論文 / 著書情報
Article / Book Information

題目(和文)	秘密分散とその変換プロトコルに関する研究
Title(English)	A Study on Secret Sharing with Share Conversion
著者(和文)	菊池亮
Author(English)	Ryo Kikuchi
出典(和文)	学位:博士（工学）, 学位授与機関:東京工業大学, 報告番号:甲第9976号, 授与年月日:2015年9月25日, 学位の種別:課程博士, 審査員:尾形　わかは,植松　友彦,山田　功,松本　隆太郎,田中　圭介
Citation(English)	Degree:, Conferring organization: Tokyo Institute of Technology, Report number:甲第9976号, Conferred date:2015/9/25, Degree Type:Course doctor, Examiner:,,,,
学位種別(和文)	博士論文
Category(English)	Doctoral Thesis
種別(和文)	要約
Type(English)	Outline

A Study on Secret Sharing with Share Conversion (Abstract)

Ryo Kikuchi
Supervisor: Wakaha Ogata

Department of Communications and Integrated Systems
Tokyo Institute of Technology

August 17, 2015

1 Background

Secret Sharing (SS) was proposed by Blakley [6] and Shamir [32]. In the SS model, a dealer first divides a *secret* into *shares* and distributes them among *parties*. A qualified coalition of parties can reconstruct the secret, and unqualified coalition of parties can obtain information about that secret. The (t, n)-threshold SS is a common class of SS. In this class, the number of shares is n, any coalition of more than t shares can reconstruct the secret,[1] and t or fewer shares are independent of the secret.

SS has been studied as not only a way to store data securely but also a primitive of other cryptographic protocols such as threshold cryptosystems [18] and fuzzy identity-based encryption for biometrics [31]. Among such applications, multiparty computation (MPC) has been well studied. An MPC aims to compute a function of inputs such as statistical analysis and data mining while any party obtain only the output of the function. Although there are many techniques to construct MPC, we focus on SS-based MPC and we use the term, MPC, as SS-based MPC in the thesis. MPC is typically executed as follows. First, input data are distributed to the parties via SS. When the parties want to compute the function, they interact with each other and obtain a share of the function result. Then the result is reconstructed if needed. Throughout the above steps, the input data are

[1] Notice that $t + 1$ shares are required to reconstruct the secret. In the latter sentences of the thesis, we use t as the maximal number of corrupted parties.

never reconstructed. Therefore, by using MPC, the parties can analyze data without leaking any information of the data except the output.

Although MPC has such attractive sense, early studies on MPC [4, 11] had been mainly considered as theoretical interest due to its inefficiency. However, there have been many studies on MPC e.g., [2, 3, 5, 9, 13, 16, 17, 20, 22–24, 27–29], and recently MPC is considered as practical interest. In fact, some practical implementations have been published and confirmed that MPC is efficient enough for certain applications [8–10].

2 Our Application Scenario

Let us consider the system that uses SS to secure data storage with MPC, which is a common model for MPC. For example, a data aggregation of network traffic statistics [10] belongs to the model. In this application, one first shares data for storing securely, and performs MPC that computes statistical analysis when it is needed. In other words, the system has two roles: Secure data storage and secure analyzing system. From the viewpoint of the secure data storage, the storage-size is desired to be small. It means that the share-size of SS should be small. On the other hand, from the viewpoint of the secure analyzing system, it is essential that the parties can perform MPC on the SS efficiently.

Another application is an MPC system with backup. In the application, one shares data via SS and performs MPC by ordinary. When the data are renewed, old shares are still stored for some purpose such as audit. Also in the application, the share-size of SS should be small and the parties can perform MPC on the SS efficiently.

3 Compatibility of Small Share-size and Efficient MPC

We call an SS whose share-size is small as *compact* and an SS on which the parties can perform MPC efficiently as *MPC-friendly.* Our application scenario requires a compact *and* MPC-friendly SS. We survey existing SSs and discuss if an SS satisfies both simultaneously.

3.1 Compact SS

Several compact SSs have been proposed. One of them is a computationally secure SS. Krawczyk [26] proposed a computationally secure SS that uses symmetric-key encryption and information dispersal algorithm (IDA) [30]. In Krawczyk's scheme, a dealer encrypts a secret with the symmetric-key encryption, the key is distributed through some SS, and the ciphertext is distributed through IDA. If the key size is much smaller than the size of the secret, Krawczyk's scheme is compact. Although a computationally secure SS such as Krawczyk's scheme is secure against only polynomially bounded adversaries, the share-size is almost the same as the optimum one, $\frac{1}{t+1}\|\mathscr{S}\|$ where $\|\mathscr{S}\|$ denotes the size of the secret.

