the scheme is secure only when the secret is an element of a finite field with odd characteristic, that is, the scheme is insecure when the secret is a element of $\mathbb{F}_{2^N}$, a finite field of characteristic two. Though there are several schemes which are secure when the secret is an element of $\mathbb{F}_{2^N}$ [3, 2], few schemes are known to be optimum with respect to the size of share. Since $\mathbb{F}_{2^N}$ is the most natural representation of data in computer systems, an efficient scheme supporting $\mathbb{F}_{2^N}$ is highly desired.

In this talk, we present cheating detectable secret sharing schemes which are secure even if the secret is an element of $\mathbb{F}_{2^N}$. When the secret is uniformly distributed and $|\mathcal{S}| \geq \epsilon^{-2}$ holds, the size of share of the proposed schemes are almost optimum in the seance that the bit length of the share meets the lower bound with equality. Moreover, the proposed schemes are applicable to any any linear secret schemes. We also present a negative result of cheating detectable secret sharing scheme for supporting $\mathbb{F}_{2^N}$ when $\epsilon = 1/|\mathcal{S}|$ holds.

References

[1] S. Cabello, C. Padró, G. Sáez, "Secret Sharing Schemes with Detection of Cheaters for a General Access Structure", Designs, Codes and Cryptography, 25, pp. 175-188, 2002.
[2] S. Obana and K. Tsuchida, "Cheating Detectable Secret Sharing Schemes Supporting an Arbitrary Finite Field", Advances in Information and Computer Security, Lecture Notes in Computer Science vol. 8639, Springer Verlag, pp 88-97, 2014.
[3] W. Ogata, T. Araki, "Cheating Detectable Secret Sharing Schemes for Random Bit Strings", IEICE TRANS. FUNDAMENTALS, VOL.E96-A, NO.11, NOVEMBER 2013.
[4] W. Ogata, K. Kurosawa and D. R. Stinson, "Optimum Secret Sharing Scheme Secure against Cheating", SIAM Journal on Discrete Mathematics, vol. 20, no. 1, pp. 79-95, 2006.

Cheating Detectable Secret Sharing Schemes Supporting Finite Fields of Characteristic Two

IMI Workshop: Secret Sharing for Dependability, Usability and Security of Network Storage and Its Mathematical Modeling

Sep 5—7, 2016

Satoshi OBANA
Hosei University, Japan
(Joint work with Hidetaka HOSHINO)

Overview of this talk

- Models of Secret Sharing against Cheating
- Methodology for Constructing Cheating Detectable Secret Sharing Schemes
- Constructions of Cheating Detectable k-out-of-n Threshold SSs
  - Capable of detecting cheating in the presence of k-1 cheaters who possibly submit forged shares
  - Secure even when a secret is an element of $\mathbb{F}_{2^N}$
  - Optimal with respect to the size of share
- A negative result…

2

Several Models of SS against Cheating

- Cheater Identifiable (CISS)
  - Reconstruction algorithm identifies cheaters who submit forged shares
- Cheating Detectable SS (CDSS: this talk)
  - Reconstruction algorithm just detects the presence of cheaters
  - CDV model: Assume powerful cheaters who somehow know the value of the secret
  - OKS model (this talk): Only deal with *natural* cheaters who do not know the secret in forging their shares

3

Model of CDSS (1)

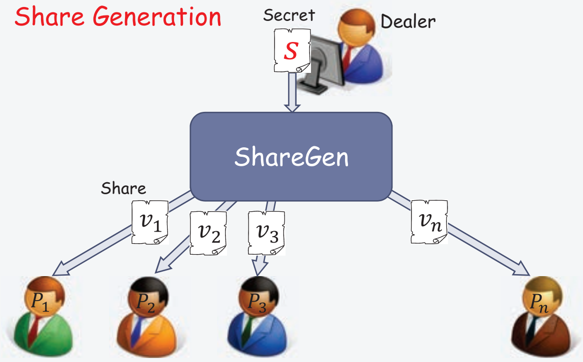
Two Types of Participants

- Dealer D
  - Dealer is honest (i.e., do not cheat)
  - Participate in the protocol only at share generation
- Users $P_1, P_2, \ldots, P_n$
  - Each user P_i obtains a share v_i from D
  - At most $k-1$ users are malicious
  - Malicious users open their shares each other and at least one of them submits forged share v'_i $(\neq v_i)$ in reconstructing a secret to make honest users reconstruct forged secret $s'(\neq s)$

4

Model of CDSS (2)

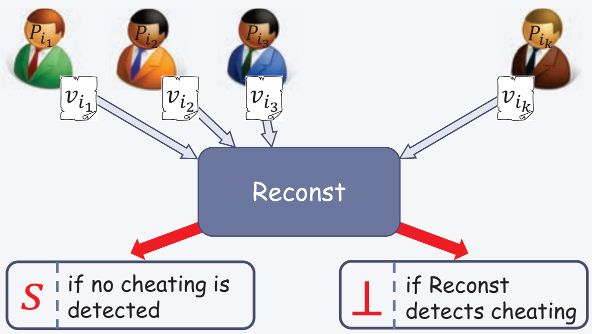
Share Generation



5

Model of CDSS (3)

Secret Reconstruction



6

Definition of Secure CDSS

Cheaters submitting forged share succeed in cheating if

- Reconst fails to detect cheating
- The value s' output by Reconst is different from what was input to ShareGen

Definition

A (k, n) threshold secret sharing scheme is called (k, n, ϵ)-secure if no $k - 1$ or less cheaters succeed in cheating with probability better than ϵ

7

A Methodology for Constructing (k, n, ϵ)-secure scheme (in the OKS model)

- Protocol Design Phase
  - Choose a fixed verification function A
- ShareGen
  1. Compute shares $v_{s,1}, \dots, v_{s,n}$ for a secret s using Shamir's (k, n) threshold scheme
  2. Compute shares $v_{a,1}, \dots, v_{a,n}$ for $A(s)$ using Shamir's (k, n) threshold scheme
  3. Output $v_i = (v_{s,i}, v_{a,i})$ as the share for user P_i
- Reconst
  1. Reconstruct $\hat{s}$ and $\hat{a}$ from $v_{s,*}$ and $v_{a,*}$, respectively
  2. Output $\hat{s}$ if $\hat{a} = A(\hat{s})$ holds, otherwise output $\perp$

8

Security of CDSS with verification func. A

Suppose that the secret is uniformly distributed. Then CDSS constructed based on such methodology is proven to be (k, n, ϵ)-secure where

$$\epsilon = \max_{\delta,\Delta} \frac{|\{s \mid A(s+\delta) = A(s) + \Delta\}|}{|S|}$$

Our Goal: To find GOOD verification function with desired properties

9

Desired Properties of Verification Func.

- Must be non-linear (otherwise, $\epsilon = 1$...)
- The degree of polynomial representation of $A(s+\delta) - A(s)$ is low since

$$\epsilon = \max_{\delta,\Delta} \frac{|\{s \mid A(s+\delta) = A(s) + \Delta\}|}{|S|}$$

- Share size of resulting scheme is small (as small as the following lower bound)

$$|V_i| \geq \frac{|S|-1}{\epsilon} + 1$$

- Applicable to a secret of a finite field of characterisic two (i.e., $\mathbb{F}_{2^N}$) since the most natural representation of data in computer systems is bit string

10

Known (k, n, ϵ)-secure schemes (OKS model)

	Verification Function $A(s)$	ϵ	Size of Shares $\lvert V_i \rvert$	Supported Mathematical Structures
Ogata-Kurosawa Eurocrypt '98	N/A different methodology		$\lvert V_i\rvert = \frac{\lvert S\rvert - 1}{\epsilon} + 1$ meet lower bound	Parameters are very much limited
Cabello-Padro-Saez DCC (2002)	$A(s) = s^2$	$\frac{1}{\lvert S\rvert}$	$\lvert V_i\rvert = \frac{\lvert S\rvert}{\epsilon}$ almost optimum	Arbitrary Finite Fields except for $\mathbb{F}_{2^N}$
Araki-Ogata IEICE Trans. Fund. (2013)	$A(s) = s^3$	$\frac{2}{\lvert S\rvert}$	$\lvert V_i\rvert = \frac{2\lvert S\rvert}{\epsilon}$	Finite Fields of Charasteristic 2 (i.e., $\mathbb{F}_{2^N}$)
Araki-Ogata IEICE Trans. Fund. (2012)	$A(s_1, \ldots, s_{N+1}) = s_N \cdot s_{N+1}^{N+1} + \sum s_i \cdot s_{N+1}^i$	$\frac{\log\lvert S\rvert}{\lvert S\rvert^{\frac{1}{\log\lvert S\rvert}}}$	$\lvert V_i\rvert = \frac{\lvert S\rvert \log\lvert S\rvert}{\epsilon}$	Arbitrary Finite Fields

11

Why CPS02 is insecure when $s \in \mathbb{F}_{2^N}$

- The share v_i of CPS02: $v_i = (v_{s,i}, v_{a,i})$ where
  - $v_{s,i}$: share of the secret s
  - $v_{a,i}$: share of the check value $A(s) = s^2$
- Cheaters can choose δ_s and δ_a arbitrarily such that
  - The secret reconstructed from shares $= s + \delta_s$
  - The check value reconstructed from shares $= A(s) + \delta_a$
- Cheaters win if $A(s+\delta_s) = A(s) + \delta_a$ holds, that is, if $(s+\delta_s)^2 = s^2 + \delta_a$ holds

$$\Updownarrow$$

$$2\delta_s \cdot s + \delta_s^2 = \delta_a$$

$$\Updownarrow$$

$\delta_s^2 = \delta_a$ (if $s \in \mathbb{F}_{2^N}$)

Cheaters succeeds in cheating with probability 1 by choosing δ_s and δ_a such that $\delta_s^2 = \delta_a$

12

Our Contribution

Construct three (k, n, ϵ)-secure SSs with the following properties:

- The scheme deals with the secret of a finite field of characteristic two
- The size of share is close to the following lower bound

$$|V_i| \geq \frac{|S| - 1}{\epsilon} + 1$$

Construction 1

- Let a secret $\mathrm{s} = (s_1, s_2)$ be a element of $\mathbb{F}_{2^N}^2$
- Employ $A(s_1, s_2) = s_1 \cdot s_2$ $(A\colon \mathbb{F}_{2^N}^2 \to \mathbb{F}_{2^N})$ as a verification function
- Properties of Construction 1:
  - $|S| = 2^{2N}$
  - $\epsilon = 1/2^N$
  - $|V_i| = 2^{3N} = |S|/\epsilon$

When $|S| = \epsilon^{-2}$ holds, Construction 1 is almost optimum with respect to the size of share

Construction 2 (Generalization)

- Let a secret $\mathrm{s} = (s_1, s_2)$ be a element of $\mathbb{F}_{2^N}^2$
- Employ $A(s_1, s_2) = \phi(s_1 \cdot s_2)$ $(A\colon \mathbb{F}_{2^N}^2 \to \mathbb{F}_{2^M})$ as a verification function ($\phi\colon \mathbb{F}_{2^N} \to \mathbb{F}_{2^M}$: linear function with $N \leq M$)
- Properties of Construction 2:
  - $|S| = 2^{2N}$
  - $\epsilon = 1/2^M$
  - $|V_i| = 2^{2N+M} = |S|/\epsilon$

When $|S| \geq \epsilon^{-2}$ holds, Construction 1 is almost optimum with respect to the size of share

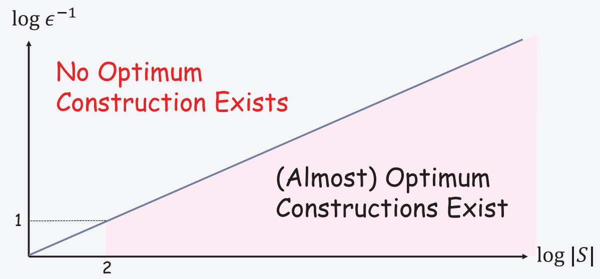
Construction 3 (Another Generalization)

- Let a secret $s = (s_1, s_2, \dots, s_{2\ell})$ be a element of $\mathbb{F}_{2^N}^{2\ell}$
- Employ $A(s_1, \dots, s_{2\ell}) = \sum_{i=1}^{\ell} s_{2i-1} \cdot s_{2i}$ $(A: \mathbb{F}_{2^N}^{2\ell} \to \mathbb{F}_{2^N})$ as a verification function
- Properties of Construction 3:
  - $|S| = 2^{2\ell N}$
  - $\epsilon = 1/2^N$
  - $|V_i| = 2^{(2\ell+1)N} = |S|/\epsilon$

Construction 3 is not only almost optimum but also easier to implement & efficiently implementable

16

What we have obtained



Natural Question:
Does optimum construction exist even when $|S| < \epsilon^{-2}$?

17

A negative result when $|S| = \epsilon^{-1}$

- For all $2^{3 \cdot 2^3}$ functions $A: \mathbb{F}_{2^3} \to \mathbb{F}_{2^3}$, we have checked the security of CDSS when using A as a verification function
- If optimum construction exists, the successful cheating probability of resulting CDSS becomes 1/8

18

A negative result when $|S| = \epsilon^{-1}$ (cont'd)

- Interestingly, no function which gives optimum construction exists!!

ϵ	# of functions
1/8	0
2/8	688128
3/8	0
4/8	10838016
5/8	0
6/8	5046272
7/8	0
1	204800

Concluding Remarks and Open Problems

- In this talk, we have presented three (k, n, ϵ)-secure CDSSs with the following properties
  - The scheme deals with the secret of a finite field of characteristic two
  - When $|S| \geq \epsilon^{-2}$ holds, the size of share is close to the lower bound
- Open Problems: Construct (k, n, ϵ)-secure CDSS
  - with optimal share size even when $|S| < \epsilon^{-2}$

Thank you!!

Fast ({1,k},n) Hierarchical Secret Sharing Schemes

Hiroshi DOI (Joint work with Koji SHIMA)
Institute of Information Security
doi@iisec.ac.jp

Shamir[1] and Blakley[2] independently introduce the basic idea of a (k, n) threshold secret sharing scheme in 1979. Shamir also recognize the concept of a hierarchical scheme, and suggests accomplishing the scheme by giving the participants of the more capable levels a greater number of shares. Some of hierarchical secret sharing schemes are known in the way that the secret is shared among a group of participants that is partitioned into levels. We look at hierarchical secret sharing schemes (HSSS) in the purpose of the ease of deleting the secret after it is distributed, that is, the reliability of data deletion depends on the deletion of the shares of the indispensable participants, and focus on providing a fast method and practicality.

In this talk, we propose two $(\{1, k\}, n)$ hierarchical secret sharing schemes. The first scheme[6, 7] inherits Tassa's idea[3, 4] of using derivatives and Birkhoff interpolation. The second scheme[6, 8] inherits XOR-based secret sharing scheme proposed by Fujii et al.'s[5]. The former provides any $(\{1, k\}, n)$ HSSS in finite fields of characteristic 2. On the otherhand, the latter provides only $(\{1, 3\}, n)$ HSSS for a small number of indispensable participants.

We also report the evaluation result of the above two schemes on a PC with Intel Celeron G1820 2.70GHz and 3.6GB RAM. The $(\{1, 3\}, n)$ HSSS using Birkhoff interpolation can recover the secret in the processing of around 0.97Gbps. On the otherhand $(\{1, 3\}, n)$ HSSS using XOR operations can recover the secret in the processing of around 7.0Gbps.

References

[1] Shamir, How to share a secret, Commun. ACM Vol.22, Issue 11, pp.612-613,1979.
[2] Blakley, Safeguarding cryptographic keys, AFIPS, Vol.48, pp.313-317, 1979.
[3] Tassa, Hierarchical Threshold Secret Sharing, TCC 2004, LNCS 2951, pp.473-490, 2004.
[4] Tassa, Hierarchical Threshold Secret Sharing, Journal of Cryptology, Vol.20, No.2, pp.237-264, 2007.
[5] Fujii, Tada, Hosaka, Tochikubo, Kato, A Fast $(2, n)$-Threshold Scheme and Its Application, CSS 2005, pp.631-636, 2005. [in Japanese]
[6] Shima, Doi, A study on fast hierarchical secret sharing schemes, CSS2014, 2E2-4, pp.1327-1334, 2015. [in Japanese]
[7] Shima, Doi, A Study on $(\{1, k\}, n)$ hierarchical secret sharing schemes over finite fields of characteristic 2, IPSJ CSEC, 2016-CSEC-72(5), pp.1-7, 2016. [in Japanese]
[8] Shima, Doi, $(\{1, 3\}, n)$ hierarchical secret sharing scheme based on XOR operations for a small number of indispensable participants, AsiaJCIS 2016, pp.108-114, 2016.

INSTITUTE of INFORMATION SECURITY

Fast ({1,k},n) Hierarchical Secret Sharing Schemes

Hiroshi DOI (Joint work with Koji SHIMA)
Institute of Information Security

Secret Sharing for Dependability, Usability and Security of Network Storage and Its Mathematical Modeling
IMI Workshop of the Joint Research Projects, Sep. 5-7, 2016

INSTITUTE of INFORMATION SECURITY

Fast ({1,k},n) HSSS

This presentation is composed of the following three works.

[SD15] Shima, Doi, "A study on fast hierarchical secret sharing schemes," Computer Security Symposium CSS2014, 2E2-4, pp.1327-1334, 2015. [in Japanese]
[SD16a] Shima, Doi, "A Study on ({1,k},n) hierarchical secret sharing schemes over finite fields of characteristic 2," IPSJ CSEC, 2016-CSEC-72(5), pp.1-7, 2016. [in Japanese]
[SD16b] Shima, Doi, "({1,3},n) hierarchical secret sharing scheme based on XOR operations for a small number of indispensable participants," AsiaJCIS 2016, pp.108-114, 2016.

HSSS: Hierarchical Secret Sharing Scheme
SSS: Secret Sharing Scheme

INSTITUTE of INFORMATION SECURITY

Outline

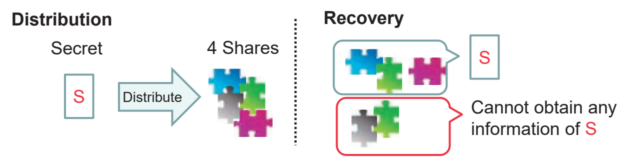
1. Background and Motivation
2. ({1,k},n) HSSS based on Birkhoff Interpolation
   i. Related Works
   ii. Our method
3. ({1,3},n) HSSS based on XOR operations for a small number of indispensable participants
   i. Related Works
   ii. Our method
4. Evaluation of Software Implementation
5. Conclusion

1. Background and Motivation

- Secret Sharing Scheme
- Hierarchical Secret Sharing Scheme
- Our Goal

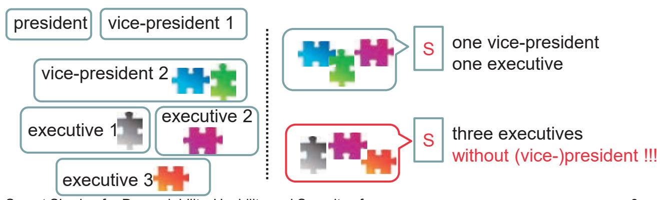
Secret Sharing Scheme(1/2)

- Methods for distributing and managing the secret information [S79,B79]
  - Prevention of both information theft and information loss
- (3,4) threshold secret sharing scheme



Secret Sharing Scheme(2/2)

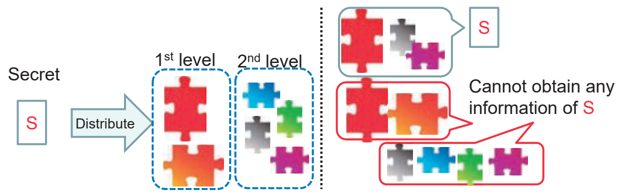
- Shamir also recognized the concept of a hierarchical SSS [S79]
  - The shares of (3,n) SSS are distributed
    - the company's president : three shares,
    - each vice-president : two shares,
    - each executive : one share



Hierarchical Secret Sharing Scheme

INSTITUTE of INFORMATION SECURITY

- The secret is shared among a group of participants that is partitioned into levels.
- $(\{1,3\},6)$ Hierarchical Secret Sharing Scheme [T04, T07]
  - Minimal number of 1st –level participants is 1
  - Minimal number of 2nd or higher–level participants is 3



Goal: Fast HSSS

INSTITUTE of INFORMATION SECURITY

- The method can be used in the purpose of the ease of deleting the secret after the secret is distributed
  - The deletion of the secret is guaranteed with the deletion of the indispensable (1st level) participants' shares
    - 1 or 2 indispensable participants will be practical for that purpose
- We focus on providing a fast method and practicality
  - Using fast operations
    - Operations in GF(2^l) / (Only) XOR
  - For fast construction, we restrict the method
    - e.g. For specific access structure (e.g. only ({1,3},n}))
    - e.g. The number of indispensable participants is 1 or 2

2. ({1,k},n) HSSS based on Birkhoff Interpolation

INSTITUTE of INFORMATION SECURITY

- Shamir's SSS and Lagrange Interpolation
- Tassa's HSSS and Birkhoff Interpolation
- Our Method

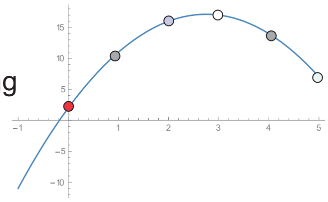
Shamir's SSS and Lagrange Interpolation

- Shamir proposed (k,n) secret sharing scheme[S79]
  - Using Lagrange Interpolation to recover the secret

Secret Sharing for Dependability, Usability and Security of Network Storage and Its Mathematical Modeling, 2016 10

Example (Lagrange Interpolation)

- Distribution:
  - $(x_1, f(x_1)), (x_2, f(x_2)), (x_3, f(x_3)), \ldots$
    - $f(x)$: polynomial with degree 2
- Recovery:
  - Calculate (a, b, c) using
    - $f(x_1) = ax_1^2 + bx_1 + c$
    - $f(x_2) = ax_2^2 + bx_2 + c$
    - $f(x_3) = ax_3^2 + bx_3 + c$



Secret Sharing for Dependability, Usability and Security of Network Storage and Its Mathematical Modeling, 2016 11

Lagrange Interpolation

- $f(x) = \sum_{i \in S} f(i) \prod_{j \in S \setminus \{i\}} \frac{x-j}{i-j}$
  - Degree of $f(x)$ is $k-1$
  - $|S| = k$, e.g. $S = \{x_1, x_2, x_3\}$
- Secret $f(0) = \sum_{i \in S} f(i) \prod_{j \in S \setminus \{i\}} \frac{-j}{i-j}$

[S79] Lagrange Interpolation in GF(p)

Secret Sharing for Dependability, Usability and Security of Network Storage and Its Mathematical Modeling, 2016 12

Tassa's HSSS and Birkhoff Interpolation

- Tassa proposed $(\{k_0, k_1, \cdots\}, n)$ hierarchical SSS [T04,T07]
  - Using Birkhoff Interpolation to recover the secret

$(\{k_0, k_1, \cdots\}, n)$ and $(\{1,3\}, n)$

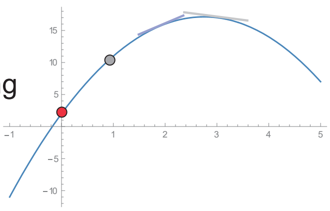
- Access Structure: $(\{k_0, k_1, \cdots\}, n)$
  - $0 < k_0 < k_1 < \cdots$
  - $\boldsymbol{\mathcal{U}} = \cup_{i=0}^{m} \boldsymbol{\mathcal{U}}_i, \boldsymbol{\mathcal{U}}_i \cap \boldsymbol{\mathcal{U}}_j = \emptyset \ (0 \le i < j \le m)$
  - $\Gamma = \{\boldsymbol{\mathcal{V}} \subset \boldsymbol{\mathcal{U}} : \left|\boldsymbol{\mathcal{V}} \cap \left(\cup_{j=0}^{i} \boldsymbol{\mathcal{U}}_j\right)\right| \ge k_i \ \forall i \in \{0,1,\ldots,m\}\}$

- $(\{1,3\}, n)$ HSSS
  - The minimal number of 1st levels participants is 1
  - The minimal number of 2nd or higher levels participants is 3
  - $(\{1,3\}, n)$ is sufficient for our goal

Example (Birkhoff Interpolation)

- Distribution:
  - $(x_1, f(x_1)), (x_2, f'(x_2)), (x_3, f'(x_3))$
  - $f(x)$: polynomial with degree 2

- Recovery:
  - Calculate (a, b, c) using
    - $f(x_1) = ax_1^2 + bx_1 + c$
    - $f'(x_2) = 2ax_2 + b$
    - $f'(x_3) = 2ax_3 + b$



Birkhoff Interpolation (1/2)

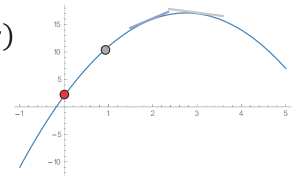
- $\mathcal{G} = \{g_0, g_1, g_2\},\ g_0(x) = 1, g_1(x) = x, g_2(x) = x^2$
- $D(E, X, \mathcal{G}) = \begin{vmatrix} g_0(x_1) & g_1(x_1) & g_2(x_1) \\ g_0{}'(x_2) & g_1{}'(x_2) & g_2{}'(x_2) \\ g_0{}'(x_3) & g_1{}'(x_3) & g_2{}'(x_3) \end{vmatrix} = \begin{vmatrix} 1 & x_1 & x_1^2 \\ 0 & 1 & 2x_2 \\ 0 & 1 & 2x_3 \end{vmatrix}$
- $D(E, X, \mathcal{G}_0) = \begin{vmatrix} f(x_1) & g_1(x_1) & g_2(x_1) \\ f'(x_2) & g_1{}'(x_2) & g_2{}'(x_2) \\ f'(x_3) & g_1{}'(x_3) & g_2{}'(x_3) \end{vmatrix}$
- $D(E, X, \mathcal{G}_1) = \begin{vmatrix} g_0(x_1) & f(x_1) & g_2(x_1) \\ g_0{}'(x_2) & f'(x_2) & g_2{}'(x_2) \\ g_0{}'(x_3) & f'(x_3) & g_2{}'(x_3) \end{vmatrix}$
- $D(E, X, \mathcal{G}_2) = \cdots$

$E = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 1 \end{bmatrix}$

$X = \{x_1, x_2, x_3\}$

Birkhoff Interpolation (2/2)

- $f(x) = \sum_{j=0}^{2} \frac{D(E,X,\mathcal{G}_j)}{D(E,X,\mathcal{G})} \cdot g_j(x)$
- Secret $f(0) = \frac{D(E,X,\mathcal{G}_0)}{D(E,X,\mathcal{G})}$

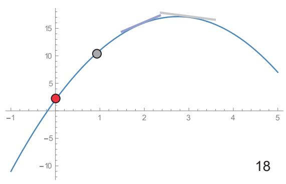


- Birkhoff Interpolation works if $D(E, X, \mathcal{G}) \neq 0$.

[T04,T07] Birkhoff Interpolation in GF(p)

Birkhoff Interpolation (Example)

- Example
  - $f(1) = 11, f'(2) = 3, f'(3) = -1$
  - $D(E, X, \mathcal{G}) = 2,$
  - $D(E, X, \mathcal{G}_0) = 4,\ D(E, X, \mathcal{G}_1) = 22,\ D(E, X, \mathcal{G}_2) = -4$
  - $f(x) = \sum_{j=0}^{2} \frac{D(E,X,\mathcal{G}_j)}{D(E,X,\mathcal{G})} \cdot g_j(x) = 2 + 11x - 2x^2$



Fast ({1,k},n) HSSS based on Birkhoff Interpolation

INSTITUTE of INFORMATION SECURITY

- In GF(2^l), constructing ({1,3},n) HSSS based on Birkhoff Interpolation is not straightforward

GF(p) where p is large

$$f(x_1) = ax_1^2 + bx_1 + c$$
$$f'(x_2) = 2ax_2 + b$$
$$f'(x_3) = 2ax_3 + b$$

Unknowns is (a, b, c)
Number of Eq. is 3
Solvable

GF(2^l)

$$f(x_1) = ax_1^2 + bx_1 + c$$
$$f'(x_2) = b$$
$$f'(x_3) = b$$

Unknowns is (a, b, c)
Number of Eq. is 2
Cannot Solve

Our Improvement for ({1,k},n)

INSTITUTE of INFORMATION SECURITY

- Using Polynomial with odd degree + constant(secret)
  - $f(x) = \sum_{i=1}^{k-1} a_i x^{2(i-1)+1} + s$
  - $f'(x) = \sum_{i=1}^{k-1} a_i x^{2(i-1)}$
- Birkhoff Interpolation for ({1,3},n)
  - $\mathcal{G} = \{g_0, g_1, g_2\},\ g_0(x) = 1, g_1(x) = x, g_2(x) = x^3$
  - $D(E, X, \mathcal{G}) = \begin{vmatrix} g_0(x_1) & g_1(x_1) & g_2(x_1) \\ g_0{}'(x_2) & g_1{}'(x_2) & g_2{}'(x_2) \\ g_0{}'(x_3) & g_1{}'(x_3) & g_2{}'(x_3) \end{vmatrix} = \begin{vmatrix} 1 & x_1 & x_1^3 \\ 0 & 1 & x_2^2 \\ 0 & 1 & x_3^2 \end{vmatrix}$
  - $D(E, X, \mathcal{G}_j) = \cdots$

Improvement for ({1,3},n)

INSTITUTE of INFORMATION SECURITY

- ({1,3},n) HSSS where the number of Indispensable participants is 1

$$s = f(0) = \frac{D(E, X, \mathcal{G}_0)}{D(E, X, \mathcal{G})}$$
$$= \frac{f(x_1)(x_2^2 + x_3^2) + f'(x_2)x_1(x_1^2 + x_3^2) + f'(x_3)x_1(x_1^2 + x_2^2)}{x_2^2 + x_3^2}$$

  - x_i^2 can be reused for fast computing

3. ({1,3},n) HSSS based on XOR operations for a small number of indispensable participants

- Fujii et al.'s $(2, n)$ threshold scheme [FTHTK05]
- Our Method
  - Case: one indispensable participant
  - Case: two indispensable participants

INSTITUTE of INFORMATION SECURITY

Fujii et al.'s (2,n) threshold scheme (1/3)

- The secret $s \in \{0,1\}^{d(n_p-1)}$ is equally divided into $n_p - 1$ blocks
- n_p is a prime number such that $n_p \geq n$
- $s_0 = \{0\}^d$

$s = \{0,\ 1\}^{d(n_p-1)}$

s_0	s_1	s_2	...	s_{n_p-2}	s_{n_p-1}

$\{0\}^d$

INSTITUTE of INFORMATION SECURITY

Fujii et al.'s (2,n) threshold scheme (2/3)

- The dealer
  - chooses $n_p - 1$ pieces of d-bit random number $r_0, \ldots, r_{n_p-2}$
  - distributes each share w_i to the participant P_i
- e.g. n=5

	$j=0$	$j=1$	$j=2$	$j=3$
w_0	r_0	$s_4 \oplus r_1$	$s_3 \oplus r_2$	$s_2 \oplus r_3$
w_1	$s_1 \oplus r_0$	r_1	$s_4 \oplus r_2$	$s_3 \oplus r_3$
w_2	$s_2 \oplus r_0$	$s_1 \oplus r_1$	r_2	$s_4 \oplus r_3$
w_3	$s_3 \oplus r_0$	$s_2 \oplus r_1$	$s_1 \oplus r_2$	r_3
w_4	$s_4 \oplus r_0$	$s_3 \oplus r_1$	$s_2 \oplus r_2$	$s_1 \oplus r_3$

Fujii et al.'s (2,n) threshold scheme (3/3)

- P_1 and P_3 cooperate to recover the secret using w_1, w_3
- From r_1 as a starting point, we obtain s_2 with r_1 and $s_2 \oplus r_1$
- From r_3 as a starting point, we obtain s_3,r_0,s_1,r_2,s_4
- Finally, we obtain $s = s_1 \parallel s_2 \parallel s_3 \parallel s_4$.

	$j = 0$	$j = 1$	$j = 2$	$j = 3$
w_1	$s_1 \oplus r_0$	r_1	$s_4 \oplus r_2$	$s_3 \oplus r_3$
w_3	$s_3 \oplus r_0$	$s_2 \oplus r_1$	$s_1 \oplus r_2$	r_3

Details of Our Method: Distribution

- We use intermediate shares $R_1, \cdots, R_4$
- Secret s is XORed in the shares of level 1 (i.e. w_0, w_1)
  - $R_1, \cdots, R_4$ are used as intermediate shares

	$j = 0$	$j = 1$	$j = 2$	$j = 3$
w_0	$r_0 \oplus s_1$	$R_4 \oplus r_1 \oplus s_2$	$R_3 \oplus r_2 \oplus s_3$	$R_2 \oplus r_3 \oplus s_4$
w_1	$R_1 \oplus r_0 \oplus s_1$	$r_1 \oplus s_2$	$R_4 \oplus r_2 \oplus s_3$	$R_3 \oplus r_3 \oplus s_4$
w_2	$R_2 \oplus r_0$	$R_1 \oplus r_1$	r_2	$R_4 \oplus r_3$
w_3	$R_3 \oplus r_0$	$R_2 \oplus r_1$	$R_1 \oplus r_2$	r_3
w_4	$R_4 \oplus r_0$	$R_3 \oplus r_1$	$R_2 \oplus r_2$	$R_1 \oplus r_3$

Details of Our Method: Recovery(1/2)

- Case: one indispensable participant
  - w_0, w_2, w_3 are used to recover the secret
    - $R_1, \cdots, R_4$ (and $r_1, \cdots, r_4$) are recovered using w_2, w_3
      - Fujii et al.'s (2,n) threshold scheme
    - $s_1, \cdots s_4$ are recovered using $R_1, \cdots, R_4, r_1, \cdots, r_4$

	$j = 0$	$j = 1$	$j = 2$	$j = 3$
w_0	$r_0 \oplus s_1$	$R_4 \oplus r_1 \oplus s_2$	$R_3 \oplus r_2 \oplus s_3$	$R_2 \oplus r_3 \oplus s_4$
w_2	$R_2 \oplus r_0$	$R_1 \oplus r_1$	r_2	$R_4 \oplus r_3$
w_3	$R_3 \oplus r_0$	$R_2 \oplus r_1$	$R_1 \oplus r_2$	r_3

Details of Our Method: Recovery(2/2)

- Case: two indispensable participants
  - w_0, w_1, w_2 are used to recover the secret
    - $R_1, \cdots, R_4$ are recovered using w_0, w_1
      - Fujii et al.'s (2,n) threshold scheme
    - $r_1, \cdots, r_4$ are recovered using $w_2, R_1, \cdots, R_4$
    - $s_1, \cdots s_4$ are recovered using $R_1, \cdots, R_4, r_1, \cdots, r_4$

	$j=0$	$j=1$	$j=2$	$j=3$
w_0	$r_0 \oplus s_1$	$R_4 \oplus r_1 \oplus s_2$	$R_3 \oplus r_2 \oplus s_3$	$R_2 \oplus r_3 \oplus s_4$
w_1	$R_1 \oplus r_0 \oplus s_1$	$r_1 \oplus s_2$	$R_4 \oplus r_2 \oplus s_3$	$R_3 \oplus r_3 \oplus s_4$
w_2	$R_2 \oplus r_0$	$R_1 \oplus r_1$	r_2	$R_4 \oplus r_3$

4. Evaluation of Software Implementation

- Environment
  - General purpose machine

CPU	Intel Celeron CPU G1820 @ 2.70GHz × 2 (2MB Cashe)
RAM	3.6GB
OS	CentOS 7 Linux 3.10.0-229.20.1.el7.x86_64
Programming Language	The C language
Compiler System	GCC 4.8.3 (-O3 –flto –DNDEBUG)

Details for ({1,k},n) HSSS based on Birkhoff Interpolation

- Operations in $GF(2^l)$

Irreducible Polynomials

$GF(2^8)$	$x^8+x^4+x^3+x+1$
$GF(2^{16})$	$x^{16}+x^{12}+x^3+x+1$
$GF(2^{32})$	$x^{32}+x^{22}+x^2+x+1$
$GF(2^{64})$	$x^{64}+x^4+x^3+x+1$
$GF(x^{128})$	$x^{128}+x^7+x^2+x+1$
$GF(2^{256})$	$x^{256}+x^{10}+x^5+x^2+1$

- Lookup Table in $GF(2^8)$
  - Precomputing $b_i \times b_j$ and b_i/b_j in $GF(2^8)$
  - Creating Table

    char mul[256][256], div[256][256]; // 128KB is needed.
  - $b_i \times b_j$ operation is implemented by referring mul[b_i][b_j]

Evaluation Result (Birkhoff Interpolation)

INSTITUTE of INFORMATION SECURITY

- ({1,3},n) HSSS based on Birkhoff Interpolation
  - Recovery (1 indispensable participant)

$GF(2^8)$ using Lookup Table	971.7

$GF(2^8)$	40.1
$GF(2^{16})$	20.0
$GF(2^{33})$	7.6
$GF(2^{64})$	3.7
$GF(2^{128})$	0.5
$GF(2^{256})$	0.2

Mbps

Details for ({1,3},n) HSSS based on XOR operations

INSTITUTE of INFORMATION SECURITY

- The secret $s \in \{0,1\}^{d(n_p-1)}$ is divided into $n_p - 1$ blocks
- d= 64 is used for the evaluation
  - We try out four values of d= 8, 16, 32, 64 and have found d= 64 is the fastest and roughly twice as fast as d= 32

Evaluation Result (XOR operations)

INSTITUTE of INFORMATION SECURITY

- ({1,3},n) HSSS based on XOR operations
  - Recovery (1 indispensable Participant)

({1,3},5)	8.37
({1,3},13)	7.65
({1,3},23)	7.38
({1,3},59)	7.65
({1,3},109)	7.32

Gbps (= 1000Mbps)

INSTITUTE of INFORMATION SECURITY

5. Conclusion

- We proposed two schemes
  - Both schemes are ideal and perfect (Omitted the proofs)
- ({1,k},n) HSSS based on Birkhoff Interpolation
  - k is selectable but effects the performance
  - The performance does not depend on n
  - 0.97Gbps (using Lookup Table)
- ({1,3},n) HSSS based on XOR operations
  - The performance depends on n
  - Only ({1,3},n) and small number of indispensable participants
    - The number of indispensable participants is one or two
  - around 7.0Gbps

INSTITUTE of INFORMATION SECURITY

References

[S79] Shamir, “How to share a secret,” Commun. ACM 22(11), pp. 612–613,1979.

[B79] Blakley, “Safeguarding cryptographic keys,” AFIPS, Vol.48, pp.313-317, 1979.

[T04] Tassa, “Hierarchical Threshold Secret Sharing,” TCC 2004, LNCS 2951, pp. 473–490, 2004.

[T07] Tassa, “Hierarchical Threshold Secret Sharing,” Journal of Cryptology 20 (2), pp. 237–264, 2007.

[FTHTK05] Fujii, Tada, Hosaka, Tochikubo, Kato, “A Fast (2,n)-Threshold Scheme and Its Application,” CSS 2005, pp. 631-636, 2005. [in Japanese]

[KKFT08a] Kurihara, Kiyomoto, Fukushima, Tanaka, “A Fast (3,n)-Threshold Secret Sharing Scheme Using Exclusive-OR Operations,” IEICE Trans. Fundamentals, E91-A(1), pp. 127-138, 2008.

[KKFT08b] Kurihara, Kiyomoto, Fukushima, Tanaka, “On a Fast (k,n)-Threshold Secret Sharing Scheme,” IEICE Trans. Fundamentals, E91-A(9), pp. 2365-2378, 2008.

[SD15] Shima, Doi, “A study on fast hierarchical secret sharing schemes,” Computer Security Symposium CSS2014, 2E2-4, pp.1327-1334, 2015. [in Japanese]

[SD16a] Shima, Doi, “A Study on ({1,k},n) hierarchical secret sharing schemes over finite fields of characteristic 2,” IPSJ CSEC, 2016-CSEC-72(5), pp.1-7, 2016. [in Japanese]

[SD16b] Shima, Doi, “({1,3},n) hierarchical secret sharing scheme based on XOR operations for a small number of indispensable participants,” AsiaJCIS 2016, pp.108-114, 2016.

SHSS: "Super High-speed (or, Sugoku Hayai) Secret Sharing" Library for Object Storage Systems

Ryo KIKUCHI (Joint work with Dai Ikarashi, Kota Tsuyuzaki, and Yuto Kawahara)
NTT Corporation
`kikuchi.ryo@lab.ntt.co.jp`

Recently, as a measure for the information security and the disaster recovery regarding on-line storage systems, the research of secret sharing technology has become quite active. On the other hand, in the research field of storages, erasure codes has been widely studied and quickly spread over practical storage systems recently.

In this work [1, 2], we point out that secret sharing has a merit from the aspect of information security as an upward compatible function of erasure codes when it is applied for object storage systems, which are becoming popular today, and propose an efficient secret sharing scheme suitable for object storage systems. Furthermore, we implemented a secret sharing library called SHSS (Super High-speed / Sugoku Hayai Secret Sharing), and report it's performance. It is about 50 times faster itself than that in the existing report for object storage systems [3], and combined with OpenStack Swift [4], it performs about 10 Gbps, which is as the same level as the standard erasure code library [5] without security.

References

[1] Dai Ikarashi, Kota Tsuyuzaki, and Yuto Kawahara. SHSS: "Super High-speed (or, Sugoku Hayai) Secret Sharing" Library for Object Storage Systems. In *IPJS SIG Technical Report* (in Japanese), 2015.

[2] Dai Ikarashi, Ryo Kikuchi, Koki Hamada, and Koji Chida. Fast Implementations of Extension Field Operations and Secret Sharing Scheme for 10GEther and Infiniband. In *SCIS* (in Japanese), 2014.

[3] J. K. Resch and J. S. Plank. AONT-RS: blending security and performance in dispersed systems. In *USENIX Conference on File and Storage Technologies*, 2011.

[4] OpenStack Swift Erasure Code Support. `http://docs.openstack.org/developer/swift/overview_erasure_code.html`.

