Poster: Secure Multiparty Computations from XOR-based Secret Sharing Schemes

Yuji Suga suga@iij.ad.jp
Internet Initiative Japan Inc.,
Iidabashi Grand Bloom, 2-10-2 Fujimi, Chiyoda-ku, 102-0071, Japan

***Abstract*—Fast (k, n)-threshold secret sharing schemes with XOR operations have proposed by Kurihara et al. and Fujii et al. 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 the only XOR operations at distribution and reconstruction processes. In these cases for the number of shares n, target data must be equally divided into individual $n_p - 1$ pieces where n_p is a prime more than n. The existing methods described above are configured using the cyclic matrices with prime order. On the other hand, a new method in WAIS2013 have proposed, this leads to general constructions of $(2, p + 1)$-threshold secret sharing schemes. In this paper, we use m-dimensional vector spaces over $\mathbb{Z}_2$ on having bases that meet certain conditions in order to construct proposed methods. This paper defines a new notion "2-propagation bases set" as a bases set to be used in the configuration and specifies the existence of $(2, 2^m)$-threshold secret sharing schemes and also proposes construction of straightforward (but fast) secret multiparty computation from XOR-based SSS.**

I. System requirements

In deployments of secret sharing schemes in cloud storage, we have to consider new proprietary system requirements: **transparency** of data flows and **lightweight process cost**. When cloud suppliers replicate customer's data into different cloud suppliers (in Figure 1), one of suppliers can obtain the qualified sets unintentionally, so we require the data-route transparency.

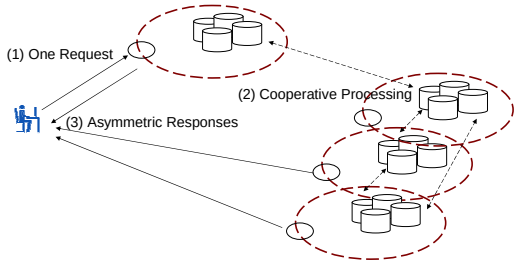

Figure 1. Asymmetric Cloud Services

Secondly, we need to reduce cryptographic process because of comfortable responses/operations in open storage. In this paper, we consider data flow model in Figure 2 that encryption process and secret-sharing process are commutative where M is a plain data, C is an encrypted (using symmetric-key cryptosystem) of M, and $X \to \{x_i\}$ means that $\{x_i\}$ are shares with related to X by applying the secret sharing scheme.

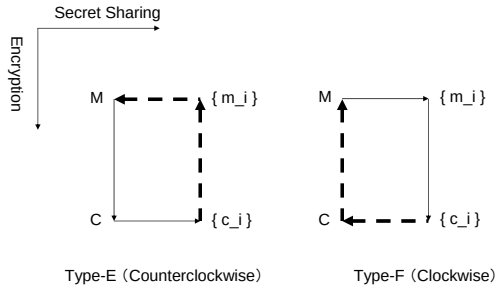

Figure 2. Data Flow Model

II. A construction of XOR-$(2, n_p + 1)$-SSS proposed in WAIS2013

A new method have proposed in WAIS2013 [2], this leads to general constructions of $(2, n_p+1)$-threshold secret sharing schemes using only exclusive-OR operations with the same assumption of previous XOR-(k,n)-SSS. Let $n^{'}$ be the number of pieces of blocks, previous schemes [1] have a restriction about $n^{'}$, that is $n^{'}$ must equal $n_p - 1$ for a certain prime n_p. For example, XOR-(2,4)-SSS with $n^{'} = 3$ must be used part of shares from XOR-(2,5)-SSS with $n^{'}$ = 4.

For given prime n_p, the following is a set of shares $\{W_i\}$ of XOR-(2,n)-SSS with $n^{'} = n_p - 1$ such that $n = n_p + 1$. $W_i (i = 0, \ldots, n-1) = W_{i0} \,||\, \ldots \,||\, W_{ij} \,||\, \ldots \,||\, W_{in''}$ $(j = 0, \ldots, n'' := n^{'} - 1)$. Let a secret be $M = M_1 \,||\, \ldots \,||\, M_{n'}$ where $M_1, \ldots, M_{n'} \in \{0,1\}^d$, $M_0 \in \{0\}^d$ and d-bit binaries $R_0, \ldots, R_{n''}$ be generated randomly.

- $W_{i0} := M_1 \oplus M_{n'+2-i} \oplus R_0 (i = 1, \ldots, n^{'})$
  ($W_{00} = R_0$, $W_{10} = M_1 \oplus R_0$)
- $W_{0j} := M_1 \oplus M_{j+1} \oplus R_j \quad (j = 1, \ldots, n^{'} - 1)$
- $W_{1j} := W_{0,j-1} \oplus R_{j-1} \oplus R_j \quad (j = 1, \ldots, n^{'} - 1)$
- $W_{ij} := W_{i-1,j-1} \oplus R_{j-1} \oplus R_j \quad (i = 1, \ldots, n^{'}, j = 1, \ldots, n^{'} - 1)$
- $W_{n'+1,j} := M_2 \quad \oplus, \ldots, \oplus \quad M_{n'} \oplus R_j \quad (j = 0, \ldots, n^{'} - 1)$

Example 1:

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	$M_2 \oplus R_0$	$M_1 \oplus R_1$

III. New construction of XOR-$(2, 2^m)$-SSS

For a set of basis over $\mathbb{Z}_2^m$, this paper defines a new concept "2-propagation bases set", and proposed new constructions of $(2, 2^m)$-threshold secret sharing schemes using exclusive-OR operations.

Definition 2 (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 $\mathbb{Z}_2^m$.

