Strongly Multiplicative and 3-Multiplicative Linear Secret Sharing Schemes

Zhifang Zhang[1], Mulan Liu[1], Yeow Meng Chee[2], San Ling[2], and Huaxiong Wang[2,3]

[1] Key Laboratory of Mathematics Mechanization, Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Beijing, China
{zfz, mlliu}@amss.ac.cn
[2] Division of Mathematical Sciences, School of Physical and Mathematical Sciences, Nanyang Technological University, Singapore
{ymchee, lingsan, hxwang}@ntu.edu.sg
[3] Centre for Advanced Computing - Algorithms and Cryptography
Department of Computing
Macquarie University, Australia

Abstract. Strongly multiplicative linear secret sharing schemes (LSSS) have been a powerful tool for constructing secure multi-party computation protocols. However, it remains open *whether or not there exist efficient constructions of strongly multiplicative LSSS from general LSSS*. In this paper, we propose the new concept of 3-*multiplicative LSSS*, and establish its relationship with strongly multiplicative LSSS. More precisely, we show that any 3-multiplicative LSSS is a strongly multiplicative LSSS, but the converse is not true; and that any strongly multiplicative LSSS can be efficiently converted into a 3-multiplicative LSSS. Furthermore, we apply 3-multiplicative LSSS to the computation of unbounded fan-in multiplication, which reduces its round complexity to four (from five of the previous protocol based on multiplicative LSSS). We also give two constructions of 3-multiplicative LSSS from Reed-Muller codes and algebraic geometric codes. We believe that the construction and verification of 3-multiplicative LSSS are easier than those of strongly multiplicative LSSS. This presents a step forward in settling the open problem of efficient constructions of strongly multiplicative LSSS from general LSSS.

Keywords monotone span program, secure multi-party computation, strongly multiplicative linear secret sharing scheme

1 Introduction

Secure multi-party computation (MPC) [16, 9] is a cryptographic primitive that enables n players to jointly compute an agreed function of their private inputs in a secure way, guaranteeing the correctness of the outputs as well as the privacy of the players' inputs, even when some players are malicious. It has become a fundamental tool in cryptography and distributed computation. Linear secret sharing schemes (LSSS) play an important role in building MPC protocols. Cramer *et al.* [6] developed a generic method of constructing MPC protocols from LSSS. Assuming that the function to be computed is represented as an arithmetic circuit over a finite field, their protocol ensures that each player share his private input through an LSSS, and then evaluates the circuit gate by gate. The main idea of their protocol is to keep the intermediate results secretly shared among the players with the underlying LSSS. Due to the nature of linearity, secure additions (and linear operations) can be easily achieved. For instance, if player P_i holds the share x_{1i} for input x_1 and x_{2i} for input x_2, he can locally compute $x_{1i} + x_{2i}$ which is actually P_i's share for $x_1 + x_2$. Unfortunately, the above homomorphic property does not hold for multiplication. In order to securely compute multiplications, Cramer *et al.* [6] introduced the concept of *multiplicative* LSSS, where the product x_1x_2 can be computed as a linear combination of the local products of shares, that is, $x_1x_2 = \sum_{i=1}^{n} a_i x_{1i} x_{2i}$ for some constants $a_i, 1 \leq i \leq n$. Since $x_{1i}x_{2i}$ can be locally computed by P_i, the product can then be securely computed through a linear combination. Furthermore, in order to resist against an active adversary, they defined *strongly* multiplicative LSSS, where x_1x_2 can be computed as a linear combination of the local products of shares by all players excluding any corrupted subset. Therefore, multiplicativity becomes an important property in constructing secure MPC protocols. For example, using strongly multiplicative LSSS, we can construct an error-free MPC protocol secure against an active adversary in the information-theoretic model [6]. Cramer *et al.* [7] also gave an efficient reconstruction algorithm for strongly multiplicative LSSS that recovers the secret even when the shares submitted by the corrupted players contain errors. This implicit “built-in” verifiability makes strongly multiplicative LSSS an attractive building block for MPC protocols.

Due to their important role as the building blocks in MPC protocols, efficient constructions of multiplicative LSSS and strongly multiplicative LSSS have been studied by several authors in recent years. Cramer *et al.* [6] developed a generic method of constructing a multiplicative LSSS from any given LSSS with a double expansion of the shares. Nikov *et al.* [14] studied how to securely compute multiplications in a dual LSSS, without blowing up the shares. For some specific access structures there exist very efficient multiplicative LSSS. Shamir's threshold secret sharing scheme is a well-known example of an ideal (strongly) multiplicative LSSS. Besides, self-dual codes give rise to ideal multiplicative LSSS [7], and Liu *et al.* [12] provided a further class of ideal multiplicative LSSS for some kind of graph access structure. We note that for strongly multiplicative LSSS, the known general construction is of exponential complexity. Käsper *et al.* [11] gave some efficient constructions for specific access structures (hierarchical threshold structures). It remains open whether there exists an efficient transformation from a general LSSS to a strongly multiplicative one.

On the other hand, although in a multiplicative LSSS, multiplication can be converted into a linear combination of inputs from the players, each player has to *reshare* the product of his shares, that is, for $1 \leq i \leq n$, P_i needs to reshare the product $x_{1i}x_{2i}$ to securely compute the linear combination $\sum_{i=1}^{n} a_i x_{1i} x_{2i}$. This resharing process involves costly interactions among the players. For example, if the players are to securely compute multiple multiplications, $\prod_{i=1}^{l} x_i$, the simple sequential multiplication requires interaction of round complexity proportional to l. Using the technique developed by Bar-Ilan and Beaver [1], Cramer *et al.* [4] recently showed that the round complexity can be significantly reduced to a constant of five for unbounded fan-in multiplications. However, the method does not seem efficient when l is small. For example, considering x_1x_2 and $x_1x_2x_3$, extra rounds of interactions seem unavoidable for computing $x_1x_2x_3$ even though we apply the method of Cramer *et al.* [4].

1.1 Our Contribution

In this paper, we propose the concept of 3-multiplicative LSSS. Roughly speaking, a 3-multiplicative LSSS is a generalization of multiplicative LSSS, where the product $x_1x_2x_3$ is a linear combination of the local products of shares. As one would expect, a 3-multiplicative LSSS achieves better round complexity for the computation of $\prod_{i=1}^{l} x_i$ compared to a multiplicative LSSS, if $l \geq 3$. Indeed, it is easy to see that computing the product $\prod_{i=1}^{9} x_i$ requires two rounds of interaction for a 3-multiplicative LSSS but four rounds for a multiplicative LSSS. We also extend the concept of a 3-multiplicative LSSS to the more general λ-multiplicative LSSS, for all integers $\lambda \geq 3$, and show that λ-multiplicative LSSS reduce the round complexity by a factor of $\frac{1}{\log \lambda}$ from multiplicative LSSS. In particular, 3-multiplicative LSSS reduce the constant round complexity of computing the unbounded fan-in multiplication from five to four, thus improving a result of Cramer *et al.* [4].

More importantly, we show that 3-multiplicative LSSS are closely related to strongly multiplicative LSSS. The latter is known to be a powerful tool for constructing secure MPC protocols against active adversaries. More precisely, we show the following:

(i) 3-multiplicative LSSS are also strongly multiplicative;
(ii) there exists an efficient algorithm that transforms a strongly multiplicative LSSS into a 3-multiplicative LSSS;
(iii) an example of a strongly multiplicative LSSS that is not 3-multiplicative.

Our results contribute to the study of MPC in the following three aspects:

- The 3-multiplicative LSSS outperform strongly multiplicative LSSS with respect to round complexity in the construction of secure MPC protocols.
- The 3-multiplicative LSSS are easier to construct than strongly multiplicative LSSS. First, the existence of an efficient transformation from a strongly multiplicative LSSS to a 3-multiplicative LSSS implies that efficiently constructing 3-multiplicative LSSS is not a harder problem. Second, verification of a strongly multiplicative LSSS requires checking the linear combinations for all possibilities of adversary sets, while the verification of a 3-multiplicative LSSS requires only one checking. We give two constructions of LSSS based on Reed-Muller codes and algebraic geometric codes that can be

easily verified for 3-multiplicativity, but it does not seem easy to give direct proofs of their strong multiplicativity.