[5] Jrasure. `http://github.com/tsuraan/Jerasure`.

SHSS: “Super High-Speed (or, Sugoku Hayai) Secret Sharing” Library for Object Storage Systems

Ryo Kikuchi (NTT Corporation)

Joint work with Dai Ikarashi, Kota Tsuyuzaki, and Yuto Kawahara

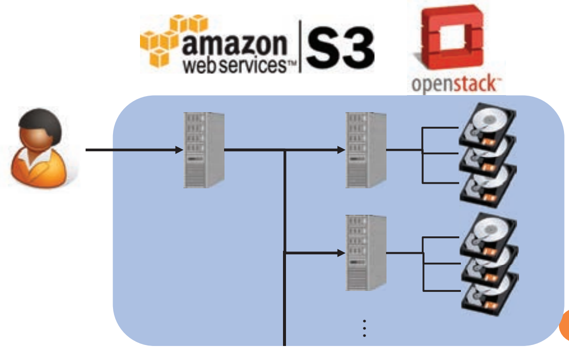
Summary

- We implement SHSS: Super High-speed (Sugoku Hayai) Secret Sharing Library
  - Sharing/reconstruction is 20Gbps on large (k, n)
- We embed SHSS into OpenStack Swift

2

Object storage

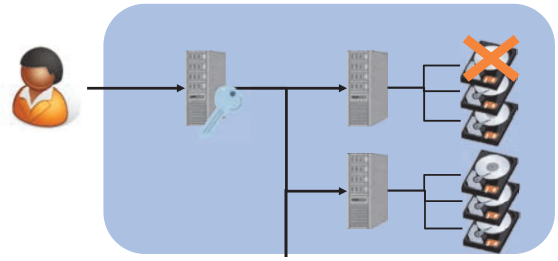
- Storage system for cloud network
  - Data is stored into several nodes and disks



3

Several features in objective storage

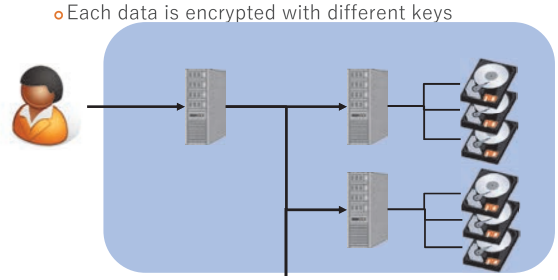
- Durability
  - Replication or Erasure code (a.k.a IDA)
- Secrecy
  - Proxy server encrypts all data as an optional feature



4

Advantage of applying (threshold) SS

- Durability, same as erasure code
  - Endure against $n-k$ disks failure
- Cost of secrecy
  - Provide secrecy w/o strict key management
    - Each data is encrypted with different keys



5

Is SS enough efficient?

☺ Enough, for small n and 1G network

- 4.7Gbps in both sharing and reconstruction on $(k,n)=(2,3)$ [IKHC14]

☹ Not enough, for object storage

- Each node is connected high-speed network
  - 10Gbps and more
- Large (k,n) such as $(6,9),(10,14)$ are used

We need more efficient SS library

[IKHC14] D. Ikarashi, R. Kikuchi, K. Hamada, and K. Chida.: Fast Implementations of Extension Field Operations and Secret Sharing Scheme for 10GEther and Infiniband. SCIS 2014.

6

Speeding up SS (on recent Intel CPU)

- Field operation
  - Multiplication
- Sharing/reconstruction algorithm

7

Field operation

- We employed $GF(2^{64})$
  - $a = \sum_{i<64} a_i x^i \in GF(2^{64})$
  - Irreducible polynomial: $f = x^{64} + x^4 + x^3 + x + 1$
- Multiplication
  - Input: $a, b \in GF(2^{64})$,
  - Output: $ab = \sum_{i<64} \sum_{i<64} a_i b_j x^{i+j} \bmod f$
    1. Multiplication (w/o reduction)
       - **MULT** (general case)
       - **BMULT** (specific case)
    2. Reduction
       - **RED**
       - **BRED**

8

MULT: Multiplication in general case

- Input: $a, b \in GF(2^{64})$
- Output: $\ell + x^{64}h \coloneqq ab = \sum_{i<64} \sum_{i<64} a_i b_j x^{i+j}$
  - $ab = \underbrace{\boxed{\quad \ell \quad}}_{64}\underbrace{\boxed{\quad h \quad}}_{63}$
- **MULT**:
  1. $\ell + x^{64}h = \sum_{i<64} \sum_{i<64} a_i b_j x^{i+j} = PCLMUL(a, b)$
     - PCLMUL is implemented as AES-NI
- Cost: 1 PCLMUL (2 clocks on Haswell CPU)

9

BMUL: Multiplication if b is monomial

- Input: $a = \sum_{i<64} a_i x^i$, $b = x^{b'}$
- Output: $\ell + x^{64} h \coloneqq ab = ax^{b'}$

- **BMUL:**
  1. $ax^{b'} = (a \ll b') + x^{64}(a \gg (64 - b'))$

- Cost: 2 SHIFT

10

RED: Reduction over GF(2^{64})

- Input: h (s.t. $ab = \ell + x^{64} h$)
  - polynomial of degree 62
- Output: $x^{64} h \bmod f$

- Intuition of algorithm
  - $x^{64} h = (x^4 + x^3 + x + 1)h = (h \ll 4) \oplus (h \ll 3) \oplus (h \ll 1) \oplus h$
    - Irreducible polynomial: $f = x^{64} + x^4 + x^3 + x + 1$
  - h is degree 62 so $h \ll 4$ and $h \ll 3$ overflow
    - This part can be computed as $(h \gg 60) \oplus (h \gg 61)$
  - $(x^4 + x^3 + x + 1) = (x^3 + 1)(x + 1)$

11

RED: Reduction over GF(2^{64})

- Input: h (s.t. $ab = \ell + x^{64} h$)
  - polynomial of degree 62
- Output: $x^{64} h \bmod f$

- **RED**:
  1. $h' \coloneqq h \oplus (h \gg 60) \oplus (h \gg 61)$
  2. $h'' \coloneqq h' \oplus (h' + h')$
  3. Output $h''' \coloneqq h'' \oplus (h'' \ll 3)$

- 3 SHIFT, 1 ADD, and 4 XOR

12

BRED: Reduction if b is 61bit or smaller

- Input: h (s.t. $ab = \ell + x^{64}h$)
  - polynomial with degree **59**

    $(h \gg 60) \oplus (h \gg 61) = 0$
- Output: $x^{64}h \bmod f$

- **BRED**:
  1. $h' \coloneqq h \oplus (h + h)$
  2. Output $h'' \coloneqq h' \oplus (h' \ll 3)$

- 1 SHIFT, 1 ADD, and 2 XOR

13

Throughput (on Haswell CPU)

faster

- **BMUL**+**BRED**: 2.25
  - if b is monomial and smaller than 61bit
- **MUL**+**BRED**: 3.25
  - if b is smaller than 61bit
- **BMUL**+**RED**: 3.75
  - if b is monomial
- **MUL**+**RED**: 4.75
  - Otherwise

14

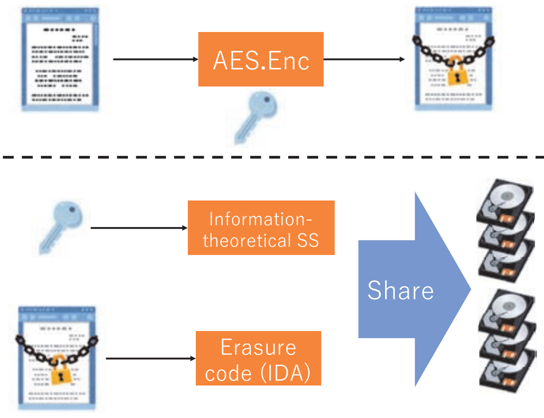
Speeding up SS

- Field operation
  - Multiplication

- Sharing/reconstruction algorithm

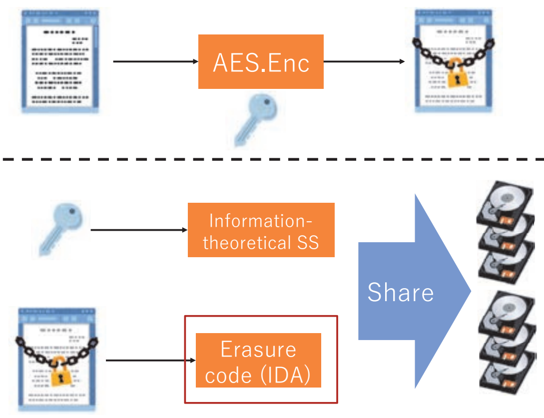
15

(A variant of) Krawczyk scheme



16

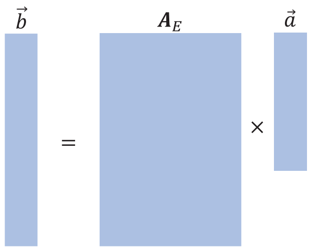
(A variant of) Krawczyk scheme



17

Erasure code

- Input: (Encrypted) data $\vec{a} = \begin{pmatrix} a_0 \\ \vdots \\ a_{k-1} \end{pmatrix}$
- Output: $\vec{b} = \begin{pmatrix} b_0 \\ \vdots \\ b_{n-1} \end{pmatrix}$
- Share: $\vec{b} = \boldsymbol{A}_E \vec{a}$
- Reconstruction:
  - explain later



18

Principle to choose $\boldsymbol{A}_E$

- Any (k, n) is acceptable
- Reduce multiplication as possible
  - 0 or 1 element is preferable
- Prefer **BMUL+BRED** rather than **MULT+RED**

19

Share (erasure code)

0 or 1

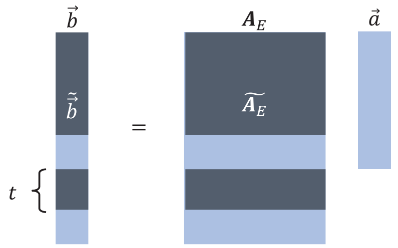
$$\begin{pmatrix} b_0 \\ \vdots \\ b_{k-1} \\ \vdots \\ b_{n-1} \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & & & \ddots & \\ 0 & 0 & 0 & \cdots & 1 \\ 1 & 1 & 1 & \cdots & 1 \\ 1 & x & x^2 & \cdots & x^{k-1} \\ 1 & x^2 & x^4 & \cdots & x^{2(k-1)} \\ \vdots & & & \ddots & \\ 1 & x^{m-1} & x^{2(m-1)} & \cdots & x^{(m-1)(k-1)} \end{pmatrix} \begin{pmatrix} a_0 \\ \vdots \\ a_{k-1} \end{pmatrix} \quad \begin{array}{l} \}\, k \\ \}\, n-k \end{array}$$

BMUL + BRED

20

Reconstruction (erasure code) 1/2

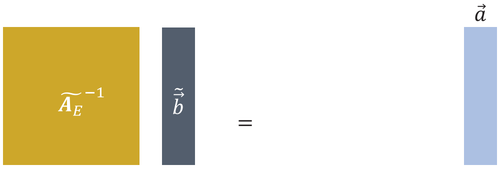
- Reconstruction is trivial if $b_0, \ldots, b_{k-1}$ are available
  - $b_0, \ldots, b_{k-1}$ are the input itself
- If not, a natural way:



21

Reconstruction (erasure code) 1/2

- Decode is trivial if $b_0, \dots, b_{k-1}$ are available
  - $b_0, \dots, b_{k-1}$ are the data itself
- If not, a normal way:

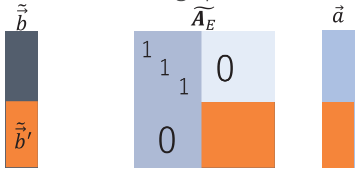


- It costs $t(k\mathbf{MUL} + \mathbf{RED})$ + "$k \times k$ matrix inversion"

22

Reconstruction (erasure code) 2/2

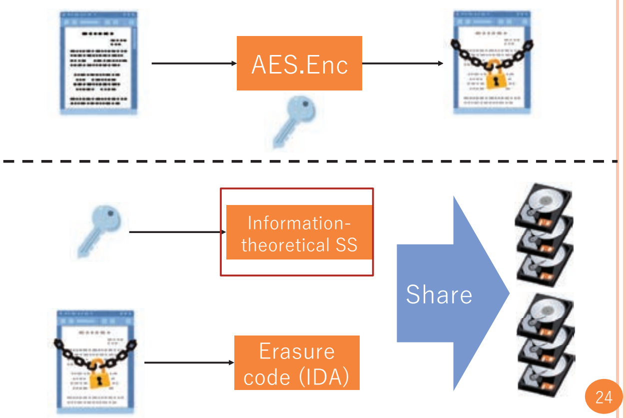
- Our approach: Eliminating "plaintext" first



- Elimination can be computed by **BMUL+BRED**
  - $b_j' = b_j - \sum x^{p_i} b_i$ for $j \geq k$ and $i < k$
- Total cost: $t((k-t)\mathbf{BMUL} + t\mathbf{MUL} + \mathbf{BRED} + \mathbf{RED})$ + "$t \times t$ matrix inversion"
  - faster than $t(k\mathbf{MUL} + \mathbf{RED})$ + "$k \times k$ matrix inversion"
- Further optimization has been applied

23

(A variant of) Krawczyk scheme



24

Information theoretical SS

- Input: key and randomness $\vec{a} = \begin{pmatrix} r_0 \\ \vdots \\ r_{k-2} \\ key \end{pmatrix}$
- Output: $\vec{b} = \begin{pmatrix} b_0 \\ \vdots \\ b_{n-1} \end{pmatrix}$
- Share: $\vec{b} = \boldsymbol{A}_I \vec{a}$

25

Share (information theoretical SS)

$$\begin{pmatrix} b_0 \\ \vdots \\ b_{k-1} \\ \vdots \\ b_{n-1} \end{pmatrix} = \begin{pmatrix} \overbrace{\boxed{\begin{matrix} 1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 1 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 1 & \cdots & 0 & 0 \\ \vdots & & & \ddots & & \\ 0 & 0 & 0 & \cdots & 1 & 0 \\ 1 & 1 & 1 & \cdots & 1 & 1 \end{matrix}}}^{\text{0 or 1}} \\ \underbrace{\boxed{\begin{matrix} 1 & x & x^2 & \cdots & x^{k-2} & x^{k-1} \\ 1 & x^2 & x^4 & \cdots & x^{2(k-2)} & x^{2(k-1)} \\ \vdots & & & \ddots & & \\ 1 & x^m & x^{2m} & \cdots & x^{m(k-2)} & x^{m(k-1)} \end{matrix}}}_{\text{BMUL + BRED}} \end{pmatrix} \begin{pmatrix} r_0 \\ \vdots \\ r_{k-2} \\ key \end{pmatrix}$$

26

Reconstruction (information theoretical SS)

- key can be computed by linear equation
  - Costs k**MUL+RED**

27

Experiment

- Parameters
  - $(k, n) = (6,9), (10,14), (11,18), (20,24)$
  - Achieve eleven nines (99,999999999%) durability
    - Same as amazon S3
    - MTTDL (Mean Time To Data Loss): 10 million years
    - Estimated by Markov model [XMS+03, GPW10]

[XMS+03] Q. Xin et al. Reliability Mechanisms for Very Large Storage Systems, MSST, 2003
[GPW10] K. M. Greenan, J. S. Plank, and J. J. Wylie. Mean time to meaningless: MTTDL, Markov models, and storage system reliability, Hot Storage, 2010.

Experiment

- Software
  - OS: Ubuntu 14.04.1 Server
  - Language: C++
  - Compiler: gcc 4.7.3
- Data properties
  - Random 1MB objects
  - The object is from/to main memory
  - $t = 1$, e.g., decode from $b_0, \ldots, b_{k-2}, b_k$

Comparison as a library

- SHSS achieves about 20Gbps

	(6,9)	(10,14)	(11,18)	(20,24)
encode	21.8	21.8	15.4	21.6
decode	23.2	18.7	17.3	19.4

Core i7 2710MQ (2.5GHz * 4core), 16GB RAM

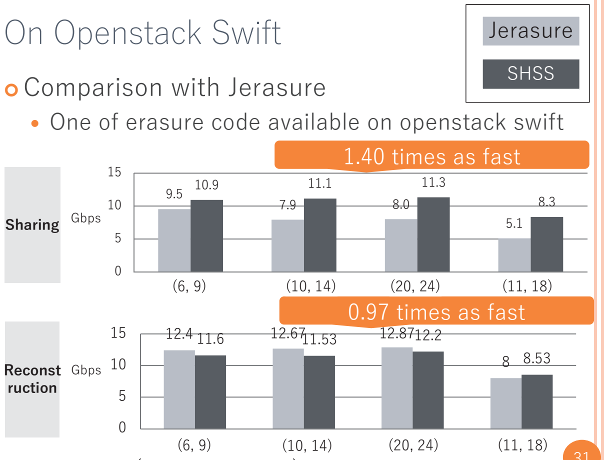
- Comparison with existing implementation
  - AONT-RS$_{\text{secure}}$ [RP11]
    - 0.46Gbps on $(k, n) = (20,24)$

Our library is 47 times faster

[RP11] J. K. Resch and J. S. Plank, AONT-RS: Blending Security and Performance in Dispersed Storage Systems, USENIX FAST, 2011.

On Openstack Swift

- Comparison with Jerasure
  - One of erasure code available on openstack swift



Xeon E5-2630(2.4GHz * 8core), 32G RAM

31

Conclusion

- Motivation: Adopting SS to object storage
  - More efficient SS is needed

- We implement SHSS: Super High-speed (or, Sugoku-Hayai) Secret Sharing
  - achieves 20Gbps on large n
    - 47 times faster than the existing SS library
  - as fast as Jerasure on openstack swift
    - We can add data secrecy with no performance decrease

32

Unequal Secret Sharing Scheme - a Proposal

(Abstract)

Rocki H. Ozaki*1
Real Technology Inc.

Kouichi Sakurai*2
Kyushu University

1. Preface

Various ideas and effort has been put into the works of "secret sharing scheme" (SSS,) initially invented independently by Shamir[1] and Blakley[2] in 1979. While the original version was "perfect" in that it assured information theoretic security, it also had some drawbacks that subsequent scholars and researchers had/ tried to improve.

This work of Ozaki/Sakurai (call it USSS in short) is one of such wherein most (if not all, to the best of our knowledge) of the SSS generate "shares" that are of equal importance and authority. USSS introduces "unequality" to the shares, wherein, for example, if shares are generated under (3, 8) threshold USSS, let us call them $\{S_1, S_2, ...S_8\}$ making S_1, S_2, S_3 as "privileged" and the rest as "non-privileged" shares. The non-privileged shares need at least one of the privileged share to reconstruct the original data.

2. The Effect

This USSS has an effect of making certain shareholders indispensable to reconstructing the original data, while 5 of the non-privileged shareholders cannot reconstruct any data or assume any part thereof. Assume A, B and C are bank staff and each given non-privileged share S_4, S_5, S_6, while M is a manager and given a privileged share S_3. In (3, 8) USSS, $A+B+C$ cannot reconstruct the original data because they are all non-privileged. They need a share from M in order to reconstruct. Either of $A+B+M$ or $A+C+M$ or $B+C+M$ will successfully reconstruct. If D is a director and given privileged share S_2, then $A+M+D$ will also reconstruct. This is the effect of USSS and conceived to have practical usage in many business environment.

3. Basic Theory

The basic theory of USSS can be explained as follows. It is a three-step process. Let S be the original data, and

Step-1: encrypt S and generate E, using key K. The encryption algorithm does not matter so long as it uses one key.)

Step-2: using any secret sharing algorithm, generate shares of E and then generate shares of K. In other words, two sets of shares are generated. (The algorithm of secret sharing could be any; if size is important it could be IDA (Information Dispersal Algorithm, Rabin [3]) or if perfection is important then it could be any of the Shamir's SSS or its descendents.)

Step-3: linking of the shares. For sake of easy explanation, let us assume (3, 8) SS for E $\{E_1, E_2, ...E_8\}$and (3,4) for K $\{K_1, K_2, ...K_4\}$. Then we link the shares E_1 and K_1; we shall write this $[E_1{:}K_1]$. The method for linking of the shares (files) does not matter so it could be a straight concatenation or some "secret" way of linking under some algorithm. Since there are 8 of Es and 4 of Ks, they will be linked as follows.

$[E_1{:}K_1]$ ... $[E_4{:}K_4]$ $[E_5{:}K_4]$ $[E_6{:}K_4]$ $[E_7{:}K_4]$ $[E_8{:}K_4]$ and given to staffs (shareholders) A, B, C, D, E, F, G, H respectively. Then, $A+B+C$ or $A+B+D$ will be able to reconstruct the original data, but $A+E+G$ or $F+G+H$ cannot since it is short of 3 parts of K to reconstruct the key.

**Step-1 and 2 resembles the method of SSMS by Krawczyk[4] but differ completely in Step-3.

4. Conclusion

This mechanism of USSS can be used in many variations depending on the applications. It can make layers of privileges and/or to limit access structures using file servers, local or on the cloud.

References

[1] Shamir, A., "How to Share a Secret," Communications of the ACM, Vol.22, No.11 (1979) pp. 612-613.

[2] Blakley, G. R. "Safeguarding cryptographic keys". Proceedings of the National Computer Conference 48 (1979) pp. 313–317

[3] Rabin, M.O., "Efficient Dispersal of Information for Security, Load Balancing, and Fault Tolerance," Journal of ACM, Vol. 36, No. 2, 1989, pp. 335-348

[4] Krawczyk, H., "Secret Sharing Made Short," D.R. Stinson (Ed.): Advances in Cryptology, CRYPT0 '93, LNCS 773, PP. 136-146, 1994. (c) Springer-Verlag Berlin Heidelberg 1994

Unequal Secret Sharing Scheme

- a proposal -

Rocki H. Ozaki
Real Technology Inc.
Yokohama, Japan

Kouichi Sakurai
Kyushu University
Fukuoka, Japan

Prepared by Rocki ozaki

Secret Sharing – Brief History

1979 Shamir[1], Blakely[2] invent the basics
Perfect ("information theoretic security" is assured)

1989 Rabin[3] IDA (Information Dispersal Algorithm)
Use less resource and faster (Computational Security)

1993 Krawczyk [4] combine perfection and speed
Encrypt data first, and then use SSS to assure perfection

1997 Rivest[5] AONT (all or nothing transform)
Protect against brute force attack (an alternative to SSS)

And many others' work; in area of Verifiable SS, Robust SS, Hierarchical SS, Rational SS, Multi-SS, Ideal SS, Proactive SS, etc. and research on the "access structure."

So what now? Yet another…

The problem is…

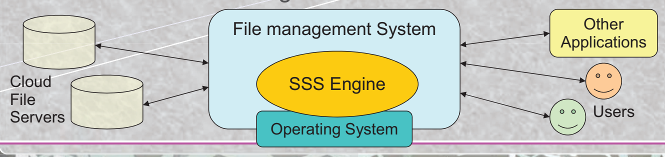
It is quite unsure "how" to use SSS in practical application.

- Who should be the dealer? Who should be the combiner?
- How to distribute the share? How to keep the share safely?
- How to send to combine? Can you trust the combiner?
- Can you trust other shareholders?
- Would others trust me? How can I proove I'm a good guy?
- I lost my share. Someone stole it? Can it be invalidated?
- I am a manager and need some authority to combine data by myself, without requesting the combiner each time.
- How can I manage hundreds of shares of hundreds of files?

We need a practical solution

A SSS solution that can:

- Utilize fully the cloud environment and servers.
- Manage hundreds (if not thousands) of files and shares.
- Manage shareholders (users) under levels of "privileges."
- Assure information theoretic security.
- Fast, reliable and easy to use.
- Ultimately, a SSS application should be in the center of a cloud based file management software.



4

Functions needed in next-gen SSS

In order to meet the criteria, per previous slide…

We must implement the following functions.

- No need for a dealer, no need for a combiner. Therefore,
- Fully server stored. Shares distributed over the Internet.
- Split and combine is automatic, but only within user's PC.
- Keep no original file on PC, split and send to the servers immediately and automatically.
- Basic access structure managed by server access. But,
- Authorization is managed by the engine, in layers.
- Higher layer managers can combine by themselves, lower layer staff need approval from higher layer manager(s).
- All shares should not be made equal = Need for unequality.

5

Implementing Unequality into SSS

How to implement "unequality" into SSS shares?
The easiest is to use a tag; but tags can be read from outside (or further, manipulated.)

authorization tag	share (split data of the original file)

So we have to use a tag that is not readable.
But, if we encrypt the tag, we have to manage an extra key, which is contrary to the concept of SSS.

So the idea is; to let the tag itself be a SSS split data of the authorization data.

6

Here's how it works

Let S be the original data ("secret" as is often called.)

Original data **S** (readable)

Encrypt this by some algorithm

Encrypted data **E** (un-readable) + Key

Split this by some SSS algorithm

- Share **E1** (un-readable)
- Share **E2** (un-readable)
- Share **E3** (un-readable)
- Share **E4** (un-readable)
- Share **E5** (un-readable)
- Share **E6** (un-readable)

- Key **K1** (un-readable)
- Key **K2** (un-readable)
- Key **K3** (un-readable)
- Key **K4** (un-readable)

7

The trick is…

So up to here, it looks like Krawczykz's SSMS ?!
The trick is; we link the two elements and make one file, as below: **This is a sample of (3,6) SSS for **E** and (3,4) SSS for **K**.

Share **E1** (un-readable)	Key **K1** (un-readable)	give to user A
Share **E2** (un-readable)	Key **K2** (un-readable)	give to user B
Share **E3** (un-readable)	Key **K3** (un-readable)	give to user C
Share **E4** (un-readable)	Key **K4** (un-readable)	give to user D
Share **E5** (un-readable)	Key **K4** (un-readable)	give to user E
Share **E6** (un-readable)	Key **K4** (un-readable)	give to user F

The trick is these users hold the same K4 so they cannot combine the original file.

Users A+B+C can combine, B+C+D can combine, but users D+E+F cannot combine unless they get 2 shares from A,B or C.

8

and users can hold multiple shares

A sample of USSS: Data= (2, 6) Key=(2,4)

Share **E1**	Key **K1**	give to CEO
Share **E2**	Key **K2**	give to CEO
Share **E3**	Key **K3**	give to Manager
Share **E4**	Key **K4**	give to Manager
Share **E5**	Key **K4**	give to Staff 1
Share **E6**	Key **K4**	give to Staff 2

Users \ Servers	①	②	③
CEO	[E1:K1]	[E2:K2]	
Manager	[E4:K4]		[E3:K3]
Staff 1	[E5:K4]	Allocation Table	↑
Staff 2	[E6:K4]		↑

Give 2 shares to CEO, 2 shares to Manager.
But [E4:K4] [E5:K4] [E6:K4] has same K4 so they cannot combine.
This is a sample of two layers;
Layer 1 (CEO and Manager) = can combine by themselves
Layer 2 (Staff 1 and 2) = need to access [E3:K3] of Manager

9

A sample of deployment

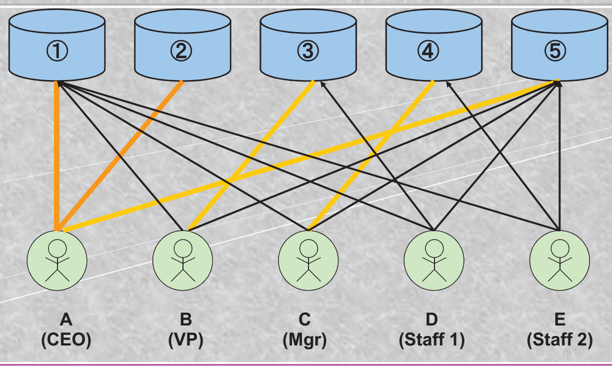
Under this scheme, it can be deployed in many different variations. Below is a sample of how to make this combination of servers and shares, according to the pre-set numbers of users. (Diagram shows we need (3,9) for data and (3,6) for the key, and 5 servers.

Users \ Servers	①	②	③	④	⑤
CEO	[E1:K1]	[E2:K2]			[E3:K3]
VP	[E6:K6]		[E4:K4]		↑
Manager	[E7:K6]			[E5:K5]	↑
Staff 1	[E8:K6]		↑		↑
Staff 2	[E9:K6]			↑	↑

is exclusive access (full control)
allow others to access (full control)
is non-privileged (no control)
↑is server access only (no control)

Allocation Table

Access to Servers



In the previous sample…

In the sample of slide 10, we are managing a company (or group) with CEO, VP, Manager and 2 staffs, D and E. The CEO has access and full control over servers ①②and⑤ so he/she can combine the original data any time. The VP has access and full control over server ③ but need access right to servers ①and⑤ so the CEO could shut out access if needed. Likewise, the Manager need ①and⑤. But then, the VP and Manager has full control over ③and④ respectively, so they can shut out staff A and B if needed. Staffs A and B have no control over servers, so they cannot manupulate to combine the data without being permitted to access the servers. So in this sample, there are three levels of hierarchy, the CEO level, the VP and Manager level, and the staffs level.

Another sample of deployment

Users \ Device	PD	①	②	③	④	⑤	⑥	⑦	⑧
CEO	[E1:K1]	[E2:K2]	[E3:K3]			[E4:K5]			
VP	[E5:K5]					↑	[E6:K6]		
Project Mgr 1				[E7:K7]			↑	[E8:K8]	
Project Mgr 2					[E9:K9]		↑		[E10:K10]
Staff 1 (G1)	[E11:K11]						↑	↑	
Staff 2 (G1)	[E12:K11]						↑	↑	
Staff 3 (G1)	[E13:K11]						↑	↑	
Staff 4 (G2)	[E14:K11]						↑		↑
Staff 5 (G2)	[E15:K11]						↑		↑
Staff 6 (G2)	[E16:K11]						↑		↑

Combine=3: Two new concept, “coverage” and PD = Private Device (e.g. USB mem.)

Hierarchy in Layers

This mechanism allows hierarchical management in layers. (Introduction of PD=Private Device)

Layer 1 CEO
Coverage 133%, can shut out VP

Layer 2 VP
Coverage 66%, can shut out all lower layers

Layer 3 Managers
Coverage 66%, can shut out project team members

Layer 4 Staff Group A and B
Coverage 33%, must have access rights from bosses.
They cannot combine using their PDs.

Defining the Security Policy

Security Policy. (The constitution for “deployment.”)

- who becomes the master controller?
- whether to use PD (private device) or not?
- whether to use a CS (common server) or not?
- how many layers? Define the hierarchy.
- define coverage for each layer or section.
- overlap this with “access management” of information for example:

level definition		who can accesss
Level-3	Top secret information	top management only
Level-2	Corporate secret info	directors and above
Level-1	Limited access info	managers and above
Level-0	General info	no restriction (sectional)

And more, but the base is…

This deployment can be extended further, but the base is, using "unequal SSS" mechanism, as explained in slides 7 and 8.

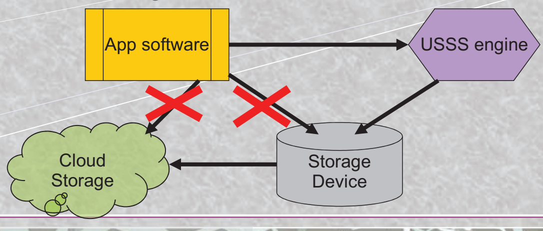
Adding further,

1. The encryption of original data S can be a fast but somewhat weak encryption or can be a strong AONT, as the needs may be.
2. The SSS of encrypted data can be (or better be) a fast but only secure computationally, such as IDA or similar, while the SSS of the key can be (or better be) a perfect SSS that is information theoretically secure.
3. Having said that, however, it is up to the development of further optimized SSS engines by researchers of the future.

 1

Other Considerations-1

Application link – Files made by application software (such as Word, Excel, etc.) should be restricted not to write out the document files directly on PCs and/or cloud storages. They should be handed over to the USSS engine and then split before being written out to local storage or cloud server.



 1

Other Considerations-2

User authentication – The only protection on the server side is the user authentication. Accordingly, an advanced yet simple authentication method should be used together.

Automatic generation of the Allocation Table – With tens of managers and hundreds of staffs in hierarchical structure, it is not easy to make the allocation table. A program should be made to do this automatically.

Off-line combine – In certain case, on-line access may be (should be) restricted. More research on combining off-line should be studied.

 1

Reference

[1] Shamir, A., "How to Share a Secret," Communications of the ACM, Vol.22, No.11 (1979) pp. 612-613.

[2] Blakley, G. R. "Safeguarding cryptographic keys". Proceedings of the National Computer Conference 48 (1979) pp. 313–317

[3] Rabin, M.O., "Efficient Dispersal of Information for Security, Load Balancing, and Fault Tolerance," Journal of ACM, Vol. 36, No. 2, 1989, pp. 335-348

[4] Krawczyk, H., "Secret Sharing Made Short," D.R. Stinson (Ed.): Advances in Cryptology, CRYPT0 '93, LNCS 773, PP. 136-146, 1994. (c) Springer-Verlag Berlin Heidelberg 1994

[5] Rivest R. Rivest, “All-or-nothing encryption and the package transform,” Fast Software Encryption ’97, Lecture Notes in Computer Science Vol. 1267, E. Biham ed., Springer-Verlag, 1997

1

Thanking for your attention.

2

IMI Workshop: Secret Sharing for Dependability, Usability and Security of Network Storage and Its Mathematical Modeling

September 5–7, 2016, Kyushu University

Integration of IoT and big data security by using asymmetric secret sharing scheme

Keiichi IWAMURA

Tokyo University of Science

iwamura@ee.kagu.tus.ac.jp

In recent years, the research on big data security and IoT (Things of Internet) security is prosperous. Especially, to realize utilization and privacy protection of big data, research on secrecy computation or searchable encryption which calculates or retrieves without restoring the data enciphered is done briskly. However, research on such Big data security is premised on that there are enough calculation resources in many cases. On the other hand, since IoT data is main data which constitutes Big data, the data enciphered by the IoT device is desired to turn into the data which can carry out secrecy computation or secrecy retrieval without being restored as it is, i.e., data compatible with big data security. However, since an IoT device is the "thing" which was not connected with a network until now, and calculation resource and communication capability are given and it is made into the part of a network, it is difficult in cost to give a big calculation resource, electric power, etc. to the "thing." Therefore, it is difficult to reconcile big data security and IoT security.

In this research, the mechanism of realizing Big data security and IoT security simultaneously using a secret sharing scheme is proposed. In this research, we use Asymmetric Secret Scharing Scheme [TKI14] by which owner of secret can control the restoration and the secrecy computation and retrieval of the secret. In addition, we propose the secrecy computation [SIK16] which can be performed in n¡2k-1. By these, a mechanism with the following features is realized. IoT device can generates the share by light processing. The secret is not revealed, since the number of output from IoT devices is less than k-1, even if all communication paths are intercepted. IoT device of relay can perform secrecy computation by light processing. The share from IoT device is saved or used for restoration, secrecy computation and retrieval as it is. The owner of secret can control the restoration and use after secret sharing only by managing one key. The secret is not revealed in secrecy computation, even if all the players except the owner collude.

References

[1] S. Takahashi, H. Kang, K. Iwamura:Asymmetric Secret Sharing Scheme Suitable for Cloud Systems, 2014 IEEE 11th Consumer Communications and Networking Conference (CCNC), pp.798-80

[2] T. Shingu, K. Iwamura, K. Kaneda: Secrecy Computation without Changing Polynomial Degree in Shamirs (k,n) Secret Sharing Scheme, DCNET2016.

Integration of IoT and big data security by using asymmetric secret sharing scheme

Keiichi Iwamura
Tokyo University of Science

Outline

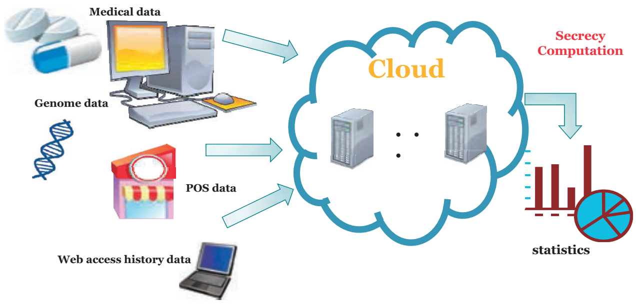
- Background
- Asymmetric Secret Sharing Scheme
- Secrecy Computation & retrieval
- Integration of IoT and big data security

Outline

- Background
- Asymmetric Secret Sharing Scheme
- Secrecy Computation & retrieval
- Integration of IoT and big data security

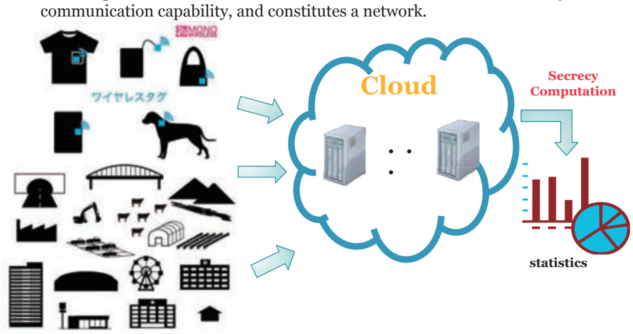
Bigdata

Utilization of Big data derives new knowledge for business and so on.
Secrecy computation is needed, since Big data includes many privacy information.



IoT(Internet of Things)

- The thing which was not connected with a network has calculation ability and communication capability, and constitutes a network.



Difference of IoT and Bigdata Security

- IoT
  - IoT device has poor calculation ability and memory.
  - IoT device works by a battery in many cases.
  - IoT device is unsuitable for the processing which needs large computational complexity such as public key cryptosystem.
- Bigdata
  - requires secrecy computation which has kept the input secret.
  - The privacy protection which respected the owner's intention is required.
  - A sufficient calculation resource and memory are prepared.

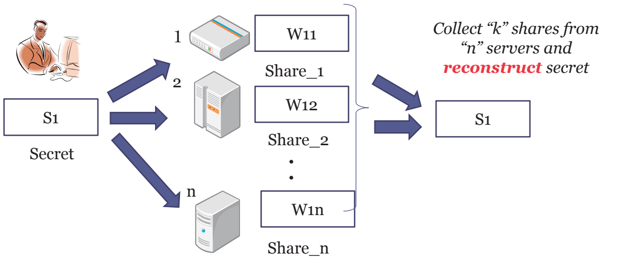
Integration of IoT and Bigdata security

- Encryption is possible also for an IoT device by lightweight processing.
- Secrecy calculation of the data from IoT devices are directly possible without conversion.
- Secrecy retrieval of the data from IoT devices are directly possible without conversion.
- Secret does not leak and the owner of the secret can control the use (restoration, computation, etc).

Outline

- Background
- Asymmetric Secret Sharing Scheme
- Secrecy Computation & retrieval
- Integration of IoT and big data security

Secret Sharing Scheme (SSS)



Feature
- ☐ Even if n-k shares are lost, **the secret can be restored**.
- ☐ Even if k-1 shares are stolen, **the secret does not leak**.

Shamir's Secret Sharing Scheme

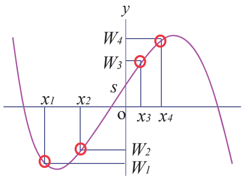
Distribution

1. Pick k-1 random integers $\boldsymbol{a_1}, \dots, \boldsymbol{a_{k-1}}$
2. Build the polynomial of the degree k-1:

$$W_i = s + a_1 x_i + a_2 x_i{}^2 + \cdots + a_{k-1} x_i{}^{k-1}$$

(Generally, $\boldsymbol{x_i}$ is a server ID. ***s*** is a secret.)

3. Compute shares W_i (i=1,…, n).
4. Send a point $(\boldsymbol{x_i}, \boldsymbol{W_i})$ to each sever.



←Determine the curve

←Determine shares (point (x_i, W_i)***)***

(determined share is a coordinate of each point)

[1] A.Shamir. How to share a secret. Communications of the ACM, 22, (11), pp.612-613 (1979)

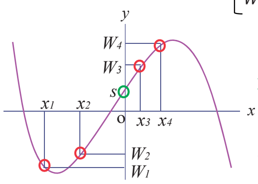
11

Shamir's Secret Sharing Scheme

Reconstruction

1. Correct k shares from servers.
2. Compute ***s*** by solving k polynomials in simultaneous equations.

k polynomials
$$\begin{cases} W_1 = s + a_1 x_1 + a_2 x_1{}^2 + \cdots + a_{k-1} x_1{}^{k-1} \\ \vdots \\ W_k = s + a_1 x_k + a_2 x_k{}^2 + \cdots + a_{k-1} x_k{}^{k-1} \end{cases}$$



←Reconstruct the curve

y-intercept is the secret

←Correct k points (x_i, W_i)

[1] A.Shamir. How to share a secret. Communications of the ACM, 22, (11), pp.612-613 (1979)

Secrecy Computation in secret sharing scheme

- Secrecy addition/subtraction

$$W_{a1} = a + a_1 x_1 + a_2 x_1{}^2 + \cdots + a_{k-1} x_1^{k-1}$$

$$+ \quad W_{b1} = b + b_1 x_1 + b_2 x_1{}^2 + \cdots + b_{k-1} x_1^{k-1}$$

$$W_{a1} + W_{b1} = (a+b) + (a_1 + b_1) x_1 + (a_2 + b_2) x_1{}^2 + \cdots + (a_{k-1} + b_{k-1}) x_1^{k-1}$$

$$W_{a2} = a + a_1 x_2 + a_2 x_2{}^2 + \cdots + a_{k-1} x_2^{k-1}$$

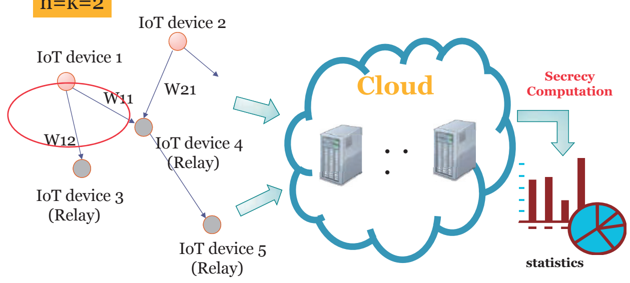
$$+ \quad W_{b1} = b + b_1 x_2 + b_2 x_2{}^2 + \cdots + b_{k-1} x_2^{k-1}$$

$$W_{a1} + W_{b1} = (a+b) + (a_1 + b_1) x_2 + (a_2 + b_2) x_2{}^2 + \cdots + (a_{k-1} + b_{k-1}) x_2^{k-1}$$

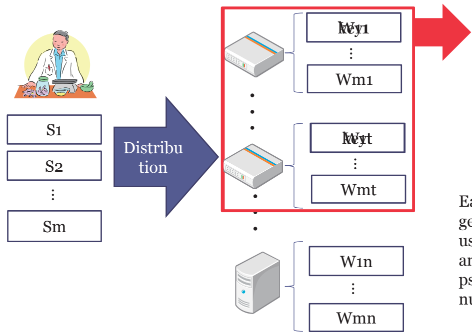
$$\vdots$$

Application to IoT of Shamir's SSS

n=k=2



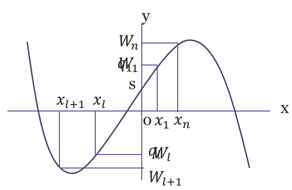
Asymmetric Secret Sharing Scheme



t(<k) servers from n servers are selected as Key server.
(The key server has only a key, and is a maximum of k-1)

Each key server generates each share by using ID of the secret and its own key as pseudo-random number.