Lemma 3: The order of 2-propagation bases set $\{b_i\}$ over $\mathbb{Z}_2^m$ is presented as 2^t (optimal case: 2^m). A set $\{b_i\}$ has t generator bases $\{c_i\}(i = 1, \ldots, t)$, for all b_i it satisfies that $b_i = \sum_{j=1}^{t} \alpha_j c_j$.

Theorem 4 (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. ∎

Theorem 4 indicates the existence of 2-propagation bases sets is important. Here are some concrete examples of 2-propagation bases sets for small order. Note that W_0 corresponds to the zero vector bases and $W_1, \ldots, W_m$ are generator bases $c_i$related to Lemma3. All shares are constructed by $\bigoplus_{s=1}^{m} w_l^{*(s)} M_s$ mentioned in Theorem 4.

Example 5 ($m = 4$: 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	$(0,1,0,1)$	$(0,1,1,0)$	$(1,1,0,0)$	$(1,0,0,0)$

IV. A construction of secure multiparty computations from SSS

The following method seems trivial. A requester distributes shares values by using XOR-SSS in advance, such that the inputs A and B should be partitioned to some shares for the distributed entities. In the reconstruction phase, the requester can get $A \oplus B$ from the delegator. The following example is based on XOR-(3,4)-SSS.

Example 6 (XOR-(3,4)-SSS):

W_0	$M_0 \oplus R_0 \oplus R_{01}$	$M_0 \oplus R_1 \oplus R_{11}$
W_1	$M_1 \oplus M_2 \oplus R_0 \oplus R_{11}$	$M_2 \oplus R_1 \oplus R_{21}$
W_2	$M_1 \oplus R_0 \oplus R_{31}$	$M_1 \oplus M_2 \oplus R_1 \oplus R_{31}$
W_3	$M_2 \oplus R_0 \oplus R_{21}$	$M_1 \oplus R_1 \oplus R_{01}$

For all entities U_i, the requester distributes the shares W_i for each A, B, for example U_0 have $A_0 \oplus R_0^A \oplus R_{01}^A$, $A_0 \oplus R_1^A \oplus R_{11}^A$ and $B_0 \oplus R_0^B \oplus R_{01}^B$, $B_0 \oplus R_1^B \oplus R_{11}^B$ in advance.

In the request phase, each entity computes XOR-ed data and send that to U_∞, for example U_0 calculates $(A_0 \oplus R_0^A \oplus R_{01}^A) \oplus (B_0 \oplus R_0^B \oplus R_{01}^B)$, $(A_0 \oplus R_1^A \oplus R_{11}^A) \oplus (B_0 \oplus R_1^B \oplus R_{11}^B)$. These values are interpreted as $(A_0 \oplus B_0) \oplus R_0^{AB} \oplus R_{01}^{AB}$, $(A_0 \oplus B_0) \oplus R_1^{AB} \oplus R_{11}^{AB}$. The last 2 value are share data of $A_i \oplus B_i$ for the entity U_0, the delegator U_∞ can compute $A_i \oplus B_i$ finally by using the ordinary routine program.

Note that this scheme avoids that all distributed entity and also the delegator archive the secrets A, B themselves, however a part of the data about A and B are leaked for only delegator, for example the density of A, B from $A \oplus B$.

A. Applying in $\mathbb{Z}/L\mathbb{Z}$

For input A, B, we consider calculating $A \pm B$ in $\mathbb{Z}/L\mathbb{Z}$ (where $L = 2^l$) by using previous XOR-VSSS method, that is, we consider matrices of numeric-version in $\mathbb{Z}/L\mathbb{Z}$ instead of XOR-ed operations.

- R_j is decided in $\mathbb{Z}/L\mathbb{Z}$ randomly.
- Instead of XOR-SSS (used operation is only XOR),
  - "+" if the sum of indices of share W_i is odd
  - "-" if the sum of indices of share W_i is even

Example 7 (A numeric-version of Example 1):

W_0	$m_0 + r_0$	$m_0 + r_1$
W_1	$m_1 - m_2 - r_0$	$m_2 - r_1$
W_2	$m_1 + r_0$	$m_1 - m_2 + r_1$
W_3	$m_2 - r_0$	$m_1 - r_1$

where m_1, m_2 and r_0, r_1 are elements of $\mathbb{Z}/L\mathbb{Z}$ and m_0:= 0. r_0, r_1 are selected randomly as same as that in XOR-SSS scheme. We don't care about the sign in the previous XOR-based scheme, however in this case, the elements of r_i should be canceled in the reconstruction phase. This construction is derived form the rule: use "+" if the sum of indices is odd, "-" if otherwise.

Example 8 (In-progress-reconstructed data of Example 7):

$F(W_0, W_1)$	$+$	$m_1 - m_2$	m_2
$F(W_0, W_2)$	$-$	$-m_1$	$-m_1 + m_2$
$F(W_0, W_3)$	$+$	m_2	m_1
$F(W_1, W_2)$	$+$	$2m_1 - m_2$	m_1
$F(W_1, W_3)$	$-$	$m_1 - 2m_2$	$-m_1 + m_2$
$F(W_2, W_3)$	$+$	$m_1 + m_2$	$2m1 - m_2$

Note that 2nd column indicates used sign (addition or subtraction) in reconstruction phase. This method seems straightforward, however reconstruction speed is fast because of setting $L = 2^l$, on the other hand, Ben-Or methods with Shamir SSS need operations in Z_p.

References

[1] J. Kurihara, S. Kiyomoto, K. Fukushima, and T. Tanaka, On a fast (k, n)-threshold secret sharing scheme, IEICE Trans. Fundamentals, vol.91-A, no.9, Sep. 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, "A Fast $(2, 2^m)$-Threshold Secret Sharing Scheme Using m Linearly Independent Binary Vectors", The 16th International Conference on Network-Based Information Systems, NBiS 2013, pp.539-544, 2013.