- This work provides two possible directions toward solving the open problem of determining the existence of efficient constructions for strongly multiplicative LSSS. On the negative side, if we can prove that in the information-theoretic model and with polynomial size message exchanged, computing $x_1x_2x_3$ inevitably needs more rounds of interactions than computing x_1x_2, then we can give a negative answer to this open problem. On the positive side, if we can find an efficient construction for 3-multiplicative LSSS, which also results in strongly multiplicative LSSS, then we will have an affirmative answer to this open problem.

1.2 Organization

Section 2 gives notations, definition of multiplicative LSSS, and general constructions for strongly multiplicative LSSS. Section 3 defines 3-multiplicative LSSS. Section 4 shows the relationship between 3-multiplicative LSSS and strongly multiplicative LSSS. Section 5 gives two constructions of 3-multiplicative LSSS from error-correcting codes, and Section 6 discusses the implications of 3-multiplicative LSSS in MPC. Section 7 concludes the paper.

2 Preliminaries

Throughout this paper, let $P = \{P_1, \ldots, P_n\}$ denote the set of n players and let $\mathcal{K}$ be a finite field. In a secret sharing scheme, the collection of all subsets of players that are authorized to recover the secret is called its *access structure*, and is denoted by AS. An access structure possesses the monotone ascending property: if $A' \in AS$, then for all $A \subseteq P$ with $A \supseteq A'$, we also have $A \in AS$. Similarly, the collection of subsets of players that are possibly corrupted is called the *adversary structure*, and is denoted as $\mathcal{A}$. An adversary structure possesses the monotone descending property: if $A' \in \mathcal{A}$, then for all $A \subseteq P$ with $A \subseteq A'$, we also have $A \in \mathcal{A}$. Owing to these monotone properties, it is often sufficient to consider the *minimum access structure* AS_{min} and the *maximum adversary structure* $\mathcal{A}_{max}$ defined as follows:

$$AS_{min} = \{A \in AS \mid \forall B \subseteq P, \text{ we have } B \subsetneq A \Rightarrow B \notin AS\},$$
$$\mathcal{A}_{max} = \{A \in \mathcal{A} \mid \forall B \subseteq P, \text{ we have } B \supsetneq A \Rightarrow B \notin \mathcal{A}\}.$$

In this paper, we consider the *complete* situation, that is, $\mathcal{A} = 2^P - AS$. Moreover, an adversary structure $\mathcal{A}$ is called Q^2 (respectively, Q^3) if any two (respectively, three) sets in $\mathcal{A}$ cannot cover the entire player set P. For simplicity, when an adversary structure $\mathcal{A}$ is Q^2 (respectively, Q^3) we also say the corresponding access structure $AS = 2^P - \mathcal{A}$ is Q^2 (respectively, Q^3).

2.1 Linear Secret Sharing Schemes and Monotone Span Programs

Suppose S is the secret-domain, R is the set of random inputs, and S_i is the share-domain of P_i, where $1 \leq i \leq n$. Let S and R denote random variables taking values in S and R, respectively. Then $\Pi : S \times R \rightarrow S_1 \times \cdots \times S_n$ is called a *secret sharing scheme* (SSS) with respect to the access structure AS, if the following two conditions are satisfied:

1. for all $A \in AS$, $H(\mathsf{S} \mid \Pi(\mathsf{S}, \mathsf{R})|_A) = 0$;
2. for all $B \notin AS$, $H(\mathsf{S} \mid \Pi(\mathsf{S}, \mathsf{R})|_B) = H(\mathsf{S})$,

where $H(\cdot)$ is the entropy function. Furthermore, the secret sharing scheme Π is called *linear* if we have $S = \mathcal{K}$, $R = \mathcal{K}^{l-1}$, and $S_i = \mathcal{K}^{d_i}$ for some positive integers l and d_i, $1 \leq i \leq n$, and the reconstruction of the secret can be performed by taking a linear combination of shares from the authorized players. The quantity $d = \sum_{i=1}^{n} d_i$ is called the *size* of the LSSS.

Karchmer and Wigderson [10] introduced monotone span programs (MSP) as a linear model for computing monotone Boolean functions. We denote an MSP by $\mathcal{M}(\mathcal{K}, M, \psi, \boldsymbol{v})$, where M is a $d \times l$ matrix over $\mathcal{K}$, $\psi : \{1, \ldots, d\} \rightarrow \{P_1, \ldots, P_n\}$ is a surjective labeling map, and $\boldsymbol{v} \in \mathcal{K}^l$ is a nonzero vector. We call d the *size* of the MSP and $\boldsymbol{v}$ the *target vector*. A monotone Boolean function $f : \{0,1\}^n \rightarrow \{0,1\}$ satisfies $f(\boldsymbol{\delta'}) \geq f(\boldsymbol{\delta})$ for any $\boldsymbol{\delta'} \geq \boldsymbol{\delta}$, where $\boldsymbol{\delta} = (\delta_1, \ldots, \delta_n)$, $\boldsymbol{\delta'} = (\delta'_1, \ldots, \delta'_n) \in \{0,1\}^n$, and $\boldsymbol{\delta'} \geq \boldsymbol{\delta}$ means $\delta'_i \geq \delta_i$ for $1 \leq i \leq n$. We say that an MSP $\mathcal{M}(\mathcal{K}, M, \psi, \boldsymbol{v})$ *computes the monotone Boolean function* f if $\boldsymbol{v} \in span\{M_A\}$ if and only if $f(\boldsymbol{\delta_A}) = 1$, where A is a set of players, M_A denotes the matrix constricted to the rows labeled by players in A, $span\{M_A\}$ denotes the linear space spanned by the row vectors of M_A, and $\boldsymbol{\delta_A}$ is the characteristic vector of A.

Theorem 1 (Beimel [2]). *Suppose AS is an access structure over P and f_{AS} is the characteristic function of AS, that is, $f_{AS}(\boldsymbol{\delta}) = 1$ if and only if $\boldsymbol{\delta} = \boldsymbol{\delta}_A$ for some $A \in AS$. Then there exists an LSSS of size d that realizes AS if and only if there exists an MSP of size d that computes f_{AS}.*

Since an MSP computes the same Boolean function under linear transformations, we can always assume that the target vector is $\boldsymbol{e}_1 = (1, 0, \ldots, 0)$. From an MSP $\mathcal{M}(\mathcal{K}, M, \psi, \boldsymbol{e}_1)$ that computes f_{AS}, we can derive an LSSS realizing AS as follows: to share a secret $s \in \mathcal{K}$, the dealer randomly selects $\boldsymbol{\rho} \in \mathcal{K}^{l-1}$, computes $M(s, \boldsymbol{\rho})^\tau$ and sends $M_{P_i}(s, \boldsymbol{\rho})^\tau$ to P_i as his share, where $1 \leq i \leq n$ and τ denotes the transpose. The following property of MSP is useful in the proofs of our results.

Proposition 1 (Karchmer and Wigderson [10]). *Let $\mathcal{M}(\mathcal{K}, M, \psi, \boldsymbol{e_1})$ be an MSP that computes a monotone Boolean function f. Then for all $A \subseteq P$, $\boldsymbol{e}_1 \notin span\{M_A\}$ if and only if there exists $\boldsymbol{\rho} \in \mathcal{K}^{l-1}$ such that $M_A(1, \boldsymbol{\rho})^\tau = \boldsymbol{0}^\tau$.*

2.2 Multiplicative Linear Secret Sharing Schemes

From Theorem 1, an LSSS can be identified with its corresponding MSP in the following way. Let $\mathcal{M}(\mathcal{K}, M, \psi, \boldsymbol{e}_1)$ be an LSSS realizing the access structure AS. Given two vectors $\boldsymbol{x} = (x_1, \ldots, x_d)$, $\boldsymbol{y} = (y_1, \ldots, y_d) \in \mathcal{K}^d$, we define $\boldsymbol{x} \diamond \boldsymbol{y}$ to be the vector containing all entries of the form $x_i \cdot y_j$ with $\psi(i) = \psi(j)$. More precisely, let

$$\begin{aligned}\boldsymbol{x} &= (x_{11}, \ldots, x_{1d_1}, \ldots, x_{n1}, \ldots, x_{nd_n}),\\ \boldsymbol{y} &= (y_{11}, \ldots, y_{1d_1}, \ldots, y_{n1}, \ldots, y_{nd_n}),\end{aligned}$$

where $\sum_{i=1}^{n} d_i = d$, and $(x_{i1}, \ldots, x_{id_i})$, $(y_{i1}, \ldots, y_{id_i})$ are the entries distributed to P_i according to ψ. Then $\boldsymbol{x} \diamond \boldsymbol{y}$ is the vector composed of the $\sum_{i=1}^{n} d_i^2$ entries $x_{ij}y_{ik}$, where $1 \le j, k \le d_i, 1 \le i \le n$. For consistency, we write the entries of $\boldsymbol{x} \diamond \boldsymbol{y}$ in some fixed order. We also define $(\boldsymbol{x} \diamond \boldsymbol{y})^\tau = \boldsymbol{x}^\tau \diamond \boldsymbol{y}^\tau$.

