

On Multiplicative Linear Secret Sharing Schemes

Ventzislav Nikov¹ *, Svetla Nikova² **, and Bart Preneel²

¹ Department of Mathematics and Computing Science,
Eindhoven University of Technology
P.O. Box 513, 5600 MB, Eindhoven, the Netherlands
`v.nikov@tue.nl`

² Department Electrical Engineering, ESAT/COSIC,
Katholieke Universiteit Leuven, Kasteelpark Arenberg 10,
B-3001 Heverlee-Leuven, Belgium
`svetla.nikova,bart.preneel@esat.kuleuven.ac.be`

Abstract. We consider both information-theoretic and cryptographic settings for Multi-Party Computation (MPC), based on the underlying linear secret sharing scheme. Our goal is to study the Monotone Span Program (MSP), that is the result of local multiplication of shares distributed by two given MSPs as well as the access structure that this *resulting* MSP computes. First, we expand the construction proposed by Cramer *et al.* for multiplying two different general access structures and we prove some properties of the resulting MSP. We prove that using two (different) MSPs to compute their resulting MSP is more efficient than building a multiplicative MSP. Next we define a (strongly) multiplicative resulting MSP and we prove that when one uses dual MSPs only all players together can compute the product. An analog of the algebraic simplification protocol of Gennaro *et al.* is presented. We show which conditions the resulting access structure should fulfill in order to achieve MPC secure against an adaptive, active adversary in the zero-error case in both the computational and the information-theoretic model.

1 Introduction

Background. The concept of *secret sharing* was introduced by Shamir [14] as a tool to protect a secret simultaneously from exposure and from being lost. It allows a so called *dealer* to share the secret among a set of entities, usually called *players*, in such a way that only certain specified subsets of the players are able to reconstruct the secret while smaller

* The research has been supported by a Marie Curie Fellowship of the European Community Programme under contract number HPMT-CT-2000-00093.

** The author was partially supported by the IWT STWW project on Anonymity and Privacy in Electronic Services and the Concerted Research Action GOA-MEFISTO-666 of the Flemish Government; part of the work was done during the author's visit at the Ruhr University, Bochum.

subsets have no information about it. Denote by P the set of participants in the scheme. The groups who are allowed to reconstruct the secret are called *qualified* (denoted by Γ), and the groups who should not be able to obtain any information about the secret are called *forbidden* (or *curious*) (denoted by Δ). Γ is *monotone increasing* and can be described by the set Γ^- consisting of its *minimal elements* (sets). Δ is *monotone decreasing* and similarly, the set Δ^+ consists of the *maximal elements* (sets) in Δ . The tuple (Γ, Δ) is called an *access structure* if $\Gamma \cap \Delta = \emptyset$. If $\Gamma = \Delta^c$ is the complement of Δ , then we say that (Γ, Δ) is *complete* and we denote it only by Γ . The *dual* Γ^\perp of a monotone access structure Γ , defined on P , is the collection of sets $A \subseteq P$ such that $A^c \notin \Gamma$. An access structure Γ is *connected* if each player belongs to at least one minimal set.

It is common to model cheating by considering an *adversary* who may corrupt some subset of the players. One can distinguish between *passive* and *active* corruption, see Fehr *et al.* [6] for recent results. Passive corruption means that the adversary obtains the complete information held by the corrupt players, but the players execute the protocol correctly. Active corruption means that the adversary takes full control of the corrupt players. Thus in a so called *mixed adversary model* an adversary is characterized by a *privacy structure* Δ (the curious players) and an *adversary structure* $\Delta_A \subseteq \Delta$ (the corrupt players). Denote the complement $\Gamma_A = \Delta_A^c$ and call its dual access structure Γ_A^\perp the *honest* (or *good*) players structure. Both passive and active adversaries may be *static*, meaning that the set of corrupt players is chosen once and for all before the protocol starts, or *adaptive* meaning that the adversary can at any time during the protocol choose to corrupt a new player based on all the information he has at the time, as long as the total set is in Δ_A .

A wide range of general approaches for designing Secret Sharing Schemes (SSS) is known, but most of these techniques result in *linear* SSS (LSSS). Since late 80's many efforts has been put into finding better presentations (algebraic, geometric, combinatorial) which allow to compute any monotone access structure. In this paper we will use an algebraic computational device introduced by Karchmer and Wigderson [10] called *Monotone Span Program*. It is well known that there is one-to-one correspondence between LSSS and MSPs and that MSPs can compute any complete monotone access structure.

Since an LSSS neither guarantees reconstructability when some shares are incorrect, nor verifiability of a shared value a stronger primitive called *verifiable secret sharing* (VSS) has been introduced in [1, 5]. In a VSS a dealer distributes a secret value among the players, where the dealer

and/or some of the players may be cheating. It is guaranteed that if the dealer is honest, then the cheaters obtain no information about the secret, and all honest players will later be able to reconstruct it, without the help of the dealer. Even if the dealer cheats, a unique value will be determined and is reconstructible without the help of the cheaters.