Asymmetric secret sharing scheme



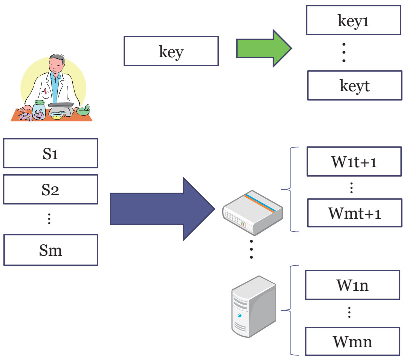
W=s+a_1 x+...+a_{k-1} x^{k-1} (1)

[Shamir's SSS]
decide a polynomial (1)
⇒obtain shares

[Asymmetric SSS]
decides shares up to k-1
using pseudo-random number
⇒decide the plynomial (1)
⇒obtain the remaining shares

The security of A-SSS is depend on that of pseudo-random number

Asymmetric Secret Sharing Scheme



The owner generates t(<k) keys for key servers from one owner's key.

[Feature of A-SSS]
1. Secure, even if all the data servers are attacked, since n-t<k.
2. The owner can control huge secrets at a smart device, since he just manages one key.
3. Secrecy calculation like Shamir's scheme is possible, since a share is the same form.

Asymmetric Secret Sharing Scheme

$Enc(a,b)$: pseudo-random number generation using a snd b

[Distribution protocol]

1. An owner of secrets generates key_j for key server x_j from the owner's key key_o. $key_j = Enc(x_j, key_o)$ (1)
2. The key server generates pseudo-random number $q_{ij}(j = 1, \dots, t)$ as shares of secret s_i using key_j. $q_{ij} = Enc(dID[s_i], key_j)$ (2)
3. The owner determines $k-1-t$ coefficients $[a_{it+1}, \dots, a_{ik-1}]$ in the following polynomial.
$$W_{ij} = s_i + a_{i1}x + \dots + a_{it}x^t + a_{it+1}x^{t+1} + \dots + a_{ik-1}x^{k-1} \quad (3)$$
4. The owner solves the following equations using $S = [s_i, \dots, s_i]^T$ and $Q = [q_{i1}, \dots, q_{it}]^T$, and determines the remaining t coefficients $A(i)_{k-1} = [a_{i1}, \dots, a_{it}]^T$. $A(i)_{k-1} = X'^{-1}(Q - S)$ (4)
5. The owner calculates the remaining shares $W_{it+1}, \dots, W_{in}$ using the polynomial (3) with the determined coefficients, and sends W_{ij} and $dID[s_i]$ to data servers x_j $(j = t+1, \dots, n)$
6. The data server stores each share and $dID[s_i]$.

Asymmetric secret sharing scheme

[Reconstruction protocol]

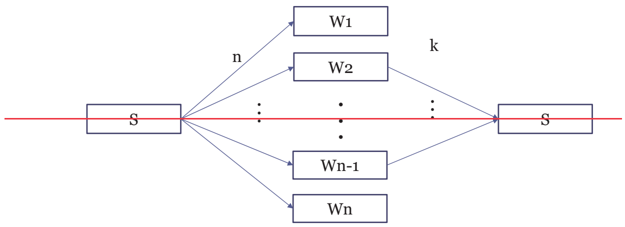
1. The user who reconstructs secret s_i selects any k servers from the n servers and sends the $dID[s_i]$.
2. If key server x_j is selected, the server generates q_{ij} using equation (2) and sends it to the user.
3. If a data server is selected, the server sends the shares W_{ij} corresponding to $dID[s_i]$ to the user.
4. The user reconstructs secret s_i as in Shamir's scheme.

[Notation]

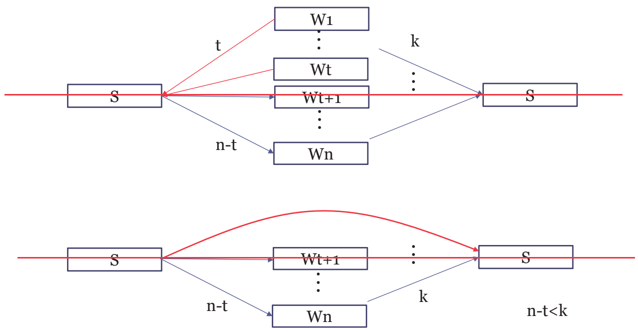
- $Enc(a,b)$: pseudo-random number generation using a snd b
- $dID[s_i]$: data ID of s_i $H(s_i) = H(s_i \mid dIS(s_i))$
-

$$X = \begin{bmatrix} x_1 & \cdots & x_1^{t-1} \\ \vdots & \ddots & \vdots \\ x_t & \cdots & x_t^{t-1} \end{bmatrix}$$

Secret Sharing Scheme (Symmetric)



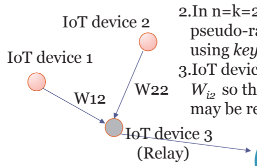
Asymmetric Secret Sharing Scheme



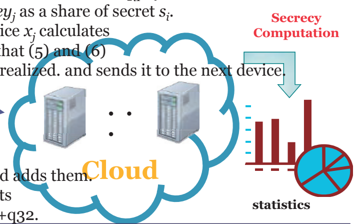
Application to IoT of A-SSS

1. The owner sets each key to each IoT device

$q_{i1} = s_i + a_{i1}c1$ (5)
$W_{i2} = s_i + a_{i1}c2$ (6)
c1, c2 are the constants defined in advance.



2.In n=k=2, IoT device x_j generates pseudo-random number q_{ii} by using key_j as a share of secret s_i.
3.IoT device x_j calculates W_{i2} so that (5) and (6) may be realized. and sends it to the next device.

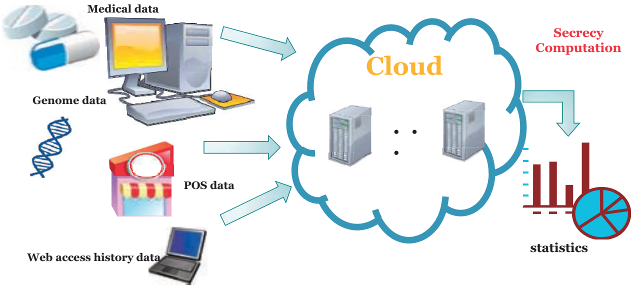


4.IoT device 3 adds W12+W22+W32, and sends it to the cloud.
5.The owner generates q12,q22,q32 and adds them.
6. The owner restores the sum of secrets from W12+W22+W32 and q12+q22+q32.

- **IoT device can calculate k-1 or less share by light processing.**
- **The secrets are not revealed even if all the communication paths are intercepted.**
- **Secrecy addition is also possible at an IoT device of relay.**

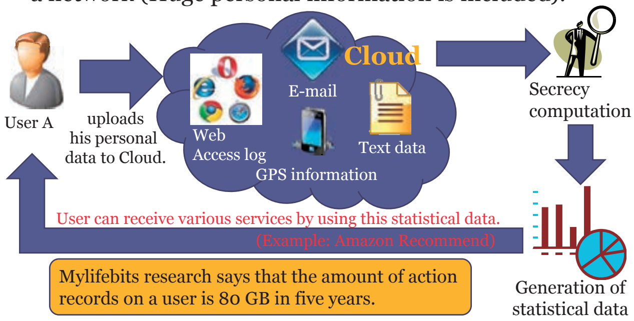
Application to Bigdata of A-SSS

When n=3 and k=3, if the owner manages 3 keys for key servers, and the data server in cloud is 3, the secret can be restored, even if two data server breaks.



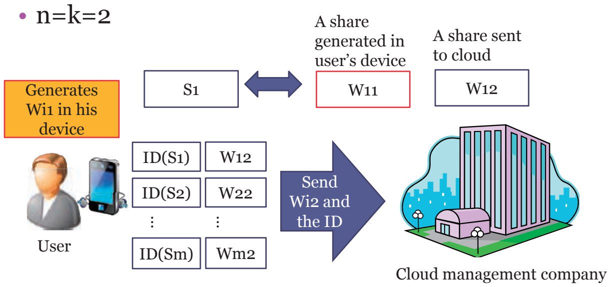
Life Log System

- A life log is action record generated from the user data on a network (Huge personal information is included).



Life Log System using A-SSS

- n=k=2



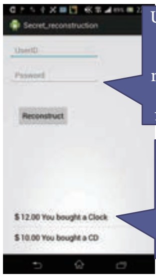
User's merit：He just manages a key to a vast quantity of data
He can control the secret restoration and use.
Cloud merit：The secrets is not revealed only from itselves.
The cloud holds users by little server investment.

Example of Life Log system using A-SSS

- The scene of performing net shopping by using smart phone which the application downloaded.

User inputs his/her ID and password.
⇒The smart phone generates key for key server by using the password.

User selects items.
⇒This calculates share for cost as secret by using A-SSS, and sends it to cloud server

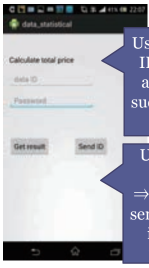


User inputs his/her ID and password, and touches reconstruct button.
⇒This gets share from cloud server.

This phone restores the cost as secret from the share and pseudo-random number using the key

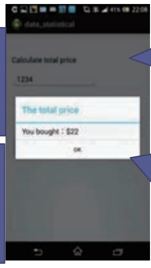
Mounting of Life Log system using A-SSS

- The scene of checking the total amount used by net shopping



User inputs his/her ID and password, and sends the ID such as the range of date and time.

User touches Get result.
⇒The cloud server sends IDs of secrets included in the period to user.



The cloud server adds the shares of sent IDs and send the results to user.

The phone generates each share from each ID, adds them, and restores the results as the total price .

Comparison between SSS and A-SSS

- SSS
  - does not have a mechanism for k shares not to be revealed.
  - In order not to reveal k shares, it is necessary to enlarge k (<n) but, and cost starts maintenance of a server.
  - The owner cannot control his secret after distribution.
  - realizes information theoretic security.
- A-SSS
  - has a mechanism for k shares not to be revealed.
  - Even if k=2, since the number of data server can be done in 1, secrets are not revealed.
  - The owner can control his secret after distribution.
  - realizes computational security.

Outline

- Background
- Asymmetric Secret Sharing Scheme
- Secrecy Computation & retrieval
- Integration of IoT and big data security

The conventional secrecy multiplication

Multiplication of polynomial with degree k-1

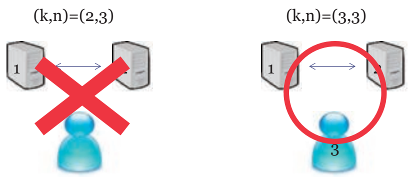
$$\boxed{W_i = s + a_1 x_i + \cdots + a_{k-1} x_i^{k-1}}$$

$$\begin{aligned} W_{ai} &= a + a_1 x_i + \cdots + a_{k-1} x_i^{k-1} \\ \times\ W_{bi} &= b + b_1 x_i + \cdots + b_{k-1} x_i^{k-1} \\ \hline W_{(ab)i} &= ab + (ba_1 + ab_1) x_i + \cdots + (a_{k-1} b_{k-1}) x_i^{2k-2} \end{aligned}$$

The degree changes from k-1 to 2k-2.
⇒2k-1 shares are required for restoration.
⇒Secrecy multiplication cannot be performed in n<2k-1.
⇒The threshold changes only in multiplication.

Problem of the conventional scheme

Consider the case of n=3,k=2 (n≧2k-1).
If 2 servers collude, the secret will be revealed.
⇒The conventional secrecy computation does not have a mechanism for k shares not to be revealed like the conventional SSS.
⇒If secrecy computation in n<2k-1 realizes, the secret does not reveal.



31

Our proposed secrecy multiplication

A threshold dose not changes in multiplication

$W_{ai} = \boxed{\alpha a} + a_1 x_i + \cdots + a_{k-1} x_i^{k-1}$ → αa is temporarily restored as scalar value.

$W_{bi} = \boxed{\beta b} + b_1 x_i + \cdots + b_{k-1} x_i^{k-1}$

The concealed secret

Multiplication

$\alpha a W_{bi} = \alpha a\,(\beta b + b_1 x_i + \cdots + b_{k-1} x_i^{k-1})$

Scalar quantity × Polynomial
→The degree of the result polynomial does not change.
→The secrecy computation in n<2k-1 is realized.

32

Distribution

$\overline{[s]}_i$: A share of s which player P_i holds

$[s]_i$: A set of shares on s which player P_i holds

Distribution

1: generates random numbers $\alpha_0, \cdots, \alpha_{k-1} \in GF(q)$

2: $\alpha = \prod_{j=0}^{k-1} \alpha_j$

3: $\alpha \times a$

4: distributes $\alpha a, \alpha_0, \cdots, \alpha_{k-1}$ by using Shamir's scheme

$P_i \left\{ \begin{array}{c} \overline{[\alpha a]}_i \\ \overline{[\alpha_0]}_i \\ \vdots \\ \overline{[\alpha_{k-1}]}_i \end{array} \right. \rightarrow [a]_i$

33

Restoration

$\overline{[s]}_i$: A share of s which player P_i holds

$[s]_i$: A set of shares on s which player P_i holds

Restoration

1: collects k set of shares $[a]_0, \cdots, [a]_{k-1}$

2: restores $\alpha a, \alpha_0, \cdots, \alpha_{k-1}$.

3: $\alpha = \prod_{j=0}^{k-1} \alpha_j$

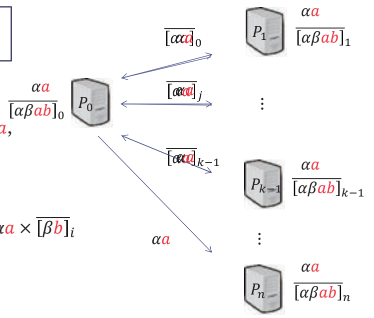
4: obtains a by $a = \alpha a / \alpha$

$P_i \left\{ \begin{array}{c} \overline{[\alpha a]}_i \\ \overline{[\alpha_0]}_i \\ \vdots \\ \overline{[\alpha_{k-1}]}_i \end{array} \right. \rightarrow [a]_i$

34

Secrecy Multiplication

P_i = *All server*
P_j = *k servers which process*



1: P_0 collects k shares of αa, and restores αa.

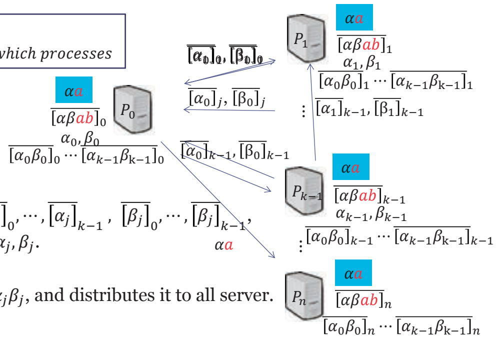
2: P_0 sends it to all server.

3: P_i calculates $\overline{[\alpha\beta ab]}_i = \alpha a \times \overline{[\beta b]}_i$

35

Secrecy Multiplication

P_i = *All server*
P_j = *k server which processes*



4: P_j collects $[\alpha_j]_0, \cdots, [\alpha_j]_{k-1}$, $[\beta_j]_0, \cdots, [\beta_j]_{k-1}$, and restores α_j, β_j.

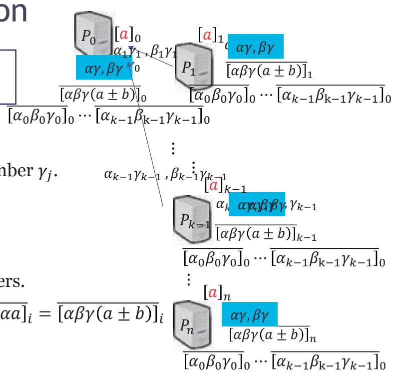
5: P_j calculates $\alpha_j\beta_j$, and distributes it to all server.

6: P_i holds $\left(\overline{[\alpha\beta ab]}_j, \overline{[\alpha_0\beta_0]}_i, \cdots, \overline{[\alpha_{k-1}\beta_{k-1}]}_i\right)$ as a set of shares of ab.

36

Secrecy Addition

P_i = *All server*
P_j = *k servers which process*



1: P_j restores α_j, β_j.

2: P_j generates a random number γ_j.

3: P_j calculates $\alpha_j\gamma_j, \beta_j\gamma_j$, and sends them to P_0.

4: P_0 calculates $\alpha\gamma, \beta\gamma$, and sends them to all servers.

5: P_i calculates $\alpha\gamma\overline{[\beta b]}_i \pm \beta\gamma\overline{[\alpha a]}_i = \overline{[\alpha\beta\gamma(a \pm b)]}_i$

6: P_j distributes $\alpha_j\beta_j\gamma_j$.

7: P_i holds $\left(\overline{[\alpha\beta\gamma(a \pm b)]}_j{}', \overline{[\alpha_0\beta_0\gamma_0]}_i, \cdots, \overline{[\alpha_{k-1}\beta_{k-1}\gamma_{k-1}]}_i\right)$ as the set of share of $a + b$

Type of attacks

- Players on attack
  - 1 .Attaker

    -This player tries to obtain the inputted secret and the result of the secrecy computation. He can know k-1 players' information.
  - 2.Inputter

    -This player inputs a secret.
  - 3.Restorer

    -This player knows the information that k players send to restore the result of the secrecy computation.
  - 4.Combination of players

    -Attacker only, Attacker=inputter, Attacker=Restore, (Attacker=Inputter=Restorer in multi-inputs computation)

Security of our scheme

- It can prove that our scheme in 2-inputs has information theoretic security in four-operations independently under the following conditions.
  (1) Zero is not included in secrecy multiplication.
  (2) Set of shares using different random number is used for the operation of different type.

- T. Shingu, K. Iwamura, K. Kaneda: Secrecy Computation without Changing Polynomial Degree in Shamir's (k,n) Secret Sharing Scheme, DCNET2016.

The problem of our scheme

- αa is known in secrecy multiplication.
- α is known in secrecy addition.

⇒secret a is known by the combination of secrecy multiplication and addition.

The combination of secrecy multiplication and addition

We generate different sets using different random numbers.

$$\begin{array}{cc} \alpha^{(1)} & \alpha^{(2)} \\ [\alpha^{(1)}a]_i & [\alpha^{(2)}a]_i \\ [\alpha_0^{(1)}]_i & [\alpha_0^{(2)}]_i \\ \vdots & \vdots \\ [\alpha_{k-1}^{(1)}]_i & [\alpha_{k-1}^{(2)}]_i \\ \text{Addition} & \text{Multiplication} \end{array}$$

$$\alpha^{(i)} = \prod_{j=0}^{k-1} \alpha_j^{(i)}$$

$\alpha^{(1)}$ is known in secrecy addition.

$\alpha^{(2)}a$ is known in secrecy multiplication.

a is not revealed,
since $\alpha^{(1)}$ and $\alpha^{(2)}a$ are independent

The Multi-inputs secrecy computation

- 2-inputs : a, b
  ab, $a + b$ are calculated securely.
- Multi-inputs : $a, b \cdots, z$ $(\alpha a, \beta b, \cdots, \zeta z)$
  $ab \cdots z$, $a + b + \cdots + z$ are also calculated securely

It can prove that our scheme in multi-inputs has also information theoretic security in four-operations independently under the same conditions.
(1) Zero is not included in secrecy multiplication.
(2) Set of shares using different random number is used for the operation of different type.

Secrecy retrieval

- Secrecy retrieval is realized by secrecy subtraction
  - The registered keyword :P=a
  - The keyword for search :P'=b
- Input: sets of shares of a and b
- Output: $\gamma(a - b)$
  - The output is 0 if $a = b$.
  - The output is a random number if $a \neq b$.
- This scheme has one problem.
  - The solution is presented in CSS2016.

Outline

- Background
- Asymmetric Secret Sharing Scheme
- Secrecy Computation & retrieval
- Integration of IoT and big data security

Integration of IoT and Bigdata security using A-SSS

- IoT device can generates the share by light processing.
- The secret is not revealed, since the number of output is less than k-1, even if all communication paths are intercepted.
- IoT device of relay can perform secrecy computation by light processing.
- The share from IoT device is saved or used for restoration, secrecy computation and retrieval as it is.
- The owner of secret can control the restoration and use after secret sharing only by managing one key.
- The secret is not revealed in secrecy computation, even if all the players except the owner collude.

Thank you for your attention.

Security of our scheme

The security against Attacker only

- Concealing random number $\left(\alpha = \prod_{j=0}^{k-1} \alpha_j\right)$ in secrecy multiplication

→ α is not revealed, even if k-1 random number α_j $(\alpha_0, \cdots, \alpha_{k-2})$ is known.
$H(\alpha) = H(\alpha | \alpha_0, \cdots, \alpha_{k-2})$

- Secret (a) in secrecy multiplication

→ a is not revealed, even if αa is known, if α is unknown.
$H(a) = H(a | \alpha a)$

- Secret (a) in secrecy addition

→ a is not revealed, even if $\alpha\gamma$ and $\beta\gamma$ is known.
$H(a) = H(a | \alpha\gamma, \beta\gamma)$

Security of our scheme

The security against Attacker=Inputter

- Inputter of b knows b and β in addition to αa in secrecy multiplication.

→ a is not revealed, since α, a and β, b are independent.
$H(a) = H(a | \alpha a, \beta, b)$

- Inputter of b knows b and β in addition to $\alpha\gamma$ and $\beta\gamma$ in secrecy addition.

→ a is not revealed, although α, β, γ are known.
$H(a) = H(a | \alpha, \beta, \gamma)$

Security of our scheme

The security against Attacker=Restorer

- Restorer knows $\alpha\beta$ and $\alpha\beta ab$ in addition to αa in secrecy multiplication.

→ a is not revealed, since $\alpha\beta$ cannot be decomposed.
$H(a) = H(a | \alpha\beta, \alpha a)$

- Restorer knows$\alpha\beta\ \gamma$ in addition to $\alpha\gamma$ and $\beta\gamma$ in secrecy addition.

→ a is not revealed, although α, β, γ are known.
$H(a) = H(a | \alpha, \beta, \gamma)$

Our scheme in 2-inputs has information theoretic security in four-operations independently, where zero is not included in secrecy multiplication.

Secret sharing schemes based on additive codes

Jon-Lark Kim
Sogang University
jlkim@sogang.ac.kr

A secret sharing scheme (SSS) was introduced by Shamir in 1979 using polynomial interpolation. It was shown that it is equivalent to an SSS based on a Reed-Solomon code. SSSs based on linear codes have been studied by numerous researchers. However there is little research on SSSs based on additive codes (that is, codes closed under addition). In this talk, we study SSSs based on additive codes, in particular, over $GF(4)$. We show that they provide higher security level than linear codes based SSSs since they require at least two steps of calculations to reveal the secret. We also describe our theorems using several interesting additive codes over $GF(4)$ including the hexacode of length 6, the dodecacode of length 12 and S_{18}, all of which contain generalized 2-designs. This is a joint work with Nari Lee.

Secret sharing schemes based on additive codes

Jon-Lark Kim

This is a joint work with Nari Lee.

Department of Mathematics
Sogang University, S. Korea

IMI Workshop: Secret Sharing for Dependability, Usability and Security of Network Storage and Its Mathematical Modeling
Sep. 5-7, 2016

Outline

1. Introduction to SSS
   - SSS based on linear codes
2. Main Results
   - SSS based on additive codes over $GF(4)$
   - Examples on additive codes over $GF(4)$
3. Conclusion

Outline

1. Introduction to SSS
   - SSS based on linear codes
2. Main Results
   - SSS based on additive codes over $GF(4)$
   - Examples on additive codes over $GF(4)$
3. Conclusion

Main Reference

This talk is based on the following paper.

J.-L. Kim and N. Lee, *Secret sharing schemes based on additive codes over GF(4)*, Applicable Algebra in Engineering, Communication and Computing, DOI: 10.1007/s00200-016-0296-5 (2016).

Introduction to SSS

A secret sharing scheme(SSS) is

- a method of distributing a secret to a finite set of participants
- all the participants receive a piece of the secret, a share
- only qualified subsets of the participants can have access to the secret by pooling the shares of their members.

Introduction to SSS

About Secret Sharing Scheme...

- It was introduced by Shamir and Blakley independently in 1979.
- Shamir used polynomial interpolation for constructing secret sharing scheme.
- Blakley used hyperplane geometry.
- Shamir's SSS turned out to be equivalent to a SSS based on a Reed-Solomon code.
- It is natural to think about SSSs based on codes.
- SSSs based on linear codes are widely studied for a long time by numerous people.

Introduction to SSS

Table: History of secret sharing schemes

Year	Author	Contribution using
1979	A. Shamir	a polynomial interpolation
1979	G. R. Blakley	a hyperplane geometry
1981	R.J. McEliece, D.V. Sarwate	a linear code
1983	C. Asmuth, J. Bloom	a Chinese Remainder Theorem
1985	G. R. Blakley	ramp schemes
1993	J.L. Massey	minimal codewords

Introduction to SSS

Some of secret sharing schemes were applied to numerous fields such as

(i) controling nuclear weapons in military
(ii) cloud computing
(iii) recovering information from multiple servers
(iv) controling access in banking system

Motivation

- Secret sharing has been focused for decades.
- The access structure of the scheme can be simply defined as long as the scheme is based on codes holding 1-designs.
- There has been less attention to SSSs based on additive codes.
- What if the properties of additive codes were translated into SSSs?
- Why codes over GF(4)?
- Self-dual codes over $GF(2)$, $GF(3)$, and $GF(4)$ have the property that they are divisible by the Gleason-Pierce-Ward Theorem.
- A code C whose codewords have weights divisible by an integer $c > 1$ is said to be divisible by a divisor c.

SSS based on linear codes

- $G = (\mathbf{g}_0, \mathbf{g}_1, \cdots, \mathbf{g}_{n-1})$: a generator matrix of an $[n, k, d]$ code over $GF(q)$
- The secret $s \in GF(q)$ in SSS is constructed from an $[n, k, d]$ linear code C
- There are $n-1$ participants $P_1, P_2, \cdots, P_{n-1}$ and a dealer P_0

SSS based on linear codes

- A dealer randomly takes an element
$$\mathbf{u}=(u_0, u_1, \cdots, u_{k-1}) \in GF(q)^k$$
- Let
$$\mathbf{t} = (t_0, t_1, \cdots, t_{n-1})=\mathbf{u}G$$
- The secret s is defined as
$$s = \mathbf{u}\mathbf{g}_0 = t_0$$
- The dealer gives the share t_i to participant P_i , $i \geq 1$.

SSS based on linear codes

Lemma 1 (Massey (1993)).

Let C be a $[n, k, d]$ linear code over the finite field $GF(q)$ and let $C^{\perp}$ be its dual code. In the secret sharing scheme based on C, a subset of shares $\{t_{i_1}, t_{i_2}, \cdots, t_{i_m}\}$, $1 \leq i_1 < \cdots < i_m \leq n-1$, determines the secret if and only if there is a codeword

$$(1, 0, \cdots, 0, c_{i_1}, 0, \cdots, 0, c_{i_m}, 0, \cdots, 0) \qquad (1)$$

in $C^{\perp}$ with $c_{i_j} \neq 0$ for at least one j.

Secret Sharing Schemes Based on Additive Codes

If there is a codeword

$$(1, 0, \cdots, 0, c_{i_1}, 0, \cdots, 0, c_{i_m}, 0, \cdots, 0)$$

in $C^{\perp}$, then the vector $\mathbf{g}_0$ is a linear combination of $\mathbf{g}_{i_1}, \ldots, \mathbf{g}_{i_m}$,

$$\mathbf{g}_0 = \sum_{j=1}^{m} x_j \mathbf{g}_{i_j}, \quad x_j \in GF(q).$$

Then the secret s is recovered by computing

$$s = \sum_{j=1}^{m} x_j t_{i_j}.$$

SSS based on linear codes

Definition 2.

- An access group is a subset of a set of participants thst can recover the secret from its shares.
- A collection Γ of access groups is called an access structure of the scheme.
- An element $A \in \Gamma$ is called a minimal access group if no element of Γ is a proper subset of A.
- We let $\bar{\Gamma} = \{A | A \ \text{ is a minimal access group}\}$. We call $\bar{\Gamma}$ the minimal access structure.

Outline

1. Introduction to SSS
   - SSS based on linear codes
2. Main Results
   - SSS based on additive codes over $GF(4)$
   - Examples on additive codes over $GF(4)$
3. Conclusion

SSS based on additive codes

- An additive code $\mathcal{C}$ over $GF(4)$ of length n is an additive subgroup of $GF(4)^n$
- The trace map for x in $GF(4)$: $\mathrm{Tr}(x) = x + x^2 \in GF(2)$
- The trace inner product of two vectors $\mathbf{x} = (x_1 x_2 \cdots x_n)$ and $\mathbf{y} = (y_1 y_2 \cdots y_n)$ in $GF(4)^n$:

$$\mathbf{x} \star \mathbf{y} = \sum_{i=1}^{n} \mathrm{Tr}(x_i \overline{y_i}) \in GF(2),$$

where $\overline{y_i}$ denotes the conjugate of y_i.

SSS based on additive codes

Lemma 3.
Let $\mathcal{C}$ be an $(n, 2^k)$ code over $GF(4)$ and $\mathcal{C}^\perp$ its dual code defined by the trace inner product. Let

$$\begin{aligned} H_1 &= \{x \mid x = (1, \cdots, 0, x_{i_1}, 0, \cdots, 0, x_{i_m}, 0, \cdots, 0) \in \mathcal{C}^\perp \\ &\qquad x_{ij} \neq 0 \text{ for at least one } j\}, \\ H_2 &= \{y \mid y = (\omega, \cdots, 0, y_{i_1}, 0, \cdots, 0, y_{i_l}, 0, \cdots, 0) \in \mathcal{C}^\perp \\ &\qquad y_{ij} \neq 0 \text{ for at least one } j\}, \\ H_3 &= \{z \mid z = (\overline{\omega}, \cdots, 0, z_{i_1}, 0, \cdots, 0, z_{i_r}, 0, \cdots, 0) \in \mathcal{C}^\perp \\ &\qquad z_{ij} \neq 0 \text{ for at least one } j\}. \end{aligned}$$

In the secret sharing scheme based on C, two subsets of shares $\{t_{i_1}, t_{i_2}, \cdots, t_{i_m}\}$ and $\{t_{i_1}, t_{i_2}, \cdots, t_{i_l}\}$, $1 \le i_1 < \cdots < i_m \le n-1$, $1 \le i_1 < \cdots < i_l \le n-1$, determine the secret if and only if there are at least two codewords from distinct sets among H_i's, $1 \le i \le 3$.

SSS based on additive codes

The secret s can be recovered by computing any two of the following:

$$\alpha_1 = \sum_{j=1}^{m} \left(t_{i_j}\bar{x}_j + (t_{i_j}\bar{x}_j)^2\right), \quad \alpha_2 = \sum_{j=1}^{l} \left(t_{i_j}\bar{y}_j + (t_{i_j}\bar{y}_j)^2\right), \quad \alpha_3 = \sum_{j=1}^{r} \left(t_{i_j}\bar{z}_j + (t_{i_j}\bar{z}_j)^2\right).$$

Now we can recover the secret s with the values of α_i's, $1 \le i \le 3$, as the table below.

α_1	0	0	1	1
α_2	0	1	0	1
α_3	0	1	1	0
s	0	1	ω	$\overline{\omega}$

SSS based on additive codes

Let

$$\Gamma_{H_1} = \{\text{the set of supports for } x \in H_1 \text{ excluding 1 from each support}\},$$
$$\Gamma_{H_2} = \{\text{the set of supports for } y \in H_2 \text{ excluding 1 from each support}\},$$
$$\Gamma_{H_3} = \{\text{the set of supports for } z \in H_3 \text{ excluding 1 from each support}\}.$$

- The access structure for a linear code based SSS is Γ_{H_1}.
- For an additive code based SSS we need at least two sets of Γ_{H_1}, Γ_{H_2}, or Γ_{H_3} to define the access structures.

SSS based on additive codes

- The access structures for SSS based on additive codes are defined in a different way from those of linear codes.
- SSSs from additive codes provide higher security level than those from linear codes since it requires at least two steps of calculations to reveal the secret.
- We call this process as a 2-step SSS.
- The previous SSS based on linear codes can be regarded as a 1-step SSS.

Assmus-Mattson Theorem for additive codes over $GF(4)$

Theorem 4 (K. and V. Pless (2003)).

Let $\mathcal{C}$ be an additive $(n, 2^k)$ code over $GF(4)$ with minimum weight d. Let $\mathcal{C}^\perp$ be its dual $(n, 2^{n-k})$ code with minimum weight d'. Let $0 < t < d$. Let s be the number of weights $B_i \neq 0$ in $\mathcal{C}^\perp$ where $0 < i \leq n - t$. Suppose that $s \leq d - t$. Then the following hold.

(i) *For each weight u $(d \leq u \leq n)$, the set of supports of codewords of weight u in $\mathcal{C}$ holds a $t-$design with possibly repeated blocks.*

(ii) *The set of supports of vectors of weight w in $\mathcal{C}^\perp$ where $B_w \neq 0$ and $d' \leq w \leq n - t$ hold a $t-$design with possibly repeated blocks.*

(iii) *The supports of minimum weight vectors are either simple blocks or have repetition number 3.*

Corollary 5.
Let $n_i := 6m + 2(i-1)$ with $m \geq 1$ any integer and $i = 1, 2,$ or 3. Let $\mathcal{C}$ be an extremal additive even self-dual $(n_i, 2^{n_i})$ code over $GF(4)$ with minimum weight $d = 2m + 2 \geq 6$. Then the vectors of each weight w in $\mathcal{C}$ where $A_w \neq 0$ and $d \leq w \leq n_i$ hold a $(7-2i)$-design with possibly repeated blocks.

Lemma 6.
Let C be an additive $(n, 2^k)$ self-dual code over $GF(4)$. Then the supports of codewords for all non-trivial weights hold a 1-design with possible repeated blocks if $d \geq \frac{n+2}{3}$.

Proof.
An additive $(n, 2^k)$ self-dual code over $GF(4)$ has $\frac{n}{2} - 1$ possible non-trivial weights. Then $\frac{d}{2} - 1$ of these possible weights have no vectors since d is the minimum weight. Therefore we need $d - 1 \geq (\frac{n}{2} - 1) - (\frac{d}{2} - 1)$ for the Assmus-Mattson theorem for additive codes over $GF(4)$ to apply. This gives that $d \geq \frac{n+2}{3}$. □

A generalized t-design

(Delsarte (1973))

- An element $a \in GF(q)^n$ is said to be covered componentwisely (*c-covered*) by an element $b \in GF(q)^n$ if each nonzero component a_i of a is equal to the corresponding component b_i of b.
- It is denoted as $a \leq b$.
- For example, $a = (1, 1, \omega, 0)$ is c-covered by $b = (1, 1, \omega, \overline{\omega})$ for $a, \ b \in GF(4)^4$.
- $\mu(i, e)$=the number of codewords of weight i that c-cover e, for $e \in GF(q)^n$
- If $i < wt(e)$, then $\mu(i, e) = 0$.

(Delsarte (1973))

Definition 7.
A subset S of $GF(q)^n$ is called a generalized t-design of type $q-1$, with parameters t-(n, k, μ_t), $0 \leq t \leq k \leq n$, $\mu_t \geq 1$, if the following two conditions are satisfied:

(i) all elements of S have the same weights k,

(ii) each element of weight t in $GF(q)^n$ is c-covered by a constant number μ_t of elements of S. If a subset S of $GF(q)^n$ holds a generalized t-design of type q-1, then it holds a generalized $(t$-1)-design of type q-1.

Outline

SSS using the $(6, 2^6)$ hexacode

Let $\mathcal{G}_6$ be the $[6, 3, 4]$ *hexacode* whose generator matrix as linear $GF(4)$-code is

$$\begin{bmatrix} 1 & 0 & 0 & 1 & \omega & \omega \\ 0 & 1 & 0 & \omega & 1 & \omega \\ 0 & 0 & 1 & \omega & \omega & 1 \end{bmatrix}.$$

As an additive code, the generator matrix of $\mathcal{G}_6$ is

$$\begin{bmatrix} 1 & 0 & 0 & 1 & \omega & \omega \\ \omega & 0 & 0 & \omega & \overline{\omega} & \overline{\omega} \\ 0 & 1 & 0 & \omega & 1 & \omega \\ 0 & \omega & 0 & \overline{\omega} & \omega & \overline{\omega} \\ 0 & 0 & 1 & \omega & \omega & 1 \\ 0 & 0 & \omega & \overline{\omega} & \overline{\omega} & \omega \end{bmatrix}.$$

SSS using the $(6, 2^6)$ hexacode

- The weight distribution of the $(6, 2^6)$ hexacode is :

$$1 + 45y^4 + 18y^6.$$

- The vectors of weight 4 hold a 2-design with possibly repeated blocks by A-M theorem for additive codes.
- The vectors of weight 6 hold 1-design.
- There are simple blocks and those with multiplicity 3.
- Note that $45 = \lambda_2 \binom{6}{2} / \binom{4}{2}$, whence $\lambda_2 = 18$.
- Thus $\lambda_1 = 18 \binom{5}{1} / \binom{3}{1} = 30$.

SSS using the $(6, 2^6)$ hexacode

Since the hexacode $\mathcal{G}_6$ is extremal even additive self-dual, the set of codewords of weight 4 forms a generalized 2-design of type 3 by Corollary[1].

It implies that

- $\mu(4, e1) = |\Gamma_{H_1}| = |\Gamma_{H_2}| = |\Gamma_{H_3}| = 10$
- $\mu(6, e1) = |\Gamma_{H_1}| = |\Gamma_{H_2}| = |\Gamma_{H_3}| = 6$,

where $e1$ denotes any vector of weight 1 in $GF(4)^n$

[1] Corollary
Let $\mathcal{C}$ be an extremal even additive self-dual code over $GF(4)$ of length $n = 6m$ (respectively, $n = 6m + 2$). Then the set of codewords of weight w in $\mathcal{C}$ with $A_w \neq 0$ forms a generalized 2-design (respectively, 1-design) of type 3.

The access structure for the hexacode $\mathcal{G}_6$

	$\Gamma_{H_1}(x_0 = 1)$	$\Gamma_{H_2}(y_0 = \omega)$	$\Gamma_{H_3}(z_0 = \overline{\omega})$
wt4	$\{2,3,4\}$	$\{2,3,4\}$	$\{2,3,4\}$
	$\{2,3,5\}$	$\{2,3,5\}$	$\{2,3,5\}$
	$\{2,3,6\}$	$\{2,3,6\}$	$\{2,3,6\}$
	$\{2,4,5\}$	$\{2,4,5\}$	$\{2,4,5\}$
	$\{2,4,6\}$	$\{2,4,6\}$	$\{2,4,6\}$
	$\{2,5,6\}$	$\{2,5,6\}$	$\{2,5,6\}$
	$\{3,4,5\}$	$\{3,4,5\}$	$\{3,4,5\}$
	$\{3,4,6\}$	$\{3,4,6\}$	$\{3,4,6\}$
	$\{3,5,6\}$	$\{3,5,6\}$	$\{3,5,6\}$
	$\{4,5,6\}$	$\{4,5,6\}$	$\{4,5,6\}$
# of wt 4	10	10	10
wt 6	$\{2,3,4,5,6\}$	$\{2,3,4,5,6\}$	$\{2,3,4,5,6\}$
# of wt 6	6	6	6
Total #	16	16	16

The size distribution of the access structure of the hexacode $\mathcal{G}_6$:

$$\sum_{i\in\{4,6\}} \sum_{j\in\{4,6\}} \mu(i, e1)\mu(j, e1)y^{(i-1,j-1)} = 100y^{(3,3)} + 60y^{(3,5)} + 60y^{(5,3)} + 36y^{(5,5)}.$$

- These pairs of groups comprise the 256 elements of the access structure.
- A group of size 6 does not c-cover any group of size 4.
- If a vector of weight 4 were c-covered by a vector of weight 6, then the sum of the two vectors will yield a weight 2 vector, which is a contradiction.
- Thus 256 pairs of supports are in the minimal access structure.