Definition 1 (Multiplicativity). *Let $\mathcal{M}(\mathcal{K}, M, \psi, \boldsymbol{e_1})$ be an LSSS realizing the access structure AS over P. Then $\mathcal{M}$ is called* multiplicative *if there exists a recombination vector $\boldsymbol{z} \in \mathcal{K}^{\sum_{i=1}^{n} d_i^2}$, such that for all $s, s' \in \mathcal{K}$ and $\boldsymbol{\rho}, \boldsymbol{\rho}' \in \mathcal{K}^{l-1}$, we have*

$$ss' = \boldsymbol{z}(M(s, \boldsymbol{\rho})^\tau \diamond M(s', \boldsymbol{\rho}')^\tau).$$

Moreover, $\mathcal{M}$ is strongly multiplicative *if for all $A \in \mathcal{A} = 2^P - AS$, $\mathcal{M}_{\overline{A}}$ is multiplicative, where $\mathcal{M}_{\overline{A}}$ denotes the MSP $\mathcal{M}$ constricted to the subset $\overline{A} = P - A$.*

Proposition 2 (Cramer *et al.* [6]). *Let AS be an access structure over P. Then there exists a multiplicative (respectively, strongly multiplicative) LSSS realizing AS if and only if AS is Q^2 (respectively, Q^3).*

2.3 General Constructions of Strongly Multiplicative LSSS

For all Q^2 access structure AS, Cramer *et al.* [6] gave an efficient construction to build a multiplicative LSSS from a general LSSS realizing the same AS. It remains *open* if we can *efficiently* construct a strongly multiplicative LSSS from an LSSS. However, there are general constructions with exponential complexity, as described below.

Since Shamir's threshold secret sharing scheme is strongly multiplicative for all Q^3 threshold access structure, a proper composition of Shamir's threshold secret sharing schemes results in a general construction for strongly multiplicative LSSS [6]. Here, we give another general construction based on multiplicative LSSS.

Let AS be any Q^3 access structure and $\mathcal{M}(\mathcal{K}, M, \psi, \boldsymbol{e}_1)$ be an LSSS realizing AS. For all $A \in \mathcal{A} = 2^P - AS$, it is easy to see that $\mathcal{M}_{\overline{A}}$ realizes the restricted access structure $AS_{\overline{A}} = \{B \subseteq \overline{A} \mid B \in AS\}$. The access structure $AS_{\overline{A}}$ is Q^2 over $\overline{A}$ because AS is Q^3 over $\overline{A} \cup A$. Thus, we can transform $\mathcal{M}_{\overline{A}}$ into a multiplicative LSSS following the general construction of Cramer *et al.* [6] to obtain a strongly multiplicative LSSS realizing AS. The example in Section 4.3 gives an illustration of this method.

We note that both constructions above give LSSS of exponential sizes, and hence are not *efficient* in general.

3 3-Multiplicative and λ-Multiplicative LSSS

In this section, we give an equivalent definition for (strongly) multiplicative LSSS. We then define 3-multiplicative LSSS and give a necessary and sufficient condition for its existence. The notion of 3-multiplicativity is also extended to λ-multiplicativity for all integer $\lambda > 1$. Finally, we present a generic (but inefficient) construction of λ-multiplicative LSSS.

Under the same notations used in Section 2.2, it is straightforward to see that we have an induced labeling map $\psi' : \{1, \ldots, \sum_{i=1}^{n} d_i^2\} \to \{P_1, \ldots, P_n\}$ on the entries of $\boldsymbol{x} \diamond \boldsymbol{y}$, distributing the entry $x_{ij}y_{ik}$ to P_i, since both x_{ij} and y_{ik} are labeled by P_i under ψ. For an MSP $\mathcal{M}(\mathcal{K}, M, \psi, \boldsymbol{e}_1)$, denote $M = (M_1, \ldots, M_l)$, where $M_i \in \mathcal{K}^d$ is the i-th column vector of M, $1 \le i \le l$. We construct a new matrix $M_\diamond$ as follows:

$$M_\diamond = (M_1 \diamond M_1, \ldots, M_1 \diamond M_l, M_2 \diamond M_1, \ldots, M_2 \diamond M_l, \ldots, M_l \diamond M_1, \ldots, M_l \diamond M_l).$$

For consistency, we also denote $M_\diamond$ as $M \diamond M$. Obviously, $M_\diamond$ is a matrix over $\mathcal{K}$ with $\sum_{i=1}^{n} d_i^2$ rows and l^2 columns. For any two vectors $\boldsymbol{u}, \boldsymbol{v} \in \mathcal{K}^l$, it is easy to verify that

$$(M\boldsymbol{u}^\tau) \diamond (M\boldsymbol{v}^\tau) = M_\diamond(\boldsymbol{u} \otimes \boldsymbol{v})^\tau,$$

where $\boldsymbol{u} \otimes \boldsymbol{v}$ denotes the tensor product with its entries written in a proper order. Define the induced labeling map ψ' on the rows of $M_\diamond$. We have the following proposition.

Proposition 3. *Let $\mathcal{M}(\mathcal{K}, M, \psi, \boldsymbol{e}_1)$ be an LSSS realizing the access structure AS, and let $M_\diamond$ be with the labeling map ψ'. Then $\mathcal{M}$ is multiplicative if and only if $\boldsymbol{e}_1 \in span\{M_\diamond\}$, where $\boldsymbol{e}_1 = (1, 0, \ldots, 0)$. Moreover, $\mathcal{M}$ is strongly multiplicative if and only if $\boldsymbol{e}_1 \in span\{(M_\diamond)_{\overline{A}}\}$ for all $A \in \mathcal{A} = 2^P - AS$.*

Proof. By Definition 1, $\mathcal{M}$ is multiplicative if and only if $ss' = \boldsymbol{z}(M(s, \boldsymbol{\rho})^\tau \diamond M(s', \boldsymbol{\rho}')^\tau)$ for all $s, s' \in \mathcal{K}$ and $\boldsymbol{\rho}, \boldsymbol{\rho}' \in \mathcal{K}^{l-1}$. Obviously,

$$M(s, \boldsymbol{\rho})^\tau \diamond M(s', \boldsymbol{\rho}')^\tau = M_\diamond((s, \boldsymbol{\rho}) \otimes (s', \boldsymbol{\rho}'))^\tau = M_\diamond(ss', \boldsymbol{\rho}'')^\tau, \tag{1}$$

where $(ss', \boldsymbol{\rho}'') = (s, \boldsymbol{\rho}) \otimes (s', \boldsymbol{\rho}')$. On the other hand, $ss' = \boldsymbol{e}_1(ss', \boldsymbol{\rho}'')^\tau$. Thus $\mathcal{M}$ is multiplicative if and only if

$$(\boldsymbol{e}_1 - \boldsymbol{z}M_\diamond)(ss', \boldsymbol{\rho}'')^\tau = 0. \tag{2}$$

Because of the arbitrariness of $s, s', \boldsymbol{\rho}$ and $\boldsymbol{\rho}'$, equality (2) holds if and only if $\boldsymbol{e}_1 - \boldsymbol{z}M_\diamond = \boldsymbol{0}$. Thus $\boldsymbol{e}_1 \in span\{M_\diamond\}$. The latter part of the proposition can be proved similarly. □

Now we are ready to give the definition of 3-multiplicative LSSS. We extend the diamond product "$\diamond$" and define $\boldsymbol{x} \diamond \boldsymbol{y} \diamond \boldsymbol{z}$ to be the vector containing all entries of the form $x_i y_j z_k$ with $\psi(i) = \psi(j) = \psi(k)$, where the entries of $\boldsymbol{x} \diamond \boldsymbol{y} \diamond \boldsymbol{z}$ are written in some fixed order.