Secure *multi-party computation* (MPC) can be defined as follows: n players compute an agreed function of their inputs in a “secure” way, where “secure” means guaranteeing the correctness of the output as well as the privacy of the players’ inputs, even when some players cheat. VSS is a key tool for secure MPC.

The Model. We will consider the standard *secure-channels model*, where the players are connected by bilateral, synchronous, reliable secure channels. We assume also the availability of a broadcast channel. By default, we consider *unconditional* security against an adaptive, active adversary (mixed adversary model) and error-free protocols.

Organization. In the first part of the next section we give some notations and linear algebra techniques, then we describe our results. In Section 3 we propose the main construction diamond \diamond and investigate its properties. Then in Section 4 conditions for the existence of MPC based on LSSS, which are secure against adaptive, active adversaries are considered.

2 Preliminaries

Related Works. We briefly recall some definitions and observations. The following operation (called element-wise union) for monotone decreasing sets was introduced in [6, 12].

Definition 1. [6, 12] We define the operation \uplus for any monotone **decreasing** sets Δ_1, Δ_2 as follows: $\Delta_1 \uplus \Delta_2 = \{A = A_1 \cup A_2; A_1 \in \Delta_1, A_2 \in \Delta_2\}$ and the operation \uplus for any monotone **increasing** sets Γ_1, Γ_2 as follows: $\Gamma_1 \uplus \Gamma_2 = \{A = A_1 \cup A_2; A_1 \notin \Gamma_1, A_2 \notin \Gamma_2\}^c$.

For an arbitrary matrix M over a finite field \mathbb{F} , with m rows labelled by $1, \dots, m$ let M_A denote the matrix obtained by keeping only those rows i with $i \in A$. Let M_A^T denote the transpose of M_A , and let $Im(M_A^T)$ denote the \mathbb{F} -linear span of the rows of M_A . We use $Ker(M_A)$ to denote the kernel of M_A , i.e., all linear combinations of the columns of M_A , leading to 0. Let $\mathbf{v} = (v_1, \dots, v_{t_1}) \in \mathbb{F}^{t_1}$ and $\mathbf{w} = (w_1, \dots, w_{t_2}) \in \mathbb{F}^{t_2}$ be two vectors. By $\langle \mathbf{v}, \mathbf{w} \rangle$ we denote the standard inner product. The tensor vector product $\mathbf{v} \otimes \mathbf{w}$ is defined as a vector in $\mathbb{F}^{t_1 t_2}$ such that the j -coordinate in \mathbf{v}

(denoted by v_j) is replaced by $v_j \mathbf{w}$, i.e., $\mathbf{v} \otimes \mathbf{w} = (v_1 \mathbf{w}, \dots, v_{t_1} \mathbf{w}) \in \mathbb{F}^{t_1 t_2}$. The tensor matrix product $\mathbf{v} \bar{\otimes} \mathbf{w}$ is defined as a $t_1 \times t_2$ matrix such that the j -column is equal to $v_j \mathbf{w}$.

Definition 2. [3, 10] *A Monotone Span Program (MSP) \mathcal{M} is a quadruple $(\mathbb{F}, M, \varepsilon, \psi)$, where \mathbb{F} is a finite field, M is a matrix (with m rows and $d \leq m$ columns) over \mathbb{F} , $\psi : \{1, \dots, m\} \rightarrow \{1, \dots, n\}$ is a surjective function and ε is a fixed vector, called **target vector**, e.g., a column vector $(1, 0, \dots, 0) \in \mathbb{F}^d$. The size of \mathcal{M} is the number m of rows.*

As ψ labels each row with a number from $\{1, \dots, m\}$ corresponding to a fixed player, we can think of each player as being the “owner” of one or more rows. For every player we consider a function φ which gives the set of rows owned by the player, i.e., φ is “inverse” of ψ . Note the difference between $M_{\varphi(G)}$ for $G \subseteq P$ and M_N for $N \subseteq \{1, \dots, m\}$, but for the sake of simplicity we will write M_G instead of $M_{\varphi(G)}$.

An MSP is said to *compute* a (complete) access structure Γ when $\varepsilon \in \text{Im}(M_G^T)$ if and only if G is a member of Γ . Hence, the players can reconstruct the secret precisely if the rows they own contain in their linear span the target vector of \mathcal{M} , and otherwise they get no information about the secret, i.e., there exists a so called *recombination vector* \mathbf{r} such that $M_G^T \mathbf{r} = \varepsilon$. Thus $\langle \mathbf{r}, M_G(s, \mathbf{c}) \rangle = \langle M_G^T \mathbf{r}, (s, \mathbf{c}) \rangle = \langle \varepsilon, (s, \mathbf{c}) \rangle = s$ for any secret s and any vector \mathbf{c} . It is well known that the vector $\varepsilon \notin \text{Im}(M_N^T)$ if and only if there exists a vector $\mathbf{k} \in \mathbb{F}^d$ such that $M_N \mathbf{k} = 0$ and $\mathbf{k}_1 = 1$.