Theorem 8.
In SSS produced from the hexacode we have the following:

- *The access structure consists of 100 pairs of groups of size (4,4), 60 pairs of groups of size (4,6), 60 pairs of groups of size (4,6), 36 pairs of groups of size (6,6).*
- *This access structure becomes the minimal access structure.*
- *No group of size less than 3 can be used in recovering the secret.*

SSS based on an extremal additive even self-dual $(12, 2^{12}, 6)$ dodecacode QC_12

The dodecacode QC_12 has the following generator matrix.

$$\begin{bmatrix}
0&0&0&0&0&0&1&1&1&1&1&1\\
0&0&0&0&0&0&\omega&\omega&\omega&\omega&\omega&\omega\\
1&1&1&1&1&1&0&0&0&0&0&0\\
\omega&\omega&\omega&\omega&\omega&\omega&0&0&0&0&0&0\\
0&0&0&1&\omega&\overline{\omega}&0&0&0&1&\omega&\overline{\omega}\\
0&0&0&\omega&\overline{\omega}&1&0&0&0&\omega&\overline{\omega}&1\\
1&\overline{\omega}&\omega&0&0&0&1&\overline{\omega}&\omega&0&0&0\\
\omega&1&\overline{\omega}&0&0&0&\omega&1&\overline{\omega}&0&0&0\\
0&0&0&1&\overline{\omega}&\omega&\omega&\overline{\omega}&1&0&0&0\\
0&0&0&\omega&1&\overline{\omega}&1&\omega&\overline{\omega}&0&0&0\\
1&\omega&\overline{\omega}&0&0&0&0&0&0&\overline{\omega}&\omega&1\\
\overline{\omega}&1&\omega&0&0&0&0&0&0&1&\overline{\omega}&\omega
\end{bmatrix}$$

SSS based on the dodecacode QC_12

The weight distribution of the dodecacode QC_12 is :

$$1 + 396y^6 + 1485y^8 + 1980y^{10} + 234y^{12}.$$

We use the generalized t-design to determine the size distribution of the access structure for SSS based on the dodecacode QC_12.

- $\mu(6, e1) = |\Gamma_{H_1}| = |\Gamma_{H_2}| = |\Gamma_{H_3}| = 66$ for weight 6 codewords.
- For weight 8 codewords with $\lambda_5 = 105$, $\lambda_1 = 105\binom{11}{4}/\binom{7}{4} = 990$.
- $\mu(8, e1) = |\Gamma_{H_1}| = |\Gamma_{H_2}| = |\Gamma_{H_3}| = 330$.
- For weight 10 with $\lambda_5 = 630$, $\lambda_1 = 630\binom{11}{4}/\binom{9}{4} = 1650$.
- $\mu(10, e1) = |\Gamma_{H_1}| = |\Gamma_{H_2}| = |\Gamma_{H_3}| = 550$.
- For weight 12 with $\lambda_5 = 234$, $\lambda_1 = 234\binom{11}{4}/\binom{11}{4} = 234$.
- $\mu(12, e1) = |\Gamma_{H_1}| = |\Gamma_{H_2}| = |\Gamma_{H_3}| = 78$.

SSS based on the dodecacode QC_12

The size distribution of the access structure of the dodecacode QC_12 is

$$\sum_{i\in\{6,8,10,12\}} \sum_{j\in\{6,8,10,12\}} \mu(i,e1)\mu(j,e1)y^{(i-1,j-1)}$$
$$= 4356y^{(5,5)} + 21780y^{(5,7)} + 36300y^{(5,9)} + y^{(5,11)} + 21780y^{(7,5)} + 108900y^{(7,7)}$$
$$+ 181500y^{(7,9)} + 25740y^{(7,11)} + 36300y^{(9,5)} + 181500y^{(9,7)} + 302500y^{(9,9)}$$
$$+ 42900y^{(9,11)} + 5148y^{(11,5)} + 25740y^{(11,7)} + 42900y^{(11,9)} + 6084y^{(11,11)}. \qquad (2)$$

SSS based on the dodecacode QC_12

Theorem 9.
In SSS produced from the dodecacode QC_12 we have the following:

- *The access structure consists of the pairs of groups as in Equation (2).*
- *All the pairs of groups with the sizes $\in \{5,7,9\}$ are contained in the minimal access structure.*
- *No group of size less than 5 can be used in recovering the secret.*

Conclusion

- In this talk, we have introduced secret sharing schemes based on linear codes and generalized them to based on additive codes.
- To construct SSS based on additive codes over $GF(4)$, we used
  - Assmus-Mattson theorem for additive code over $GF(4)$
  - generalized t-designs.
- Using these two theorems, SSS based on additive codes can be constructed in a different way, generalizing SSS based on linear codes.

References

[1] E. F. Assmus, H. F. Mattson, *New 5-designs*, J. Combin. Theory, 6 (1969), 122-151.

[2] G. R. Blakley, *Safeguarding cryptographic keys*, American Federation of Information Processing Societies, National Computer Conference, (1979), 313-317.

[3] F. Carreras, A. Magaña, C. Munuera, *The accessibility of an access structure*, RAIRO-Theoretical Informatics and Applications 40.04, 559-567 (2006)

[4] S.T. Dougherty, S. Mesnager, and P Sol. *Secret-sharing schemes based on self-dual codes*, Information Theory Workshop, 2008. ITW'08. IEEE. IEEE, 2008.

[5] G. Höhn, *Self-dual codes over the Kleinian four group*,preprint (1996). An updated version in http://xxx.lanl.gov/pdf/math.co/0005266.

[6] J.-L. Kim, V. Pless, *Designs in additive codes over* $GF(4)$,Designs, Codes and Cryptography , vol. 30 (2003), Issue 2, 187-199

[7] J. L. Massey, *Minimal codewords and secret sharing*, Proceedings 6th Joint Swedish-Russian International Workshop on Information Theory, (1993), 276-279

[8] A. Shamir, *How to share a secret*, Communications of the ACM, 22 (1979), 612-613

THANK YOU FOR YOUR ATTENTION!

Access Structures of Weighted Threshold Ideal Secret Sharing Schemes

Arkadii SLINKO (Joint work with Ali HAMEED)

The University of Auckland

a.slinko@auckland.ac.nz

One of the most important challenges of the theory of secret sharing is to characterize access structures that can carry an ideal secret sharing scheme. Finding such a description appeared to be quite difficult. A result that generated much hope in this direction was the paper by Brickell and Davenport [2] who showed that all ideal secret sharing schemes can be obtained from matroids. Not all matroids, however, define ideal schemes so the problem was reduced to classifying those matroids that do. There was little further progress, if any, in this direction.

In his pioneering paper Shamir [5] introduced the notion of weighted threshold access structure. In such a structure every agent is given a weight and a coalition is authorised if their combined weight is at least a certain predefined threshold. Beimel, Tassa and Weinreb [1] and Farras and Padro [3] partially characterized access structures of ideal weighted threshold secret sharing schemes in terms of the operation of composition introduced by Shapley [4]. They proved that any weighted threshold ideal access structure is a composition of indecomposable ones. Farras and Padro gave a list of seven classes of access structures—one unipartite, three bipartite and three tripartite—to which all weighted threshold ideal indecomposable access structures may belong. Hameed and Slinko [6] determine exactly which access structures from those classes are indecomposable. They also determined which compositions of indecomposable weighted threshold access structures are again weighted threshold and obtained an if and only if characterization of ideal weighted threshold secret sharing schemes. They used game-theoretic techniques to achieve this. In my talk I will summarize the aforementioned developments and give a complete characterization of weighted threshold access structures.

References

[1] A. Beimel, T. Tassa, and E. Weinreb, Characterizing ideal weighted threshold secret sharing, SIAM J. Discrete Math. 22 (2008), no. 1, 360–397.

[2] E. Brickell and D. Davenport, On the classification of ideal secret sharing schemes, Journal of Cryptology 4 (1991), 123–134.

[3] O. Farràs and C. Padró, Ideal hierarchical secret sharing schemes, Theory of Cryptography (Daniele Micciancio, ed.), Lecture Notes in Computer Science, vol. 5978, Springer Berlin / Heidelberg, 2010, pp. 219–236.

[4] L. S. Shapley, Simple games: An outline of the descriptive theory, Behavioral Science 7 (1962), no. 1, 5966.

[5] A. Shamir, How to share a secret, Commun. ACM 22 (1979), 612613.

[6] A. Hameed, A. Slinko A characterisation of ideal weighted secret sharing schemes. Journal of Mathematical Cryptology 9 (4), 227–244.

Access Structures of Weighted Threshold Ideal Secret Sharing Schemes

Arkadii Slinko

(joint research with Ali Hameed)

Department of Mathematics
The University of Auckland

IMI Workshop: Secret Sharing for Dependability, Usability and Security of Network Storage and Its Mathematical Modeling

September 5–7, 2016, Kyushu University

Plan for the Talk

- The idea of Secret Sharing
- Access Structure
- Weighted and Hierarchical Access Structures
- Linear and Ideal Secret Sharing
- Composition of Access Structures
- Classification Weighted Ideal Secret Sharing Schemes

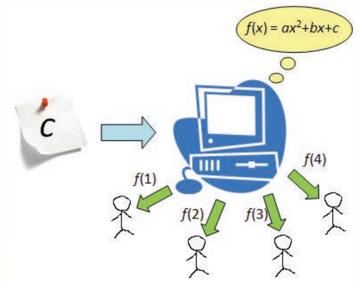
Shamir's idea of storing sensitive data

In 1979 Shamir suggested that for security valuable data can be stored on several servers so that if some servers are compromised the data cannot be stolen and can be recovered from the remaining servers.

He suggested the now classical k-out-of-n scheme based on Lagrange's interpolation.

Shamir's Scheme

Here is a pictorial interpretation of 3-out-of 4 scheme.



Any three would know the whole polynomial including c.

Attribute-based encryption

One of the chalanges is to be able to broadcast encrypted messages which will be meaningful only to a certain category of users defined by a set of attributes.

The idea of secret sharing

A secret sharing scheme 'divides' the secret S into 'shares' —one for each user—in such a way that:

- S can be easily reconstructed by any authorised coalition of users, but
- an unauthorised coalition of users cannot determine S.

The idea of secret sharing

A secret sharing scheme 'divides' the secret S into 'shares' —one for each user—in such a way that:

- S can be easily reconstructed by any authorised coalition of users, but
- an unauthorised coalition of users cannot determine S.

In the first example the 'users' were computers and in the second they were attributes.

6 / 31

The idea of secret sharing

A secret sharing scheme 'divides' the secret S into 'shares' —one for each user—in such a way that:

- S can be easily reconstructed by any authorised coalition of users, but
- an unauthorised coalition of users cannot determine S.

In the first example the 'users' were computers and in the second they were attributes.

Any secret sharing scheme has the following main ingredients:

- the access structure to the secret;
- mechanism of generating the shares;
- secret recovery algorithm.

6 / 31

Access structure

The set $U = \{1, 2, \ldots, n\}$ denotes the set of users.

7 / 31

Access structure

The set $U = \{1, 2, \ldots, n\}$ denotes the set of users.

Definition

An access structure is a pair $G = (U, W)$, where W is a subset of the power set 2^U, different from $\emptyset$, which satisfies the monotonicity condition:

if $X \in W$ and $X \subset Y \subseteq U$, then $Y \in W$.

Coalitions from W are called authorised. We also denote

$$L = 2^U \setminus W$$

and call coalitions from L unauthorised.

7 / 31

Access structure

The set $U = \{1, 2, \ldots, n\}$ denotes the set of users.

Definition

An access structure is a pair $G = (U, W)$, where W is a subset of the power set 2^U, different from $\emptyset$, which satisfies the monotonicity condition:

if $X \in W$ and $X \subset Y \subseteq U$, then $Y \in W$.

Coalitions from W are called authorised. We also denote

$$L = 2^U \setminus W$$

and call coalitions from L unauthorised.

The access structure is a simple game in the sense of von-Neumann and Morgenstern (1944).

7 / 31

Why do we need general access structures?

8 / 31

Why do we need general access structures?

- Participating agents might have different status, some more important then the others. The access structure must reflect this.

8 / 31

Why do we need general access structures?

- Participating agents might have different status, some more important then the others. The access structure must reflect this.
- In some scenarios like dynamic distributed encryption, or attribute-based encryption the sender should be allowed to choose a decryption policy for each ciphertext.

8 / 31

Why do we need general access structures?

- Participating agents might have different status, some more important then the others. The access structure must reflect this.
- In some scenarios like dynamic distributed encryption, or attribute-based encryption the sender should be allowed to choose a decryption policy for each ciphertext.
- This decryption policy can be seen as an access structure Γ over the set of all attributes.

8 / 31

Why do we need general access structures?

- Participating agents might have different status, some more important then the others. The access structure must reflect this.
- In some scenarios like dynamic distributed encryption, or attribute-based encryption the sender should be allowed to choose a decryption policy for each ciphertext.
- This decryption policy can be seen as an access structure Γ over the set of all attributes.
- Since different attributes may have different significance, it is not reasonable to restrict the sender to the threshold access structures only.

Examples of access structures 1

Shamir (1979) suggested two types of structures:

Example (k-out-of-n structure)

$X \subseteq U$ is authorised iff $|X| \geq k$.

Examples of access structures 1

Shamir (1979) suggested two types of structures:

Example (k-out-of-n structure)

$X \subseteq U$ is authorised iff $|X| \geq k$.

Example (weighted threshold structure)

An access structure G is called a weighted threshold structure if there exists a weight function $w\colon U \to \mathcal{R}^+$, where $\mathcal{R}^+$ is the set of all non-negative reals, and a real number q, called the quota, such that

$$X \in W \iff \sum_{i \in X} w_i \geq q.$$

We also call $[q; w_1, \ldots, w_n]$ as a weighted representation for G.

Examples of access structures 2

Suppose now $U = U_1 \cup U_2$ with $|U_1| = n_1$, $|U_2| = n_2$ and players within each part are equivalent. For a coalition X let $X_i = X \cap U_i$, $i \in \{1, 2\}$.

Example (hierarchical disjunctive structure, Simmons, 1990)

A hierarchical disjunctive structure $H_\exists(\mathbf{n}, \mathbf{k})$ with $\mathbf{n} = (n_1, n_2)$ and $\mathbf{k} = (k_1, k_2)$, $k_1 < k_2$, is defined by the set of authorised coalitions

$$W_\exists = \{X \subseteq U \mid (|X_1| \geq k_1) \vee (|X_1| + |X_2| \geq k_2)\},$$

where $1 \leq k_1 \leq n_1$ and $k_2 - k_1 < n_2$ (if these conditions are not satisfied all users becomes equivalent).

Examples of access structures 3

Suppose now $U = U_1 \cup U_2$ and players within each part are equivalent. For a coalition X let $X_i = X \cap U_i$, $i \in \{1, 2\}$.

Example (hierarchical conjunctive structure, Tassa, 2007)

A hierarchical conjunctive structure $H_\exists(\mathbf{n}, \mathbf{k})$ with $\mathbf{n} = (n_1, n_2)$ and $\mathbf{k} = (k_1, k_2)$, $k_1 < k_2$, is defined by the set of authorised coalitions

$$W_\forall = \{X \subseteq U \mid (|X_1| \geq k_1) \wedge (|X_1| + |X_2| \geq k_2)\},$$

where $1 \leq k_1 \leq n_1$ and $k_2 - k_1 < n_2$ (if these conditions are not satisfied all users becomes equivalent).

UN Security Council

The 15 member UN Security Council consists of five permanent and 10 non-permanent countries. A passage requires:

- approval of at least nine countries,
- subject to a veto by any one of the permanent members.

UN Security Council

The 15 member UN Security Council consists of five permanent and 10 non-permanent countries. A passage requires:

- approval of at least nine countries,
- subject to a veto by any one of the permanent members.

This is a conjunctive hierarchical game, it is also a weighted game with

$$[39; 7, 7, 7, 7, 7, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1].$$

12 / 31

Money Bank Transfert

If a significant sum of money is being transferred, an approval requires:

- signitures of two vice-presidents, or
- three senior tellers; or
- a vice-president and two senior tellers.

13 / 31

Money Bank Transfert

If a significant sum of money is being transferred, an approval requires:

- signitures of two vice-presidents, or
- three senior tellers; or
- a vice-president and two senior tellers.

This disjunctive hierarchical game is also weighted:

$$[6; 3, \ldots 3, 2, \ldots, 2].$$

13 / 31

Opening the vault

The secret combination opening the vault key must be distributed among bank employees. The bank policy requires the presence of three employees in opening the vault, but at least one of them must be a departmental manager.

14 / 31

Opening the vault

The secret combination opening the vault key must be distributed among bank employees. The bank policy requires the presence of three employees in opening the vault, but at least one of them must be a departmental manager.

Opening the vault game is not weighted:

$$\{m_1, t_1, t_2\} \cup \{m_2, t_3, t_4\} = \{m_1, m_2\} \cup \{t_1, t_2, t_3, t_4\}$$

is a trading transform, which is a certificate of nonweightedness.

14 / 31

Linear secret sharing

Let $\mathbf{h}_0, \mathbf{h}_1, \ldots, \mathbf{h}_n \in F^k$ be row vectors with coefficients in a finite field F. Let

$$H = \begin{bmatrix} \mathbf{h}_0 \\ \mathbf{h}_1 \\ \vdots \\ \mathbf{h}_n \end{bmatrix}$$

be an $(n+1) \times k$ matrix. We can define the access structure for $P = \{1, 2, \ldots, n\}$ related to this sequence of vectors as

$$W_H = \{\{\, i_1, i_2, \ldots i_k\} \mid \mathbf{h}_0 \in \mathrm{span}(\mathbf{h}_{i_1}, \mathbf{h}_{i_2}, \ldots, \mathbf{h}_{i_k})\}.$$

15 / 31

Linear secret sharing

Let $\mathbf{h}_0, \mathbf{h}_1, \ldots, \mathbf{h}_n \in F^k$ be row vectors with coefficients in a finite field F. Let

$$H = \begin{bmatrix} \mathbf{h}_0 \\ \mathbf{h}_1 \\ \vdots \\ \mathbf{h}_n \end{bmatrix}$$

be an $(n+1) \times k$ matrix. We can define the access structure for $P = \{1, 2, \ldots, n\}$ related to this sequence of vectors as

$$W_H = \{\{\, i_1, i_2, \ldots i_k\} \mid \mathbf{h}_0 \in \mathrm{span}(\mathbf{h}_{i_1}, \mathbf{h}_{i_2}, \ldots, \mathbf{h}_{i_k})\}.$$

Both types of hierarchical structures are linear but weighted threshold structures are seldom linear.

15 / 31

Linear secret sharing

The shares for the linear schemes are generated as follows:

$$\begin{bmatrix} s_0 \\ s_1 \\ \vdots \\ s_n \end{bmatrix} = H \begin{bmatrix} t_1 \\ t_2 \\ \vdots \\ t_k \end{bmatrix}$$

where $t_1, \ldots, t_k$ are randomly generated. Then if $\{\, i_1, i_2, \ldots i_k\}$ is authorised and

$$\mathbf{h}_0 = a_1 \mathbf{h}_{i_1} + a_2 \mathbf{h}_{i_2} + \ldots + a_k \mathbf{h}_{i_k},$$

then

$$s_0 = a_1 s_{i_1} + a_2 s_{i_2} + \ldots + a_k s_{i_k}.$$

16 / 31

Ideal secret sharing

Linear schemes have two important properties:

- they are secure, i.e., unauthorised coalitions get no information about the secret;
- the length of any share (in bits) is the same as the length of the secret.

Such schemes are called ideal.

17 / 31

Ideal secret sharing

Linear schemes have two important properties:

- they are secure, i.e., unauthorised coalitions get no information about the secret;
- the length of any share (in bits) is the same as the length of the secret.

Such schemes are called ideal.

Some very simple access structures, like $\{\{1,2\},\{2,3\},\{3,4\}\}$, are not linear and not even ideal.

17 / 31

Ideal secret sharing

Linear schemes have two important properties:

- they are secure, i.e., unauthorised coalitions get no information about the secret;
- the length of any share (in bits) is the same as the length of the secret.

Such schemes are called ideal.

Some very simple access structures, like $\{\{1,2\},\{2,3\},\{3,4\}\}$, are not linear and not even ideal.

Classification of access structures that can carry an ideal secret sharing scheme is an important problem.

17 / 31

Non-ideal secret sharing

It is believed that secure schemes on some access structures may need very long shares.

18 / 31

Non-ideal secret sharing

It is believed that secure schemes on some access structures may need very long shares.

Conjecture (Beimel, 2010)

There exists $\epsilon > 0$ such that for every integer n there is an access structure with n users for which every secret sharing scheme distributes shares of length $\ell 2^{\epsilon n}$, where ℓ is the length of the secret.

Non-ideal secret sharing

It is believed that secure schemes on some access structures may need very long shares.

Conjecture (Beimel, 2010)

There exists $\epsilon > 0$ such that for every integer n there is an access structure with n users for which every secret sharing scheme distributes shares of length $\ell 2^{\epsilon n}$, where ℓ is the length of the secret.

Csirmaz (1994) proved that for sharing ℓ-bit secret shares of the length $\Omega(\ell n/\log n)$ may be necessary.

How to describe ideal access structures?

Characterising access structures that can carry an ideal secret sharing scheme (ideal structures) is an important problem in secret sharing.

We need ideas from algebra and game theory to start doing this.

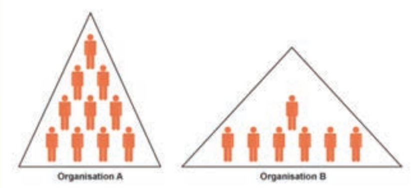
In this talk I will give a description of weighted threshold ideal access structures.

This is a combined effort of Beimel-Tassa-Weinreb (2008), Farras-Padro (2010) and Hameed-Slinko (2016).

Composition of games (example)

It expresses the idea that a collective member may be a player in a larger game.

We can take a unanimity game as a higher level game, i.e., both organisations must approve the decision.

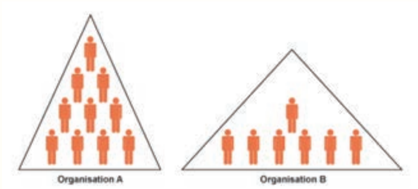


Within each organisation we may its own rule of approval. This is how the European Union works.

Composition of games (example)

It expresses the idea that a collective member may be a player in a larger game.

We can take a unanimity game as a higher level game, i.e., both organisations must approve the decision.



Within each organisation we may its own rule of approval. This is how the European Union works.

Introduced by Shapley (1962), rediscovered by Martin (1993).

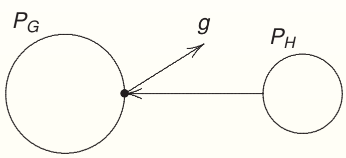
Composition of simple games (formal definition)

Definition

Let $G = (P_G, W_G)$ and $H = (P_H, W_H)$ be two games defined on disjoint sets of players and $g \in P_G$. We define the composition game $C = G \circ_g H$ by defining $P_C = (P_G \setminus \{g\}) \cup P_H$ and

$$W_C = \{X \subseteq P_C \mid X_G \in W_G \text{ or } X_G \cup \{g\} \in W_G \text{ and } X_H \in W_H\},$$

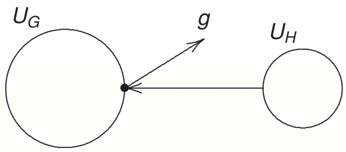
where $X_G = X \cap P_G$ and $X_H = X \cap P_H$.



Composition of access structures

Theorem (Beimel-Tassa-Weinreb, 2008)

Composition $C = G \circ_g H$ of any two access structures is ideal if and only if g is not a dummy in G and G and H are ideal.



Proof (one way): If G and h are ideal and s is the secret, then: distribute shares in G, then take the share of user g and make it the secret for H and distribute shares in H accordingly.

Associativity of composition

Proposition

Let G, H, K be three games defined on the disjoint set of players and $g \in P_G$, $h \in P_H$. Then

$$(G \circ_g H) \circ_h K \cong G \circ_g (H \circ_h K),$$

that is the two products are isomorphic.

Associativity of composition

Proposition

Let G, H, K be three games defined on the disjoint set of players and $g \in P_G$, $h \in P_H$. Then

$$(G \circ_g H) \circ_h K \cong G \circ_g (H \circ_h K),$$

that is the two products are isomorphic.

Definition

A game G is said to be indecomposable if there does not exist two games H and K and $h \in P_H$ such that $\min(|H|, |K|) > 1$ and $G \cong H \circ_h K$.

Associativity of composition

Proposition

Let G, H, K be three games defined on the disjoint set of players and $g \in P_G$, $h \in P_H$. Then

$$(G \circ_g H) \circ_h K \cong G \circ_g (H \circ_h K),$$

that is the two products are isomorphic.

Definition

A game G is said to be indecomposable if there does not exist two games H and K and $h \in P_H$ such that $\min(|H|, |K|) > 1$ and $G \cong H \circ_h K$.

Theorem (Freeman-Slinko, 2013)

Every weighted simple game can be expressed uniquely as a product of indecomposable weighted simple games.

First classification theorem

Beimel et al (2008) idea was that it is enough to characterise weighted ideal indecomposable access structures.

Definition

We call an access structure *m*-partite if the set of users can be split into m classes of equivalent users.

Theorem (Beimel et al, 2008)

Any weighted threshold ideal access structure is either 1*-partite or* 2*-partite or* 3*-partite.*

1-partite indecomposable access structures

Since all n players are equivalent, there exist k such that it takes k or more players to win. Such a game is called k-out-of-n game, denoted $H_{n,k}$.

1-partite indecomposable access structures

Since all n players are equivalent, there exist k such that it takes k or more players to win. Such a game is called k-out-of-n game, denoted $H_{n,k}$.

H: $H_{n,k}$ is indecomposable for $1 < k < n$.

1-partite indecomposable access structures

Since all n players are equivalent, there exist k such that it takes k or more players to win. Such a game is called k-out-of-n game, denoted $H_{n,k}$.

H: $H_{n,k}$ is indecomposable for $1 < k < n$.

The game $U_n = H_{n,n}$ is special and is called the unanimity game on n players. Only U_2 is indecomposable.

1-partite indecomposable access structures

Since all n players are equivalent, there exist k such that it takes k or more players to win. Such a game is called k-out-of-n game, denoted $H_{n,k}$.

H: $H_{n,k}$ is indecomposable for $1 < k < n$.

The game $U_n = H_{n,n}$ is special and is called the unanimity game on n players. Only U_2 is indecomposable.

The game $A = H_{n,1}$ does not have a name in the literature. We will call it anti-unanimity game. Only A_2 is indecomposable.

Bipartite weighted indecomposable access structures

Farras and Padro (2010) classified these:

B1: The family of bipartite conjunctive hierarchical games $H_\forall(\mathbf{n},\mathbf{k})$ with $\mathbf{n} = (n_1, n_2)$ and $\mathbf{k} = (k_1, k_2)$ such that $1 \leq k_1 < n_1$ and $k_2 - k_1 = n_2 - 1 \geq 1$.

Bipartite weighted indecomposable access structures

Farras and Padro (2010) classified these:

B1: The family of bipartite conjunctive hierarchical games $H_\forall(\mathbf{n},\mathbf{k})$ with $\mathbf{n} = (n_1, n_2)$ and $\mathbf{k} = (k_1, k_2)$ such that $1 \leq k_1 < n_1$ and $k_2 - k_1 = n_2 - 1 \geq 1$.

B2: The family of bipartite disjunctive hierarchical games $H_\exists(\mathbf{n},\mathbf{k})$ with $\mathbf{n} = (n_1, n_2)$ and $\mathbf{k} = (k, k+1)$ with $2 \leq k \leq n_1$ and $n_2 \geq 3$.

Bipartite weighted indecomposable access structures

Farras and Padro (2010) classified these:

B1: The family of bipartite conjunctive hierarchical games $H_\forall(\mathbf{n},\mathbf{k})$ with $\mathbf{n} = (n_1, n_2)$ and $\mathbf{k} = (k_1, k_2)$ such that $1 \leq k_1 < n_1$ and $k_2 - k_1 = n_2 - 1 \geq 1$.

B2: The family of bipartite disjunctive hierarchical games $H_\exists(\mathbf{n},\mathbf{k})$ with $\mathbf{n} = (n_1, n_2)$ and $\mathbf{k} = (k, k+1)$ with $2 \leq k \leq n_1$ and $n_2 \geq 3$.

They also had a third type that appeared to be reducible.

Tripartite weighted indecomposable access structures

T_1: Let $\mathbf{n} = (n_1, n_2, n_3)$ and $\mathbf{k} = (k_1, k_2, k_3)$, where n_1, n_2, n_3 and k_1, k_2, k_3 are positive integers. The game $\Delta_1(\mathbf{n}, \mathbf{k})$ is defined on the multiset $P = U_1 \cup U_2 \cup U_3$ with the set of authorised coalitions $X \subseteq U$ satisfying

$$(|X_1| \geq k_1) \vee [(|X_1| + |X_2| \geq k_2) \wedge (|X_1| + |X_2| + |X_3| \geq k_3)].$$

T_2: The game $\Delta_2(\mathbf{n}, \mathbf{k})$ has authorised coalitions $X \subseteq U$ satisfying

$$(|X_1| + |X_2| \geq k_2) \vee [(|X_1| \geq k_1) \wedge (|X_1| + |X_2| + |X_3| \geq k_3)].$$

In both cases there are restrictions on $\mathbf{n}$ and $\mathbf{k}$ to prevent them to have dummies or become bipartite.

Second Classification Theorem

Theorem (Farras-Padro, 2010)

Let U be a set of users and Γ be an ideal weighted threshold access structure. Then one of the following three conditions holds:

Second Classification Theorem

Theorem (Farras-Padro, 2010)

Let U be a set of users and Γ be an ideal weighted threshold access structure. Then one of the following three conditions holds:

1. *Γ is onepartite, i.e., k-out-of-n access structure;*

Second Classification Theorem

Theorem (Farras-Padro, 2010)

Let U be a set of users and Γ be an ideal weighted threshold access structure. Then one of the following three conditions holds:

1. *Γ is onepartite, i.e., k-out-of-n access structure;*
2. *Γ is bipartite of types $\mathbf{B}_1$, $\mathbf{B}_2$, ($\mathbf{B}_3$);*

Second Classification Theorem

Theorem (Farras-Padro, 2010)

Let U be a set of users and Γ be an ideal weighted threshold access structure. Then one of the following three conditions holds:

1. *Γ is onepartite, i.e., k-out-of-n access structure;*
2. *Γ is bipartite of types $\mathbf{B}_1$, $\mathbf{B}_2$, ($\mathbf{B}_3$);*
3. *Γ is tripartite of types $\mathbf{T}_1$, $\mathbf{T}_2$, ($\mathbf{T}_3$);*

Second Classification Theorem

Theorem (Farras-Padro, 2010)

Let U be a set of users and Γ be an ideal weighted threshold access structure. Then one of the following three conditions holds:

1. *Γ is onepartite, i.e., k-out-of-n access structure;*
2. *Γ is bipartite of types $\mathbf{B}_1$, $\mathbf{B}_2$, ($\mathbf{B}_3$);*
3. *Γ is tripartite of types $\mathbf{T}_1$, $\mathbf{T}_2$, ($\mathbf{T}_3$);*
4. *Γ is a composition of Γ_1 and Γ_2, where Γ_1 and Γ_2 are ideal weighted access structures defined over sets of users smaller than U.*

Second Classification Theorem

Theorem (Farras-Padro, 2010)

Let U be a set of users and Γ be an ideal weighted threshold access structure. Then one of the following three conditions holds:

1. *Γ is onepartite, i.e., k-out-of-n access structure;*
2. *Γ is bipartite of types $\mathbf{B}_1$, $\mathbf{B}_2$, ($\mathbf{B}_3$);*
3. *Γ is tripartite of types $\mathbf{T}_1$, $\mathbf{T}_2$, ($\mathbf{T}_3$);*
4. *Γ is a composition of Γ_1 and Γ_2, where Γ_1 and Γ_2 are ideal weighted access structures defined over sets of users smaller than U.*

Moreover, there exists a linear ideal secret sharing scheme that realises Γ.

Second Classification Theorem

Theorem (Farras-Padro, 2010)

Let U be a set of users and Γ be an ideal weighted threshold access structure. Then one of the following three conditions holds:

1. *Γ is onepartite, i.e., k-out-of-n access structure;*
2. *Γ is bipartite of types $\mathbf{B}_1$, $\mathbf{B}_2$, ($\mathbf{B}_3$);*
3. *Γ is tripartite of types $\mathbf{T}_1$, $\mathbf{T}_2$, ($\mathbf{T}_3$);*
4. *Γ is a composition of Γ_1 and Γ_2, where Γ_1 and Γ_2 are ideal weighted access structures defined over sets of users smaller than U.*

Moreover, there exists a linear ideal secret sharing scheme that realises Γ.

Comment: Those in brackets later appeared to be reducible.

Counterexample

Example (Hameed, Slinko, 2015)

Let G be defined on $P_G = A \cup B$ and H on $P_H = C$ with $A = \{a_1, a_2\}$, $B = \{b_1, b_2, b_3\}$, $C = \{c_1, c_2, c_3, c_4\}$ and weighted representations

$$[7; 3, 3, 2, 2, 2] \quad \text{and} \quad H = [2; 1, 1, 1, 1],$$

respectively. Let $g = b_3$ be the player to be replaced with H. Then we have the certificate of nonweightedness for $G \circ_g H$:

$$\{a_1, b_1, c_1, c_2\} \cup \{a_2, b_2, c_3, c_4\} = \{a_1, a_2, c_1\} \cup \{b_1, b_2, c_2, c_3, c_4\},$$

i.e., the union of two authorised coalitions is equal to the union of two unauthorised (which cannot happen in a weighted case).

The Main Theorem

Theorem (Hameed-Slinko, 2015)

An access structure G with no dummies is ideal and weighted if and only if it is a composition

$$G = H_1 \circ \cdots \circ H_s \circ I \circ A_1 \circ \cdots \circ A_t,$$

where H_i is a k_i-out-of-n_i access structure for each $i = 1, 2, \ldots, s$, A_j is an indecomposable access structure of type **A** *for each $j = 1, 2, \ldots, t$, and I is an indecomposable access structure of types* $\mathbf{B_1}$, $\mathbf{B_2}$, $\mathbf{T_1}$, $\mathbf{T_2}$.

In this composition we may have $s = 0$, $t = 0$ and I also may be absent. Moreover, we can have $t > 0$ only if I is of type $\mathbf{B_2}$.

Our paper is published in:

> *A Characterization of Ideal Weighted Secret Sharing Schemes. Journal of Mathematical Cryptology, 9(4): 227-244 (2015).*

Any comments will be greatly appreciated.

Our paper is published in:

> *A Characterization of Ideal Weighted Secret Sharing Schemes. Journal of Mathematical Cryptology, 9(4): 227-244 (2015).*

Any comments will be greatly appreciated.

Thank you for your attention!

Toward Highly Secure Metering Data Management in the Smart Grid

Yuichi KOMANO, Shinji YAMANAKA and Satoshi ITO

TOSHIBA Corporation

yuichi1.komano@toshiba.co.jp

In the smart grid [1], several information systems collaborate to efficiently manage electricity. Smart meter is an equipment located in each home (and office) to monitor the electric power usage of each home and to periodically send the metering data to upper stream. The metering data is transferred from the smart meter to metering data management (MDM) through some communication channel. There are two well-known use cases: (i) MDM system statistically estimates the total power usage of some area in each time to control the power generation, and (ii) MDM system statistically summarizes the metering data through some time period in each home to charge users electricity bills, respectively.

MDM system might store huge amount of metering data for lots of sites (such as home and office) and for time period (such as for several years). As shown in NIST IR 7628 [2], such metering data include users' privacy information, such as life cycle and electric equipment held in the home. Once the stored data in MDM system is leaked, it causes a big security and privacy issue.

Our motivation is to propose a concept of highly secure MDM system. We believe that the secret sharing is one of promising solutions for this purpose. We assume that each metering data is divided into multiple shares and several MDM servers store one of shares, respectively. Under this assumption, even though some of servers leak stored data (share) by malicious attack or human mistake, the corresponding metering data still remains secret and no security nor privacy issue happens. In this scenario, as shown (i) and (ii) above, MDM system should two types of statistical computations.

In this talk, we give a system model of such MDM system and its requirements. Then, we show a construction as an example based on [3]. Of course, if we combine a multiparty computation protocol with secret sharing, we can achieve such system; however, for simplicity (and to reduce implementation costs), we give an example based on modular addition and homomorphic message authentication.

References

[1] NIST, "NIST Framework and Roadmap for Smart Grid Interoperability Standards, Release 3.0", SP 1108, 2014

[2] NIST, "Guidelines for Smart Grid Cybersecurity", IR 7628 Rev.1, 2014

[3] Shinji YAMANAKA, Yuichi KOMANO and Satoshi ITO, "A Privacy Protection Scheme for Smart Grid using Secret Sharing Scheme", SCIS 2013 (in Japanese)

TOSHIBA
Leading Innovation >>>

Toward highly secure metering data management in the smart grid

Yuichi Komano, Shinji Yamanaka, Satoshi Ito
(Toshiba corporation)

5-7, September, 2016
IMI Joint Research Project in 2016

Profile

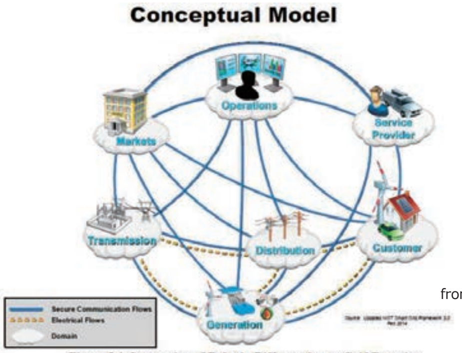
- **Personal information**
  - Yuichi Komano, Dr. Science (Waseda univ., 2007)
    - 2003.3 Waseda univ., Master of Science (Mathematics)
    - 2003.4 Toshiba ((present) Senior Research Scientist)
- **My interest**
  - Cryptography and information security
    - Public key cryptography and provable security
    - Applied cryptography (smart grid, vehicle comm., etc.)
    - Secure implementation (side channel attack and counter)
- **Volunteer work (recent)**
  - IEICE Trans. A, area editor in security (2014-2016)
  - IEICE ISEC Tech. Committee, Secretary (2013-)
  - CARDIS PC (2015,2016)

Contents

- **Introduction on Smart Grid (SG)**
- **Security and Privacy Issues on SG**
- **Our Proposal for Highly Secure MDM**
- **Concluding Remarks**

Smart Grid (SG)

- SG makes power supply efficient by IT
- NIST divides it 7 domains:

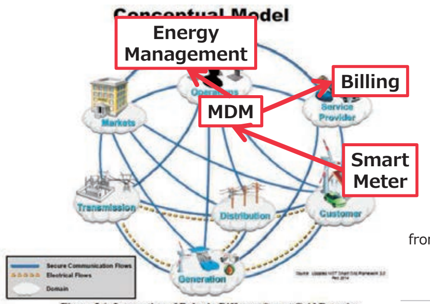


from NIST SP1108r3

Figure 5-1. Interaction of Roles in Different Smart Grid Domains through Secure Communication

Meter Data Management (MDM)

- Smart meter collects metering data periodically
- MDM collects the data
  - (i) to manage power supply in each area, and
  - (ii) to charge customers their electricity bills



from NIST SP1108r3

Figure 5-1. Interaction of Roles in Different Smart Grid Domains through Secure Communication

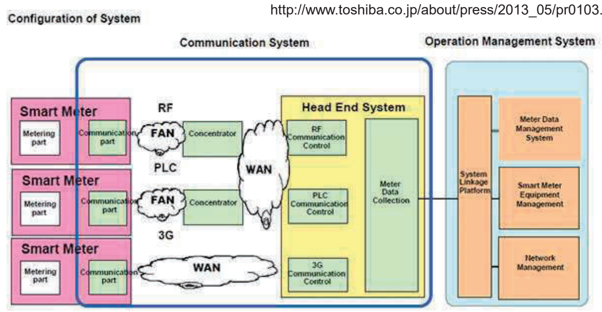
Merits of Meter Data Management

- MDM makes power supply efficient and robust
  - The amount and time for power generation is well-managed
- MDM enhances electricity deregulation
  - Precise data in MDM enhances various business model
  - MDM can be infrastructure to share data among companies

Toshiba's activities on SG

- **TEPCO's smart meter communication systems**
  - including 27millions smart meters
  - from 2013 (up) to 2023

http://www.toshiba.co.jp/about/press/2013_05/pr0103.htm



Security and Privacy Issues in SG

- **Security is mandatory everywhere in SG**

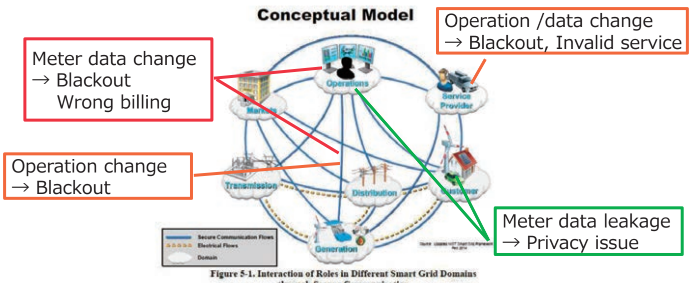


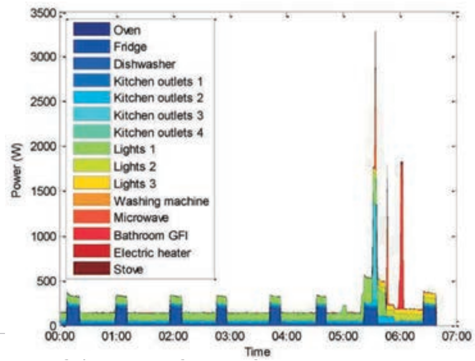
Figure 5-1. Interaction of Roles in Different Smart Grid Domains through Secure Communication

from NIST SP1108r3

- **In this talk**
  - We focus on the security at MDM

Privacy Information in Meter Data

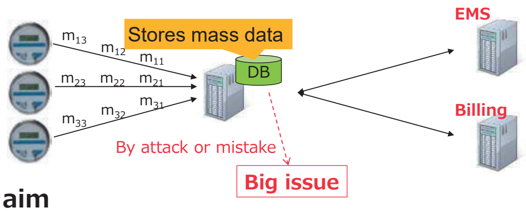
- **Meter data leaks life style of customer**
  - what kind of equipment, eg. electronic vehicle
  - when customers leave home



from NIST IR7628r1

Motivation

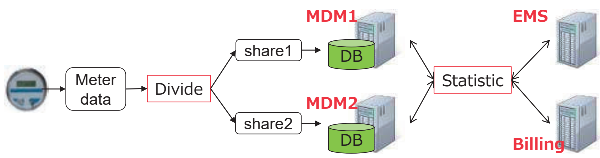
- **Our concern**
  - MDM stores huge amount of data
    - For lots of customers and for long time period
  - Leakage from MDM causes big security and privacy issue



- **Our aim**
  - Propose privacy enhanced MDM
    - Not only clears security and privacy issues
    - But enables applications for energy management and billing

Our Solution ~ Privacy Enhanced MDM

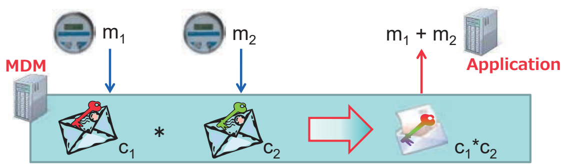
- **Strategy**
  - Use secret sharing to protect meter data (into “shares”)
  - Theoretically, MPC enables statistically operation on “shares”



- **Our construction**
  - Data is “additively” divided into shares
    - “Additive division” enables operations on divided data
    - “Additive division” and operations are easier than MPC

Preliminaries

- **Homomorphic encryption**
  - Enables mathematical operation on cipher w/o decryption
    - $\mathrm{Enc}(key, m_1)*\mathrm{Enc}(key, m_2) = \mathrm{Enc}(key, m_1+m_2)$

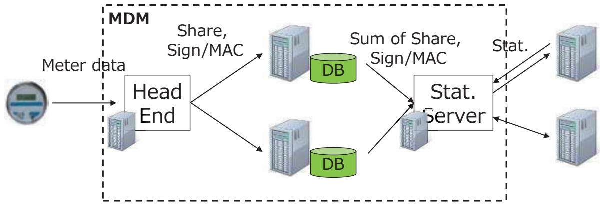


- **Homomorphic Signature/MAC***
  - Enables aggregate Signature/MACs w/o mac key
    - $\mathrm{Sign}(key, m_1)*\mathrm{Sign}(key, m_2) = \mathrm{Sign}(key, m_1+m_2)$

*Message authentication code

Our System Model

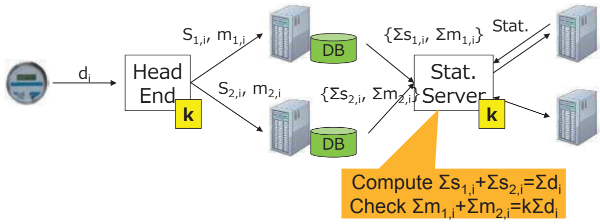
- **Divide meter data into share at head end**
- **Head end also generates Signature/MACs**
  - Shares and Sign/MACs have homomorphic property



- **Requirements**
  - From shares, no information about meter data leaks
  - Data change between head-end and stat. server is detected

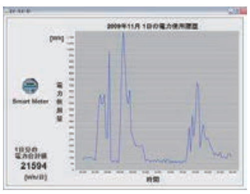
Concrete Example

- **To reduce comp. cost, we use simple solutions**
  - Apply simple (2,2)-SS (mod P) to hide data
    - Divide meter data d into shares s_1,s_2, st, $s_1+s_2 \equiv d$ (mod P)
  - Modify SS with secret key k to authenticate data
    - Compute MACs m_1,m_2, st, $m_1+m_2 \equiv dk$ (mod P)
      - Eg. m_1 is random in (0,P), $m_2=dk - m_1$ mod P

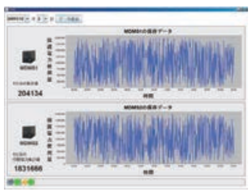


Demo for Privacy Enhancement (1)

- **We developed demo based on previous example**



Divide



Meter data for 1 day (simulation)

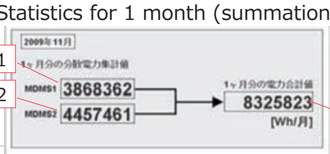
Shares for MDM1 and MDM2

Statistics for 1 month (summation)

Stat.