Another compact SS is a ramp scheme that was independently proposed by Blakley and Meadows [7], and Yamamoto [34]. In the ramp scheme, one share can contain multiple secrets so the share-size is small in total. If we set parameters of the ramp scheme so that one share contains L secrets, the share-size is $\frac{1}{L}\|\mathscr{S}\|$.

3.2 MPC-Friendly SS

Next we discuss an SS on which the parties can perform MPC efficiently. Cramer et al. [15] showed that MPC can be conducted on a wide class of SSs called linear SS. However, most practical results of MPC are on the specific SSs, Shamir's SS or replicated SS [14, 25]. They have certain preferred properties, perfect privacy, homomorphicity, and simple arithmetic structure. There have been many practically useful protocols that compute not an arithmetic circuit but a "high-level" function such as bit-decomposition [16, 29], comparison [16, 29], devision [9], shuffling [28], sorting [22], floating point [9] and join [27]. These protocols are based on Shamir's SS, replicated SS, or linear SS including both Shamir's and replicated SSs. Therefore, Shamir's SS and replicated SS are MPC-friendly. In fact, to our knowledge, all implementation results of MPC have been constructed based on either these two SSs [8–10].

3.3 Compatibility

To our knowledge, no efficient MPC based on computationally secure SSs such as Krawczyk's SS has been proposed since most of them have no homomorphism. On the other hands, some MPCs based on the ramp scheme have been proposed so far. Franklin and Yung [19] proposed the protocol that computes parallel multiplications $(a_0b_0, \ldots, a_Lb_L)$, where $(a_0, \ldots, a_L)$ and $(b_0, \ldots, b_L)$ are secretly shared via the ramp scheme. However, the computation is restricted to the pair-wise multiplication, i.e., we cannot compute $a_ib_{i'}$ $(i \neq i')$ with this protocol. Cramer et al. [12] presented the protocol that computes $(\sum_{i+j=0} a_ib_j, \ldots, \sum_{i+j=2L} a_ib_j)$, where $(a_0, \ldots, a_L)$ and $(b_0, \ldots, b_L)$ are secretly shared via the ramp scheme. This protocol can perform wider class of computations compared to [19]. However, their protocol only computes arithmetic circuits. For practical use, protocols that compute high-level functions are essential but have not been proposed on the ramp scheme. Therefore, the compact SSs are not MPC-friendly.

On the other hand, the share-size of Shamir's SS is $\|\mathscr{S}\|$, which is larger than the ones of the compact SSs. The share-size of Replicated SS is much larger than the ones of the compact SSs since it is $\binom{n-1}{t}\|\mathscr{S}\|$. Therefore, the MPC friendly SSs are not compact.

Consequently, there is no SS that is both compact and MPC-friendly.

4 Our Contribution

4.1 Approaches

For our application scenarios, we take an approach that one switches two SSs, a compact SS and an MPC-friendly SS, depending on scenes. We adopt the approach in the secure storage with MPC as follows.

- Each user uses an compact SS to store his data in the system.
- When a user wishes to perform MPC, servers perform a conversion protocol that converts stored shares of the compact SS to those of an MPC-friendly SS, and perform MPC on it.
- After performing MPC, the servers perform another conversion protocol that converts shares of the MPC-friendly SS to those of the compact SS.

We also adopt the conversion protocol in the MPC system with backup as follows.

- Each user shares his data via an MPC-friendly SS and performs MPC on it.
- When the data are renewed, the user shares the renewed data via the MPC-friendly SS.
- The servers perform a conversion protocol that converts the old shares of the MPC-friendly SS to those of an compact SS, and keep them for backup.

As a compact SS, we consider a computationally secure SS and the ramp scheme. As an MPC-friendly SS, we consider homomorphic SS and linear SS, which are classes of SSs and both contain Shamir's SS and replicated SS.