Because of the linearity LSSSs provide it is easy to add secrets securely: it is sufficient for each player to add up the shares he holds. Therefore, to achieve general MPC, it suffices to implement multiplication of shared secrets. That is, we need a protocol where each player initially holds shared secrets s and s' , and ends up holding a share of the product ss' . Several such protocols are known for the threshold case [1, 2, 7, 8] and for general access structure [3].

We follow the approach proposed by Cramer *et al.* in [3] to build an MPC from any LSSS, provided that the LSSS is called (*strongly*) *multiplicative*. Loosely speaking, an LSSS is (strongly) multiplicative if each player i can, from his shares of secrets s and s' , compute a value c_i , such that the product ss' can be obtained using all values (only values from honest players). Let Γ be an access structure, computed by the MSP $\mathcal{M} = (\mathbb{F}, M, \varepsilon, \psi)$. Given two m -vectors \mathbf{x} and \mathbf{y} , Cramer *et al.* in [3] denote $\mathbf{x} \diamond \mathbf{y}$ to be the vector containing all the entries of the form $x_i y_j$, where $\psi(i) = \psi(j)$. Thus, if $m_i = |\varphi(i)|$ is the number of rows owned by a player i , then $\mathbf{x} \diamond \mathbf{y}$ has $\bar{m} = \sum_i m_i^2$ entries. So, if \mathbf{x} and \mathbf{y} contain

shares resulting from sharing two secrets using \mathcal{M} , then the vector $\mathbf{x} \diamond \mathbf{y}$ can be computed using only local computations by the players, i.e., each component of the vector can be computed by one player. Denote by \mathcal{M}_A the MSP obtained from \mathcal{M} restricted to the set players A .

Definition 3. [3] A **multiplicative MSP** is an MSP \mathcal{M} for which there exists an \bar{m} -vector \mathbf{r} called a **recombination vector**, such that for any two secrets s' and s'' and any random vectors \mathbf{c}' and \mathbf{c}'' , it holds that

$$s' s'' = \langle \mathbf{r}, M(s', \mathbf{c}') \diamond M(s'', \mathbf{c}'') \rangle.$$

It is said that \mathcal{M} is **strongly multiplicative** if for any subset A of honest players \mathcal{M}_A is multiplicative.

In the recent paper of Cramer *et al.* [4] this definition is rephrased.

Definition 4. [4] The MSP \mathcal{M} is called **multiplicative** if there exists a block-diagonal matrix $D \in \mathbb{F}^{m \times m}$ such that $M^T D M = \varepsilon \varepsilon^T$, where block-diagonal is to be understood as follows. Let the rows and columns of D be labelled by ψ , then the non-zero entries of D are collected in blocks D_1, \dots, D_n such that for every player $i \in P$ the rows and columns in D_i are labelled by i . \mathcal{M} is called **strongly multiplicative** if, for any subset A of honest players \mathcal{M}_A is multiplicative.

Hirt and Maurer [9] call the adversary structure Q^2 (Q^3) if no two (three) sets in Δ_A cover the full player set P . Unconditional secure MPC for arbitrary Q^2 (in the passive case) and Q^3 (in the active case) access structures has been completely solved by Hirt and Maurer [9]. Efficient MPC (no error in the passive case and negligible error in the active case) from LSSS has been proposed by Cramer *et al.* [3]. They have also proposed a LSSS with strong multiplication, but for this case both their solution and the solution of Hirt and Maurer are not efficient. Defining complexity measure for MPC is rather subtle. For that reason the complexity of the MSP is used, which is a measure of the complexity of its adversary structure.

Define $msp_{\mathbb{F}}(f)$ to be the size of the smallest MSP over \mathbb{F} computing a monotone boolean function f . Next define $\mu_{\mathbb{F}}(f)$ to be the size of the smallest multiplicative MSP over \mathbb{F} computing f . Similarly, define $\mu_{\mathbb{F}}^*(f)$ to be the size of the smallest strongly multiplicative MSP. In other words for a given adversary \mathcal{A} with adversary structure Δ_A the requirement is for every set $B \in \Delta_A$ to have $B \notin \Gamma$, but $B^c \in \Gamma$. By definition, we have $msp_{\mathbb{F}}(f) \leq \mu_{\mathbb{F}}(f) \leq \mu_{\mathbb{F}}^*(f)$. In [3] Cramer *et al.* characterized the functions that (strongly) multiplicative MSPs can compute, and proved that the multiplication property for an MSP can be achieved without loss

of efficiency. In particular, for the passive (multiplicative) case they proved that $\mu_{\mathbb{F}}(f) \leq 2 \text{msp}_{\mathbb{F}}(f)$ provided that f is Q^2 function. Unfortunately there is no similar result for the strongly multiplicative case. Instead the authors in [3] proved that for an active adversary (strongly multiplicative) case $\mu_{\mathbb{F}}^*(f)$ is bounded by the so-called “formula complexity”, provided that f is Q^3 function.