Sum of share1

Sum of share2



Total usage

Demo for Privacy Enhancement (2)

- **We had experimental test for NEDO pj.**
  - Simulate 16,000 smart meters (SMs) with 8 PCs
  - Simulate 2 MDMs
  - Measure running time for collecting data from SMs to MDMs
  - Confirm shares in MDM are independent from meter data

Challenges

- **Robustness and ease of maintenance**
  - Retrieval from errors and faults, Redundancy
- **Tradeoff between "practical" and "highly secure"**
  - Data size, computation cost, implementation cost

Concluding Remarks

- **Introduce SG and its security & privacy issues**
- **Discuss highly secure MDM system using SS**

- **Applications of SG will be developed in future**
  - Security mechanism for sensitive data may be required

TOSHIBA

Leading Innovation >>>

Homomorphic authentication schemes for network coding

Chi Cheng (Joint work with Jemin Lee, Tao Jiang, and Tsuyoshi Takagi)

Kyushu University

chengchizz@gmail.com

Ever since the pioneer work of Ahlswede et al. [1], the introduction of network coding has sparked a flurry of research interest in designing more efficient network systems. Different from the traditional store and forward or routing mechanisms, network coding enables intermediate nodes to encode the received packets to generate output packets. However, the paradigm shift in data transmission also makes the system with network coding seriously vulnerable to pollution attacks.

In this talk, we first give a brief introduction to homomorphic authentication schemes for network coding. Then, we show that there exists an efficient multi-generation pollution attack on two recent homomorphic authentication schemes named homomorphic subspace signature (HSS) [2]and key predistribution-based tag encoding (KEPTE) [3]. Specifically, we show that by using packets and their signatures of different generations, the adversary can create invalid packets and their corresponding signatures that pass the verification of HSS and KEPTE at intermediate nodes as well as at the destination nodes. After giving a more generic attack, we analyze the cause of the proposed attack. We then propose the improved key distribution schemes for HSS and KEPTE, respectively. Next, we show that the proposed key distribution schemes can combat against the proposed multi-generation pollution attacks. Finally, we analyze the computation and communication costs of the proposed key distribution schemes for HSS and KEPTE, and by implementing experiments, we demonstrate that the proposed schemes add acceptable burden on the system.

References

[1] R. Ahlswede, N. Cai, S.-Y. R. Li and R. W. Yeung, “Network information flow,” *IEEE Transactions on Information Theory*, vol. 46, no. 4, pp. 1204-1216, Jul. 2000.

[2] P. Zhang, Y. Jiang, C. Lin, H. Yao, A. Wasef, and X. Shen, “Padding for orthogonality: Efficient subspace authentication for network coding,” in *Proceedings of the IEEE International Conference on Computer Communications (INFOCOM)*, pp. 1026-1034, Apr. 2011.

[3] X. Wu, Y. Xu, C. Yuen, and L. Xiang, “A Tag Encoding Scheme against Pollution Attack to Linear Network Coding,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 1, pp. 33-42, Jan. 2014.

Homomorphic Authentication Schemes for Network Coding

Chi Cheng

Institute of Mathematics for Industry, Kyushu University

chengchi@math.kyushu-u.ac.jp

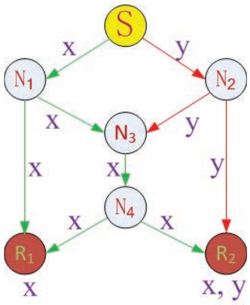
Overview

Routing and Network Coding

Routing and Network Coding

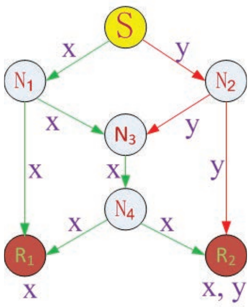
Routing and Network Coding

- Routing



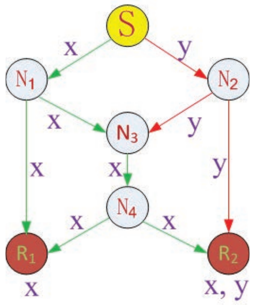
Routing and Network Coding

- Routing

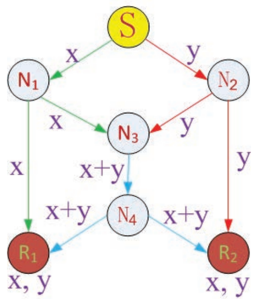


Routing and Network Coding

- Routing



- Network Coding



Random Linear Network Coding

- At Source $\mathcal{S}$: A file is divided into generations (subfiles), and each generation consists of $\bar{u}_1, \bar{u}_2, \ldots, \bar{u}_m \in \mathbb{F}_q^n$

Random Linear Network Coding

- At Source $\mathcal{S}$: A file is divided into generations (subfiles), and each generation consists of $\bar{u}_1, \bar{u}_2, \ldots, \bar{u}_m \in \mathbb{F}_q^n$
- Generate augmented vectors and forward them

$$u_i = (\overbrace{\underbrace{0, \ldots, 0, 1}_{i},\ 0, \ldots, 0}^{m}, \bar{u}_i) \in \mathbb{F}_q^{n+m}$$

Random Linear Network Coding

- At Source $\mathcal{S}$: A file is divided into generations (subfiles), and each generation consists of $\bar{u}_1, \bar{u}_2, \ldots, \bar{u}_m \in \mathbb{F}_q^n$
- Generate augmented vectors and forward them

$$u_i = (\overbrace{\underbrace{0, \ldots, 0, 1}_{i}, 0, \ldots, 0}^{m}, \bar{u}_i) \in \mathbb{F}_q^{n+m}$$

- For example,

$$2u_1 + 3u_2 = (\overbrace{2, 3, \ldots, 0, \ldots, 0}^{m}, 2\bar{u}_1 + 3\bar{u}_2)$$

The first m bits contain the coefficients used in combing the vectors, which are called the Global Encoding Vector.

Decoding

- After receiving $\{w_i = (v_i, \bar{w}_i)\}_{i=1}^m$ in which $v_1, v_2, \ldots, v_m$ are linearly independent

Decoding

- After receiving $\{w_i = (v_i, \bar{w}_i)\}_{i=1}^m$ in which $v_1, v_2, \ldots, v_m$ are linearly independent
  - Set

$$\bar{U} = \begin{pmatrix} \bar{u}_1 \\ \bar{u}_2 \\ \cdots \\ \bar{u}_m \end{pmatrix}, V = \begin{pmatrix} v_1 \\ v_2 \\ \cdots \\ v_m \end{pmatrix}, \text{and } \bar{W} = \begin{pmatrix} \bar{w}_1 \\ \bar{w}_2 \\ \cdots \\ \bar{w}_m \end{pmatrix}.$$

Decoding

- After receiving $\{w_i = (v_i, \bar{w}_i)\}_{i=1}^m$ in which $v_1, v_2, \ldots, v_m$ are linearly independent
  - Set

$$\bar{U} = \begin{pmatrix} \bar{u}_1 \\ \bar{u}_2 \\ \cdots \\ \bar{u}_m \end{pmatrix}, V = \begin{pmatrix} v_1 \\ v_2 \\ \cdots \\ v_m \end{pmatrix}, \text{and } \bar{W} = \begin{pmatrix} \bar{w}_1 \\ \bar{w}_2 \\ \cdots \\ \bar{w}_m \end{pmatrix}.$$

  - The original messages can be recovered

$$\bar{U} = V^{-1}\bar{W}$$

Network Coding Homomorphic Authentication for Network Coding Improvements on HSS and KEPTE

Decoding

- After receiving $\{w_i = (v_i, \bar{w}_i)\}_{i=1}^m$ in which $v_1, v_2, \ldots, v_m$ are linearly independent
  - Set

$$\bar{U} = \begin{pmatrix} \bar{u}_1 \\ \bar{u}_2 \\ \cdots \\ \bar{u}_m \end{pmatrix}, V = \begin{pmatrix} v_1 \\ v_2 \\ \cdots \\ v_m \end{pmatrix}, \text{and } \bar{W} = \begin{pmatrix} \bar{w}_1 \\ \bar{w}_2 \\ \cdots \\ \bar{w}_m \end{pmatrix}.$$

  - The original messages can be recovered

$$\bar{U} = V^{-1}\bar{W}$$

- $q = 2^8$ is sufficient to achieve a successful decoding probability greater than 99%

Network Coding Homomorphic Authentication for Network Coding Improvements on HSS and KEPTE

Random Linear Network Coding

- At Source $\mathcal{S}$: A file is divided into generations (subfiles), and each generation consists of $\bar{u}_1, \bar{u}_2, \ldots, \bar{u}_m \in \mathbb{F}_q^n$

Random Linear Network Coding

- At Source $\mathcal{S}$: A file is divided into generations (subfiles), and each generation consists of $\bar{u}_1, \bar{u}_2, \dots, \bar{u}_m \in \mathbb{F}_q^n$
- Generate augmented vectors and forward them

$$u_i = (\overbrace{\underbrace{0, \dots, 0, 1}_{i},\ 0, \dots, 0}^{m}, \bar{u}_i) \ \in \mathbb{F}_q^{n+m}$$

Random Linear Network Coding

- At Source $\mathcal{S}$: A file is divided into generations (subfiles), and each generation consists of $\bar{u}_1, \bar{u}_2, \dots, \bar{u}_m \in \mathbb{F}_q^n$
- Generate augmented vectors and forward them

$$u_i = (\overbrace{\underbrace{0, \dots, 0, 1}_{i},\ 0, \dots, 0}^{m}, \bar{u}_i) \ \in \mathbb{F}_q^{n+m}$$

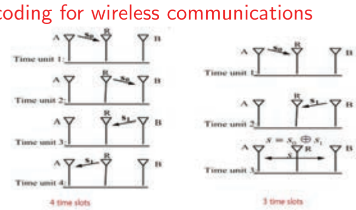
- For example,

$$2u_1 + 3u_2 = (\overbrace{2, 3, \dots, 0, \dots, 0}^{m}, 2\bar{u}_1 + 3\bar{u}_2)$$

The first m bits contain the coefficients used in combing the vectors, which are called the Global Encoding Vector.

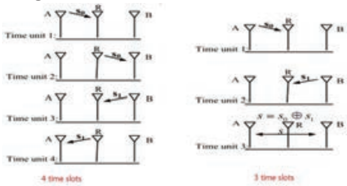
Network Coding Applications

- Network coding for wireless communications

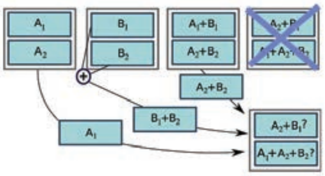


Network Coding Applications

- Network coding for wireless communications



- Network coding for distributed storage systems



Network Coding Homomorphic Authentication for Network Coding Improvements on HSS and KEPTE

First Impression of Homomorphic Authentication

- RSA is homomorphic: From $\mathrm{Sign(m)} = \mathrm{m^d}$, we know that

$$m_1^d \cdot m_2^d = (m_1 m_2)^d$$

Network Coding Homomorphic Authentication for Network Coding Improvements on HSS and KEPTE

First Impression of Homomorphic Authentication

- RSA is homomorphic: From $\mathrm{Sign(m)} = \mathrm{m^d}$, we know that

$$m_1^d \cdot m_2^d = (m_1 m_2)^d$$

- Considered to be undesirable and Hash-and-sign to eliminate it: $\mathrm{Sign\,[H(m)]} = \mathrm{[H(m)]^d}$

First Impression of Homomorphic Authentication

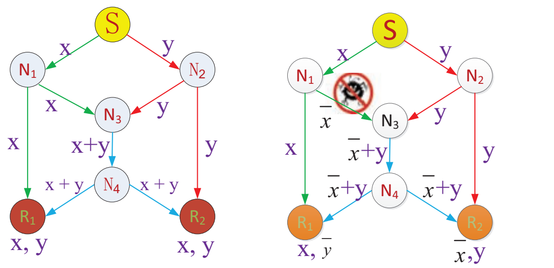
- RSA is homomorphic: From $\mathrm{Sign(m) = m^d}$, we know that

$$m_1^d \cdot m_2^d = (m_1 m_2)^d$$

- Considered to be undesirable and Hash-and-sign to eliminate it: $\mathrm{Sign\,[H(m)] = [H(m)]^d}$
- Can we find the positive side of homomorphic signatures?

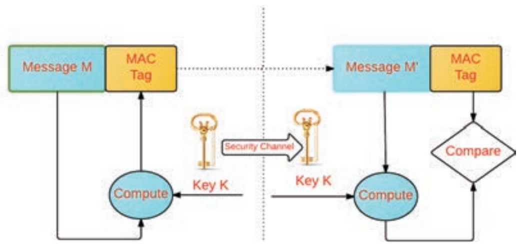
Pollution Attacks

- The adversary inject invalid packets into the network



Can Regular Cryptographic Tools Help?

- The original messages sent by the source have been modified by the intermediate nodes.



Homomorphic Hashing

- First for Rateless Erasure Codes (Krohn2004); Related to Pederson Commitment Scheme

Homomorphic Hashing

- First for Rateless Erasure Codes (Krohn2004); Related to Pederson Commitment Scheme
- Exponential Homomorphic Hash (EHH): Let $g_1, g_2, \ldots, g_n$ be generators of a cyclic group G of order p, and $x = (x_1, \ldots, x_n)$

$$h(x) = \prod_{i=1}^{n} g_i^{x_i}$$

Property of Homomorphic Hashing

- Homomorphic Property: For scalars α, β and vectors a, b

$$h(\alpha a + \beta b) = h(a)^{\alpha} h(b)^{\beta}$$

Property of Homomorphic Hashing

- Homomorphic Property: For scalars α, β and vectors a, b

$$h(\alpha a + \beta b) = h(a)^{\alpha} h(b)^{\beta}$$

- Collision Resistance: If $h(c) = h(a)^{\alpha} h(b)^{\beta}$ for vectors a, b, c, then

$$c = \alpha a + \beta b$$

Homomorphic Signatures for Network Coding

- Homomorphic Property: For scalars α, β and vectors a, b

$$\text{Sign}(\alpha a + \beta b) = \text{Sign}(a)^{\alpha} \text{Sign}(b)^{\beta}$$

Homomorphic Signatures for Network Coding

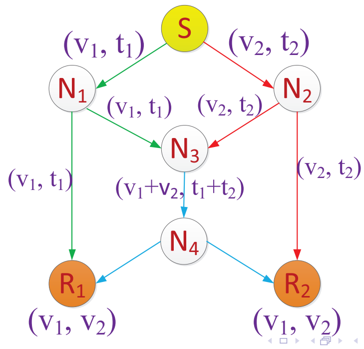
- Homomorphic Property: For scalars α, β and vectors a, b

$$\text{Sign}(\alpha a + \beta b) = \text{Sign}(a)^{\alpha} \text{Sign}(b)^{\beta}$$

- Any intermediate node can (i) verify the signatures and (ii) compute a valid signature on each outgoing vector without knowing the secret key.

Homomorphic MAC

- In homomorphic signature, the encoding coefficients are chosen from $p \approx 2^{160}$ instead of $q = 2^8$.



The Homomorphic Subspace Signature (HSS) Scheme

- Parameters: Select a cyclic group H with order p and g is the generator of H, then randomly select $\boldsymbol{\beta} = (\beta_1, \beta_2, \ldots, \beta_{N+1}) \in F_p^{N+1}$, and calculate $\boldsymbol{h} = (h_1, h_2, \ldots, h_{N+1})$, where $h_i = g^{\beta_i}$ for each $1 \leq i \leq N+1$. The public key is $(H,\ p,\ g,\ \boldsymbol{h})$, and the private key is $\boldsymbol{\beta}$.

P. Zhang et al., "Padding for orthogonality: Efficient subspace authentication for network coding," in IEEE INFOCOM 2011.

The Homomorphic Subspace Signature (HSS) Scheme

- Parameters: Select a cyclic group H with order p and g is the generator of H, then randomly select $\boldsymbol{\beta} = (\beta_1, \beta_2, \ldots, \beta_{N+1}) \in F_p^{N+1}$, and calculate $\boldsymbol{h} = (h_1, h_2, \ldots, h_{N+1})$, where $h_i = g^{\beta_i}$ for each $1 \leq i \leq N+1$. The public key is $(H,\ p,\ g,\ \boldsymbol{h})$, and the private key is $\boldsymbol{\beta}$.
- Signatures: For message $\boldsymbol{u} = (u_1, u_2, \ldots, u_N) \in F_p^N$, the signature $\sigma_{\boldsymbol{u}}$ of $\boldsymbol{u} \in F_p^N$ is calculated as

$$\sigma_{\boldsymbol{u}} = -\sum_{i=1}^{N} \beta_i \boldsymbol{u}_i / \beta_{N+1}.$$

P. Zhang et al., "Padding for orthogonality: Efficient subspace authentication for network coding," in IEEE INFOCOM 2011.

The KEy Predistribution-based Tag Encoding (KEPTE) Scheme

- Intermediate and destination nodes use pre-distributed secrets to detect and filter the corrupted packets by verifying the validity of signatures appended with the received packets.

X. Wu et al., "A Tag Encoding Scheme against Pollution Attack to Linear Network Coding," IEEE Trans on Parallel and Distributed Systems, vol. 25, no. 1, pp. 33-42, January 2014.

The KEy Predistribution-based Tag Encoding (KEPTE) Scheme

- Intermediate and destination nodes use pre-distributed secrets to detect and filter the corrupted packets by verifying the validity of signatures appended with the received packets.
- A trusted KDC selects secrets $\boldsymbol{s}_1, \boldsymbol{s}_2, \ldots, \boldsymbol{s}_l \in \mathbb{F}_q^{n+m}$, and sends them to the source $\mathcal{S}$.

X. Wu et al., "A Tag Encoding Scheme against Pollution Attack to Linear Network Coding," IEEE Trans on Parallel and Distributed Systems, vol. 25, no. 1, pp. 33-42, January 2014.

The KEy Predistribution-based Tag Encoding (KEPTE) Scheme

- Intermediate and destination nodes use pre-distributed secrets to detect and filter the corrupted packets by verifying the validity of signatures appended with the received packets.
- A trusted KDC selects secrets $\boldsymbol{s}_1, \boldsymbol{s}_2, \ldots, \boldsymbol{s}_l \in \mathbb{F}_q^{n+m}$, and sends them to the source $\mathcal{S}$.
- For node $\mathcal{N}$, KDC sends $\boldsymbol{z}_\mathcal{N}$ and $\boldsymbol{x}_\mathcal{N}$ to $\mathcal{N}$ in a secure way. Here $\boldsymbol{z}_\mathcal{N} = (z_1, z_2, \ldots, z_l) \in \mathbb{F}_q^l$ is randomly selected and

$$x_k = \sum_{j=1}^{l} z_j s_{j,k}, \quad 1 \leq k \leq n+m.$$

X. Wu et al., "A Tag Encoding Scheme against Pollution Attack to Linear Network Coding," IEEE Trans on Parallel and Distributed Systems, vol. 25, no. 1, pp. 33-42, January 2014.

A Successful Forgery Attack

If $\boldsymbol{u} \in \prod_1 \subset \mathbb{F}_p^{n+m}$ and $\boldsymbol{v} \in \prod_2 \subset \mathbb{F}_p^{n+m}$, the adversary can launch a successful forgery attack by simply setting $\boldsymbol{u}^* = \boldsymbol{u} + \boldsymbol{v}$ and $\sigma_{\boldsymbol{u}^*} = \sigma_{\boldsymbol{u}} + \sigma_{\boldsymbol{v}}$, which can pass the verification at intermediate and destination nodes. Furthermore, we can show that $\boldsymbol{u}^*$ does not belong to $\prod_1$ or $\prod_2$ with a high probability.

Network Coding Homomorphic Authentication for Network Coding Improvements on HSS and KEPTE

What We can Learn from the Attack

- Cause of The Attack: The HSS and KEPTE schemes own the homomorphic property for messages belong to two different generations.

Network Coding Homomorphic Authentication for Network Coding Improvements on HSS and KEPTE

What We can Learn from the Attack

- Cause of The Attack: The HSS and KEPTE schemes own the homomorphic property for messages belong to two different generations.
- **Query in the improved security model:** The adversary $\mathcal{A}$ can adaptively chooses vector $\boldsymbol{v} \in \prod_i \subset \mathbb{F}_q^N$ and sends it to the challenger $\mathcal{C}$. The challenger $\mathcal{C}$ randomly chooses an identifier id_i for $\prod_i$, and signs the vector $\boldsymbol{v}$. The signature $\sigma_{\boldsymbol{v}}$ and the identifier id_i are then sent to the adversary $\mathcal{A}$.

An Improved Key Distribution Scheme for HSS

- At the beginning of each generation with identification id_i, the source node $\mathcal{S}$ obtains $\beta_{\mathrm{id}_i} = F(k_2, \mathrm{id}_i) \in \mathbb{F}_p$ and updates $\boldsymbol{\beta}$ as

$$\boldsymbol{\beta}^{\mathrm{id}_i} = (\beta_1 + \beta_{\mathrm{id}_i}, \beta_2 + \beta_{\mathrm{id}_i}, \ldots, \beta_{N+1} + \beta_{\mathrm{id}_i}).$$

An Improved Key Distribution Scheme for HSS

- At the beginning of each generation with identification id_i, the source node $\mathcal{S}$ obtains $\beta_{\mathrm{id}_i} = F(k_2, \mathrm{id}_i) \in \mathbb{F}_p$ and updates $\boldsymbol{\beta}$ as

$$\boldsymbol{\beta}^{\mathrm{id}_i} = (\beta_1 + \beta_{\mathrm{id}_i}, \beta_2 + \beta_{\mathrm{id}_i}, \ldots, \beta_{N+1} + \beta_{\mathrm{id}_i}).$$

- Next, the updated $\boldsymbol{\beta}^{\mathrm{id}_i}$ is used to sign the messages. After that the source node $\mathcal{S}$ appends $g^{\beta_{\mathrm{id}_i}}$ to each message $\boldsymbol{u}$ and its corresponding signature $\sigma_{\boldsymbol{u}}$. With the received information of $g^{\beta_{\mathrm{id}_i}}$ at all the intermediate nodes and receivers, they can update $\boldsymbol{h}$ in the public key as

$$\begin{aligned}\boldsymbol{h}^{\mathrm{id}_i} &= (h_1^{\mathrm{id}_i}, h_2^{\mathrm{id}_i}, \ldots, h_{N+1}^{\mathrm{id}_i}) \in \mathbb{F}_p^{N+1} \\ &= (h_1 g^{\beta_{\mathrm{id}_i}}, h_2 g^{\beta_{\mathrm{id}_i}}, \ldots, h_{N+1} g^{\beta_{\mathrm{id}_i}}) \\ &= (g^{\beta_1+\beta_{\mathrm{id}_i}}, g^{\beta_2+\beta_{\mathrm{id}_i}}, \ldots, g^{\beta_{N+1}+\beta_{\mathrm{id}_i}}).\end{aligned}$$

An Improved Key Distribution Scheme for KEPTE: For All Nodes

- The KDC randomly selects l seeds $k_1, k_2, \ldots, k_l \in \mathcal{K}_F$ and send them in a secure way to all the nodes in the network.

An Improved Key Distribution Scheme for KEPTE: For All Nodes

- The KDC randomly selects l seeds $k_1, k_2, \ldots, k_l \in \mathcal{K}_F$ and send them in a secure way to all the nodes in the network.
- Then, for the transmission messages in the generation with identification id_i, all the nodes in the network can generate vectors $\boldsymbol{y}_{\mathrm{id}_i,1}, \boldsymbol{y}_{\mathrm{id}_i,2}, \ldots, \boldsymbol{y}_{\mathrm{id}_i,l}$ as

$$\boldsymbol{y}_{\mathrm{id}_i,j} = F(k_j, \mathrm{id}_i) \in \mathbb{F}_q^{n+m}, 1 \le j \le l$$

An Improved Key Distribution Scheme for KEPTE: At the Source

- The KDC selects l seeds $b_1, \ldots, b_l \in \mathcal{K}_{G_1}$ to generate $\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_l$ in such a way that $(\mathbf{u}_1^T, \mathbf{u}_2^T \ldots, \mathbf{u}_l^T)^T$ is of full rank. The KDC sends $b_1, \ldots, b_l$ to the source node $\mathcal{S}$, then the source node $\mathcal{S}$ can get l secret vectors $\mathbf{s}_1, \ldots, \mathbf{s}_l \in \mathbb{F}_q^{n+m}$ from

$$\begin{pmatrix} \boldsymbol{u}_1 \\ \boldsymbol{u}_2 \\ \cdots \\ \boldsymbol{u}_l \end{pmatrix} \begin{pmatrix} \boldsymbol{s}_1 \\ \boldsymbol{s}_2 \\ \cdots \\ \boldsymbol{s}_l \end{pmatrix} = \begin{pmatrix} \boldsymbol{y}_{\mathrm{id}_i,1} \\ \boldsymbol{y}_{\mathrm{id}_i,2} \\ \cdots \\ \boldsymbol{y}_{\mathrm{id}_i,l} \end{pmatrix}.$$

An Improved Key Distribution Scheme for KEPTE: For Node $\mathcal{N}$

- The KDC selects $\mathbf{c}_{\mathcal{N}}$ for $\mathcal{N}$ and then computes $\mathbf{z}_{\mathcal{N}} \in \mathbb{F}_q^l$ as

$$\mathbf{z}_{\mathcal{N}} = \mathbf{c}_{\mathcal{N}} \begin{pmatrix} \boldsymbol{a}_1 \\ \boldsymbol{a}_2 \\ \cdots \\ \boldsymbol{a}_l \end{pmatrix}.$$

An Improved Key Distribution Scheme for KEPTE: For Node $\mathcal{N}$

- The KDC selects $\mathbf{c}_{\mathcal{N}}$ for $\mathcal{N}$ and then computes $\mathbf{z}_{\mathcal{N}} \in \mathbb{F}_q^l$ as

$$\mathbf{z}_{\mathcal{N}} = \mathbf{c}_{\mathcal{N}} \begin{pmatrix} \boldsymbol{a}_1 \\ \boldsymbol{a}_2 \\ \cdots \\ \boldsymbol{a}_l \end{pmatrix}.$$

- $\mathbf{c}_{\mathcal{N}}$ and $\mathbf{z}_{\mathcal{N}}$ are then sent to the node $\mathcal{N}$ in a secure way by the KDC. With $\mathbf{c}_{\mathcal{N}}$, the node $\mathcal{N}$ can get a secret vector $\mathbf{x}_{\mathcal{N}} \in \mathbb{F}_q^{n+m}$ as

$$\mathbf{x}_{\mathcal{N}} = \mathbf{c}_{\mathcal{N}} \begin{pmatrix} \boldsymbol{y}_{\mathrm{id}_i,1} \\ \boldsymbol{y}_{\mathrm{id}_i,2} \\ \cdots \\ \boldsymbol{y}_{\mathrm{id}_i,l} \end{pmatrix}.$$

Efficiency Analysis of the Improved KEPTE

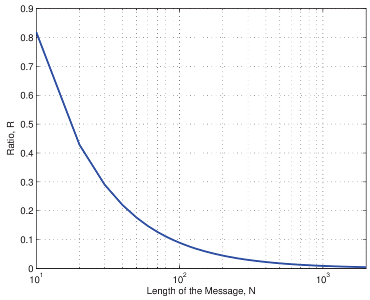
- Main difference: At the beginning of each generation, all the nodes use a PRF to produce $\boldsymbol{y}_{\mathrm{id},j} = F_1(k_j, \mathrm{id})$, $1 \leq j \leq l$, instead of one PRG computation in KEPTE.

Efficiency Analysis of the Improved KEPTE

- Main difference: At the beginning of each generation, all the nodes use a PRF to produce $\boldsymbol{y}_{\mathrm{id},j} = F_1(k_j, \mathrm{id})$, $1 \leq j \leq l$, instead of one PRG computation in KEPTE.
- Therefore, the communication and storage cost of the improved key distribution scheme is the same with that in KEPTE, while an additional PRF computation is needed during data transmission in every generation .

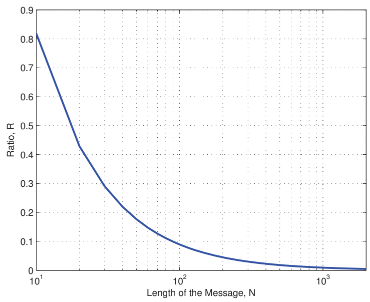
Added Communication Cost for the Improved HSS

- The ratio of the added communication cost $R = \frac{|p|+r}{(N+1)|p|}$.



Added Communication Cost for the Improved HSS

- The ratio of the added communication cost $R = \frac{|p|+r}{(N+1)|p|}$.
- If we set $N = 1000$ and $|p| = 128$, $r = 1024$, then R is less than 0.9%, which is acceptable.



Added Computation Cost for the Improved HSS

- Here, we set $N = 1000$ and $|p| = 128$

Schemes	**At source**	**At intermediate node**	**Total**
HSS	4.403 ms	803.354 ms	807.757 ms
Proposed	10.676 ms	803.954 ms	814.630 ms
Added cost	6.273 ms	0.600 ms	6.873 ms

Thanks & Questions?

Unifying Reliability, Security, and Deduplication in Cloud Storage

Patrick P. C. Lee

The Chinese University of Hong Kong

pclee@cse.cuhk.edu.hk

In this talk, we study the problem of dispersing user backup data across multiple clouds, with a primary objective of providing a unified multicloud storage solution with reliability, security, and cost-efficiency guarantees.

We first present CDStore [1], a multi-cloud storage system that builds on an augmented secret sharing scheme called *convergent dispersal*, which supports deduplication by using deterministic content-derived hashes as inputs to secret sharing. We describe how CDStore combines convergent dispersal with two-stage deduplication to achieve both bandwidth and storage savings and be robust against side-channel attacks. We evaluate the performance of our CDStore prototype using real-world workloads on LAN and commercial cloud testbeds. Our cost analysis also demonstrates that CDStore achieves a monetary cost saving of 70% over a baseline cloud storage solution using state-of-the-art secret sharing.

We next present REED [2], a cloud storage system that further addresses the rekeying problem for cloud storage that combines both encryption and deduplication. Rekeying renews security protection, so as to protect against key compromise and enable dynamic access control in cryptographic storage. However, it is non-trivial to realize efficient rekeying in the context of encrypted deduplication. REED is rekeying-aware by extending the CDStore design, such that it enables secure and lightweight rekeying, while preserving the deduplication capability. We propose two REED encryption schemes that trade between performance and security, and extend REED for dynamic access control. We implement a REED prototype with various performance optimization techniques. Our trace-driven testbed evaluation shows that our REED prototype maintains high performance and storage efficiency.

References

[1] M. Li, C. Qin, and P. P. C. Lee. CDStore: Toward Reliable, Secure, and Cost-Efficient Cloud Storage via Convergent Dispersal. In *USENIX ATC*, July 2015.

[2] J. Li, C. Qin, P. P. C. Lee, and J. Li. Rekeying for Encrypted Deduplication Storage. In *IEEE/IFIP DSN*, June 2016.

Unifying Reliability, Security, and Deduplication in Cloud Storage

Patrick P. C. Lee

The Chinese University of Hong Kong

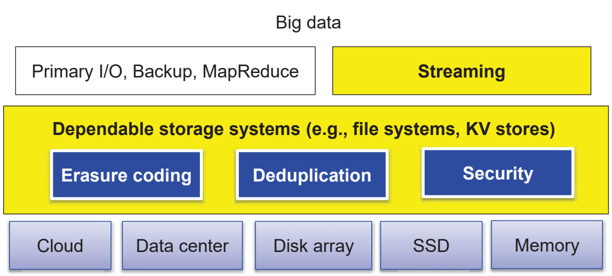
1

Our Research Focus

- **Dependable storage systems**
  - Improve fault tolerance, recovery, security, and performance of storage systems
  - Architectures: clouds, data centers, disk arrays, SSDs, memory
- **Fault-tolerant distributed stream analytics**
  - Applications
    - Anomaly detection in network traffic monitoring
    - Distributed machine learning
  - Fault tolerance of computation and storage
- Our approach:
  - Build prototypes, backed by experiments and theoretical analysis
  - Open-source software: http://www.cse.cuhk.edu.hk/~pclee

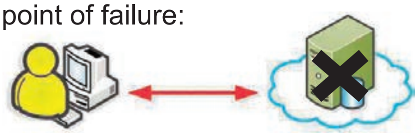
2

Our Research Focus

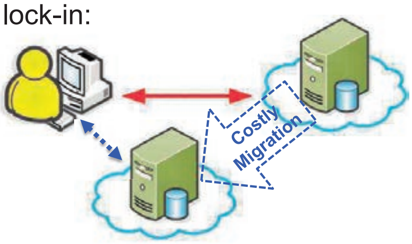


Single Cloud Problems

- Single point of failure:

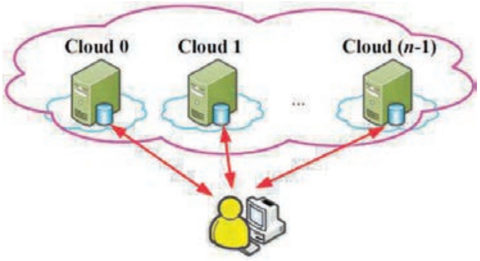


- Vendor lock-in:



4

Cloud-of-Clouds



- Exploits diversity of multiple-cloud storage:
  - Reliability (or fault tolerance)
  - No vendor lock-in
  - Security

5

Secret Sharing

- Input: secret; output: multiple shares
- Secret is recoverable from enough shares
  → Reliability
- Secret is inaccessible without enough shares
  → Security

6

Examples

- **Shamir's [CACM'79]**
  - Information-theoretic security
  - Same storage overhead as replication
- **IDA [JACM'89]**
  - Weakest security
  - Low storage overhead
- **Ramp's [Crypto'84]**
  - Trade between Shamir's and IDA
- **Secret sharing made short [Crypto'93]**
  - Computational security – Shamir's for keys and IDA for data
  - Low storage overhead
- **AONT-RS [FAST'11]**
  - Computational security with even smaller overhead
  - Allow integrity checking

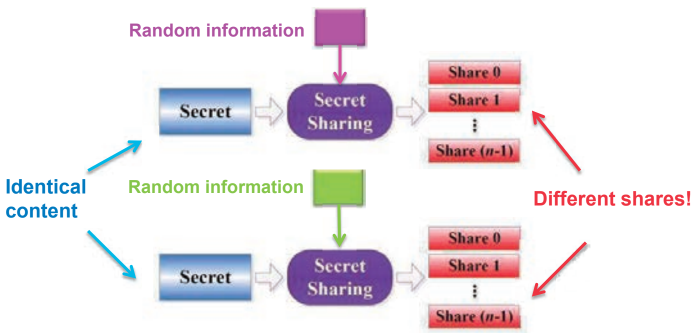
7

Challenges

- Cloud storage uses **deduplication** to save cost
- Deduplication avoids storing multiple data copies with identical content
  - Saves storage space
  - Saves write bandwidth
- However, secret sharing breaks deduplication
  - Root cause: **security builds on embedded randomness**

8

Challenges

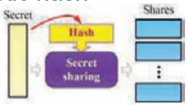


Q: Can we unify secret sharing and deduplication in a seamless way?

9

Challenges

- ➢ Secret sharing prohibits deduplication
  - Reason: Security builds on embedded randomness
  - → Identical secrets lead to different shares
  - → High bandwidth and storage overhead
- ➢ **Convergent dispersal**[*]:
  - Replaces random input with deterministic hash derived from original secret
  - → Reliability, security, cost efficiency
- ➢ How to deploy?