Definition 2 (3-Multiplicativity). *Let $\mathcal{M}(\mathcal{K}, M, \psi, \boldsymbol{e}_1)$ be an LSSS realizing the access structure AS. Then $\mathcal{M}$ is called* 3-multiplicative *if there exists a recombination vector $\boldsymbol{z} \in \mathcal{K}^{\sum_{i=1}^{n} d_i^3}$ such that for all $s_1, s_2, s_3 \in \mathcal{K}$ and $\boldsymbol{\rho}_1, \boldsymbol{\rho}_2, \boldsymbol{\rho}_3 \in \mathcal{K}^{l-1}$, we have*

$$s_1 s_2 s_3 = \boldsymbol{z}(M(s_1, \boldsymbol{\rho}_1)^\tau \diamond M(s_2, \boldsymbol{\rho}_2)^\tau \diamond M(s_3, \boldsymbol{\rho}_3)^\tau).$$

We can derive an equivalent definition for 3-multiplicative LSSS, similar to Proposition 3: $\mathcal{M}$ is 3-multiplicative if and only if $\boldsymbol{e}_1 \in span\{(M \diamond M \diamond M)\}$. The following proposition gives a necessary and sufficient condition for the existence of 3-multiplicative LSSS.

Proposition 4. *For all access structures AS, there exists a 3-multiplicative LSSS realizing AS if and only if AS is Q^3.*

Proof. Suppose $\mathcal{M}(\mathcal{K}, M, \psi, \boldsymbol{e}_1)$ is a 3-multiplicative LSSS realizing AS, and suppose to the contrary, that AS is not Q^3, so there exist $A_1, A_2, A_3 \in \mathcal{A} = 2^P - AS$ such that $A_1 \cup A_2 \cup A_3 = P$. By Proposition 1, there exists $\boldsymbol{\rho}_i \in \mathcal{K}^{l-1}$ such that $M_{A_i}(1, \boldsymbol{\rho}_i)^\tau = \boldsymbol{0}^\tau$ for $1 \leq i \leq 3$. Since $A_1 \cup A_2 \cup A_3 = P$, we have $M(1, \boldsymbol{\rho}_1)^\tau \diamond M(1, \boldsymbol{\rho}_2)^\tau \diamond M(1, \boldsymbol{\rho}_3)^\tau = \boldsymbol{0}^\tau$, which contradicts Definition 2.

On the other hand, a general construction for building a 3-multiplicative LSSS from a strongly multiplicative LSSS is given in the next section, thus sufficiency is guaranteed by Proposition 2. □

A trivial example of 3-multiplicative LSSS is Shamir's threshold secret sharing scheme that realizes any Q^3 threshold access structure. Using an identical argument for the case of strongly multiplicative LSSS, we have a general construction for 3-multiplicative LSSS based on Shamir's threshold secret sharing schemes, with exponential complexity.

For any λ vectors $\boldsymbol{x}_i = (x_{i1}, \ldots, x_{id}) \in \mathcal{K}^d, 1 \leq i \leq \lambda$, we define $\diamond_{i=1}^{\lambda} \boldsymbol{x}_i$ to be the $\sum_{i=1}^{n} d_i^{\lambda}$-dimensional vector which contains entries of the form $\prod_{i=1}^{\lambda} x_{ij_i}$ with $\psi(j_1) = \cdots = \psi(j_\lambda)$.

Definition 3 ($\boldsymbol{\lambda}$-Multiplicativity). *Let $\mathcal{M}(\mathcal{K}, M, \psi, \boldsymbol{e}_1)$ be an LSSS realizing the access structure AS, and let $\lambda > 1$ be an integer. Then $\mathcal{M}$ is* λ-multiplicative *if there exists a recombination vector $\boldsymbol{z}$ such that for all $s_1, \ldots, s_\lambda \in \mathcal{K}$ and $\boldsymbol{\rho}_1, \ldots, \boldsymbol{\rho}_\lambda \in \mathcal{K}^{l-1}$, we have*

$$\prod_{i=1}^{\lambda} s_i = \boldsymbol{z}(\diamond_{i=1}^{\lambda} M(s_i, \boldsymbol{\rho}_i)^\tau).$$

Moreover, $\mathcal{M}$ is strongly λ-multiplicative *if for all $A \notin AS$, the constricted LSSS $\mathcal{M}_{\overline{A}}$ is λ-multiplicative.*

Again, we can define a new matrix by taking the diamond product of λ copies of M. This gives an equivalence to (strongly) λ-multiplicative LSSS. Also, since Shamir's threshold secret sharing scheme is trivially λ-multiplicative and strongly λ-multiplicative, a proper composition of Shamir's threshold secret sharing schemes results in a general construction for both λ-multiplicative LSSS and strongly λ-multiplicative LSSS. Let Q^λ be a straightforward extension of Q^2 and Q^3, that is, an access structure AS is Q^λ if the player set P cannot be covered by λ sets in $\mathcal{A} = 2^P - AS$. The following corollary is easy to prove.

Corollary 1. *Let AS be an access structure over P. Then there exists a λ-multiplicative (respectively, strongly λ-multiplicative) LSSS realizing AS if and only if AS is Q^λ (respectively, $Q^{\lambda+1}$).*

Since a λ-multiplicative LSSS transforms the products of λ entries into a linear combination of the local products of shares, it can be used to simplify the secure computation of sequential multiplications. In particular, when compared to using only the multiplicative property (which corresponds to the case when $\lambda = 2$), a λ-multiplicative LSSS can lead to reduced round complexity by a factor of $\frac{1}{\log \lambda}$ in certain cases.

We also point out that Q^λ is not a necessary condition for secure computation. Instead, the necessary condition is Q^2 for the passive adversary model, or Q^3 for the active adversary model [6]. The condition Q^λ is just a necessary condition for the existence of λ-multiplicative LSSS which can be used to simplify computation. In practice, many threshold adversary structures satisfy the Q^λ condition for some appropriate integer λ, and the widely used Shamir's threshold secret sharing scheme is already λ-multiplicative. By using this λ-multiplicativity, we can get more efficient MPC protocols. However, since the special case $\lambda = 3$ shows a close relationship with strongly multiplicative LSSS, a fundamental tool in MPC, this paper focuses on 3-multiplicative LSSS.

4 Strong Multiplicativity and 3-Multiplicativity

In this section, we show that strong multiplicativity and 3-multiplicativity are closely related. On the one hand, given a strongly multiplicative LSSS, there is an *efficient* transformation that converts it to a 3-multiplicative LSSS. On the other hand, we show that any 3-multiplicative LSSS is a strongly multiplicative LSSS, but the converse is not true. It should be noted that strong multiplicativity, as defined, has a combinatorial nature. The definition of 3-multiplicativity is essentially algebraic, which is typically easier to verify.

4.1 From Strong Multiplicativity to 3-Multiplicativity

We show a general method to efficiently build a 3-multiplicative LSSS from a strongly multiplicative LSSS, for all Q^3 access structures. As an extension, the proposed method can also be used to efficiently build a $(\lambda + 1)$-multiplicative LSSS from a strongly λ-multiplicative LSSS.

Theorem 2. *Let AS be a Q^3 access structure and $\mathcal{M}(\mathcal{K}, M, \psi, \boldsymbol{e}_1)$ be a strongly multiplicative LSSS realizing AS. Suppose that $\mathcal{M}$ has size d and $|\psi^{-1}(P_i)| = d_i$, for $1 \leq i \leq n$. Then there exists a 3-multiplicative LSSS for AS of size $O(d^2)$.*

Proof. We give a constructive proof. Let $M_\diamond$ be the matrix defined in Section 3, and ψ' be the induced labeling map on the rows of $M_\diamond$. Then we have an LSSS $\mathcal{M}_\diamond(\mathcal{K}, M_\diamond, \psi', \boldsymbol{e}_1)$ that realizes an access structure $AS_\diamond$. Because $\mathcal{M}$ is strongly multiplicative, by Proposition 3 we have $\boldsymbol{e}_1 \in span\{(M_\diamond)_{\overline{A}}\}$ for all $A \notin AS$. Therefore $\overline{A} \in AS_\diamond$ and it follows that $AS^* \subseteq AS_\diamond$, where AS^* denotes the dual access structure of AS, defined by $AS^* = \{A \subseteq P \mid P - A \notin AS\}$.