4.2 New Compact SS: Variants of Krawczyk's Scheme

Although Krawczyk's scheme is compact, we propose two variants of Krawczyk's scheme, (t,n)-Comp and (t,n)-$\mathsf{Comp2}$. The reason why we show the variants of Krawczyk's scheme is that Krawczyk's scheme cannot be easily converted. Suppose a is distributed through Krawczyk's scheme with the key key. In this situation, $\mathsf{Enc}_{key}(a)$ is distributed via IDA and key is distributed via an SS, where Enc is the encryption algorithm of a symmetric-key encryption. To convert to an MPC-friendly SS, we have to decrypt $\mathsf{Enc}_{key}(a)$ but the decrypted value should be kept secret. One approach is masking with a randomness: Generate $\mathsf{Enc}_{key}(r)$ and compute $\mathsf{Enc}_{key}(a-r)$ before the decryption. However, this approach cannot be used since a ciphertext is not homomorphic.[2] Another approach is performing the protocol that computes the decryption algorithm. However, it tends to be inefficient since the decryption algorithm should not have a simple arithmetic structure.

Therefore, we propose the variants of Krawczyk's scheme so as to convert their shares to MPC-friendly SSs efficiently. Our approach is making use of multiple secret keys and distribute them so that an adversary cannot obtain all keys.

4.3 Conversion Protocol

We propose several conversion protocols between compact SS and MPC-friendly SS. Before explaining individual protocols, we introduce two evaluation criteria of the conversion protocol.

The first criterion is the computational power of adversaries. There are mainly two types of the adversary's computational power, *computational* and *information-theoretical* security. The former means that the protocol is secure against only polynomially bounded adversaries, and the latter means that the protocol is secure against any (unbounded) adversaries. The latter is stronger security notion so information-theoretical security is preferable from the viewpoint of security. On the other hand, the share-size of a compact SS with computational security tends to be smaller than that of a compact SS with information-theoretic security.

The second criterion is adversary's behavior. There are mainly two types of the adversary's behavior,[3] called *passive* and *active* security. The passive security means that the protocol is secure against only restricted adversaries that follow the protocol. If the protocol is actively secure, it is secure against adversaries whose behavior is not restricted at all. Passively secure conversion protocols are more efficient than actively secure ones so the passively secure ones are preferable if the passive security is enough. For example, if MPC performed after/before conversion are passively secure such as [8–10], passively secure conversion protocols are suitable. On the other hand, if MPC are actively secure such as [5, 20, 24], actively secure conversion protocols are suitable to achieve the active security in the whole system.

From the above criteria, we have four settings: Computational and

[2] Note that even if the symmetric key encryption is a stream cipher, a ciphertext is homomorphic only when the key is the same.

[3] Covert security [1] that is an emerging notion, weaker than active security and stronger than the passive security, have been proposed. However, in the thesis we focus on active and the passive security.

passive, computational and active, information-theoretic and passive, and information-theoretic and active. For each setting, we propose two conversion protocols. One converts a compact SS to MPC-friendly SS and the other is its converse. Therefore, we propose eight conversion protocols in total.

References

[1] Yonatan Aumann and Yehuda Lindell. Security against covert adversaries: Efficient protocols for realistic adversaries. *J. Cryptology*, 23(2):281–343, 2010.

[2] Zuzana Beerliová-Trubíniová and Martin Hirt. Efficient multi-party computation with dispute control. In Halevi and Rabin [21], pages 305–328.

[3] Zuzana Beerliová-Trubíniová and Martin Hirt. Perfectly-secure MPC with linear communication complexity. In Ran Canetti, editor, *TCC*, volume 4948 of *Lecture Notes in Computer Science*, pages 213–230. Springer, 2008.

[4] Michael Ben-Or, Shafi Goldwasser, and Avi Wigderson. Completeness theorems for non-cryptographic fault-tolerant distributed computation (extended abstract). In Simon [33], pages 1–10.

[5] Eli Ben-Sasson, Serge Fehr, and Rafail Ostrovsky. Near-linear unconditionally-secure multiparty computation with a dishonest minority. In Reihaneh Safavi-Naini and Ran Canetti, editors, *Advances in Cryptology - CRYPTO 2012 - 32nd Annual Cryptology Conference, Santa Barbara, CA, USA, August 19-23, 2012. Proceedings*, volume 7417 of *Lecture Notes in Computer Science*, pages 663–680. Springer, 2012.

[6] G. R. Blakley. Safeguarding cryptographic keys. In *Proceedings of the National Computer Conference*, volume 48, pages 313–317, 1979.

[7] G. R. Blakley and Catherine Meadows. Security of ramp schemes. In G. R. Blakley and David Chaum, editors, *CRYPTO*, volume 196 of *Lecture Notes in Computer Science*, pages 242–268. Springer, 1984.