[*] "Convergent Dispersal: Toward Storage-Efficient Security in a Cloud-of-Clouds", **HotStorage'14**

10

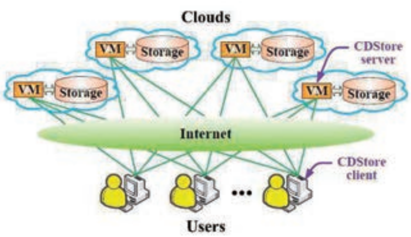
CDStore

- ➢ **CDStore**[*]: a unified cloud storage system with reliability, security, and cost efficiency
- ➢ A new instantiation of convergent dispersal
  - Higher throughput than our prior approach
- ➢ Two-stage deduplication
  - Bandwidth and storage savings
  - Secure
- ➢ Trace-driven experiments and cost analysis

[*] "CDStore: Toward Reliable, Secure, and Cost-Efficient Cloud Storage via Convergent Dispersal", **USENIX ATC'15, IEEE Internet Computing 16**

11

CDStore Architecture



- ➢ Client-server model
- ➢ Target audience: an organization that needs storage outsourcing for users' data
- ➢ Target workload: backup and archival

12

Goals

- Reliability:
  - Availability if some clouds are operational
  - No metadata loss if CDStore clients fail
- Security:
  - Confidentiality (i.e., data is secret)
  - Integrity (i.e., data is uncorrupted)
  - Robust against side-channel attacks
- Deduplication:
  - Low bandwidth and storage costs via deduplication
  - Low VM computation and metadata overheads

13

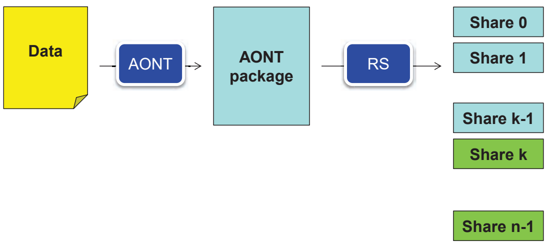
Assumptions

- Reliability:
  - Efficient repair is not considered
- Security:
  - Secrets drawn from large message space, so brute-force attacks are infeasible [Bellare, Security'13]
  - Encrypted and authenticated client-server channels
- Cost efficiency:
  - No billing for communication between co-locating VMs and storage

14

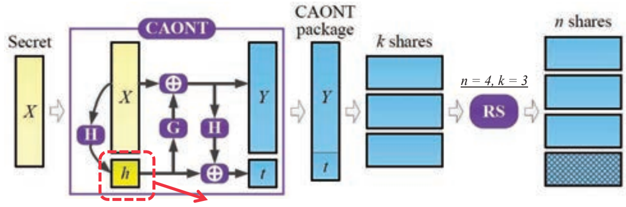
[Resch and Plank, FAST'11]

AONT-RS



15

Convergent AONT-RS (CAONT-RS)

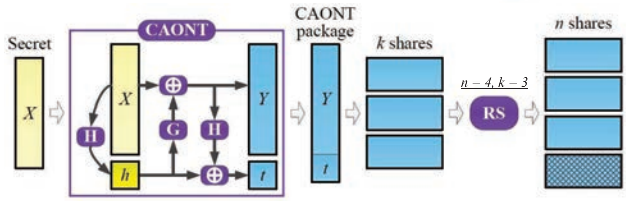


➢ Extension of AONT-RS

➢ Optimal asymmetric encryption padding (OAEP) AONT
- Single encryption on a large block

16

CAONT-RS Encoding



➢ Generate CAONT package (Y, t):
- $h = \mathbf{H}(X)$
- $Y = X \oplus \mathbf{G}(h)$
- $\mathbf{G}(h) = \mathbf{E}(h, C)$
- $t = h \oplus \mathbf{H}(Y)$

$\mathbf{H}(.)$: hash function (e.g., SHA-256)
$\mathbf{G}(.)$: generator function
$\mathbf{E}(.)$: encryption function (e.g., AES-256)
C: constant value block

➢ Encode CAONT package with Reed-Solomon codes

17

Deduplication

➢ Deduplication at the secret level
- Same secret → same shares that are dedup'ed
- Ensure the same share in the same cloud
  - Share i stored in cloud i, where $i = 0, 1, \ldots, n\text{-}1$

➢ Naïve approach: client-side global deduplication
- Saves most upload bandwidth and storage
- Susceptible to side-channel attacks
  - Attackers can infer if other users have stored same data

18

Two-Stage Deduplication

- Decomposes deduplication into two stages:
  - **Client-side intra-user deduplication**
    - Each CDStore client uploads unique shares of same user
    - Effective for backup workloads
  - **Server-side Inter-user deduplication**
    - Each CDStore server dedups same shares from different users
    - Effective if many users share similar data (e.g., VM images)
- Fingerprint index maintained by CDStore servers

19

CDStore Implementation

- C++ implementation on Linux
- Features:
  - Content-defined chunking
  - Parallelization of encoding and I/O operations
  - Batched network and storage I/Os
- Open issues:
  - Storage reclaim via garbage collection and compression
  - Multiple CDStore servers per cloud
  - Consistency due to concurrent updates

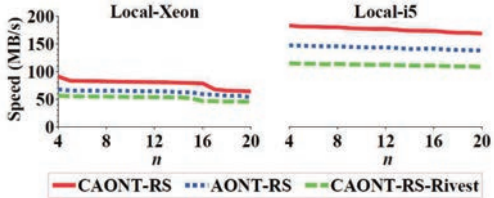
20

Experimental Setup

- Testbeds:
  - **Local machines**: i5 3.4GHz (fast), Xeon 2.4GHz (slow)
  - **LAN**: Multiple i5 machines via 1Gb switch
  - **Cloud**: Google, Azure, AWS and Rackspace
- Datasets:
  - Synthetic unique and fully duplicate data
  - FSL dataset from Stony Brook University
  - Our own VM images of 156 students

21

Encoding Speeds

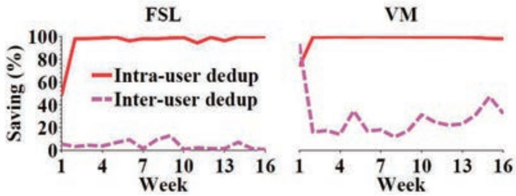


Largest k with $k/n \le 3/4$

- OAEP-based AONT brings high performance gain
  - CAONT-RS achieves 183MB/s on Local-i5
- Encoding speed slightly decreases with n
  - RS coding has small overhead
- Multi-threading boosts speed (details in paper)

22

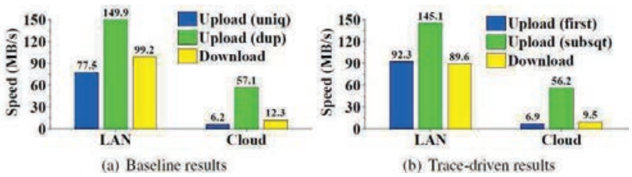
Storage Savings



- Intra-user dedup achieves high saving
  - At least 98% after Week 1
- Inter-user dedup is effective for VM dataset
  - Week 1: 93.4%
  - After Week 1: 11.8% - 47.0%

23

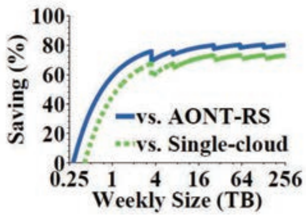
Transfer Speeds

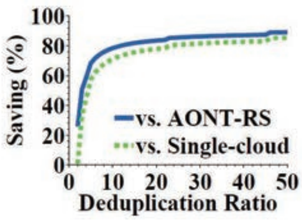


- (Single-client) upload speeds in LAN:
  - Unique data ~ 77MB/s (network bound)
  - Duplicate data ~ 150MB/s (bounded by encoding + chunking)
- Performance in cloud bounded by Internet bandwidth
- Aggregate upload speeds increase with number of clients (details in paper)

24

Cost Analysis





- Compared to solutions w/o dedup:
  - (1) single cloud; (2) multiple clouds with AONT-RS
- At least 70% savings when dedup ratio is 10x – 50x
- Jagged curves due to switching cheapest VM instances

25

Summary

- **CDStore**: a unified multi-cloud storage system with three goals in mind: **reliability**, **security**, and **deduplication**
- Building blocks:
  - Convergent dispersal
  - Two-stage deduplication
- Source code:
  - http://ansrlab.cse.cuhk.edu.hk/software/cdstore

26

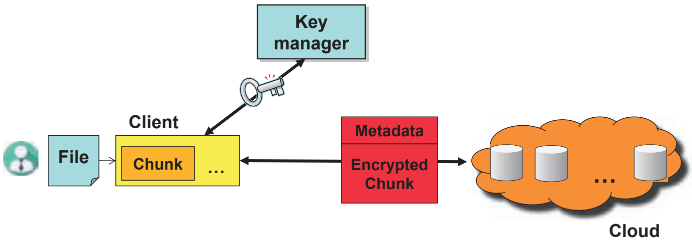
Encrypted Deduplication

- **Message-locked encryption** [Bellare, EUROCRYPT'13]
  - Derive encryption key from message itself
  - Same message → Identical cipher text
  - e.g., **convergent encryption**: key = message hash
- **DupLESS** [Bellare, USENIX Security'13]
  - Realizes server-aided message-locked encryption
  - A dedicated **key manager** for key generation
  - **MLE key** generated by a **derivation function**
    - Same messages → same ciphers
    - Ciphers appear random

27

Encrypted Deduplication

- Server-aided message-locked encryption:



28

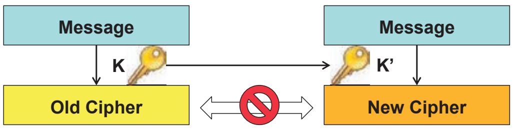
Rekeying

- **Rekeying** renews security protection
  - Replaces an existing key with a new encryption key
- Benefits:
  - Protects against key compromise
  - Revokes unauthorized users from accessing data
- Challenges:
  - Renewing derivation function makes new data fail to be dedup'ed with old data
  - Cipher re-encryption is expensive

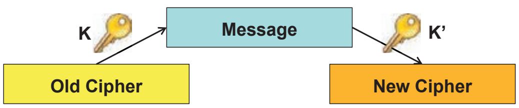
29

Rekeying Challenges

- Renewing derivation function (e.g., K → K'): new data can't be dedup'ed with old data:



- Cipher re-encryption with new key K':



30

REED

- REED[*], a **Re**keying-aware **E**ncrypted **D**eduplication storage system
  - Provides secure and lightweight rekeying
  - Preserves content similarity for deduplication
- Two encryption schemes for REED
  - **Basic**: high performance (203MB/s)
  - **Enhanced**: resilient against key leakage (155MB/s)
- Enabling dynamic access control
- Testbed Experiments

[*] "Rekeying for Encrypted Deduplication Storage", DSN'16

31

Threat Model

- **Honest-but-curious** adversary, who can:
  - Compromise storage backend
  - Collude with any revoked client
  - Attempt to learn files beyond access scope
  - Monitor keys returned by key manager
- Assumptions:
  - Encrypted and authenticated communication between client and key manager (e.g., by SSL/TLS)
  - Key manager cannot infer message content (OPRF)
  - Key manager is deployed in protected zone

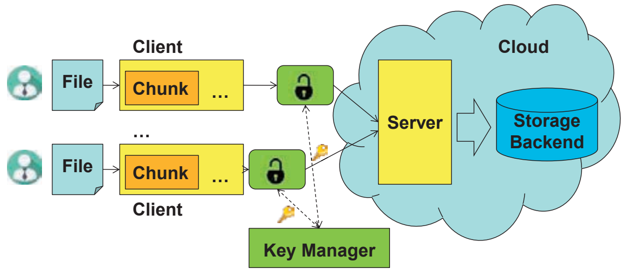
32

Main Idea

- Build security on two symmetric keys:
  - **File key**: **renewable**, controlling access for files
  - **MLE key**: **unchanged**, preserving deduplication
- Extends convergent all-or-nothing transform (CAONT) [Li, USENIX ATC'15]
  - Encrypts entire message using MLE key; and further encrypts a small part (**stub**) using file key
  - Performs deduplication on large part; yet message is unrecoverable with any small part unavailable
  - Rekeying on stub (64 bytes, 0.78% for 8KB chunks)

33

REED Overview

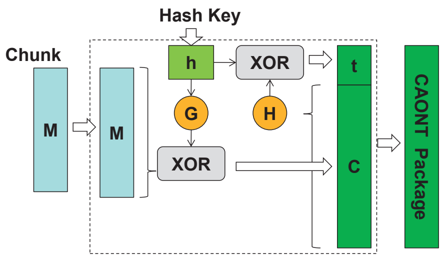


➢ Target workload: backup and archival storage

34

[Li, USENIX ATC'15]

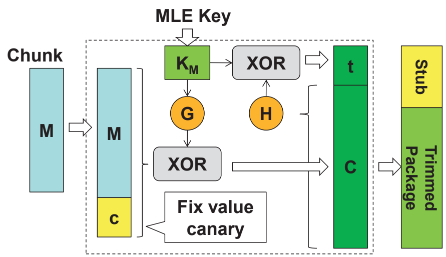
CAONT



➢ Limitation:
- Secure for unpredictable messages only (otherwise, vulnerable to brute-force dictionary attacks)

35

Basic Encryption

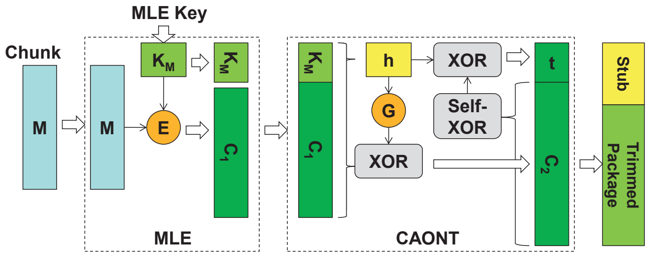


➢ Two modifications to CAONT
- Replaces hash key by MLE key from key manager
- Add a canary for integrity checking

➢ Limitation: vulnerable to MLE key compromise

36

Enhanced Encryption



➢ Resilient against MLE key leakage:
- First applies MLE to form a ciphertext
- Then applies CAONT to the MLE ciphertext

➢ Rationale: MLE ciphertext is further protected by CAONT

37

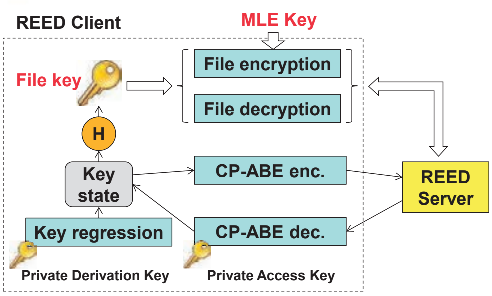
Comparison

➢ Basic encryption:
- Vulnerable to MLE key compromise
  - Adversary can recover large part (trimmed package) of the original message with MLE key obtained
- Faster encryption

➢ Enhanced encryption:
- Higher security level
  - Adversary needs both MLE key and file key to recover a message
  - Even if MLE key is disclosed, remains secure for unpredictable messages
- Slower encryption

38

Dynamic Access Control



➢ Uses CP-ABE for access control [Bethencourt, S&P'07]

➢ Uses key regression for lazy revocation [Kamara, NDSS'06]

39

Dynamic Access Control

- **Lazy revocation**
  - Current key state can derive previous states
  - Revoked user cannot access future key states
  - Allows user to access not-yet-updated files
  - Defers file re-encryption (e.g. midnight update)
- **Active revocation**
  - Re-encryption happens immediately with new key

40

Confidentiality

- Level 1: same as DupLESS
  - Adversary can access all trimmed packages, encrypted stubs, and encrypted key states
- Level 2: colluding with revoked users
  - Adversary can learn a set of private access keys from any revoked user
- Level 3: monitoring key generation
  - Adversary can monitor a subset of revoked users and identify MLE keys returned by key manager

41

Integrity

- Basic encryption
  - By checking the canary attached to recovered chunks
- Enhanced encryption
  - By comparing the hash of MLE ciphertext

42

Implementation

➢ Entities:
- **Client**: chunking, encryption/decryption, upload/download
- **Key manager**: MLE key generation
- **Server**: deduplication, metadata storage
- **Cloud**: file recipe, stub, key states

➢ Optimization:
- **Batch** key generation requests to mitigate I/O
- **Cache** previous MLE keys to reduce computation
- **Parallelize** key generation, encryption and upload via multi-threading

43

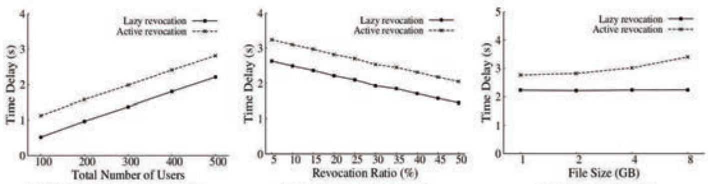
Evaluation

➢ Datasets
- Synthetic dataset (2 GB files with unique chunks)
- FSL data trace (147 daily snapshots, 56.2 TB in total)

➢ Testbed
- Servers connected over a Gigabit LAN

44

Rekeying Performance



(a) Varying the total number of users (b) Varying the revocation ratio (c) Varying the file size

➢ Rekeying delays remain small
- 3.4s for 8 GB data

48

Summary

- REED:
  - Enables rekeying for encrypted deduplication storage
  - Proposes two encryption schemes
  - Enables dynamic access control
  - Implements a prototype
  - Conducts extensive trace-driven evaluation
- Software: http://ansrlab.cse.cuhk.edu.hk/software/reed

51

On The Robustness of Secret Sharing Schemes

Partha Sarathi Roy (Joint work with Avishek Adhikari, Kirill Morozov, Satoshi Obana, Kouchi Sakurai, Rui Xu)

Faculty of Information Science and Electrical Engineering, Department of Informatics, Kyushu University

`royparthasarathi0@gmail.com`

In the basic form of secret sharing schemes, it was assumed that everyone involved with the protocol is semi-honest. But for the real life scenario, this assumption may not hold good due to the presence of adversary. This idea leads to the development of secret sharing under various adversarial models. It may happen that some participants behave maliciously during the execution of the protocol. Malicious participants may submit incorrect shares resulting in incorrect secret reconstruction. This observation led to *robust secret sharing schemes* [4]. Informally, robust secret sharing schemes allow the correct secret to be recovered even when some of the shares presented during an attempted reconstruction are incorrect.

Here, we consider the problem of (t, δ) robust secret sharing secure against rushing adversary. We present a simple t-out-of-n secret sharing scheme, which can reconstruct the secret in presence of t cheating participants except with probability at most δ, provided $t < n/2$. The later condition on cheater resilience is optimal for the case of public reconstruction of the secret, on which we focus our work.

Our construction improves the share size of Cevallos et al. (EUROCRYPT-2012) robust secret sharing scheme by applying the "authentication tag compression" technique devised by Carpentieri in 1995. Our improvement is by a constant factor that does not contradict the asymptotic near-optimality of the former scheme. Finally, we discuss the further improvement of our construction.

References

[1] Adhikari A., Morozov K., Obana S., Roy P.S., Sakurai K., Xu R.: *Efficient Threshold Secret Sharing Schemes Secure against Rushing Cheaters*. eprint.iacr.org/2015/1115.pdf.

[2] Carpentieri, M.: *A perfect threshold secret sharing scheme to identify cheaters*. Design Codes Cryptography 5(3), 183-187 (1995)

[3] Cevallos, A., Fehr, S., Ostrovsky, R., Rabani, Y.: *Unconditionally-secure robust secret sharing with compact shares*. EUROCRYPT 2012, 195-208 (2012)

[4] Rabin, T., Ben-Or, M.: *Verifiable secret sharing and multiparty protocols with honest majority (extended abstract)*. STOC 1989, 73-85 (1989)

[5] Roy P. S., Adhikari A., Xu R., Morozov K., Sakurai K.: *An Efficient Robust Secret Sharing Scheme with Optimal Cheater Resiliency*. Lecture Notes in Computer Sciences, Springer, Volume 8804, 2014, pp 47-58 (SPACE 2014)

On The Robustness of Secret Sharing Schemes

Partha Sarathi Roy

Graduate School of Information Science and Electrical Engineering
Department of Informatics
Kyushu University
Joint Work with
Avishek Adhikari, Kirill Morozov, Satoshi Obana, Kouichi Sakurai, Rui Xu

IMI Workshop: Secret Sharing for Dependability, Usability and Security of Network Storage and Its Mathematical Modeling

Outline

Outline

Outline

KYUSHU UNIVERSITY

1. Secret Sharing
2. Robust Secret Sharing
3. Preliminaries
4. State of The Art
   - Comparison
   - Share Authentication
5. Our Contribution
   - Roy et al. [27]
   - Adhikari et al. [1]
6. Appendix

Partha Sarathi Roy (Kyushu University) Robust Secret Sharing Schemes September 6, 2016 2 / 36

Outline

KYUSHU UNIVERSITY

1. Secret Sharing
2. Robust Secret Sharing
3. Preliminaries
4. State of The Art
   - Comparison
   - Share Authentication
5. Our Contribution
   - Roy et al. [27]
   - Adhikari et al. [1]
6. Appendix

Partha Sarathi Roy (Kyushu University) Robust Secret Sharing Schemes September 6, 2016 2 / 36

Outline

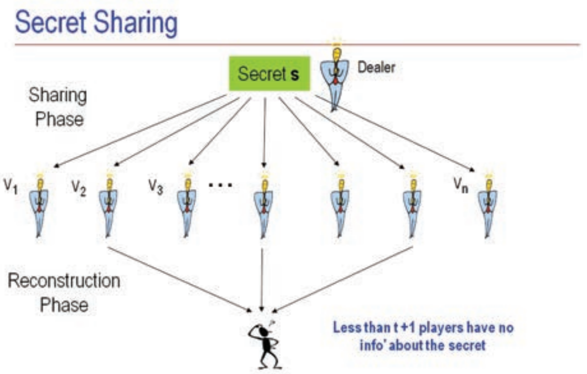
KYUSHU UNIVERSITY

1. Secret Sharing
2. Robust Secret Sharing
3. Preliminaries
4. State of The Art
   - Comparison
   - Share Authentication
5. Our Contribution
   - Roy et al. [27]
   - Adhikari et al. [1]
6. Appendix

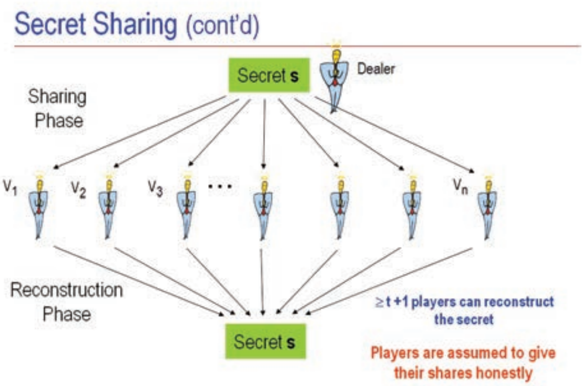
Partha Sarathi Roy (Kyushu University) Robust Secret Sharing Schemes September 6, 2016 2 / 36

Outline

Secret Sharing



Secret Sharing



Secret Sharing

(t-out-of-n) Secret Sharing

secret: s

shares: s_1 s_2 $\cdots$ s_n

- **Privacy**: any t shares give no information on s

  $s_1 \quad s_2 \cdots s_t \quad \rightarrow \quad ?$

- **Reconstructability**: any $t+1$ shares uniquely determine s

  $s_1 \quad s_2 \quad \cdots \quad s_{t+1} \rightarrow s$

Secret Sharing

Shamir's Secret Sharing Scheme [Sha79]

secret: $s \in \mathbb{F}$

$f(X) = s + a_1X + \ldots + a_tX^t \in \mathbb{F}[X]$

shares: $s_1 = f(x_1)$ $\cdots$ $s_n = f(x_n)$

- **Privacy** and **reconstructability** follow from Lagrange interpolation

- Here and in general:
  reconstructability requires correct shares

Robust Secret Sharing

Robust Secret Sharing

secret: s

Note:
assume dealer to be honest

shares: s_1 s_2 $\cdots$ s_n

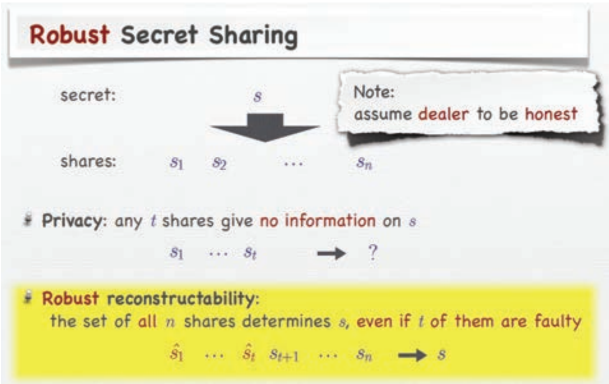
- **Privacy**: any t shares give no information on s

  $s_1 \cdots s_t \quad \rightarrow \quad ?$

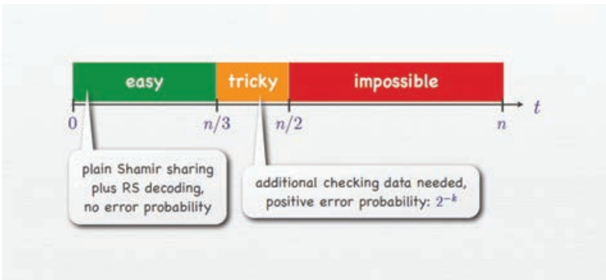
- **Robust reconstructability:**
  the set of all n shares determines s, even if t of them are faulty

  $\hat{s}_1 \quad \cdots \quad \hat{s}_t \quad s_{t+1} \quad \cdots \quad s_n \rightarrow s$

Robust Secret Sharing



Robust Secret Sharing

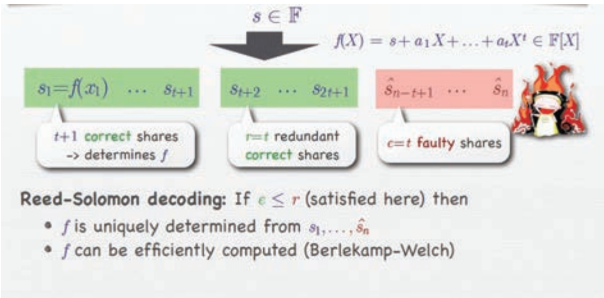


The Reed-Solomon Code

The R-S Code

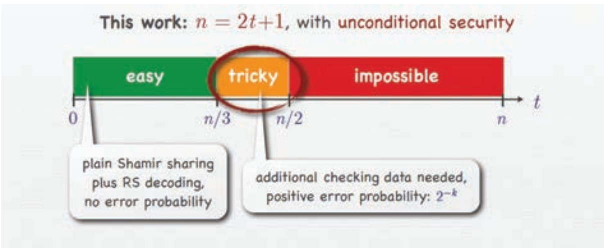
- Let $(a_0, \ldots, a_t) \in \mathbb{F}^{t+1}$ and $f(x) = a_0 + a_1 x + \ldots + a_t x^t \in \mathbb{F}[X]$ be a polynomial of degree at most t. Let $x_1, x_2, \ldots, x_n \in \mathbb{F} \setminus \{0\}$, for $n > t$, be distinct elements.
- Then $C = (f(x_1), f(x_2), \ldots, f(x_n))$ is a codeword of Reed-Solomon error correcting code [20] of the message $(a_0, \ldots, a_t)$.
- Reed-Solomon code can correct up to e erroneous symbols, i.e. when e out of n evaluation points $f(x_i)$ $(1 \leq i \leq n)$ are manipulated, the polynomial (i.e., the message) can be uniquely determined if and only if $n \geq t + 1 + 2e$.
- There exist efficient algorithms implementing Reed-Solomon decoding, such as Berlekamp-Welch algorithm [4].

Robust Secret Sharing for $t < n/3$



Robust Secret Sharing

KYUSHU UNIVERSITY



Message Authentication Codes

KYUSHU UNIVERSITY

MAC

A message authentication code (or MAC) for a finite message space $\mathcal{M}$ consists of a function $MAC : \mathcal{M} \times \mathcal{K} \to \mathcal{T}$ for finite sets $\mathcal{K}$ and $\mathcal{T}$. It is called ϵ-secure if for all $m, m' \in \mathcal{M}$ with $m \neq m'$ and for all $\tau, \tau' \in \mathcal{T}$:

$$P[MAC(m', K) = \tau' | MAC(m, K) = \tau] \leq \epsilon,$$

where the random variable K is uniformly distributed over $\mathcal{K}$.

Example

$$MAC : \mathbb{F}^l \times \mathbb{F}^2 \to \mathbb{F}, ((m_1, \ldots, m_l), (\alpha, \beta)) \to \Sigma_{k=1}^{l} \alpha^i . m_i + \beta \qquad (1)$$

is a ϵ-secure MAC with $\epsilon = l/|\mathbb{F}|$.

State of The Art

Table: Comparison Among Existing Efficient RSS

Scheme	Overhead (bits)
Rabin and Ben-Or [26]	$3(n-1)(2\log(k+1)+\mu)$
Cevallos et al. [6]	$3(n-1)(\log(k+1)+\log(m)+\frac{2}{k+1}(\mu+\log(e)))$
Roy et al. [27]	$(2n+k-2)(\log(k+1)+\log(l)+\frac{2}{k+1}(\mu+\log(e)))$
Adhikari et al. [1]	$(n+k)(\log(k)+\log(l)+\frac{2}{k+1}(\mu+\log(e)))$

Here, m is the bit length of the secret and m is an integer multiple of l, k is the number of cheaters, $n = 2k+1$ is the number of total participants, $e = exp(1)$, and μ is the security parameter *s.t.* the scheme fails to reconstruct the authentic secret with probability at most $2^{-\mu}$.

Rabin and Ben-Or [26] Technique

Let s_i be the Shamir share for the player P_i.

$\Downarrow$

For every pair of players P_i and P_j, P_i's Shamir share s_i is authenticated to the player P_j with an authentication tag $\tau_{i,j}$ obtained by message authentication code, where the corresponding authentication key $k_{j,i}$ is held by player P_j.

$\Downarrow$

Specifically, this step may be done by choosing $k_{j,i} = (g_{j,i}, b_{j,i})$ uniformly at random from $\mathbb{F} \times \mathbb{F}$ and then computing $\tau_{j,i} = s_i g_{j,i} + b_{j,i}$.

$\Downarrow$

Each player will get $n-1$ keys and $n-1$ tags.

State of The Art | Share Authentication

Rabin and Ben-Or [26] Technique

Let s_i be the Shamir share for the player P_i.

$\Downarrow$

For every pair of players P_i and P_j, P_i's Shamir share s_i is authenticated to the player P_j with an authentication tag $\tau_{i,j}$ obtained by message authentication code, where the corresponding authentication key $k_{j,i}$ is held by player P_j.

$\Downarrow$

Specifically, this step may be done by choosing $k_{j,i} = (g_{j,i}, b_{j,i})$ uniformly at random from $\mathbb{F} \times \mathbb{F}$ and then computing $\tau_{j,i} = s_i g_{j,i} + b_{j,i}$.

$\Downarrow$

Each player will get $n-1$ keys and $n-1$ tags.

Partha Sarathi Roy (Kyushu University) | Robust Secret Sharing Schemes | September 6, 2016 | 15 / 36

Rabin and Ben-Or [26] Technique

Let s_i be the Shamir share for the player P_i.

$\Downarrow$

For every pair of players P_i and P_j, P_i's Shamir share s_i is authenticated to the player P_j with an authentication tag $\tau_{i,j}$ obtained by message authentication code, where the corresponding authentication key $k_{j,i}$ is held by player P_j.

$\Downarrow$

Specifically, this step may be done by choosing $k_{j,i} = (g_{j,i}, b_{j,i})$ uniformly at random from $\mathbb{F} \times \mathbb{F}$ and then computing $\tau_{j,i} = s_i g_{j,i} + b_{j,i}$.

$\Downarrow$

Each player will get $n-1$ keys and $n-1$ tags.

State of The Art Share Authentication

Rabin and Ben-Or [26] Technique

Let s_i be the Shamir share for the player P_i.

$\Downarrow$

For every pair of players P_i and P_j, P_i's Shamir share s_i is authenticated to the player P_j with an authentication tag $\tau_{i,j}$ obtained by message authentication code, where the corresponding authentication key $k_{j,i}$ is held by player P_j.

$\Downarrow$

Specifically, this step may be done by choosing $k_{j,i} = (g_{j,i}, b_{j,i})$ uniformly at random from $\mathbb{F} \times \mathbb{F}$ and then computing $\tau_{j,i} = s_i g_{j,i} + b_{j,i}$.

$\Downarrow$

Each player will get $n-1$ keys and $n-1$ tags.

Partha Sarathi Roy (Kyushu University) Robust Secret Sharing Schemes September 6, 2016 15 / 36

Cevallos et al. [6] Technique

Use small tags and keys.

$\Downarrow$

MAC has weak security.

$\Downarrow$

Incorrect shares may be approved by some honest players and Rabin & Ben-Or reconstruction fails.

$\Downarrow$

Cevallos et al. introduce a novel reconstruction technique by using R-S error correcting code where $t < n/2$.

$\Downarrow$

Still, each player will get $n-1$ keys and $n-1$ tags.

Cevallos et al. [6] Technique

Use small tags and keys.

⇓

MAC has weak security.

⇓

Incorrect shares may be approved by some honest players and Rabin & Ben-Or reconstruction fails.

⇓

Cevallos et al. introduce a novel reconstruction technique by using R-S error correcting code where $t < n/2$.

⇓

Still, each player will get $n - 1$ keys and $n - 1$ tags.

State of The Art Share Authentication

Cevallos et al. [6] Technique

Use small tags and keys.

⇓

MAC has weak security.

⇓

Incorrect shares may be approved by some honest players and Rabin & Ben-Or reconstruction fails.

⇓

Cevallos et al. introduce a novel reconstruction technique by using R-S error correcting code where $t < n/2$.

⇓

Still, each player will get $n - 1$ keys and $n - 1$ tags.

Partha Sarathi Roy (Kyushu University) Robust Secret Sharing Schemes September 6, 2016 16 / 36

State of The Art Share Authentication

Cevallos et al. [6] Technique

Use small tags and keys.

⇓

MAC has weak security.

⇓

Incorrect shares may be approved by some honest players and Rabin & Ben-Or reconstruction fails.

⇓

Cevallos et al. introduce a novel reconstruction technique by using R-S error correcting code where $t < n/2$.

⇓

Still, each player will get $n - 1$ keys and $n - 1$ tags.

Partha Sarathi Roy (Kyushu University) Robust Secret Sharing Schemes September 6, 2016 16 / 36

Cevallos et al. [6] Technique

KYUSHU UNIVERSITY

Use small tags and keys.

$\Downarrow$

MAC has weak security.

$\Downarrow$

Incorrect shares may be approved by some honest players and Rabin & Ben-Or reconstruction fails.

$\Downarrow$

Cevallos et al. introduce a novel reconstruction technique by using R-S error correcting code where $t < n/2$.

$\Downarrow$

Still, each player will get $n-1$ keys and $n-1$ tags.

Technique to Reduce the Number of Auth. Tags [8]

KYUSHU UNIVERSITY

Instead of sending $n-1$ tags to each player, send a *seed* c_i to player P_i.

$\Downarrow$

The necessary authentication tags will be generated from the *seed* c_i together with some public information.

$\Downarrow$

The *seed* for P_i is $c_i = (d_{i,1}, \ldots, d_{i,t})$, where $d_{i,j}$ for $j \in \{1, \ldots, t\}$ is randomly chosen from $\mathbb{F}$ and the authentication tag of P_i against P_j's key is $\tau_{i,j} = \alpha_i d_{j,1} + \alpha_i^2 d_{j,2} + \cdots + \alpha_i^t d_{j,t}$.

$\Downarrow$

Compared to the setting of Rabin and Ben-Or, each player now gets a *seed* of t field elements from which the $n-1$ authentication tags are generated. Thus, the share size of each player is reduced by $n-t-1$ field elements.

Our Contribution

Technique to Reduce the Number of Auth. Tags [8]

KYUSHU UNIVERSITY

Instead of sending $n-1$ tags to each player, send a *seed* c_i to player P_i.

$\Downarrow$

The necessary authentication tags will be generated from the *seed* c_i together with some public information.

$\Downarrow$

The *seed* for P_i is $c_i = (d_{i,1}, \ldots, d_{i,t})$, where $d_{i,j}$ for $j \in \{1, \ldots, t\}$ is randomly chosen from $\mathbb{F}$ and the authentication tag of P_i against P_j's key is $\tau_{i,j} = \alpha_i d_{j,1} + \alpha_i^2 d_{j,2} + \cdots + \alpha_i^t d_{j,t}$.

$\Downarrow$

Compared to the setting of Rabin and Ben-Or, each player now gets a *seed* of t field elements from which the $n-1$ authentication tags are generated. Thus, the share size of each player is reduced by $n-t-1$ field elements.

Our Contribution

Technique to Reduce the Number of Auth. Tags [8]

Instead of sending $n-1$ tags to each player, send a *seed* c_i to player P_i.

$\Downarrow$

The necessary authentication tags will be generated from the *seed* c_i together with some public information.

$\Downarrow$

The *seed* for P_i is $c_i = (d_{i,1}, \ldots, d_{i,t})$, where $d_{i,j}$ for $j \in \{1, \ldots, t\}$ is randomly chosen from $\mathbb{F}$ and the authentication tag of P_i against P_j's key is $\tau_{i,j} = \alpha_i d_{j,1} + \alpha_i^2 d_{j,2} + \cdots + \alpha_i^t d_{j,t}$.

$\Downarrow$

Compared to the setting of Rabin and Ben-Or, each player now gets a *seed* of t field elements from which the $n-1$ authentication tags are generated. Thus, the share size of each player is reduced by $n-t-1$ field elements.

Partha Sarathi Roy (Kyushu University) Robust Secret Sharing Schemes September 6, 2016 17 / 36

Our Contribution

Technique to Reduce the Number of Auth. Tags [8]

Instead of sending $n-1$ tags to each player, send a *seed* c_i to player P_i.

$\Downarrow$

The necessary authentication tags will be generated from the *seed* c_i together with some public information.

$\Downarrow$

The *seed* for P_i is $c_i = (d_{i,1}, \ldots, d_{i,t})$, where $d_{i,j}$ for $j \in \{1, \ldots, t\}$ is randomly chosen from $\mathbb{F}$ and the authentication tag of P_i against P_j's key is $\tau_{i,j} = \alpha_i d_{j,1} + \alpha_i^2 d_{j,2} + \cdots + \alpha_i^t d_{j,t}$.

$\Downarrow$

Compared to the setting of Rabin and Ben-Or, each player now gets a *seed* of t field elements from which the $n-1$ authentication tags are generated. Thus, the share size of each player is reduced by $n-t-1$ field elements.

Partha Sarathi Roy (Kyushu University) Robust Secret Sharing Schemes September 6, 2016 17 / 36

Our Contribution Roy et al. [27]

Adversarial Model

- The dealer $\mathcal{D}$ and the reconstructor $\mathcal{R}$ are assumed to be honest. The dealer delivers the shares to respective participants over point-to-point private channels.
- We assume that $\mathcal{A}$ is computationally unbounded, active, adaptive, rushing adversary who can corrupt up to $t < n/2$ participants (but neither $\mathcal{D}$ nor $\mathcal{R}$).
- Note that assuming $\mathcal{R}$ to be honest is equivalent to assuming a broadcast channel available to each participant.

Partha Sarathi Roy (Kyushu University) Robust Secret Sharing Schemes September 6, 2016 18 / 36

Roy et al. [27]

- **Initialization:** For $i = 1, \ldots, n$, let the distinct elements $\alpha_i \in \mathbb{F}_{2^m} \setminus \{0\}$ be fixed and public. Moreover, let α_i be also non-zero and distinct in $\mathbb{F}_{2^q}$, where m, q are two positive integers and the cardinalities of both fields are larger than n.

Sharing Phase

- The dealer $\mathcal{D}$ chooses randomly a polynomial $f(x) \in \mathbb{F}_{2^m}[X]$ of degree at most t, where $f(0) = s$ is the secret to be shared, and computes $f(\alpha_i) = s_i$ in $\mathbb{F}_{2^m}$, where $i = 1, \ldots, n$.
- If $q < m$, we let $l = m/q$ (for simplicity, assuming that l is an integer) and $s_j = s_{j,1}||\ldots||s_{j,l}$.
$\mathcal{D}$ chooses randomly $d_{i,1}, \ldots, d_{i,t}$ and $g_{i,j}$ from F_{2^q}, and computes
$$b_{i,j} = \begin{cases} g_{i,j}s_j + \Sigma_{k=1}^{t} \alpha_i^k d_{j,k} & \text{for} \quad q \geq m \\ \Sigma_{k=1}^{l} g_{i,j}^k s_{j,k} + \Sigma_{k=1}^{t} \alpha_i^k d_{j,k} & \text{for} \quad q < m \end{cases}$$
where $j = 1, \ldots, i-1, i+1, \ldots, n$ and $i = 1, \ldots, n$.
- $\mathcal{D}$ privately sends to each P_i the share
$$S_i = (s_i, d_{i,1}, \ldots, d_{i,t}, g_{i,1}, \ldots, g_{i,i-1}, g_{i,i+1}, \ldots, g_{i,n}, b_{i,1}, \ldots, b_{i,i-1}, b_{i,i+1}, \ldots, b_{i,n}).$$

Sharing Phase

- The dealer $\mathcal{D}$ chooses randomly a polynomial $f(x) \in \mathbb{F}_{2^m}[X]$ of degree at most t, where $f(0) = s$ is the secret to be shared, and computes $f(\alpha_i) = s_i$ in $\mathbb{F}_{2^m}$, where $i = 1, \ldots, n$.
- If $q < m$, we let $l = m/q$ (for simplicity, assuming that l is an integer) and $s_j = s_{j,1}||\ldots||s_{j,l}$.
$\mathcal{D}$ chooses randomly $d_{i,1}, \ldots, d_{i,t}$ and $g_{i,j}$ from F_{2^q}, and computes
$$b_{i,j} = \begin{cases} g_{i,j}s_j + \Sigma_{k=1}^{t} \alpha_i^k d_{j,k} & \text{for} \quad q \geq m \\ \Sigma_{k=1}^{l} g_{i,j}^k s_{j,k} + \Sigma_{k=1}^{t} \alpha_i^k d_{j,k} & \text{for} \quad q < m \end{cases}$$
where $j = 1, \ldots, i-1, i+1, \ldots, n$ and $i = 1, \ldots, n$.
- $\mathcal{D}$ privately sends to each P_i the share
$$S_i = (s_i, d_{i,1}, \ldots, d_{i,t}, g_{i,1}, \ldots, g_{i,i-1}, g_{i,i+1}, \ldots, g_{i,n}, b_{i,1}, \ldots, b_{i,i-1}, b_{i,i+1}, \ldots, b_{i,n}).$$

Sharing Phase

- The dealer $\mathcal{D}$ chooses randomly a polynomial $f(x) \in \mathbb{F}_{2^m}[X]$ of degree at most t, where $f(0) = s$ is the secret to be shared, and computes $f(\alpha_i) = s_i$ in $\mathbb{F}_{2^m}$, where $i = 1, \ldots, n$.
- If $q < m$, we let $l = m/q$ (for simplicity, assuming that l is an integer) and $s_j = s_{j,1}||\ldots||s_{j,l}$.
  $\mathcal{D}$ chooses randomly $d_{i,1}, \ldots, d_{i,t}$ and $g_{i,j}$ from F_{2^q}, and computes
  $b_{i,j} = \begin{cases} g_{i,j}s_j + \Sigma_{k=1}^{t}\alpha_i^k d_{j,k} & \text{for} \quad q \geq m \\ \Sigma_{k=1}^{l} g_{i,j}^k s_{j,k} + \Sigma_{k=1}^{t}\alpha_i^k d_{j,k} & \text{for} \quad q < m \end{cases}$
  where $j = 1, \ldots, i-1, i+1, \ldots, n$ and $i = 1, \ldots, n$.
- $\mathcal{D}$ privately sends to each P_i the share

$$S_i = (s_i, d_{i,1}, \ldots, d_{i,t}, g_{i,1}, \ldots, g_{i,i-1}, g_{i,i+1}, \ldots, g_{i,n}, \\ b_{i,1}, \ldots, b_{i,i-1}, b_{i,i+1}, \ldots, b_{i,n}).$$

Reconstruction Phase

- **Round 1:** Each P_i sends $(s'_i, d'_{i,1}, \ldots, d'_{i,t})$ to the reconstructor $\mathcal{R}$.
- **Round 2:** Each P_i sends $(g'_{i,1}, \ldots, g'_{i,i-1}, g'_{i,i+1}, \ldots, g'_{i,n}, b'_{i,1}, \ldots, b'_{i,i-1}, b'_{i,i+1}, \ldots, b'_{i,n})$ to the reconstructor $\mathcal{R}$.