The equality (1) in the proof of Proposition 3 shows that the diamond product of two share vectors equals sharing the product of the two secrets by the MSP $\mathcal{M}_\diamond(\mathcal{K}, M_\diamond, \psi', \boldsymbol{e}_1)$, that is,

$$(M(s_1, \boldsymbol{\rho}_1')^\tau) \diamond (M(s_2, \boldsymbol{\rho}_2')^\tau) = M_\diamond(s_1 s_2, \boldsymbol{\rho})^\tau, \quad \text{for some } \boldsymbol{\rho}_1', \boldsymbol{\rho}_2', \boldsymbol{\rho} \in \mathcal{K}^{l-1}.$$

Thus, using a method similar to Nikov *et al.* [14], we can get the product $(s_1 s_2) \cdot s_3$ by sharing s_3 through the dual MSP of $\mathcal{M}_\diamond$, denoted by $(\mathcal{M}_\diamond)^*$. Furthermore, since $(\mathcal{M}_\diamond)^*$ realizes the dual access structure $(AS_\diamond)^*$ and $(AS_\diamond)^* \subseteq (AS^*)^* = AS$, we can build a 3-multiplicative LSSS by the union of $\mathcal{M}$ and $(\mathcal{M}_\diamond)^*$, which realizes the access structure $AS \cup (AS_\diamond)^* = AS$. Now following the same method of Cramer *et al.* and Fehr [6, 8], we prove the required result via the construction below.

Compute the column vector $\boldsymbol{v}_0$ as a solution to the equation $(M_\diamond)^\tau \boldsymbol{v} = \boldsymbol{e_1}^\tau$ for $\boldsymbol{v}$, and compute $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_k$ as a basis of the solution space to $(M_\diamond)^\tau \boldsymbol{v} = \boldsymbol{0}^\tau$. Note that $(M_\diamond)^\tau \boldsymbol{v} = \boldsymbol{e_1}^\tau$ is solvable because $\boldsymbol{e}_1 \in span\{(M_\diamond)_{\overline{A}}\}$ for all $A \notin AS$, while $(M_\diamond)^\tau \boldsymbol{v} = \boldsymbol{0}^\tau$ may only have the trivial solution $\boldsymbol{v} = \boldsymbol{0}$ and $k = 0$. Let

$$M' = \begin{pmatrix} m_{11} & \cdots & m_{1l} & & & \\ \vdots & \ddots & \vdots & & & \\ m_{d1} & \cdots & m_{dl} & & & \\ \boldsymbol{v_0} & & & \boldsymbol{v_1} & \cdots & \boldsymbol{v_k} \end{pmatrix},$$

where $\begin{pmatrix} m_{11} & \cdots & m_{1l} \\ \vdots & \ddots & \vdots \\ m_{d1} & \cdots & m_{dl} \end{pmatrix} = M$ and the blanks in M' denote zeros. Define a labeling map ψ'' on the rows of M' which labels the first d rows of M' according to ψ and the other $\sum_{i=1}^n d_i^2$ rows according to ψ'.

As mentioned above, $\mathcal{M}'(\mathcal{K}, M', \psi'', \boldsymbol{e}_1)$ obviously realizes the access structure AS. We now verify its 3-multiplicativity.

Let $N = (\boldsymbol{v}_0, \boldsymbol{v}_1, \ldots, \boldsymbol{v}_k)$, a matrix over $\mathcal{K}$ with $\sum_{i=1}^n d_i^2$ rows and $k+1$ columns. For $s_i \in \mathcal{K}$ and $\boldsymbol{\rho}_i = (\boldsymbol{\rho}_i', \boldsymbol{\rho}_i'') \in \mathcal{K}^{l-1} \times \mathcal{K}^k$, $1 \leq i \leq 3$, denote $M'(s_i, \boldsymbol{\rho}_i)^\tau = (\boldsymbol{u}_i, \boldsymbol{w}_i)^\tau$, where $\boldsymbol{u}_i^\tau = M(s_i, \boldsymbol{\rho}_i')^\tau$ and $\boldsymbol{w}_i^\tau = N(s_i, \boldsymbol{\rho}_i'')^\tau$. We have

$$\boldsymbol{u}_1^\tau \diamond \boldsymbol{u}_2^\tau = (M(s_1, \boldsymbol{\rho}_1')^\tau) \diamond (M(s_2, \boldsymbol{\rho}_2')^\tau) = M_\diamond(s_1 s_2, \boldsymbol{\rho})^\tau,$$

where $(s_1s_2, \boldsymbol{\rho}) = (s_1, \boldsymbol{\rho}'_1) \otimes (s_2, \boldsymbol{\rho}'_2)$. Then,

$$\begin{aligned}
(\boldsymbol{u}_1 \diamond \boldsymbol{u}_2) \cdot \boldsymbol{w}_3^{\tau} &= (s_1s_2, \boldsymbol{\rho})(M_\diamond)^{\tau} \cdot N \begin{pmatrix} s_3 \\ \\ \boldsymbol{\rho}_3''^{\,\tau} \end{pmatrix} \\
&= (s_1s_2, \boldsymbol{\rho}) \begin{pmatrix} 1 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 \end{pmatrix} \begin{pmatrix} s_3 \\ \\ \boldsymbol{\rho}_3''^{\,\tau} \end{pmatrix} \\
&= s_1s_2s_3.
\end{aligned}$$

It is easy to see that $(\boldsymbol{u}_1 \diamond \boldsymbol{u}_2) \cdot \boldsymbol{w}_3^{\tau}$ is a linear combination of the entries from $(\boldsymbol{u}_1 \diamond \boldsymbol{u}_2) \diamond \boldsymbol{w}_3$, and so is a linear combination of the entries from $M'(s_1, \boldsymbol{\rho}_1)^{\tau} \diamond M'(s_2, \boldsymbol{\rho}_2)^{\tau} \diamond M'(s_3, \boldsymbol{\rho}_3)^{\tau}$.

Hence $\mathcal{M}'$ is a 3-multiplicative LSSS for AS. Obviously, the size of $\mathcal{M}'$ is $O(d^2)$, since $d + \sum_{i=1}^{n} d_i^2 < d^2 + d$. □

If we replace the matrix $M_\diamond$ above by the diamond product of λ copies of M, using an identical argument, the construction from Theorem 2 gives rise to a $(\lambda+1)$-multiplicative LSSS from a strongly λ-multiplicative LSSS.

Corollary 2. *Let AS be a $Q^{\lambda+1}$ access structure and $\mathcal{M}(\mathcal{K}, M, \psi, \boldsymbol{e}_1)$ be a strongly λ-multiplicative LSSS realizing AS. Suppose the size of $\mathcal{M}$ is d and $|\psi^{-1}(P_i)| = d_i$, for $1 \le i \le n$. Then there exists a $(\lambda+1)$-multiplicative LSSS for AS of size $O(d^{\lambda})$.*

4.2 From 3-Multiplicativity to Strong Multiplicativity

Theorem 3. *Any 3-multiplicative LSSS is strongly multiplicative.*

Proof. Let $\mathcal{M}(\mathcal{K}, M, \psi, \boldsymbol{e}_1)$ be a 3-multiplicative LSSS realizing the access structure AS over P. For all $A \in \mathcal{A} = 2^P - AS$, by Proposition 1, we can choose a fixed vector $\boldsymbol{\rho}'' \in \mathcal{K}^{l-1}$ such that $M_A(1, \boldsymbol{\rho}'')^{\tau} = \boldsymbol{0}^{\tau}$. There exists a recombination vector $\boldsymbol{z} \in \mathcal{K}^{\sum_{i=1}^{n} d_i^3}$ such that for all $s, s' \in \mathcal{K}$ and $\boldsymbol{\rho}, \boldsymbol{\rho}' \in \mathcal{K}^{l-1}$, we have

$$ss' = \boldsymbol{z}(M(s, \boldsymbol{\rho})^{\tau} \diamond M(s', \boldsymbol{\rho}')^{\tau} \diamond M(1, \boldsymbol{\rho}'')^{\tau}).$$