[8] Dan Bogdanov, Sven Laur, and Jan Willemson. Sharemind: A framework for fast privacy-preserving computations. In Sushil Jajodia and Javier López, editors, *ESORICS*, volume 5283 of *Lecture Notes in Computer Science*, pages 192–206. Springer, 2008.

[9] Dan Bogdanov, Margus Niitsoo, Tomas Toft, and Jan Willemson. High-performance secure multi-party computation for data mining applications. *Int. J. Inf. Sec.*, 11(6):403–418, 2012.

[10] Martin Burkhart, Mario Strasser, Dilip Many, and Xenofontas A. Dimitropoulos. SEPIA: Privacy-preserving aggregation of multi-domain network events and statistics. In *USENIX Security Symposium*, pages 223–240. USENIX Association, 2010.

[11] David Chaum, Claude Crépeau, and Ivan Damgård. Multiparty unconditionally secure protocols (extended abstract). In Simon [33], pages 11–19.

[12] Ronald Cramer, Ivan Damgård, and Robbert de Haan. Atomic secure multi-party multiplication with low communication. In Moni Naor, editor, *EUROCRYPT*, volume 4515 of *Lecture Notes in Computer Science*, pages 329–346. Springer, 2007.

[13] Ronald Cramer, Ivan Damgård, Stefan Dziembowski, Martin Hirt, and Tal Rabin. Efficient multiparty computations secure against an adaptive adversary. In Jacques Stern, editor, *Advances in Cryptology - EUROCRYPT '99, International Conference on the Theory and Application of Cryptographic Techniques, Prague, Czech Republic, May 2-6, 1999, Proceeding*, volume 1592 of *Lecture Notes in Computer Science*, pages 311–326. Springer, 1999.

[14] Ronald Cramer, Ivan Damgård, and Yuval Ishai. Share conversion, pseudorandom secret-sharing and applications to secure computation. In Joe Kilian, editor, *TCC*, volume 3378 of *Lecture Notes in Computer Science*, pages 342–362. Springer, 2005.

[15] Ronald Cramer, Ivan Damgård, and Ueli M. Maurer. General secure multi-party computation from any linear secret-sharing scheme. In Bart Preneel, editor, *EUROCRYPT*, volume 1807 of *Lecture Notes in Computer Science*, pages 316–334. Springer, 2000.

[16] Ivan Damgård, Matthias Fitzi, Eike Kiltz, Jesper Buus Nielsen, and Tomas Toft. Unconditionally secure constant-rounds multi-party computation for equality, comparison, bits and exponentiation. In Halevi and Rabin [21], pages 285–304.

[17] Ivan Damgård and Jesper Buus Nielsen. Scalable and unconditionally secure multiparty computation. In Alfred Menezes, editor, *CRYPTO*, volume 4622 of *Lecture Notes in Computer Science*, pages 572–590. Springer, 2007.

[18] Yvo Desmedt and Yair Frankel. Threshold cryptosystems. In Gilles Brassard, editor, *CRYPTO*, volume 435 of *Lecture Notes in Computer Science*, pages 307–315. Springer, 1989.

[19] Matthew K. Franklin and Moti Yung. Communication complexity of secure computation (extended abstract). In S. Rao Kosaraju, Mike Fellows, Avi Wigderson, and John A. Ellis, editors, *STOC*, pages 699–710. ACM, 1992.

[20] Daniel Genkin, Yuval Ishai, Manoj Prabhakaran, Amit Sahai, and Eran Tromer. Circuits resilient to additive attacks with applications to secure computation. In David B. Shmoys, editor, *STOC*, pages 495–504. ACM, 2014.

[21] Shai Halevi and Tal Rabin, editors. *Theory of Cryptography, Third Theory of Cryptography Conference, TCC 2006, New York, NY, USA, March 4-7, 2006, Proceedings*, volume 3876 of *Lecture Notes in Computer Science*. Springer, 2006.

[22] Koki Hamada, Ryo Kikuchi, Dai Ikarashi, Koji Chida, and Katsumi Takahashi. Practically efficient multi-party sorting protocols from comparison sort algorithms. In Taekyoung Kwon, Mun-Kyu Lee, and Daesung Kwon, editors, *ICISC*, volume 7839 of *Lecture Notes in Computer Science*, pages 202–216. Springer, 2012.