Reconstruction Phase

- **Round 1:** Each P_i sends $(s'_i, d'_{i,1}, \ldots, d'_{i,t})$ to the reconstructor $\mathcal{R}$.
- **Round 2:** Each P_i sends $(g'_{i,1}, \ldots, g'_{i,i-1}, g'_{i,i+1}, \ldots, g'_{i,n}, b'_{i,1}, \ldots, b'_{i,i-1}, b'_{i,i+1}, \ldots, b'_{i,n})$ to the reconstructor $\mathcal{R}$.

Reconstruction Phase

KYUSHU UNIVERSITY

- **Computation by $\mathcal{R}$:**
  1. $\mathcal{R}$ sets v_{ij}, $i, j \in \{1, 2. \dots, n\}$, to be 1 if P_i's authentication tag is accepted by P_j, i.e., if
  $$b'_{i,j} = \begin{cases} g'_{i,j}s'_j + \Sigma_{k=1}^{t} \alpha_i^k d'_{j,k} & \text{for} \quad q \geq m \\ \Sigma_{k=1}^{l} g'^{k}_{i,j} s'_{j,k} + \Sigma_{k=1}^{t} \alpha_i^k d'_{j,k} & \text{for} \quad q < m \end{cases},$$
  otherwise she sets v_{ij} to 0.
  2. $\mathcal{R}$ computes the largest set $\mathcal{I} \subseteq \{1, 2, \dots, n\}$ with the property that
  $$\forall i \in \mathcal{I} : |\{j \in \mathcal{I} | v_{ij} = 1\}| = \Sigma_{j \in \mathcal{I}} v_{ij} \geq t + 1.$$
  Clearly, $\mathcal{I}$ contains all honest participants. Let $e = |\mathcal{I}| - (t+1)$ be the maximum number of corrupted participants in $\mathcal{I}$.
  3. Using the error correction algorithm for Reed-Solomon code, $\mathcal{R}$ computes a polynomial $f(x) \in \mathbb{F}_{2^m}[X]$ of degree at most t such that $f(\alpha_i) = s'_i$ for at least $(t+1) + \frac{e}{2}$ participants i in $\mathcal{I}$.
  If no such polynomial exists then output $\perp$,
  otherwise, output $s = f(0)$.

Reconstruction Phase

KYUSHU UNIVERSITY

- **Computation by $\mathcal{R}$:**
  1. $\mathcal{R}$ sets v_{ij}, $i, j \in \{1, 2. \dots, n\}$, to be 1 if P_i's authentication tag is accepted by P_j, i.e., if
  $$b'_{i,j} = \begin{cases} g'_{i,j}s'_j + \Sigma_{k=1}^{t} \alpha_i^k d'_{j,k} & \text{for} \quad q \geq m \\ \Sigma_{k=1}^{l} g'^{k}_{i,j} s'_{j,k} + \Sigma_{k=1}^{t} \alpha_i^k d'_{j,k} & \text{for} \quad q < m \end{cases},$$
  otherwise she sets v_{ij} to 0.
  2. $\mathcal{R}$ computes the largest set $\mathcal{I} \subseteq \{1, 2, \dots, n\}$ with the property that
  $$\forall i \in \mathcal{I} : |\{j \in \mathcal{I} | v_{ij} = 1\}| = \Sigma_{j \in \mathcal{I}} v_{ij} \geq t + 1.$$
  Clearly, $\mathcal{I}$ contains all honest participants. Let $e = |\mathcal{I}| - (t+1)$ be the maximum number of corrupted participants in $\mathcal{I}$.
  3. Using the error correction algorithm for Reed-Solomon code, $\mathcal{R}$ computes a polynomial $f(x) \in \mathbb{F}_{2^m}[X]$ of degree at most t such that $f(\alpha_i) = s'_i$ for at least $(t+1) + \frac{e}{2}$ participants i in $\mathcal{I}$.
  If no such polynomial exists then output $\perp$,
  otherwise, output $s = f(0)$.

Reconstruction Phase

KYUSHU UNIVERSITY

- **Computation by $\mathcal{R}$:**
  1. $\mathcal{R}$ sets v_{ij}, $i, j \in \{1, 2. \dots, n\}$, to be 1 if P_i's authentication tag is accepted by P_j, i.e., if
  $$b'_{i,j} = \begin{cases} g'_{i,j}s'_j + \Sigma_{k=1}^{t} \alpha_i^k d'_{j,k} & \text{for} \quad q \geq m \\ \Sigma_{k=1}^{l} g'^{k}_{i,j} s'_{j,k} + \Sigma_{k=1}^{t} \alpha_i^k d'_{j,k} & \text{for} \quad q < m \end{cases},$$
  otherwise she sets v_{ij} to 0.
  2. $\mathcal{R}$ computes the largest set $\mathcal{I} \subseteq \{1, 2, \dots, n\}$ with the property that
  $$\forall i \in \mathcal{I} : |\{j \in \mathcal{I} | v_{ij} = 1\}| = \Sigma_{j \in \mathcal{I}} v_{ij} \geq t + 1.$$
  Clearly, $\mathcal{I}$ contains all honest participants. Let $e = |\mathcal{I}| - (t+1)$ be the maximum number of corrupted participants in $\mathcal{I}$.
  3. Using the error correction algorithm for Reed-Solomon code, $\mathcal{R}$ computes a polynomial $f(x) \in \mathbb{F}_{2^m}[X]$ of degree at most t such that $f(\alpha_i) = s'_i$ for at least $(t+1) + \frac{e}{2}$ participants i in $\mathcal{I}$.
  If no such polynomial exists then output $\perp$,
  otherwise, output $s = f(0)$.

Proof of Security

KYUSHU UNIVERSITY

Lemma 1

The above scheme provides perfect secrecy, i.e. the adversary $\mathcal{A}$ controlling any t participants during the sharing phase will get no information about the secret s. Proof

Lemma 2

Any corrupted participant P_i who submits $s'_i \neq s_i$ in Round 1 of the reconstruction phase will be accepted by an honest participant with probability at most $\epsilon = \begin{cases} \frac{1}{2^q} & \text{for} \quad q \geq m \\ \frac{l}{2^q} & \text{for} \quad q < m \end{cases}$.

Proof

Proof of Security

KYUSHU UNIVERSITY

Theorem

For any positive integer t such that $n = 2t + 1$, the proposed construction forms (t, δ)-robust secret sharing scheme for n participants with the space of secrets $\mathbb{F}_{2^m}$ and

$$\delta \leq e.((t+1)\epsilon)^{(t+1)/2}$$

where $e = exp(1)$ and $\epsilon = \begin{cases} \frac{1}{2^q} & \text{for} \quad q \geq m \\ \frac{l}{2^q} & \text{for} \quad q < m \end{cases}$.

Proof

Authentication Technique

KYUSHU UNIVERSITY

- $MAC : \mathbb{F}^{l \times n} \times \mathbb{F}^{n+1}$, where $\mathbb{F}$ is a finite field of size q is a authentication code to authenticate n messages. MAC is constructed as follows: the n messages are $(m_{i,1}, \ldots, m_{i,l})$ for $i \in [n]$, the authentication key is $(g, b_1, \ldots, b_n)$, where $[n] = \{1, 2, \ldots, n\}$. The tag for message i is $\tau_i = \Sigma_{k=1}^{l} g^k.m_{i,k} + b_i$.

Initialization

- For $i = 1, \ldots, n$, let the distinct elements $\alpha_i \in \mathbb{F}_{2^m} \setminus \{0\}$ be fixed and public. Moreover, let α_i be also non-zero and distinct in $\mathbb{F}_{2^q}$, where m, q are two positive integers, $m = l \cdot q$ (for simplicity, assuming that l is an integer) and the cardinality of both fields are larger than n.

Sharing Phase

- The dealer $\mathcal{D}$ chooses randomly a polynomial $f(x)$ of degree at most $(k-1)$ in x from $\mathbb{F}_{2^m}[X]$ such that $f(0) = s$, where s is the secret to be shared. Also, the dealer $\mathcal{D}$ computes $f(\alpha_i) = s_i$ in $\mathbb{F}_{2^m}$, where $i = 1, \ldots, n$ and $s_i = s_{i,1} || \ldots || s_{i,l}$.
-
  - The dealer first chooses $g_i \in_R \mathbb{F}_{2^q}$ and a polynomial of degree at most $k-1$ with free coefficient 0,
    $t_i(x) = t_{i,1}x + t_{i,2}x^2 + \cdots + t_{i,k-1}x^{k-1}$, from $\mathbb{F}_{2^q}[X]$.
  - The dealer computes, $\tau_{i,j} = t_i(\alpha_j)$ and $b_{i,j} = t_j(\alpha_i) - \Sigma_{u=1}^{l} g_i^u \cdot s_{j,u}$ for $i \in [n] \setminus j$.
- $\mathcal{D}$ sends each P_i the share
  $V_i = (s_i, t_i(x), g_i, b_{i,1}, \ldots, b_{i,i-1}, b_{i,i+1}, \ldots, b_{i,n})$.

Proof of Security

Theorem

For any positive integer k such that $n = 2k - 1$, the proposed construction forms (k, δ)-robust secret sharing scheme for n participants with the space of secrets $\mathbb{F}_{2^m}$ and

$$\delta \leq e.(k\epsilon)^{k/2}$$

where $e = exp(1)$ and $\epsilon = \frac{l}{2^q}$.

- Adhikari A., Morozov K., Obana S., Roy P. S., Sakurai K., Xu R.: *Efficient Threshold Secret Sharing Schemes Secure against Rushing Cheaters*. IACR Cryptology ePrint Archive 2015: 1115 (2015).
- Araki T., Obana S.: *Flaws in some secret sharing schemes against cheating*. ACISP 2007, 122-132 (2007)
- Araki T. *Efficient (k, n) threshold secret sharing schemes secure against cheating from n-1 cheaters*. ACISP 2007, 133-142 (2007)
- Berlekamp, E.R., Welch, L.R.: *Error correction of algebraic block codes*. U.S. Patent Number 4, 633-470 (1986)
- Blakley G.R.: *Safeguarding cryptographic keys*. AFIPS 1979, 313-317 (1979)
- Cevallos, A., Fehr, S., Ostrovsky, R., Rabani, Y.: *Unconditionally-secure robust secret sharing with compact shares*. EUROCRYPT 2012, 195-208 (2012)
- Cabello S., Padro C., Saez G.: *Secret sharing schemes with detection of cheaters for a general access structure*. Design Codes Cryptography, 25(2), 175-188 (2002)

- Carpentieri, M.: *A perfect threshold secret sharing scheme to identify cheaters*. Design Codes Cryptography 5(3), 183-187 (1995)
- Choudhury, A.: *Brief announcement: optimal amortized secret sharing with cheater identification*. PODC 2012, 101-102 (2012)
- Cramer R., Damgard I., Fehr S.: *On the cost of reconstructing a secret, or VSS with optimal reconstruction phase*. CRYPTO 2001, 503-523 (2001)
- Den Boer, B.: *A simple and key-economical unconditional authentication scheme*. Journal of Computer Security 2, 65-72 (1993)
- Cramer R., Dodis Y., Fehr S., Padro C., Wichs D.: *Detection of algebraic manipulation with applications to robust secret sharing and fuzzy extractors*. EUROCRYPT 2008, 471-488 (2008)
- Chor, B., Goldwasser, S., Micali, S., and Awerbuch, B.: *Verifiable Secret Sharing and Achieving Simultaneity in the Presence of Faults (Extended Abstract)*. FOCS 1985, 383-395 (1985)

- Dolev, D., Dwork, C., Waarts, O., Yung, M.: *Perfectly secure message transmission*. FOCS 1990, 36-45 (1990). Journal version in J. ACM 40(1), 17-47 (1993)
- Ishai, Y., Ostrovsky, R., Seyalioglu, H.: *Identifying cheaters without an honest majority*. TCC 2012, 21-38 (2012)
- Mahabir Prasad Jhanwar, Reihaneh Safavi-Naini: Unconditionally-secure ideal robust secret sharing schemes for threshold and multilevel access structure. J. Mathematical Cryptology 7(4), 279-296 (2013)
- Johansson T., Kabatianskii G., Smeets B.: *On the relation between A-codes and codes correcting independent errors*. EUROCRYPT 93, 1-11 (1994)
- Kurosawa, K., Obana, S., Ogata, W.: *t-cheater identifiable (k, n) threshold secret sharing schemes*. CRYPTO 1995, 410-423 (1995)
- Lakshmanan, S., Ahamad, M., Venkateswaran, H.: *Responsive security for stored data*. IEEE Trans. Parallel Distrib. Syst. 14(9), 818-828 (2003)
- MacWilliams, F. J., Sloane, N. J. A.: *The theory of error-correcting codes* (Vol. 16). Elsevier (1977)

Martin, K.M., Paterson, M.B., Stinson, D.R.: *Error decodable secret sharing and one-round perfectly secure message transmission for general adversary structures*. Cryptography and Communications 3(2), 65-86 (2011)

McEliece, R., Sarwate, D.: *On sharing secrets and reed-solomon codes*. Commun. ACM 24(9), 583-584 (1981)

Obana, S.: *Almost optimum t-cheater identifiable secret sharing schemes*. EUROCRYPT 2011, 284-302 (2011)

Obana S., Araki T.: *Almost optimum secret sharing schemes secure against cheating for arbitrary secret distribution*. ASIACRYPT 2006, 364-379 (2006)

Ogata W., Kurosawa K., Stinson D. R.: *Optimum secret sharing scheme secure against cheating*. SIAM J. Discrete Math., 20(1), 79-95 (2006)

Rabin, T., Ben-Or, M.: *Verifiable secret sharing and multiparty protocols with honest majority (extended abstract)*. STOC 1989, 73-85 (1989)

Roy P. S., Adhikari A., Xu R., Morozov K., Sakurai K.: *An Efficient Robust Secret Sharing Scheme with Optimal Cheater Resiliency*, Lecture Notes

in Computer Sciences, Springer, Volume 8804, 2014, pp 47-58 (SPACE 2014).

Shamir A.: *How to share a secret*. Comm. ACM 22(11), 612-613 (1979)

Taylor, R.: *An Integrity Check Value Algorithm for Stream Ciphers*. CRYPTO 1993, 40-48 (1994)

Tompa, M., Woll, H.: *How to share a secret with cheaters*. J. Cryptology 1(2), 133-138 (1988)

Waldman, M., Rubin, A.D., Cranor, L.F.: *The architecture of robust publishing systems*. ACM Trans. Internet Techn. 1(2), 199-230 (2001)

Wegman M.N., Lawrence Carter J.: *New classes and applications of hash functions*. FOCS 1979, 175-182 (1979)

Xu R., Morozov K., Takagi T.: *On Cheater Identifiable Secret Sharing Schemes Secure Against Rushing Adversary*. IWSEC 2013, 258-271 (2013)

Question

END

KYUSHU UNIVERSITY

proof

The dealer $\mathcal{D}$ shares the secret s through a polynomial $f(x)$, where the degree of the polynomial is at most t in x, and the share of each P_i is

$$S_i = (s_i, d_{i,1}, \ldots, d_{i,t}, g_{i,1}, \ldots, g_{i,i-1}, g_{i,i+1}, \ldots, g_{i,n}, \\ b_{i,1}, \ldots, b_{i,i-1}, b_{i,i+1}, \ldots, b_{i,n}).$$

Without loss of generality, we may assume that the first t participants $P_1, \ldots, P_t$ are under $\mathcal{A}$'s control. Now, according to *Lagrange's interpolation*, $t+1$ such values s_i fully define a degree-t polynomial. Thus, we need to choose one more s_i, where $i \in \{1, 2, \ldots, n\} \setminus L$ and $L = \{1, 2, \ldots, t\}$. Without loss of generality, we may assume that $i = t+1$. Let us now estimate the information regarding s_{t+1} which is available to each P_i, $i \in L$, via $(g_{i,t+1}, b_{i,t+1})$.

proof Contd.

Case 1 ($q \geq m$)**:**For all $i \in L$,

$$b_{i,t+1} = g_{i,t+1}s_{t+1} + \alpha_i d_{t+1,1} + \alpha_i^2 d_{t+1,2} + \cdots + \alpha_i^t d_{t+1,t}.$$

So, for all $i \in L$,

$$b_{i,t+1} - g_{i,t+1}s_{t+1} = \alpha_i d_{t+1,1} + \alpha_i^2 d_{t+1,2} + \cdots + \alpha_i^t d_{t+1,t}.$$

Note that the above system of linear equations is associated with the following matrix, which is non-singular in $\mathbb{F}_{2^q}$:

$$\begin{bmatrix} \alpha_1 & \alpha_1^2 & \ldots & \alpha_1^t \\ \alpha_2 & \alpha_2^2 & \ldots & \alpha_2^t \\ \ldots & \ldots & \ldots & \ldots \\ \alpha_t & \alpha_t^2 & \ldots & \alpha_t^t \end{bmatrix}.$$

Now, we conclude that $\mathcal{A}$ can guess the correct s_{t+1} with probability at most $\frac{1}{2^m}$ as $s_{t+1} \in \mathbb{F}_{2^m}$.

proof Contd.

Case 2 ($q < m$):
For all $i \in L$,

$$b_{i,t+1} = \Sigma_{k=1}^{l} g_{i,t+1}^{k} s_{t+1,k} + \Sigma_{k=1}^{t} \alpha_i^k d_{t+1,k}.$$

Here $q < m$, $l = m/q$ (for simplicity, l is assumed to be an integer) and $s_j = s_{j,1}||\dots||s_{j,l}$. So, for all $i \in L$,

$$b_{i,t+1} - \Sigma_{k=1}^{l} g_{i,t+1}^{k} s_{t+1,k} = \Sigma_{k=1}^{t} \alpha_i^k d_{t+1,k}.$$

Now, for any fixed value of $s_{t+1} = s_{t+1,1}||\dots||s_{t+1,l}$, we can use the same argument as in Case 1 in order to show that the probability for $\mathcal{A}$ to guess s_{t+1} correctly is at most $(1/2^q)^l = 1/2^m$.

Return

proof

Without loss of generality, we assume that the corrupted participant is P_1 who submits $s_i' \neq s_i$ in Round 1 of the reconstruction phase.
Case 1 ($q \geq m$):
P_1 will be accepted by honest P_j if
$b_{j,1} = g_{j,1}s_1' + \alpha_j d_{1,1}' + \alpha_j^2 d_{1,2}' + \cdots + \alpha_j^t d_{1,t}'$. Thus P_1 has to guess $g_{j,1}$ correctly. Now, let

$$g_{j,1}s_i' + \Sigma_{k=1}^{t} \alpha_j^k d_{1,k}' = g_{j,1}s_i + \Sigma_{k=1}^{t} \alpha_j^k d_{1,k}.$$

Then,

$$g_{j,1} = (s_1' - s_1)^{-1} \Sigma_{k=1}^{t} \alpha_j^k (d_{1,k} - d_{1,k}').$$

Note that $g_{j,1}$ is independent of all information that the adversary $\mathcal{A}$ has obtained and $g_{j,1} \in \mathbb{F}_{2^q}$.

proof Contd.

Thus, P_1 will be accepted by P_j with probability at most $\frac{1}{2^q} \geq Pr(v_{1j} = 1)$. Therefore, any dishonest participant P_i submitting $s_i' \neq s_i$ in Round 1 of the reconstruction phase will be accepted by a honest participant P_j with probability $Pr(v_{ij} = 1) \leq 1/2^q$.
Case 2 ($q < m$):
P_1 will be accepted by honest P_j if $b_{j,1} = \Sigma_{k=1}^{l} g_{j,1}'^{k} s_{1,k}' + \Sigma_{k=1}^{t} \alpha_j^k d_{1,k}'$.
As $s_1 \neq s_1'$, at least one of $s_{1,k} \neq s_{1,k}'$. Assume that only one $s_{1,k} \neq s_{1,k}'$. So, as in Case 1, P_1 will be accepted by P_j with probability at most $\frac{1}{2^q} \geq Pr(v_{1j} = 1)$. Taking into account the union bound, P_1 will be accepted by P_j with probability at most $\frac{l}{2^q} \geq Pr(v_{1j} = 1)$. Therefore, any dishonest participant P_i submitting $s_i' \neq s_i$ in Round 1 of the reconstruction phase will be accepted by a honest participant P_j with probability $Pr(v_{ij} = 1) \leq l/2^q$.

Return

proof

Privacy: Follows from Lemma 23.
Reconstructability: From Lemma 23, we have found that $Pr(v_{ij} = 1) \leq \epsilon$. The rest of the proof is the same as in [6, Theorem 3.1].

Return

Secret Sharing against Cheaters

Rui Xu (Joint work with Kirill Morozov and Tsuyoshi Takagi)

KDDI R&D Laboratories, Inc.
ru-xu@kddilabs.jp

Information theoretically secure secret sharing first proposed by Shamir [2] and Blarkley [1] is a useful tool for many cryptographic applications. A secret sharing scheme allows a so-called dealer to distribute his secret to a group of parties in such a way that authorized sets of parties can collaboratively reconstruct the secret, while unauthorized sets of parties get no information regarding the secret.

We consider the case where some parties may cheat while reconstructing the secret in order to fool other parties. However, the dealer is assumed to be honest in this work. We introduce two cheater identifiable secret sharing (CISS) schemes with efficient reconstruction, tolerating $t < k/2$ cheaters and one robust secret sharing scheme (RSS).

Cheater identifiable secret sharing (CISS) is an upgrade of (k, n)-threshold secret sharing schemes [2, 1] that can tolerate up to t actively corrupt participants. The dealer in CISS is assumed to be honest. The goal in this scenario is to identify cheaters from the threshold k number of players, and to recover a correct secret whenever possible. Our constructions [3], which provide public cheater identification, feature a novel application of multi-receiver authentication codes to ensure integrity of shares. The first CISS scheme, which tolerates rushing cheaters, has the share size $|S|(n-t)^{n+t+2}/\epsilon^{n+t+2}$ in the general case, that can be ultimately reduced to $|S|(k-t)^{k+t+2}/\epsilon^{k+t+2}$ assuming that all the t cheaters are among the k reconstructing players. The second CISS scheme, which tolerates non-rushing cheaters, has the share size $|S|(n-t)^{2t+2}/\epsilon^{2t+2}$. These two constructions have the smallest share size among the existing CISS schemes of the same category, when the secret is a single field element.

Robust secret sharing (RSS) differs from CISS in that it aims to assure the correct recovery of the shared secret by requiring all parties to appear in the reconstruction phase. More specifically in a (t, n, δ) RSS, the dealer shares the secret to n parties and an adversary can adaptively corrupt t of the parties and modify there shares in an arbitrary way. Finally, we use the tool of multi-receiver authentication to construct a robust secret sharing scheme, which updates the start-of-art against rushing adversary by reducing the share overhead by slightly more than one half.

References

[1] Blarkley, G.R.: Safeguarding cryptographic keys. In: Proceedings of AFIPS 1979 National Computer Conference, vol. 48, pp. 313-317 (1979)

[2] Shamir, A.: How to Share a Secret. Commun. ACM 22(11), 612-613 (1979)

[3] Xu, R., Morozov, K., and Takagi, T.: Cheater identifiable secret sharing schemes via multi-receiver authentication. In Advances in Information and Computer Security (pp. 72-87), IWSEC 2014. Springer International Publishing.

Secret Sharing Against Cheaters

Rui Xu[1], Kirill Morozov[2], Tsuyoshi Takagi[3]

[1] KDDI R&D Laboratories, Inc.
[2] School of Computing, Tokyo Institute of Technology
[2] Institute of Mathematics for Industry, Kyushu University

2016/09/07 @ Kyushu, IMI Workshop

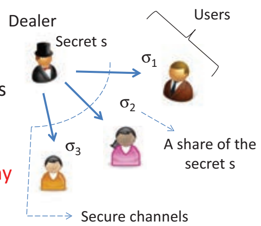
Outline

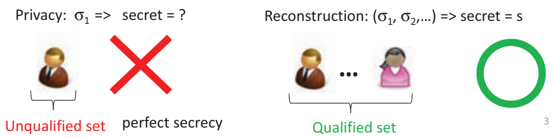
- Secret sharing
- Cheater Identification & Robustness
- Our Construction of cheater identifiable secret sharing
- Application to robust secret sharing
- Open questions

2

Secret Sharing

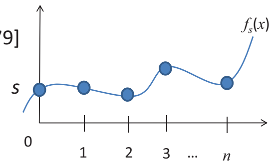
- Dealer distributes shares among n users (parties)
- A collection A of subsets of parties (access structure) can reconstruct the secret
- A subset *not in* A cannot reveal any information on the secret





3

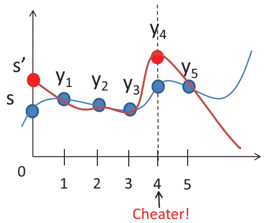
Shamir (k,n) Threshold Secret Sharing



- [Shamir, Commun. ACM 22(11) '79]
- **Share Generation:**
  Dealer chooses
  $f_s(x) \in_R F_p[X]$: $\deg(f_s) \leq k\text{-}1$,
- $f_x(x) = s + \sum_{j=1}^{k-1} b_j x^j$
  s is the secret, $b_j \leftarrow_R F_p$, $1 \leq j \leq n$
- Shares $\sigma_i = f_s(i)$, $1 \leq i \leq n$
- **Reconstruction:** Using Lagrange interpolation, any subset of k parties reconstructs "s"

4

Secret Sharing with Cheaters

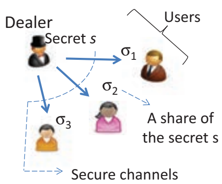


- Computational security
  - Security based on some unproven hardness assumption
- Information theoretical security
  - Adversary has unlimited computing power
- Different Models
  - Cheater Detection (do not identify cheater)
  - Robust Secret Sharing (need all shares)
  - Cheater Identification

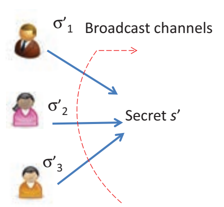
5

Cheater identifiable secret sharing

- Communication model

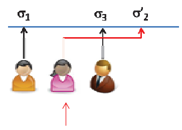


Share Generation



Secret Reconstruction

Synchronous network with rushing
1. Communication proceeds within rounds
2. Rushing is allowed

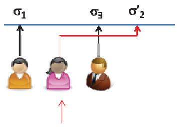


Rushing adversary speaks at last

6

Cheater identifiable secret sharing (CISS)

- Adversary model
  - A_{lis}: adaptive, computationally unbounded, passive, can control at most k-1 players. (listening adversary)
  - A_{cheat}: adaptive, computationally unbounded, active, can control t players. (cheating adversary)
- number of active cheaters: $t<k/2$
- A_{cheat} may be rushing or non-rushing
- A_{lis} and A_{cheat} do not collude.



Rushing adversary speaks at last

7

Cheater identifiable secret sharing (CISS)

- Goal: to identify the cheaters with the smallest shares possible (k)
- Assumption: dealer is honest, public identification.
- Notation:
  - (t,ε) CISS: $\leq t$ cheaters succeed with probability $\leq\varepsilon$
- At reconstruction, a list of cheaters L is output

8

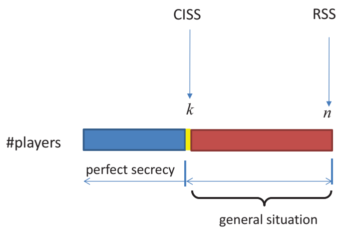
Robust secret sharing(RSS)

- Adversary model
  - A: adaptive, computationally unbounded, active, can control at most t players.
- $t < n/2$ and the threshold will be $t+1$
- Adversary may be rushing or non-rushing
- Communication model is the same as in CISS
- Goal: to recover the correct secret even in presence of cheaters
- Notation: (t, δ) RSS, in presence of t cheaters, the secret can be correctly recovered with probability at least 1- δ

9

CISS and RSS

- CISS aims to identify cheaters from minimal (k) number of shares
- RSS aims to recover the correct secret even in the presence of cheaters



But cheater identification sometimes is possible in RSS.

10

Lower Bound

- [Kurosawa, Obana, Ogata CRYPTO '95]
- In CISS, the share size is at least

$$|V_i| \geq (|S|-1)/\varepsilon + 1,$$

where $|V_i|$ is the share size for player P_i,
$|S|$ is the size of the secret,
ε - cheaters' success probability

- Note: this is a lower bound for non-rushing adversary

11

Previous Works

Refrence	Category	Contribution	Limitation
[Tompa,Woll. J. Crypt'88]	Cheater Detection	Point out the issue of cheater in secret sharing	Large share size No identification
[Rabin, Ben-Or. STOC'89]	Robust Secret Sharing (cheat identification)	First scheme with cheater identification	Large share size
[McEliece, Sarwate. Comm. ACM'81]	Cheater Identification	Connection between Shamir Scheme and RS Code	More than k shares
[Obana. EuroCrypt' 11]	CISS	Optimal share size (for t<k/3), identify cheaters by k shares	Non-rushing adversary
[Choudhury. PODC'12]	CISS	Asymptotically optimal share size against $t<k/2$ rushing cheaters	Secret is a vector, optimal only its length is large
[Jhanwar & Safavi-Naini FC'12]	Robust Secret Sharing	Ideal robust scheme against $t<n/2-1$ non-rushing cheaters	Inefficient reconstruction

12

Our results

	Non-rushing	Rushing
$t < k/3$	[Obana. EuroCrypt' 11] $\lvert V_i \rvert = \lvert S \rvert / \epsilon$ (almost optimal)	[Xu et.al. IWSEC' 13] $\lvert V_i \rvert = \lvert S \rvert / \epsilon^{n-t+1}$ [This work] $\lvert V_i \rvert = \lvert S \rvert / \epsilon^{n-\lfloor (k-1)/3 \rfloor+1}$
$t < k/2$	[Obana. EuroCrypt' 11] $\lvert V_i \rvert \approx \lvert S \rvert (nt \cdot 2^{3t})^2 / \epsilon^2$ (inefficient reconstruction) [This work] $\lvert V_i \rvert = \lvert S \rvert (n-t)^{2t+2} / \epsilon^{2t+2}$ (efficient reconstruction)	[Choudhury. PODC' 12] $\lvert V_i \rvert = \lvert S \rvert (n-t)^{3n} / \epsilon^{3n}$ [This work] $\lvert V_i \rvert = \lvert S \rvert (k-t)^{k+t+2} / \epsilon^{k+t+2}$

small improvement

tradeoff result

smallest share size

13

Our construction for (t, ε)-CISS

- Dealer authenticates each share using MAC
- Send the share and tag to each player, while sending the verification key to other players
- Determine the cheaters by a majority voting
  - Dealer honest, rushing adversary, t < k/2 and the adversary only corrupts the player showing up in the reconstruction.

votes got by P_1

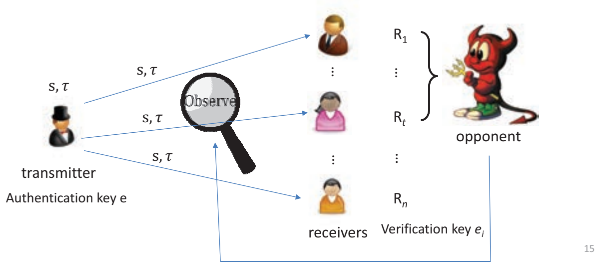
	P_1	P_2	P_3	P_4	P_5
P_1	√	√	X	X	X
P_2	√	√	X	X	X
P_3	X	X	√	√	√
P_4	?	?	√	√	√
P_5	?	?	√	√	√

P_1, P_2 are cheaters

14

Techniques

- (t,n) Multi-receiver Message Authentication Code [Desmedt et.al., Infocom'92]
  - One transmitter, one opponent, multiple receivers



15

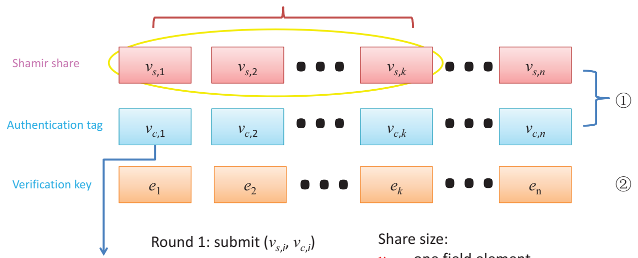
Techniques

- (t,n,w) Multi-receiver multi-message Authentication Code [Safavi-Naini&Wang, EUROCRYPT'98]
  - Authentication Key: (owned by transmitter)
    w+1 polynomials $e=(P_0(x), \ldots, P_w(x))$ of degree at most t
  - Verification Key: (owned by each receiver)
    $e_i = P_0(x_i), \ldots, P_w(x_i)$ for i=1,…,n
  - Authentication tag: (to be broadcast)
    for a message s, $A_s(x) = P_0(x) + sP_1(x) + \ldots + s^w P_w(x)$
  - Verification: (conducted by each receiver)
    $A_s(x_i) \stackrel{?}{=} P_0(x_i) + sP_1(x_i) + s^w P_w(x_i)$
  - Property: The probability that t corrupt receivers and/or the outside opponent upon seeing up to w messages and their corresponding tags succeed in deceiving any receiver R_i is at most $1/q$.

$s, x_i \in F_q$
$P(x) \in F_q[x]$

16

Our Proposal



(t,n,k)-MAC

Round 1: submit $(v_{s,i}, v_{c,i})$

Round 2: submit e_i

for i in $[k]$:
  for j in $[k]$:
    use e_j to verify $(v_{s,i}, v_{c,i})$
if less than t+1 keys admit $(v_{s,i}, v_{c,i})$
  put player R_i in the cheater list **L**

Share size:
$v_{s,i}$ -- one field element
$v_{c,i}$ -- polynomial of degree at most t
e_i -- evaluation of k+1 polynomials

$$|V_i| = q^{1+t+1+k+1}$$
$$|S| = q, \epsilon = (k-t)/q$$
$$|V_i| = |S|(k-t)^{k+t+2}/\epsilon^{k+t+2}$$

17

Apply to RSS

- We can also apply the same idea of multi-receiver multi-message authentication to RSS.
  - Cevallos et al [Cevallos, Fehr, Ostrovsky & Rabani, EUROCRYPT'12] observed that a clever reconstruction algorithm in RSS can reduce the share size.
  - The observation is simple: instead of accept a share with majority votes, accept a share as authentic iff it is accepted by t+1 honest players whose shares are considered as authentic.

Cevallos et al. scheme: (t, δ) RSS

- Dealer authenticates each share using MAC (pairwisely)
- Send the share and tag to each player, while sending the verification key to other players
- Use the new reconstruction to recover the secret
- Theorem [Cevallos et al]: If MAC is ε-secure, then $\delta \leq e \cdot \left((t+1) \cdot \epsilon\right)^{(t+1)/2}$

Using multi-receiver authentication instead

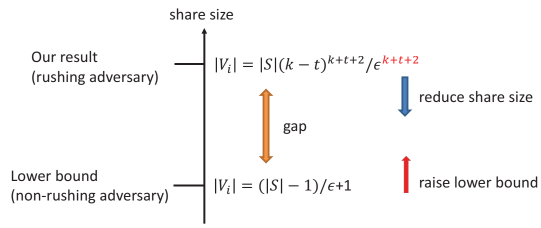
- Dealer authenticates each share using MAC (multi-receiver multi-message authentication)
- Send the share and tag to each player, while sending the verification key to other players
- Use the new reconstruction to recover the secret
- Overhead comparison: $\delta = 2^{-\lambda}, n = 2t+1, m$ is the bit length of the secret
  - Cevallos et al.: $12\lambda + 3n(\log(t+1) + \log(m) + 3)$
  - This one: $6\lambda + 1.5n(\log(t+1) + 3)$ One problem: Not flexible

λ depends on m

Length of redundant information for the purpose of robustness

Open Questions

- Lower bound for rushing adversary
- More compact share size?



21

Thank you!

22

High-Throughput Secure Computation using bit slicing

Toshinori ARAKI Joint work with J. Furukawa, Y. Lindell, A. Nof, K. Ohara.

NEC Corporation

t-araki@ek.jp.nec.com

This talk is about the result of [1]. We describe a new information-theoretic protocol (and a computationally-secure variant) for secure *three*-party computation with an honest majority. The protocol has very minimal computation and communication; for Boolean circuits, each party sends only a single bit for every AND gate (and nothing is sent for XOR gates). This protocol is efficiently parallelizable by using bit slicing method. This protocol is (simulation-based) secure in the presence of semi-honest adversaries, and achieves privacy in the client/server model in the presence of malicious adversaries.

We ran our implementation on a cluster of three mid-level servers connected by a 10Gbps LAN with a ping time of 0.13 ms. Each server has two Intel Xeon E5-2650 v3 2.3GHz CPUs with a total of 20 cores. On a cluster of three 20-core servers with a 10Gbps connection, the implementation of our protocol carries out over *1.3 million* AES computations per second, which involves processing over *7 billion gates per second*. Moreover, we developed a Kerberos extension that replaces the ticket-granting-ticket encryption on the Key Distribution Center (KDC) in MIT-Kerberos with our protocol, using keys/ passwords that are shared between the servers. This enables the use of Kerberos while protecting passwords. Our implementation is able to support a login storm of over 35,000 logins per second, which suffices even for very large organizations. Our work demonstrates that high-throughput secure computation is possible on standard hardware.

Cores	AES/sec	Latency	CPU %	Network
1	100,103 ± 1632	128.5 ± 2.1	73.3%	0.572
5	530,408 ± 7219	121.2 ± 1.7	62.2%	2.99
10	975,237 ± 3049	131.9 ± 0.4	54.0%	5.47
16	1,242,310 ± 4154	165.7 ± 0.4	49.5%	6.95
20	1,324,117 ± 3721	194.2 ± 0.9	49.6%	7.38

Table 1. Experiment results running AES-CTR. The CPU column shows the average CPU utilization per core, and the network column is in Gbps per server. Latency is given in milliseconds.

References

[1] T. Araki, J. Furukawa, Y. Lindell, Ariel Nof, K. Ohara. High-Thrhoughout Semi-Honest Secure Three-Party Computation with an Honest Majority. ACM CCS 2016.

\Orchestrating a brighter world NEC

Secret Sharing for Dependability, Usability and Security of Network Storage and Its Mathematical Modeling

High-Throughput Secure Computation using bit slicing

2016 /9/7
Toshinori Araki (NEC)

About this talk

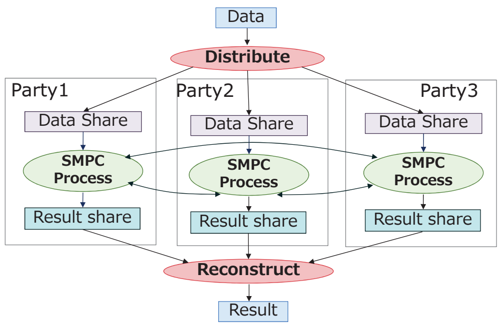
▌ This talk is about the following paper and demo.

- **High-Throughput Semi-Honest Secure Three-Party Computation with an Honest Majority(ACM-CCS 2016)** Toshinori Araki, Jun Furukawa (NEC), Yehuda Lindell, Ariel Nof (Bar-llan University) and Kazuma Ohara (NEC)
  → https://eprint.iacr.org/2016/768

- **DEMO : High-Throughput Secure Three-Party Computation of Kerberos Ticket Generation (ACM-CCS2016)**
  Toshinori Araki (NEC Corporation), Assaf Barak (Bar-Ilan University), Jun Furukawa (NEC Corporation), Yehuda Lindell (Bar-Ilan University), Ariel Nof (Bar-Ilan University) and Kazuma Ohara (NEC Corporation)

What is Secure Multi-party Computation(SMPC)?

▌ SMPC enable us to compute with respect to secret shared data without revealing data & result to Parties hold shared data.



Summary

- We developed **New SMPC protocol** for achieving High throughput.
  - ●Secure three party computation with an honest majority.
  - ●This scheme is secure in the presence of semi-honest adversary.
- By using this scheme, we can process **1.3 million AES** per sec.
  - ●This is corresponding to **40,000** Login processes of Kerberos,
  - ●This performance is sufficient even for very large organization.

The Performance of AES computation by SMPC

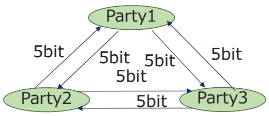
Year		Latency	Throughput
2010	I. Damgard , M. Keller.	2000sec	-
2012	J. Launchbury, I.S. Diatchki, T. DuBuisson , A. Adams-Moran.	14.28 msec	320/sec
2013	S. Laur, R. Talviste J. Willemson.	323 msec	3,450/sec
2016	R. Talviste	223 msec	25,000/sec
2016	Sharemind	-	90,000/sec
2016	**This work**	194 msec	**1,324,117 /sec**

Our approach for achieving High-Throughput

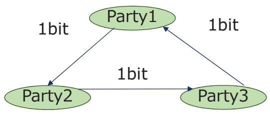
- We have tried to reduce the amount of communication.
  - ●Du-Atallah protocol (Sharemind uses)
    - •Each party sends 10 bit per AND gate.
    - •XOR gate is free from communication.
    - •Assuming AES circuit has 5000 AND gates and parties are connected by 10Gbps band, 200,000 AES per sec is the limit.

➡ Our goal is breaking this limit.

  - ●Our protocol
    - •Each party sends only 1 bit per AND gate.
    - •XOR gate is free from communication.
    - •Specialized in (2,3) access structure.



Du-Atallah Protocol



Our Protocol

About our scheme

- Secret Sharing
- Exclusive OR gates
- AND gates
- Parallelization

Secret Sharing

Share Generation $v \in \{0,1\}$

- Choose a_1, a_2, a_3 such that $a_1 \oplus a_2 \oplus a_3 = v$.
- Compute following values.
  - P_1's share : $(x_1 = a_3 \oplus a_1, a_1)$
  - P_2's share : $(x_2 = a_1 \oplus a_2, a_2)$
  - P_3's share : $(x_3 = a_2 \oplus a_3, a_3)$

Secret Reconstruction

- From any combination of two share, (a_1, a_2, a_3) can get.