Since $M_A(1, \boldsymbol{\rho}'')^{\tau} = \boldsymbol{0}^{\tau}$, and $M_{\overline{A}}(1, \boldsymbol{\rho}'')^{\tau}$ is a constant vector for fixed $\boldsymbol{\rho}''$, the vector $\boldsymbol{z}' \in \mathcal{K}^{\sum_{P_i \notin A} d_i^2}$ that satisfies

$$\boldsymbol{z}(M(s, \boldsymbol{\rho})^{\tau} \diamond M(s', \boldsymbol{\rho}')^{\tau} \diamond M(1, \boldsymbol{\rho}'')^{\tau}) = \boldsymbol{z}'(M_{\overline{A}}(s, \boldsymbol{\rho})^{\tau} \diamond M_{\overline{A}}(s', \boldsymbol{\rho}')^{\tau})$$

can be easily determined. Thus $ss' = \boldsymbol{z}'(M_{\overline{A}}(s, \boldsymbol{\rho})^{\tau} \diamond M_{\overline{A}}(s', \boldsymbol{\rho}')^{\tau})$. Hence, $\mathcal{M}$ is strongly multiplicative. □

Although 3-multiplicative LSSS is a subclass of strongly multiplicative LSSS, one of the advantages of 3-multiplicativity is that its verification admits a simpler process. For 3-multiplicativity, we need only to check that $\boldsymbol{e}_1 \in \mathit{span}\{(M \diamond M \diamond M)\}$, while strong multiplicativity requires the verification of $\boldsymbol{e}_1 \in \mathit{span}\{(M \diamond M)_{\overline{A}}\}$ for *all* $A \notin AS$.

Using a similar argument, the following results for $(\lambda + 1)$-multiplicativity can be proved:

(i) A $(\lambda + 1)$-multiplicative LSSS is a strongly λ-multiplicative LSSS.
(ii) A λ-multiplicative LSSS is a λ'-multiplicative LSSS, where $1 < \lambda' < \lambda$.

4.3 An Example of a Strongly Multiplicative LSSS that is Not 3-Multiplicative

We give an example of a strongly multiplicative LSSS that is not 3-multiplicative. It follows that 3-multiplicative LSSS are strictly contained in the class of strongly multiplicative LSSS. The construction process is as follows. Start with an LSSS that realizes a Q^3 access structure but is not strongly multiplicative. We then apply the general construction given in Section 2.3 to convert it into a strongly multiplicative LSSS. The resulting LSSS is however not 3-multiplicative.

Let $P = \{P_1, P_2, P_3, P_4, P_5, P_6\}$ be the set of players. Consider the access structure AS over P defined by

$$AS_{min} = \{(1,2),(3,4),(5,6),(1,5),(1,6),(2,6),(2,5),(3,6),(4,5)\},$$

where we use subscript to denote the corresponding player. For example, $(1,2)$ denotes the subset $\{P_1, P_2\}$. It is easy to verify that the corresponding adversary structure is

$$\mathcal{A}_{max} = \{(1,3),(1,4),(2,3),(2,4),(3,5),(4,6)\},$$

and that AS is a Q^3 access structure.

Let $\mathcal{K} = \mathbb{F}_2$. Define the matrix M over $\mathbb{F}_2$ with the labeling map ψ such that

$$M_{P_1} = \begin{pmatrix} 1\,0\,1\,0\,0 \\ 0\,0\,0\,1\,0 \\ 0\,0\,0\,0\,1 \end{pmatrix},\ M_{P_2} = \begin{pmatrix} 0\,0\,1\,0\,0 \\ 0\,0\,0\,1\,0 \\ 0\,0\,0\,0\,1 \end{pmatrix},\ M_{P_3} = \begin{pmatrix} 1\,1\,0\,0\,0 \\ 0\,0\,0\,0\,1 \end{pmatrix},$$
$$M_{P_4} = \begin{pmatrix} 0\,1\,0\,0\,0 \\ 0\,0\,0\,1\,0 \end{pmatrix},\ M_{P_5} = \begin{pmatrix} 1\,1\,1\,0\,0 \\ 1\,0\,0\,1\,0 \end{pmatrix},\ M_{P_6} = \begin{pmatrix} 0\,1\,1\,0\,0 \\ 1\,0\,0\,0\,1 \end{pmatrix}.$$

It can be verified that the LSSS $\mathcal{M}(\mathbb{F}_2, M, \psi, \boldsymbol{e}_1)$ realizes the access structure AS. Moreover, for all $A \in \mathcal{A} - \{(1,3),(1,4)\}$, the constricted LSSS $\mathcal{M}_{\overline{A}}$ is multiplicative. Thus in order to get a strongly multiplicative LSSS, we just need to expand $\mathcal{M}$ with multiplicativity when constricted to both $\{P_2, P_4, P_5, P_6\}$ and $\{P_2, P_3, P_5, P_6\}$.

Firstly, consider the LSSS $\mathcal{M}$ constricted to $P' = \{P_2, P_4, P_5, P_6\}$. Obviously, $\mathcal{M}_{P'}$ realizes the access structure $AS'_{min} = \{(5,6),(2,6),(2,5),(4,5)\}$, which is

Q^2 over P'. By the method of Cramer *et al.* [6], we can transform $\mathcal{M}_{P'}$ into the multiplicative LSSS $\mathcal{M}'_{P'}(\mathbb{F}_2, M', \psi', \boldsymbol{e}_1)$ defined as follows:

$$M'_{P_2} = \begin{pmatrix} 0&0&1&0&0&&&& \\ 0&0&0&1&0&&&& \\ 0&0&0&0&1&&&& \\ &&&&&0&1&1&1 \\ &&&&&1&1&0&0 \\ &&&&&0&0&0&1 \end{pmatrix}, \quad M'_{P_4} = \begin{pmatrix} 0&1&0&0&0&&&& \\ 0&0&0&1&0&&&& \\ &&&&&0&1&1&1 \\ &&&&&1&0&0&0 \end{pmatrix},$$

$$M'_{P_5} = \begin{pmatrix} 1&1&1&0&0&&&& \\ 1&0&0&1&0&&&& \\ 1&&&&&0&1&0&1 \\ 0&&&&&0&1&0&0 \end{pmatrix}, \quad M'_{P_6} = \begin{pmatrix} 0&1&1&0&0&&&& \\ 1&0&0&0&1&&&& \\ 1&&&&&0&0&1&0 \\ 0&&&&&0&0&0&1 \end{pmatrix},$$

where the blanks in the matrices denote zeros.

For consistency, we define

$$M'_{P_1} = (M_{P_1}\ O_{3\times 4}),$$
$$M'_{P_3} = (M_{P_3}\ O_{2\times 4}),$$

where $O_{m\times n}$ denotes the $m \times n$ matrix of all zeros. It can be verified that for the subset $P'' = \{P_2, P_3, P_5, P_6\}$, the constricted LSSS $\mathcal{M}'_{P''}$ is indeed multiplicative. Therefore, $\mathcal{M}'(\mathbb{F}_2, M', \psi', \boldsymbol{e}_1)$ is a strongly multiplicative LSSS realizing the access structure AS. Furthermore, it can be verified that $\mathcal{M}'$ is not 3-multiplicative (the verification involves checking a 443×729 matrix using Matlab).

The scheme $\mathcal{M}(\mathbb{F}_2, M, \psi, \boldsymbol{v}_1)$ given above is the first example of an LSSS which realizes a Q^3 access structure but is not strongly multiplicative.

5 Constructions for 3-multiplicative LSSS

It is tempting to find efficient constructions for 3-multiplicative LSSS. In general, it is a hard problem to construct LSSS with polynomial size for any specified access structure, and it seems to be an even harder problem to construct polynomial size 3-multiplicative LSSS with general Q^3 access structures. We mention two constructions for 3-multiplicative LSSS. These constructions are generally inefficient, which can result in schemes with exponential sizes. The two constructions are:

1. The Cramer-Damgård-Maurer construction based on Shamir's threshold secret sharing scheme [6].
2. The construction given in Subsection 4.1 based on strongly multiplicative LSSS.

There exist, however, some efficient LSSS with specific access structures that are multiplicative or 3-multiplicative. For instance, Shamir's t out of n threshold secret sharing schemes are multiplicative if $n \geq 2t+1$, and 3-multiplicative if $n \geq 3t+1$.