[23] Martin Hirt and Ueli M. Maurer. Robustness for free in unconditional multi-party computation. In Joe Kilian, editor, *Advances in Cryptology - CRYPTO 2001, 21st Annual International Cryptology Conference, Santa Barbara, California, USA, August 19-23, 2001, Proceedings*, volume 2139 of *Lecture Notes in Computer Science*, pages 101–118. Springer, 2001.

[24] Dai Ikarashi, Ryo Kikuchi, Koki Hamada, and Koji Chida. Actively private and correct MPC scheme in $t < n/2$ from passively secure schemes with small overhead. *IACR Cryptology ePrint Archive*, 2014:304, 2014.

[25] Mitsuru Ito, Akira Saito, and Takao Nishizeki. Secret sharing schemes realizing general access structure. In *Proc. of the IEEE Global Telecommunication Conf., Globecom 87*, pages 99–102, 1987. Journal version: Multiple assignment scheme for sharing secret. J. of Cryptology, 6(1):15-20, 1993.

[26] Hugo Krawczyk. Secret sharing made short. In Douglas R. Stinson, editor, *CRYPTO*, volume 773 of *Lecture Notes in Computer Science*, pages 136–146. Springer, 1993.

[27] Sven Laur, Riivo Talviste, and Jan Willemson. From oblivious AES to efficient and secure database join in the multiparty setting. In Michael J. Jacobson Jr., Michael E. Locasto, Payman Mohassel, and Reihaneh Safavi-Naini, editors, *ACNS*, volume 7954 of *Lecture Notes in Computer Science*, pages 84–101. Springer, 2013.

[28] Sven Laur, Jan Willemson, and Bingsheng Zhang. Round-efficient oblivious database manipulation. In Xuejia Lai, Jianying Zhou, and Hui Li, editors, *ISC*, volume 7001 of *Lecture Notes in Computer Science*, pages 262–277. Springer, 2011.

[29] Takashi Nishide and Kazuo Ohta. Multiparty computation for interval, equality, and comparison without bit-decomposition protocol. In *Public Key Cryptography*, pages 343–360, 2007.

[30] Michael O. Rabin. Efficient dispersal of information for security, load balancing, and fault tolerance. *Journal of the ACM*, 36(2):335–348, 1989.

[31] Amit Sahai and Brent Waters. Fuzzy identity-based encryption. In Ronald Cramer, editor, *EUROCRYPT*, volume 3494 of *Lecture Notes in Computer Science*, pages 457–473. Springer, 2005.

[32] Adi Shamir. How to share a secret. *Communications of the ACM*, 22(11):612–613, 1979.

[33] Janos Simon, editor. *Proceedings of the 20th Annual ACM Symposium on Theory of Computing, May 2-4, 1988, Chicago, Illinois, USA*. ACM, 1988.

[34] Hirosuke Yamamoto. Secret sharing system using (k,l,n) threshold scheme. *IECE. Trans.*, J68-A(9):945–952, 1985 (in Japanese). English translation: Electronics and Communications in Japan, Part I, vol. 69, no. 9, pp. 46-54, Scripta Technica, Inc., 1986.

Acknowledgment

I would like to thank my supervisor, Wakaha Ogata, for her helpful teaching. I would like to thank my co-authors, Katsumi Takahashi, Koji Chida, Dai Ikarashi, Koki Hamada, Wakaha Ogata, for helpful comments.

Author's Contributions

Journals:

- Ryo Kikuchi, Koji Chida, Dai Ikarashi, Wakaha Ogata, Koki Hamada, and Katsumi Takahashi. *Secret sharing with share-conversion: Achieving small share-size and extendibility to multiparty computation.* IEICE Transactions, 98-A(1):213-222, 2015.

- Ryo Kikuchi, Dai Ikarashi, Koki Hamada, and Koji Chida. *Adaptively and unconditionally secure conversion protocols between ramp and linear secret sharing.* IEICE Transactions, 98-A(1):223-231, 2015.

International conferences (with peer review):

- Ryo Kikuchi, Koji Chida, Dai Ikarashi, Koki Hamada, and Katsumi Takahashi. *Secret sharing schemes with conversion protocol to achieve short share-size and extendibility to multiparty computation.* In Colin Boyd and Leonie Simpson, editors, ACISP, volume 7959 of Lecture Notes in Computer Science, pages 419-434. Springer, 2013.