Properties

- The sum of former part is equal to 0.
  - $x_1 \oplus x_2 \oplus x_3 = \boxed{a_3} \oplus \boxed{a_1 \oplus a_1} \oplus \boxed{a_2 \oplus a_2} \oplus \boxed{a_3}$
- The sum of latter part is equal to v.
  - $a_1 \oplus a_2 \oplus a_3 = v$

XOR gates

Input for computing $v \oplus w$ $(v = a_1 \oplus a_2 \oplus a_3,\ w = b_1 \oplus b_2 \oplus b_3)$

	Shares of v		Shares of w
P_1's input :	(x_1, a_1) where $x_1 = a_3 \oplus a_1$	,	(y_1, b_1) where $y_1 = b_3 \oplus b_1$
P_2's input :	(x_2, a_2) where $x_2 = a_1 \oplus a_2$	,	(y_2, b_2) where $y_2 = b_1 \oplus b_2$
P_3's input :	(x_3, a_3) where $x_3 = a_2 \oplus a_3$	,	(y_3, b_3) where $y_3 = b_2 \oplus b_3$

XOR gate computation (0 replace with 3)

- Each P_i computes $(z_i, c_i) = (x_i \oplus y_i, a_i \oplus b_i)$
- $z_1 \oplus z_2 \oplus z_3 = \underbrace{a_3 \oplus a_1 \oplus a_1 \oplus a_2 \oplus a_2 \oplus a_3}_{0} \oplus \underbrace{b_3 \oplus b_1 \oplus b_1 \oplus b_2 \oplus b_2 \oplus b_3}_{0}$
- $c_1 \oplus c_2 \oplus c_3 = \underbrace{a_1 \oplus a_2 \oplus a_3}_{v} \oplus \underbrace{b_1 \oplus b_2 \oplus b_3}_{w}$
- Then, $(z_i,\ c_i)$ is the P_i's share of $v \oplus w$.

This computation can be done by each party without communication

AND gates[1/3]

Input for computing $v \cdot w$ $(v = a_1 \oplus a_2 \oplus a_3,\ w = b_1 \oplus b_2 \oplus b_3)$

	Shares of v		Shares of w
P_1's input :	(x_1, a_1) where $x_1 = a_3 \oplus a_1$	,	(y_1, b_1) where $y_1 = b_3 \oplus b_1$
P_2's input :	(x_2, a_2) where $x_2 = a_1 \oplus a_2$	,	(y_2, b_2) where $y_2 = b_1 \oplus b_2$
P_3's input :	(x_3, a_3) where $x_3 = a_2 \oplus a_3$	,	(y_3, b_3) where $y_3 = b_2 \oplus b_3$

AND gate computation (4 replace with 1. 0 replace with 3 .)

- Now suppose P_i has α_i such that $\alpha_1 \oplus \alpha_2 \oplus \alpha_3 = 0$.
- Each P_i computes $r_i = x_i y_i \oplus a_i b_i \oplus \boxed{\alpha_i}$ sends r_i to P_{i+1}.
  - Each party sends only 1bit!.
  - α_i is used as mask. P_{i+1} can not get additional information from r_i.
- Each P_i computes $(z_i,\ c_i) = (r_{i-1} \oplus r_i, r_i)$.
- Then $(z_i,\ c_i)$is P_i's share of $v \oplus w$.

Clearly, $z_1 \oplus z_2 \oplus z_3 = r_3 \oplus r_1 \oplus r_1 \oplus r_2 \oplus r_2 \oplus r_3 = 0$.

AND gates[2/3]

- Confirming $r_1 \oplus r_2 \oplus r_3 = v \cdot w$.

$$
\begin{aligned}
r_1 \oplus r_2 \oplus r_3 &= \boxed{x_1 y_1} \oplus \boxed{a_1 b_1} \oplus \boxed{\alpha_1} \oplus \boxed{x_2 y_2} \oplus \boxed{a_2 b_2} \oplus \boxed{\alpha_2} \oplus \boxed{x_3 y_3} \oplus \boxed{a_3 b_3} \oplus \boxed{\alpha_3} \\
&= \boxed{a_1 b_1 \oplus a_2 b_2 \oplus a_3 b_3} \oplus x_1 y_1 \oplus x_2 y_2 \oplus x_3 y_3 \oplus \underbrace{\alpha_1 \oplus \alpha_2 \oplus \alpha_3}_{0} \\
&= \boxed{a_1 b_1 \oplus a_2 b_2 \oplus a_3 b_3} \oplus (a_3 \oplus a_1)(b_3 \oplus b_1) \oplus (a_1 \oplus a_2)(b_1 \oplus b_2) \oplus (a_2 \oplus a_3)(b_2 \oplus b_3) \\
&= a_1 b_1 \oplus a_2 b_2 \oplus a_3 b_3 \oplus \\
&\quad a_1 b_1 \oplus a_3 b_3 \oplus a_3 b_1 \oplus a_1 b_3 \oplus \\
&\quad a_1 b_1 \oplus a_2 b_2 \oplus a_1 b_2 \oplus a_2 b_1 \oplus \\
&\quad a_2 b_2 \oplus a_3 b_3 \oplus a_2 b_3 \oplus a_3 b_2 \\
&= a_1 b_1 \oplus a_2 b_2 \oplus a_3 b_3 \oplus a_1 b_2 \oplus a_2 b_1 \oplus a_1 b_3 \oplus a_3 b_1 \oplus a_2 b_3 \oplus a_3 b_2 \\
&= (a_1 \oplus a_2 \oplus a_3)(b_1 \oplus b_2 \oplus b_3) \\
&= v \cdot w
\end{aligned}
$$

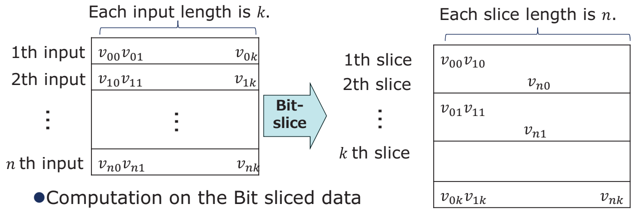
$a_i b_i$: twice
$a_i b_j (i \neq j)$: once

AND gates[3/3]

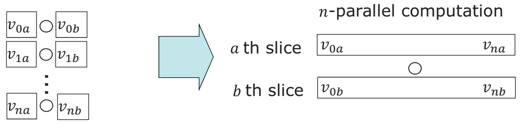
- How to generate $\alpha_1 \oplus \alpha_2 \oplus \alpha_3 = 0$ non-interactively.
  - F_k is pseudorandom function outputting a single bit. k is key.
- Init
  - Each P_i chooses a random $k_i \in \{0,1\}^\kappa$ (κ is security parameter).
  - Each P_i sends k_i to P_{i-1} . (0 replace with 3)
  - After that,
    - P_1 has (k_1, k_2).
    - P_2 has (k_2, k_3).
    - P_3 has (k_3, k_1).
- GenRandom: Given a unique identifier id,
  - P_1 computes $\alpha_1 = F_{k_1}(id) \oplus F_{k_2}(id)$.
  - P_2 computes $\alpha_2 = F_{k_2}(id) \oplus F_{k_3}(id)$.
  - P_3 computes $\alpha_3 = F_{k_3}(id) \oplus F_{k_1}(id)$.
- Note:
  - $\alpha_1 \oplus \alpha_2 \oplus \alpha_3 = \boxed{F_{k_1}(id)} \oplus \boxed{F_{k_2}(id) \oplus F_{k_2}(id)} \oplus \boxed{F_{k_3}(id) \oplus F_{k_3}(id)} \oplus \boxed{F_{k_1}(id)} = 0$

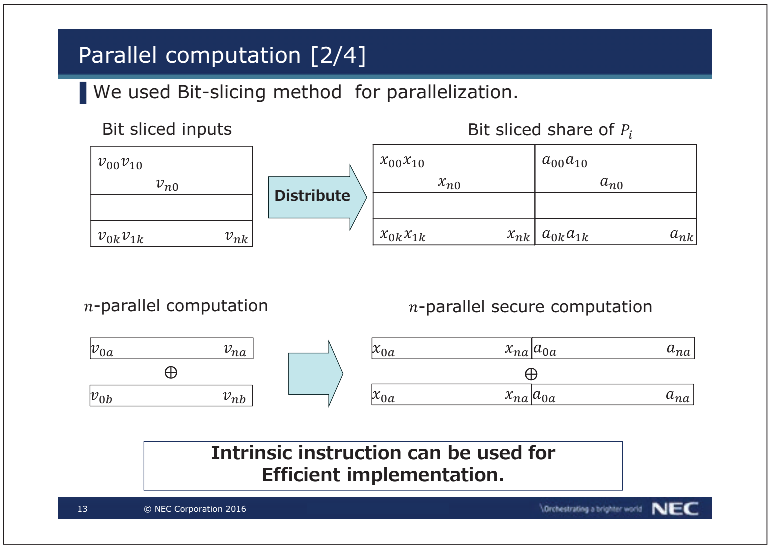
Parallel computation [1/4]

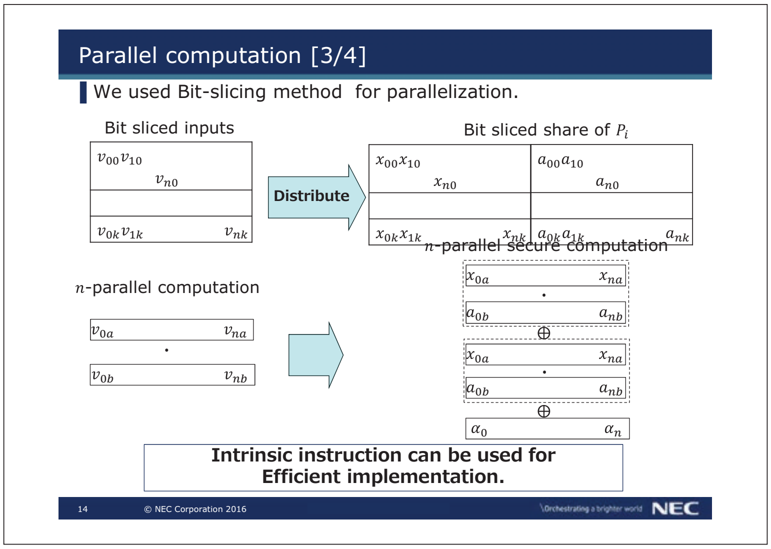
- We used Bit-slicing method for parallelization.
  - Bit-slicing

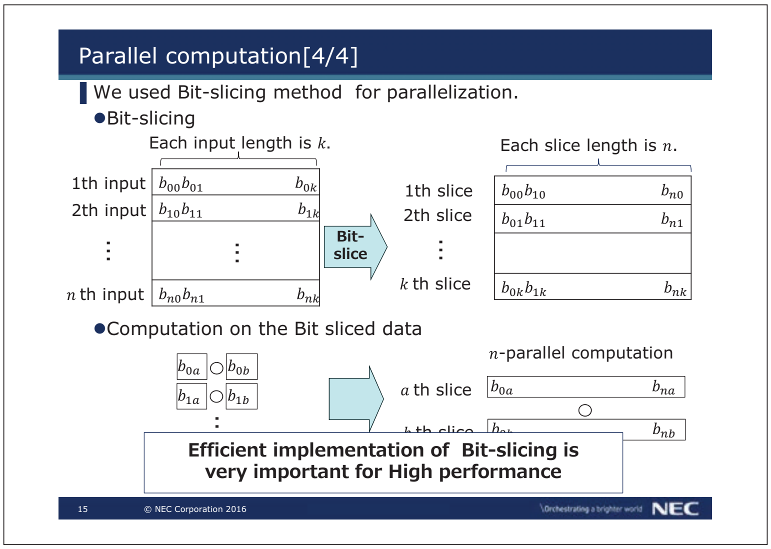


  - Computation on the Bit sliced data




Parallel computation [2/4]
We used Bit-slicing method for parallelization.
Bit sliced inputs
$v_{00} v_{10}$ v_{n0}
$v_{0k} v_{1k}$ v_{nk}
Distribute
Bit sliced share of P_i
$x_{00} x_{10}$ x_{n0}
$x_{0k} x_{1k}$ x_{nk}
$a_{00} a_{10}$ a_{n0}
$a_{0k} a_{1k}$ a_{nk}
n-parallel computation
v_{0a} v_{na}
$\oplus$
v_{0b} v_{nb}
n-parallel secure computation
x_{0a} x_{na} a_{0a} a_{na}
$\oplus$
x_{0a} x_{na} a_{0a} a_{na}
Intrinsic instruction can be used for Efficient implementation.
13
© NEC Corporation 2016
\Orchestrating a brighter world
NEC


Parallel computation [3/4]
We used Bit-slicing method for parallelization.
Bit sliced inputs
$v_{00} v_{10}$ v_{n0}
$v_{0k} v_{1k}$ v_{nk}
Distribute
Bit sliced share of P_i
$x_{00} x_{10}$ x_{n0}
$x_{0k} x_{1k}$ x_{nk}
$a_{00} a_{10}$ a_{n0}
$a_{0k} a_{1k}$ a_{nk}
n-parallel computation
v_{0a} v_{na}
$\cdot$
v_{0b} v_{nb}
n-parallel secure computation
x_{0a} x_{na}
$\cdot$
a_{0b} a_{nb}
$\oplus$
x_{0a} x_{na}
$\cdot$
a_{0b} a_{nb}
$\oplus$
α_0 α_n
Intrinsic instruction can be used for Efficient implementation.
14
© NEC Corporation 2016
\Orchestrating a brighter world
NEC

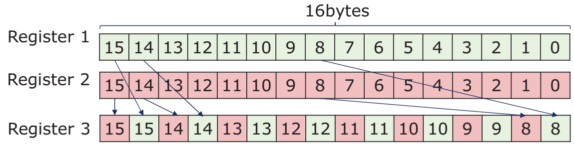

Parallel computation[4/4]
We used Bit-slicing method for parallelization.
●Bit-slicing
Each input length is k.
1th input $b_{00} b_{01}$ b_{0k}
2th input $b_{10} b_{11}$ b_{1k}
n th input $b_{n0} b_{n1}$ b_{nk}
Bit-slice
Each slice length is n.
1th slice $b_{00} b_{10}$ b_{n0}
2th slice $b_{01} b_{11}$ b_{n1}
k th slice $b_{0k} b_{1k}$ b_{nk}
●Computation on the Bit sliced data
b_{0a} ○ b_{0b}
b_{1a} ○ b_{1b}
n-parallel computation
a th slice b_{0a} b_{na}
○
b_{nb}
Efficient implementation of Bit-slicing is very important for High performance
15
© NEC Corporation 2016
\Orchestrating a brighter world
NEC

Implementation of Bit-slicing [1/3]

▌We have implemented Bit-slicing using Intel Intrinsics.
- Mainly, we used **unpack** and **movmskb.**
- The unit of our bit-slicing is 16 messages of length 8 bytes.

$$m_0 = \overbrace{(m_{0,0}, m_{0,1}, m_{0,2}, m_{0,3}, m_{0,4}, m_{0,5}, m_{0,6}, m_{0,7})}^{\text{8bytes}}$$
$$m_1 = (m_{1,0}, m_{1,1}, m_{1,2}, m_{1,3}, m_{1,4}, m_{1,5}, m_{1,6}, m_{1,7})$$
$$\vdots$$
$$m_{15} = (m_{15,0}, m_{15,1}, m_{15,2}, m_{15,3}, m_{15,4}, m_{15,5}, m_{15,6}, m_{15,7})$$

▌Unpack(VPUNPCHBW)
- This instruction can be used for mixing two input.



Implementation of Bit-slicing [2/3]

▌We have implemented Bit-slicing using Intel Intrinsics.
- Mainly, we use **unpack** and **movmskb.**
- The unit of our bit-slicing is 16 messages of length 8 bytes.

$$m_0 = \overbrace{(m_{0,0}, m_{0,1}, m_{0,2}, m_{0,3}, m_{0,4}, m_{0,5}, m_{0,6}, m_{0,7})}^{\text{8bytes}}$$
$$m_1 = (m_{1,0}, m_{1,1}, m_{1,2}, m_{1,3}, m_{1,4}, m_{1,5}, m_{1,6}, m_{1,7})$$
$$\vdots$$
$$m_{15} = (m_{15,0}, m_{15,1}, m_{15,2}, m_{15,3}, m_{15,4}, m_{15,5}, m_{15,6}, m_{15,7})$$

▌Unpack(VPUNPCHBW)
- By applying 32 unpack instruction, Byte-sliced data can be made.

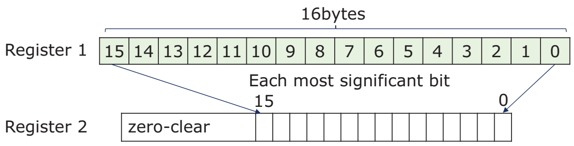
$$m'_0 = \overbrace{(m_{0,0}, m_{1,0}, m_{2,0}, \ldots, m_{15,0})}^{\text{16bytes}}$$
$$m'_1 = (m_{0,1}, m_{1,1}, m_{2,1}, \ldots, m_{15,1})$$
$$\vdots$$
$$m'_7 = (m_{0,0}, m_{1,1}, m_{2,1}, \ldots, m_{15,7})$$

Implementation of Bit-slicing [3/3]

▌Movmskb instruction can be used for making **bit-sliced data** from **byte-sliced data**.

$$m'_0 = (m_{0,0}, m_{1,0}, m_{2,0}, \ldots, m_{15,0})$$
$$m'_1 = (m_{0,0}, m_{1,1}, m_{2,1}, \ldots, m_{15,1})$$
$$\vdots$$
$$m'_7 = (m_{0,0}, m_{1,1}, m_{2,1}, \ldots, m_{15,7})$$

▌movmskb



- Register 2 contains bit-sliced data.
- By applying 64 times **movmskb** and **shift**, Bit-sliced data can be made.

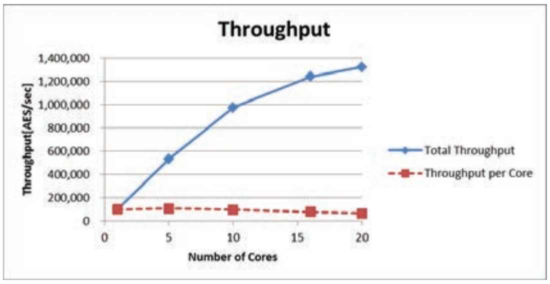
Experiment : Performance[1/3]

- Server
  - ●CPU : Two Intel Xeon E5-2650 v3 2.3GHz (Total 20 cores)
  - ●Network : 10Gbps LAN with a ping time of 0.13 ms
- Encryption scheme
  - ●AES-128 using expanded key
  - ●These computations can be with different keys and plaintexts .
  - ●Mode of operation is AES-CTR

Cores	AES/sec	Latency (ms)	CPU %	Network(Gbps)
1	100,103	128.5	73.3%	0.572
5	530,408	121.2	62.2%	2.99
10	975,237	131.9	54.0%	5.47
16	1,242,310	165.7	49.5%	6.95
20	1,324,117	194.2	49.6%	7.38

Experiment : Throughput per core[2/3]

- Up to 10 cores, the throughput is stable at approximately 100,000 AES/sec per core.

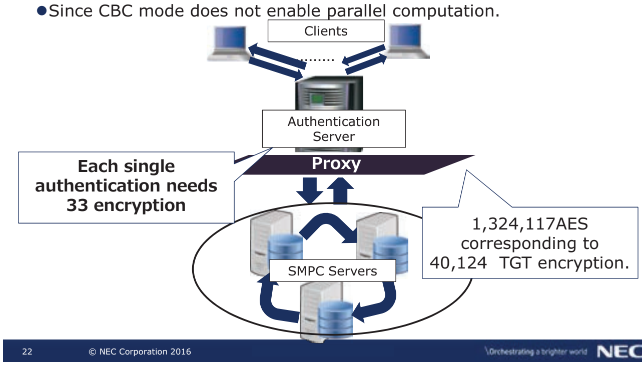


Experiment : Micro Benchmark[3/3]

Protocol part	Percentage
Server bitslice and deslice	8.70%
AND and XOR gate computation	49.82%
Randomness generation	9.54%
Comm. delays between MPC servers	27.87%
Communication delays for input/output	4.07%

Applied to Kerberos authentication server

- We took the Open Source MIT Kerberos
  - We modified the encryption mode used to encrypt the TGT to counter mode.
  - Since CBC mode does not enable parallel computation.



  \Orchestrating a brighter world NEC

Summary

We developed SMPC protocol for achieving High throughput.

- Proposed scheme can process 1.3 million AES/sec.
- This throughput corresponds to 40,000 Kerberos Authentication.
  - Single authentication needs 33 AES computations.
- This performance is sufficient even for very large organization.

  \Orchestrating a brighter world NEC

\Orchestrating a brighter world

NEC

XOR-based $(2, 2^m)$ threshold schemes

Yuji SUGA

Internet Initiative Japan Inc.

suga@iij.ad.jp

The (k,n)-threshold secret sharing schemes using exclusive-OR operations (XOR-(k,n)-SSS) are proposed by Fujii et al. and Kurihara et al. [1] independently. Their method are ideal that share size is equal to the size of the data to be distributed with the benefits that can be handled very fast for using only XOR operation at distribution and restoration processes.

A new method proposed in WAIS2013 [2]**:** A new method have proposed, this leads to general constructions of $(2, p+1)$-threshold secret sharing schemes using only exclusive-OR operations with the same assumption of previous XOR-(k,n)-SSS.

Example 1 (XOR-$(2,4)$-SSS [2])**.** $M = M_1||\ M_2\ (n' = 2), M_0 \in \{0\}^d$

W_0	$M_0 \oplus R_0$	$M_1 \oplus M_2 \oplus R_1$
W_1	$M_1 \oplus M_2 \oplus R_0$	$M_1 \oplus R_1$
W_2	$M_1 \oplus R_0$	$M_0 \oplus R_1$
W_3	$M_2 \oplus R_0$	$M_2 \oplus R_1$

Definition 2 (2-propagation bases set defined in [3])**.** *2-propagation bases set* $\{b_i\}(i = 1, \ldots, l)$ *is a set of bases over* $\mathbb{Z}_2^m$ *satisfies the following properties:* b_1 *is a set of* m *zero-vectors and for all distinct two bases* b_i, b_j*,* $b_i + b_j$ *is also a basis over* $\mathbb{Z}_2^m$*.*

Theorem 3 (Main Theorem)**.** *When an optimal 2-propagation bases set* $\{b_i\}$ $(i = 1, \ldots, 2^m)$ *over* $\mathbb{Z}_2^m$*, these exists an XOR-*$(2, 2^m)$*-SSS with vector-representation* $\{w_{ij} = b_i^j\}$ $(i = 1, \ldots, 2^m,\ i = 1, \ldots, m)$*.*

Proof. From the definition of 2-propagation bases set, for distinct u, v, $b_u + b_v$ is a basis, so $w_1^* = w_{u1} + w_{v1}\ , \ldots, w_m^* = w_{um} + w_{vm}$ are bases over $\mathbb{Z}_2^m$. The l-th element of $W_u \oplus W_v$ equals $\bigoplus_{s=1}^m w_l^{*(s)} M_s$. In this case, these exist m linearly independent simultaneous equations for $M_s (s = 1, \ldots, m)$, so we can reconstruct all M_s.

Example 4 ($m = 4 :$ XOR-$(2, 2^4)$-SSS)**.**

W_0	$(0,0,0,0)$	$(0,0,0,0)$	$(0,0,0,0)$	$(0,0,0,0)$
W_1	$(1,0,0,0)$	$(0,1,0,0)$	$(0,0,1,0)$	$(0,0,0,1)$
W_2	$(1,1,0,0)$	$(1,0,0,0)$	$(0,0,1,1)$	$(0,0,1,0)$
W_3	$(0,0,1,1)$	$(1,0,0,1)$	$(0,1,1,0)$	$(0,1,0,0)$
W_4	$(0,1,0,1)$	$(0,1,1,0)$	$(1,1,0,0)$	$(1,0,0,0)$

References

[1] J. Kurihara, S. Kiyomoto, K. Fukushima, T. Tanaka, "On a Fast (k, n)-Threshold Secret Sharing Scheme", IEICE Trans. on Fundamentals, vol.E91-A, no.9, 2008.

[2] Y. Suga, "New Constructions of (2,n)-Threshold Secret Sharing Schemes Using Exclusive-OR Operations", The 7th International Workshop on Advances in Information Security (WAIS2013), 2013.

[3] Y. Suga, "Consideration of the XOR-operation based Secure Multiparty Computationg", The Ninth International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS2015), 2015.

IMI Workshop: Secret Sharing for Dependability, Usability and Security of Network Storage and Its Mathematical Modeling, September 7th, 2016, Momochi-hama, Japan

XOR-based (2, 2^m) threshold schemes

Yuji SUGA
Internet Initiative Japan Inc.

Agenda

- Previous XOR-based secret sharing schemes
  - Using circulant permutation matrices
- A new proposal : (2,2^m)-SSS using m-dimensional vector spaces over Z_2

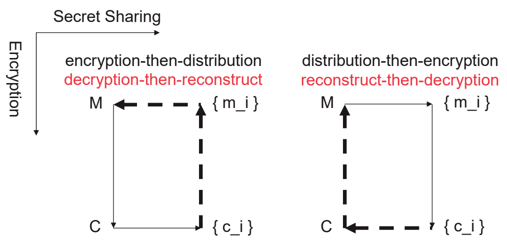
XOR-based SSS

- Very fast (k,n)-threshold secret sharing
  - uses only XOR operation in both of the distribution phase and reconstruction phase.
  - proposed by KDDI and Toshiba Solutions independently.
- From 2012, IIJ also proposed similar schemes.

Allowing cyclic flow

- Assume that 2 operations are <u>commutative</u>

Encryption: CTR mode or stream cipher



Type-E（Counterclockwise）　　Type-F（Clockwise）

A toy example: XOR-(2,3)-SSS

KDDI

Secret M is divided into M_i's

where M=M_1 || M_2 and M_0=Zero-bit-binary

|M_1| = |M_2| = |M_0| = d

W_0	$(M_0 \oplus R_0) \parallel (M_2 \oplus R_1)$
W_1	$(M_1 \oplus R_0) \parallel (M_0 \oplus R_1)$
W_2	$(M_2 \oplus R_0) \parallel (M_1 \oplus R_1)$

for random data R_0, R_1.

|R_0| = |R_1| = d

Kurihara et.al, On a fast (k,n)-threshold secret sharing scheme, IEICE TRANS. FUNDAMENTALS, VOL.E91-A, No.9, 2008.

A toy example: XOR-(2,3)-SSS

KDDI

F({W0,W1}) = {M1, M2}
F({W0,W2}) = {M2, M1+M2}
F({W1,W2}) = {M1+M2, M1}

A not-strictly-defined function F()
outputs the data to be recovered in each part.

|M_1| = |M_2| = |M_0| = d

W_0	$(M_0 \oplus R_0) \parallel (M_2 \oplus R_1)$
W_1	$(M_1 \oplus R_0) \parallel (M_0 \oplus R_1)$
W_2	$(M_2 \oplus R_0) \parallel (M_1 \oplus R_1)$

for random data R_0, R_1.

|R_0| = |R_1| = d

Kurihara et.al, On a fast (k,n)-threshold secret sharing scheme, IEICE TRANS. FUNDAMENTALS, VOL.E91-A, No.9, 2008.

Pros./Cons. of KDDI methods

- FAST!! because using only XOR-op.
- For all (k,n), there exist XOR-(k,n)-SSS
  - # of the number of pieces of block is n-1

- Target data must be equally divided into p-1 pieces where p is a prime of more than n
  - XOR-(2,4)-SSS is from XOR-(2,5)-SSS

7

Our (previous) contributions in WAIS2013

- (1) # of divisions for the original data is able to be less than n-1
- (2) the size of the share is able to be smaller than the size of target data
- (3) makes it possible to select the number of shares other than prime numbers

- A (3,2,4) ramp secret sharing scheme proposed by Matsumoto et al. announced in SCIS2012

Yuji Suga ,"New Constructions of (2, n)-Threshold Secret Sharing Schemes Using Exclusive-OR Operations", Seventh International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing, 2013

8

Proposal-1(New XOR-(k,n)-SSS)

- $W_{i0} := M_1 \oplus M_{n'+2-i} \oplus R_0 (i = 1, \ldots, n')$

  where indexes are calculated as $\mod n_p$ So, we got $W_{00} = R_0, W_{10} = M_1 \oplus R_0)$
- $W_{0j} := M_1 \oplus M_{j+1} \oplus R_j \ \ (j = 1, \ldots, n' - 1)$
- $W_{1j} := W_{0,j-1} \oplus R_{j-1} \oplus R_j \ \ (j = 1, \ldots, n' - 1)$
- $W_{ij} := W_{i-1,j-1} \oplus R_{j-1} \oplus R_j \ \ (i = 1, \ldots, n', j = 1, \ldots, n' - 1)$
- $W_{n'+1,j} := M_2 \oplus, \ldots, \oplus M_{n'} \oplus R_j \ \ (j = 0, \ldots, n' - 1)$

9

An example proposed in WAIS2013

- XOR-(2,4)-SSS with $n' = 2 \neq n - 1 = 3$

- W1: M0 + R0 || M1 + M2 + R1
- W2: M1 + M2 + R0 || M1 + R1
- W3: M1 + R0 || M0 + R1
- W4: M2 + R0 || M2 + R1

An example proposed in WAIS2013

- W0: M0 + R0 || M1 + M2 + R1
- W1: M1 + M2 + R0 || M1 + R1
- W2: M1 + R0 || M0 + R1
- W3: M2 + R0 || M2 + R1

F({W0,W1}) = {M1+M2, M2}
F({W0,W2}) = {M1, M2}
F({W0,W3}) = {M2, M1}
F({W1,W2}) = {M2, M1}
F({W1,W3}) = {M1, M1+M2}
F({W2,W3}) = {M1+M2, M2}

Introduction of a concept "isomorphism" for XOR-SSS

- For an XOR-(2,n)-SSS Ψ with matrix-representation of W_ij,
- an XOR-(2,n)-SSS following operation

W_0	$(M_0 \oplus R_0) \,\|\|\, (M_2 \oplus R_1)$
W_1	$(M_1 \oplus R_0) \,\|\|\, (M_0 \oplus R_1)$
W_2	$(M_2 \oplus R_0) \,\|\|\, (M_1 \oplus R_1)$

(1) Replace a line with some other line.

W_0	$M_0 \oplus R_0$	$M_1 \oplus R_1$	$\oplus R_1'$
W_1	$M_2 \oplus R_0$	$M_0 \oplus R_1$	$\oplus R_1'$
W_2	$M_1 \oplus R_0$	$M_2 \oplus R_1$	$\oplus R_1'$

(3) For all sub-shares of a column, add same data with XOR-operations.

Modification of a previous example

- See modified left part is added by M_1.

W_0	$M_0 \oplus R_0$	$M_1 \oplus R_1$
W_1	$M_2 \oplus R_0$	$M_0 \oplus R_1$
W_2	$M_1 \oplus R_0$	$M_2 \oplus R_1$

W_0	$M_0 \oplus R_0$	$M_0 \oplus R_1'$
W_1	$M_2 \oplus R_0$	$M_1 \oplus R_1'$
W_2	$M_1 \oplus R_0$	$M_1 \oplus M_2 \oplus R_1'$

KDDI vs. IIJ in XOR-(2,3)-SSS

- KDDI

W_0	$M_0 \oplus R_0$	$M_0 \oplus R_1'$
W_1	$M_2 \oplus R_0$	$M_1 \oplus R_1'$
W_2	$M_1 \oplus R_0$	$M_1 \oplus M_2 \oplus R_1'$

- IIJ(WAIS2013)

There must be something relevant!

W_0	$M_0 \oplus R_0$	$M_0 \oplus R_1$
W_1	$M_1 \oplus M_2 \oplus R_0$	$M_2 \oplus R_1$
W_2	$M_1 \oplus R_0$	$M_1 \oplus M_2 \oplus R_1$
W_3	$M_2 \oplus R_0$	$M_1 \oplus R_1$

Another representation

- Matrix-representation

$W_{ij} = \bigoplus_{t=1}^{n'} \alpha_t M_t$

$w_{ij} = (\alpha_1, \ldots, \alpha_{n'}) \in \mathbb{Z}_2^{n'}$

	$M_0 \oplus R_0$	$M_0 \oplus R_1$
	$M_1 \oplus M_2 \oplus R_0$	$M_2 \oplus R_1$
	$M_1 \oplus R_0$	$M_1 \oplus M_2 \oplus R_1$
	$M_2 \oplus R_0$	$M_1 \oplus R_1$

- Vector-representation

(Elements of $Z_2^{n'}$)
(Coefficients)

W_0	$(0,0)$	$(0,0)$
W_1	$(1,1)$	$(0,1)$
W_2	$(1,0)$	$(1,1)$
W_3	$(0,1)$	$(1,0)$

W_0	$(0,0)$	$(0,0)$
W_1	$(1,1)$	$(0,1)$
W_2	$(1,0)$	$(1,1)$
W_3	$(0,1)$	$(1,0)$

- We can see that…
  - w10 = w20 + w30 w11 = w21 + w31
  - (1,1) = (1,0) + (0,1) (0,1) = (1,1) + (1,0)

(+ : addition over Z_2^2)

16

W_0	$(0,0)$	$(0,0)$
W_1	$(1,1)$	$(0,1)$
W_2	$(1,0)$	$(1,1)$
W_3	$(0,1)$	$(1,0)$

- And also…
  - {w10, w11} is a basis of Z_2^2

17

W_0	$(0,0)$	$(0,0)$
W_1	$(1,1)$	$(0,1)$
W_2	$(1,0)$	$(1,1)$
W_3	$(0,1)$	$(1,0)$

- And also…
  - {w10, w11} is a basis of Z_2^2
  - {w20, w21} is a basis of Z_2^2
  - {w30, w31} is a basis of Z_2^2

18

New definition for “a set of bases”

b1	$(0,0)$	$(0,0)$
b2	$(1,1)$	$(0,1)$
b3	$(1,0)$	$(1,1)$
b4	$(0,1)$	$(1,0)$

Definition 9 (2-propagation bases set): 2-propagation bases set $\{b_i\}(i = 1, \ldots, l)$ is a set of bases over $\mathbb{Z}_2^m$ satisfies the following properties: b_1 is a set of m zero-vectors and for all distinct two bases b_i, b_j, $b_i + b_j$ is also a basis over $\mathbb{Z}_2^m$.

2-propagation bases set → XOR-SSS

Theorem 11 (Main Theorem): When an optimal 2-propagation bases set $\{b_i\}$ $(i = 1, \ldots, 2^m)$ over $\mathbb{Z}_2^m$, these exists an XOR-$(2, 2^m)$-SSS with vector-representation $\{w_{ij} = b_i^j\}$ $(i = 1, \ldots, 2^m,\ i = 1, \ldots, m)$.

∵ for distinct u, v, b_u + b_v is a basis,
there exist m linearly independent simultaneous equations for M_s.

$\boxed{w_{ij} = (\alpha_1, \ldots, \alpha_{n'}) \in \mathbb{Z}_2^{n'}}$ $\boxed{W_{ij} = \bigoplus_{t=1}^{n'} \alpha_t M_t}$

W_0	$(0,0)$	$(0,0)$
W_1	$(1,1)$	$(0,1)$
W_2	$(1,0)$	$(1,1)$
W_3	$(0,1)$	$(1,0)$

W_0	$M_0 \oplus R_0$	$M_0 \oplus R_1$
W_1	$M_1 \oplus M_2 \oplus R_0$	$M_2 \oplus R_1$
W_2	$M_1 \oplus R_0$	$M_1 \oplus M_2 \oplus R_1$
W_3	$M_2 \oplus R_0$	$M_1 \oplus R_1$

XOR-(2,2^3)-SSS

W_0	$(0,0,0)$	$(0,0,0)$	$(0,0,0)$
W_1	$(1,0,0)$	$(0,1,0)$	$(0,0,1)$
W_2	$(0,1,1)$	$(1,0,0)$	$(0,1,0)$
W_3	$(1,1,0)$	$(0,1,1)$	$(1,0,0)$

W_0	R_0	R_1	R_2
W_1	$M_1 \oplus R_0$	$M_2 \oplus R_1$	$M_3 \oplus R_2$
W_2	$M_2 \oplus M_3 \oplus R_0$	$M_1 \oplus R_1$	$M_2 \oplus R_2$
W_3	$M_1 \oplus M_2 \oplus R_0$	$M_2 \oplus M_3 \oplus R_1$	$M_1 \oplus R_2$
W_4	$M_1 \oplus M_2 \oplus M_3 \oplus R_0$	$M_1 \oplus M_2 \oplus R_1$	$M_2 \oplus M_3 \oplus R_2$
W_5	$M_2 \oplus R_0$	$M_3 \oplus R_1$	$M_1 \oplus M_3 \oplus R_2$
W_6	$M_1 \oplus M_3 \oplus R_0$	$M_1 \oplus M_2 \oplus M_3 \oplus R_1$	$M_1 \oplus M_2 \oplus R_2$
W_7	$M_3 \oplus R_0$	$M_1 \oplus M_3 \oplus R_1$	$M_1 \oplus M_2 \oplus M_3 \oplus R_2$

XOR-$(2,2^4)$-SSS

We implemented search algorithm of 2-propagation bases sets for small m.

W_0	(0,0,0,0)	(0,0,0,0)	(0,0,0,0)	(0,0,0,0)
W_1	(1,0,0,0)	(0,1,0,0)	(0,0,1,0)	(0,0,0,1)
W_2	(1,1,0,0)	(1,0,0,0)	(0,0,1,1)	(0,0,1,0)
W_3	(0,0,1,1)	(1,0,0,1)	(0,1,1,0)	(0,1,0,0)
W_4	(0,1,0,1)	(0,1,1,0)	(1,1,0,0)	(1,0,0,0)

XOR-$(2,2^5)$-SSS

W_0	(0,0,0,0,0)	(0,0,0,0,0)	(0,0,0,0,0)	(0,0,0,0,0)	(0,0,0,0,0)
W_1	(1,0,0,0,0)	(0,1,0,0,0)	(0,0,1,0,0)	(0,0,0,1,0)	(0,0,0,0,1)
W_2	(0,0,1,0,0)	(1,0,0,0,0)	(0,1,0,0,0)	(0,0,0,1,1)	(0,0,0,1,0)
W_3	(1,1,0,0,0)	(1,0,0,0,1)	(0,0,0,1,1)	(0,0,1,1,0)	(0,0,1,0,0)
W_4	(0,0,0,1,0)	(0,0,0,1,1)	(1,0,0,0,0)	(0,1,1,0,0)	(0,1,0,0,0)
W_5	(0,1,1,1,1)	(0,1,1,0,0)	(0,0,0,0,1)	(1,0,1,0,1)	(1,0,0,0,0)

XOR-$(2,2^6)$-SSS

(0,0,0,0,0,0)	(0,0,0,0,0,0)	(0,0,0,0,0,0)	(0,0,0,0,0,0)	(0,0,0,0,0,0)	(0,0,0,0,0,0)
(1,0,0,0,0,0)	(0,1,0,0,0,0)	(0,0,1,0,0,0)	(0,0,0,1,0,0)	(0,0,0,0,1,0)	(0,0,0,0,0,1)
(0,0,0,0,0,1)	(1,0,0,0,0,1)	(0,1,0,0,0,0)	(0,0,1,0,0,0)	(0,0,0,1,0,0)	(0,0,0,0,1,0)
(0,0,0,0,1,0)	(0,0,1,0,0,0)	(1,0,0,0,0,0)	(0,1,0,0,0,0)	(0,0,0,0,1,1)	(0,0,0,1,0,0)
(0,0,0,1,0,1)	(1,0,0,0,1,1)	(1,1,0,0,0,1)	(0,1,0,0,0,1)	(0,0,1,0,0,1)	(0,0,1,0,0,0)
(0,0,1,0,1,0)	(0,0,1,1,0,0)	(0,0,1,0,1,1)	(1,0,0,0,0,0)	(0,1,0,0,1,0)	(0,1,0,0,0,0)
(0,1,0,0,0,1)	(0,1,1,1,0,1)	(1,0,1,1,0,1)	(0,1,0,0,1,0)	(1,0,0,1,0,0)	(1,0,0,0,0,0)

Concluding remarks

- Consideration for next generation cloud
  - Affinity with mobile, Asymmetric cloud
  - New Issues / Problems
- Requirements for deploying SSS
  - Transparency on data flow in cloud
- Coexistence of Confidentiality and Secret sharing
  - By using XOR-based primitives
  - A light-weight proposal from m-dimensional vector spaces over Z_2

25

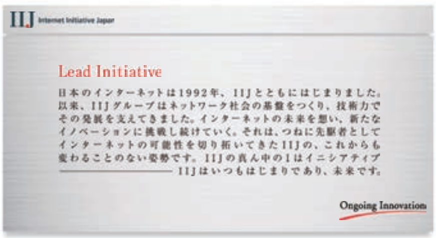
Future works

- In the cases with $k > 2$?
- I believe there exists extended scheme
  - Ex) $GF(3)^m$: random data could be cancelled as

$$R \oplus_3 R \oplus_3 R = 0$$

  - But calculation would be not efficient
    - Needs operations over GF(3)

26



お問い合わせ先　IIJインフォメーションセンター
TEL：03-5205-4466（9：30～17：30 土/日/祝日除く）
info@iij.ad.jp
http://www.iij.ad.jp/

27

Panel Discussion

Secret Sharing in Real-Life Distributed Systems: Perspectives and Challenges

Panelists: Yvo Desmedt, Jon-Lark Kim, Patrick P. C. Lee, Rocki H. Ozaki, Satoshi Obana,

Moderator: Kirill Morozov

The video of our panel discussion is available at “YouTube”:

- Video1: https://youtu.be/gpUOT43FQVM
- Video2: https://youtu.be/AuRBxiKr6lU

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 26 年 10 月
マス・フォア・インダストリ研究所
所長　福本康秀

Secret Sharing for Dependability, Usability and Security of Network Storage and Its Mathematical Modeling

発　行　　2017年３月15日
編　集　穴田啓晃, Kirill Morozov, 須賀祐治, 奥村伸也, 櫻井幸一
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 http://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

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/