On the other hand, secret sharing schemes from error-correcting codes give good multiplicative properties. It is well known that a secret sharing scheme from a linear error-correcting code is an LSSS. We know that such an LSSS is multiplicative provided the underlying code is a self dual code [7]. The LSSS from a Reed-Solomon code is λ-multiplicative if the corresponding access structure is Q^{λ}. In this section, we show the multiplicativity of two other classes of secret sharing schemes from error-correcting codes:

(i) schemes from Reed-Muller codes are λ-multiplicative LSSS; and
(ii) schemes from algebraic geometric codes are λ-multiplicative ramp LSSS.

5.1 A Construction from Reed-Muller Codes

Let $\boldsymbol{v}_0, \boldsymbol{v}_1, \ldots, \boldsymbol{v}_{2^m-1}$ be all the points in the space $\mathbb{F}_2^m$. The binary Reed-Muller code $\mathcal{R}(r,m)$ is defined as follows:

$$\mathcal{R}(r,m) = \{(f(\boldsymbol{v}_0), f(\boldsymbol{v}_1), \ldots, f(\boldsymbol{v}_{2^m-1})) \mid f \in \mathbb{F}_2[x_1, \ldots, x_m],\ \deg f \leq r\}.$$

Take $f(\boldsymbol{v}_0)$ as the secret, and $f(\boldsymbol{v}_i)$ as the share distributed to player P_i, $1 \leq i \leq 2^m - 1$. Then $\mathcal{R}(r,m)$ gives rise to an LSSS for the set of players $\{P_1, \ldots, P_n\}$, with the secret-domain being $\mathbb{F}_2$, where $n = 2^m - 1$. For any three codewords

$$\boldsymbol{c}_i = (s_i, s_{i1}, \ldots, s_{in}) = (f_i(\boldsymbol{v}_0), f_i(\boldsymbol{v}_1), \ldots, f_i(\boldsymbol{v}_n)) \in \mathcal{R}(r,m), \quad 1 \leq i \leq 3,$$

it is easy to see that

$$\begin{aligned}\boldsymbol{c}_1 \diamond \boldsymbol{c}_2 \diamond \boldsymbol{c}_3 &= (s_1 s_2 s_3, s_{11}s_{21}s_{31}, \ldots, s_{1n}s_{2n}s_{3n}) \\ &= (g(\boldsymbol{v}_0), g(\boldsymbol{v}_1), \ldots, g(\boldsymbol{v}_n)) \in \mathcal{R}(3r,m),\end{aligned}$$

where $g = f_1 f_2 f_3 \in \mathbb{F}_2[x_1, \ldots, x_m]$ and $\deg g \leq 3r$. From basic results on Reed-Muller codes [15], we know that $\mathcal{R}(3r,m)$ has dual code $\mathcal{R}(m-3r-1,m)$ when $m > 3r$, and the dual code $\mathcal{R}(m-3r-1,m)$ trivially contains the codeword $(1,1,\ldots,1)$. It follows that $s_1 s_2 s_3 = \sum_{j=1}^{n} s_{1j}s_{2j}s_{3j}$, which shows that the LSSS from $\mathcal{R}(r,m)$ is 3-multiplicative when $m > 3r$. Certainly, this LSSS is strongly multiplicative. In general, we have the following result:

Theorem 4. *The LSSS constructed above from $\mathcal{R}(r,m)$ is λ-multiplicative, provided $m > \lambda r$.*

5.2 A Construction from Algebraic Geometric Codes

Chen and Cramer [3] constructed secret sharing schemes from algebraic geometric (AG) codes. These schemes are *quasi-threshold* (or *ramp*) schemes, which means that any t out of n players can recover the secret, and any fewer than

t' players have no information about the secret, where $t' \leq t \leq n$. In this section, we show that ramp schemes from some algebraic geometric codes [3] are λ-multiplicative.

Let χ be an absolutely irreducible, projective, and nonsingular curve defined over $\mathbb{F}_q$ with genus g, and let $D = \{v_0, v_1, \ldots, v_n\}$ be the set of $\mathbb{F}_q$-rational points on χ. Let G be an $\mathbb{F}_q$-rational divisor with degree m satisfying $supp(G) \cap D = \emptyset$ and $2g - 2 < m < n + 1$. Let $\overline{\mathbb{F}}_q$ denote the algebraic closure of $\mathbb{F}_q$, let $\overline{\mathbb{F}}_q(\chi)$ denote the function field of the curve χ, and let $\Omega(\chi)$ denote all the differentials on χ. Define the linear spaces:

$$\begin{aligned}&\mathcal{L}(G) = \{f \in \overline{\mathbb{F}}_q(\chi) \mid (f) + G \geq 0\},\\ &\Omega(G) = \{\omega \in \Omega(\chi) \mid (\omega) \geq G\}.\end{aligned}$$

Then the functional AG code $C_{\mathcal{L}}(D, G)$ and residual AG code $C_{\Omega}(D, G)$ are respectively defined as follows:

$$\begin{aligned}&C_{\mathcal{L}}(D, G) = \{(f(v_0), f(v_1), \ldots, f(v_n)) \mid f \in \mathcal{L}(G)\} \subseteq \mathbb{F}_q^{n+1},\\ &C_{\Omega}(D, G) = \{(Res_{v_0}(\eta), Res_{v_1}(\eta), \ldots, Res_{v_n}(\eta)) \mid \eta \in \Omega(G - D)\} \subseteq \mathbb{F}_q^{n+1},\end{aligned}$$

where $Res_{v_i}(\eta)$ denotes the residue of η at v_i.

As above, $C_{\Omega}(D, G)$ induces an LSSS for the set of players $\{P_1, \ldots, P_n\}$, where for every codeword $(f(v_0), f(v_1), \ldots, f(v_n)) \in C_{\Omega}(D, G) = C_{\mathcal{L}}(D, D - G + (\eta))$, $f(v_0)$ is the secret and $f(v_i)$ is P_i's share, $1 \leq i \leq n$. For any λ codewords

$$\begin{aligned}\boldsymbol{c}_i &= (s_i, s_{i1}, \ldots, s_{in})\\ &= (f_i(v_0), f_i(v_1), \ldots, f_i(v_n)) \in C_{\mathcal{L}}(D, D - G + (\eta)), \quad 1 \leq i \leq \lambda,\end{aligned}$$

it is easy to see that

$$\diamond_{i=1}^{\lambda} \boldsymbol{c}_i = \left(\prod_{i=1}^{\lambda} s_i, \prod_{i=1}^{\lambda} s_{i1}, \ldots, \prod_{i=1}^{\lambda} s_{in}\right) \in C_{\mathcal{L}}(D, \lambda(D - G + (\eta))).$$

If $2g - 2 < \deg(\lambda(D - G + (\eta))) < n$, then $C_{\mathcal{L}}(D, \lambda(D - G + (\eta)))$ has the dual code $C_{\Omega}(D, \lambda(D - G + (\eta))) = C_{\mathcal{L}}(D, \lambda G - (\lambda - 1)(D + (\eta)))$. When $\deg(\lambda G - (\lambda - 1)(D + (\eta))) \geq 2g$, $C_{\Omega}(D, \lambda(D - G + (\eta)))$ has a codeword with a nonzero first coordinate, implying $\prod_{i=1}^{\lambda} s_i = \sum_{j=1}^{n} a_j \prod_{i=1}^{\lambda} s_{ij}$ for some constants $a_j \in \mathbb{F}_q$. Thus, the LSSS induced by the AG code $C_{\Omega}(D, G)$ is λ-multiplicative. It is easy to see that if $\deg G = m \geq \frac{(\lambda-1)(n-1)}{\lambda} + 2g$ then we have $2g - 2 < \deg(\lambda(D - G + (\eta))) < n$ and $\deg(\lambda G - (\lambda - 1)(D + (\eta))) \geq 2g$. Therefore, we have the following theorem.

Theorem 5. *Let χ be an absolutely irreducible, projective, and nonsingular curve defined over $\mathbb{F}_q$ with genus g, let $D = \{v_0, v_1, \ldots, v_n\}$ be the set of $\mathbb{F}_q$-rational points on χ. Let G be an $\mathbb{F}_q$-rational divisor with degree m satisfying $supp(G) \cap D = \emptyset$ and $2g - 2 < m < n + 1$. Then the LSSS induced by the AG code $C_{\Omega}(D, G)$ is λ-multiplicative, provided $m \geq \frac{(\lambda-1)(n-1)}{\lambda} + 2g$.*

6 Implications of the Multiplicativity of LSSS

The property of 3-multiplicativity implies strong multiplicativity, and so is sufficient for building MPC protocols against active adversaries. The conditions for 3-multiplicativity are easy to verify, while verification for strong multiplicativity involves checking an exponential number of equations (each subset in the adversary structure corresponds to an equation).

With 3-multiplicative LSSS, or more generally λ-multiplicative LSSS, we can simplify local computation for each player and reduce the round complexity in MPC protocols. For example, using the technique of Bar-Ilan and Beaver [1], we can compute $\prod_{i=1}^{l} x_i,\ x_i \in \mathbb{F}_q$, in a constant number of rounds, independent of l. For simplicity, we consider passive adversaries in the information-theoretic model. Suppose for $1 \leq i \leq l$, the shares of x_i, denoted by $[x_i]$, have already been distributed among the players. To compute $\prod_{i=1}^{l} x_i,\ x_i \in \mathbb{F}_q$, we follow the process of Cramer *et al.* [4]:

(1) Generate $[b_0 \in_R \mathbb{F}_q^*], [b_1 \in_R \mathbb{F}_q^*], \ldots, [b_l \in_R \mathbb{F}_q^*]$ and $[b_0^{-1}], [b_1^{-1}], \ldots, [b_l^{-1}]$, where $b_i \in_R \mathbb{F}_q^*$ means that b_i is a random element in $\mathbb{F}_q^*$.

(2) For $1 \leq i \leq l$, each player computes $[b_{i-1}x_i b_i^{-1}]$ from $[b_{i-1}], [b_i^{-1}]$ and $[x_i]$.

(3) Recover $d_i = b_{i-1}x_i b_i^{-1}$ from $[b_{i-1}x_i b_i^{-1}]$ for $1 \leq i \leq l$, and compute $d = \prod_{i=1}^{l} d_i$.

(4) Compute $[db_0^{-1}b_l]$ from $[b_0^{-1}], [b_l]$ and d.

It is easy to see that $db_0^{-1}b_l = \prod_{i=1}^{l} x_i$. Using a multiplicative LSSS, the above process takes five rounds of interactions as two rounds are required in Step (2). However, if we use a 3-multiplicative LSSS instead, then only one round is needed for Step (2). Thus, 3-multiplicative LSSS reduce the round complexity of computing unbounded fan-in multiplication from five to four. This in turn simplifies the computation of many problems, such as polynomial evaluation and solving linear systems of equations.

In general, the relationship between λ-multiplicative LSSS and strongly λ-multiplicative LSSS can be described as follows:

$$\cdots \subseteq SMLSSS_{\lambda+1} \subsetneq MLSSS_{\lambda+1} \subseteq SMLSSS_{\lambda} \subsetneq MLSSS_{\lambda} \subseteq \cdots,$$

where $MLSSS_\lambda$ (respectively, $SMLSSS_\lambda$) denotes the class of λ-multiplicative (respectively, strongly λ-multiplicative) LSSS. It is easy to see that $SMLSSS_\lambda \subsetneq MLSSS_\lambda$ because they exist under the conditions $Q^{\lambda+1}$ and Q^λ, respectively. Since $SMLSSS_\lambda$ and $MLSSS_{\lambda+1}$ both exist under the same necessary and sufficient condition of $Q^{\lambda+1}$, it is not straightforward to see whether $MLSSS_{\lambda+1}$ is strictly contained in $SMLSSS_\lambda$. For $\lambda = 2$, we already know that $MLSSS_3 \subsetneq SMLSSS_2$ (Section 4.3). It would be interesting to find out if this is also true for $\lambda > 2$. We have also given an efficient transformation from $SMLSSS_\lambda$ to $MLSSS_{\lambda+1}$. It remains open whether an efficient transformation from $MLSSS_\lambda$ to $SMLSSS_\lambda$ exists when the access structure is $Q^{\lambda+1}$. When $\lambda = 2$, this is a well-known open problem [6].

7 Conclusions

In this paper, we propose the new concept of 3-multiplicative LSSS, which form a subclass of strongly multiplicative LSSS. The 3-multiplicative LSSS are easier to construct compared to strongly multiplicative LSSS. They can also simplify the computation and reduce the round complexity in secure multiparty computation protocols. We believe that 3-multiplicative LSSS are a more appropriate primitive as building blocks for secure multiparty computations, and deserve further investigation. We stress that finding efficient constructions of 3-multiplicative LSSS for general access structures remains an important open problem.

Acknowledgement

The work of M. Liu and Z. Zhang is supported in part by the open project of the State Key Laboratory of Information Security and the 973 project of China (No. 2004CB318000). Part of the work was done while Z. Zhang was visiting Nanyang Technological University supported by the Singapore Ministry of Education under Research Grant T206B2204.

The work of Y. M. Chee, S. Ling, and H. Wang is supported in part by the Singapore National Research Foundation under Research Grant NRF-CRP2-2007-03.

In addition, the work of Y. M. Chee is also supported in part by the Nanyang Technological University under Research Grant M58110040, and the work of H. Wang is also supported in part by the Australian Research Council under ARC Discovery Project DP0665035.

References

1. J. Bar-Ilan , D. Beaver, Non-cryptographic fault-tolerant computing in constant number of rounds of interaction. PODC'89, pp. 201-209, 1989.
2. A. Beimel, Secure schemes for secret sharing and key distribution. PhD thesis, Technion - Israel Institute of Technology, 1996.
3. H. Chen, R. Cramer, Algebraic geometric secret sharing schemes and secure multi-party computations over small fields. CRYPTO'06, LNCS, vol. 4117, pp. 521-536, 2006.
4. R. Cramer, E. Kiltz, C. Padró, A note on secure computation of the Moore-Penrose pseudoinverse and its spplication to secure linear algebra. CRYPTO'07, LNCS, vol. 4622, pp. 613-630, 2007.
5. H. Chen, R. Cramer, R. de Haan and I. Cascudo Pueyo. Strongly multiplicative ramp schemes from high degree rational points on curves. EUROCRYPT'08, LNCS, vol. 4965, pp. 451-470, 2008.
6. R. Cramer, I. Damgård, U. Maurer, General secure multi-party computation from any linear secret-sharing scheme. EUROCRYPT'00, LNCS, vol. 1807, pp. 316-334, 2000.
7. R. Cramer, V. Daza, I. Gracia, J. Urroz, G. Leander, J. Martí-Farré, C. Padró, On codes, matroids and secure multi-party computation from linear secret sharing schemes. CRYPTO'05, LNCS, vol. 3621, pp. 327-343, 2005.

8. S. Fehr, Efficient construction of the dual span program. Master Thesis, the Swiss Federal Institute of Technology (ETH) Zürich, 1999 http://homepages.cwi.nl/~fehr/publications.html
9. O. Goldreich, S. Micali, A. Wigderson, How to play ANY mental game. STOC'87, pp. 218-219, 1987.
10. M. Karchmer, A. Wigderson, On span programs. Proc. 8th Ann. Symp. Structure in Complexity Theory, pp. 102-111, 1993.
11. E. Käsper, V. Nikov, S. Nikova, Strongly multiplicative hierarchical threshold secret sharing. In 2nd International Conference on Information Theoretic Security - ICITS 2007, LNCS, to appear.
12. M. Liu, L, Xiao, Z. Zhang, Multiplicative linear secret sharing schemes based on connectivity of graphs. IEEE Transactions on Information Theory 53(11), pp. 3973-3978, 2007.
13. J.L. Massey, Minimal codewords and secret sharing. Proc. 6th Joint Swedish-Russian Workshop on Information Theory, pp. 276-279, 1993.
14. V. Nikov, S. Nikova, B. Preneel, On multiplicative linear secret sharing schemes. Indocrypt'03, LNCS, vol. 2904, pp. 135-147, 2003.
15. J.H. van Lint, Introduction to coding theory. 3rd edition, Graduate Texts in Mathematics 86, Springer, 1999.
16. A. Yao, Protocols for secure computation. FOCS '82, pp. 160-164, 1982.