Recently Maurer [11] has proved that general unconditional information-theoretically MPC secure against a mixed (Δ_1, Δ_A) -adversary is possible if and only if $P \notin \Delta_1 \uplus \Delta_1 \uplus \Delta_A$ or equivalently if and only if $\Gamma_A^\perp \subseteq \Gamma_1 \uplus \Gamma_1$. Another important recent result, which gives necessary and sufficient conditions for the existence of an information-theoretically secure VSS, against a mixed (Δ_1, Δ_A) -adversary, has been proved by Fehr and Maurer in [6]: the robustness conditions for VSS are fulfilled if and only if $P \notin \Delta_1 \uplus \Delta_A \uplus \Delta_A$ or equivalently if and only if $(\Gamma_A \uplus \Gamma_A)^\perp \subseteq \Gamma_1$. We will refer to those two results as the MPC and VSS *conditions*.

Our Results. We will use the approach proposed by Cramer *et al.* in [3] for building General Secure Multi-Party Computation based on an underlying linear secret sharing scheme. First we expand the construction proposed by Cramer *et al.* in [3]. Let Γ_1 and Γ_2 be access structures, computed by MSPs $\mathcal{M}_1 = (\mathbb{F}, M1, \varepsilon1, \psi_1)$ and $\mathcal{M}_2 = (\mathbb{F}, M2, \varepsilon2, \psi_2)$. Let also $M1$ be an $m_1 \times d_1$ matrix, $M2$ be an $m_2 \times d_2$ matrix and φ_1, φ_2 are the “inverse” functions of ψ_1 and ψ_2 . Given an m_1 -vector \mathbf{x} and an m_2 -vector \mathbf{y} , we denote $\mathbf{x} \diamond \mathbf{y}$ to be the vector containing all entries of form $x_i y_j$, where $\psi_1(i) = \psi_2(j)$. Thus $\mathbf{x} \diamond \mathbf{y}$ has $m = \sum_i |\varphi_1(i)| |\varphi_2(i)|$ entries (notice that $m < m_1 m_2$). So, if \mathbf{x} and \mathbf{y} contain shares resulting from sharing two secrets using \mathcal{M}_1 and \mathcal{M}_2 , then the vector $\mathbf{x} \diamond \mathbf{y}$ can be computed using only local computation by the players, i.e., each component of the vector can be computed by one player. In other words we define the operation diamond \diamond for vectors (and analogously for matrices) as concatenation of vectors (matrices), which are the tensor multiplication (\otimes) of the sub-vectors (sub-matrices) belonging to a fixed player. In order to better characterize the multiplicative property of an MSP we introduce a new notion *multiplicative resulting MSP*.

Definition 5. Define MSP \mathcal{M} to be $(\mathbb{F}, M = M1 \diamond M2, \varepsilon = \varepsilon1 \diamond \varepsilon2, \psi)$, where $\psi(i, j) = r$ if and only if $\psi_1(i) = \psi_2(j) = r$. Given two MSPs \mathcal{M}_1 and \mathcal{M}_2 , the MSP \mathcal{M} is called their **multiplicative resulting MSP** if there exists an m -vector \mathbf{r} called a recombination vector, such that for any two secrets s' and s'' and any random vectors \mathbf{c}' and \mathbf{c}'' , it holds that

$$s' s'' = \langle \mathbf{r}, M1 (s', \mathbf{c}') \diamond M2 (s'', \mathbf{c}'') \rangle = \langle \mathbf{r}, M((s', \mathbf{c}') \otimes (s'', \mathbf{c}'')) \rangle.$$

An MSP \mathcal{M} is called a **strongly multiplicative resulting MSP** if for the access structure Γ computed by \mathcal{M} we have $\{P\} \subset \Gamma$.

This means that one can construct a multiplicative resulting MSP with which some subsets of players are able to compute the product of the secrets shared by MSPs \mathcal{M}_1 and \mathcal{M}_2 ; these subsets constitute a new access structure (called *resulting*) Γ . The difference between the multiplicative resulting MSP and the strongly multiplicative resulting MSP is that in the first one $\Gamma = \{P\}$.

Recall that in [3] the mixed adversary model is not considered, i.e. the authors consider access structures Γ_1 such that $\Delta_A = \Delta_1$ is \mathcal{Q}^2 (\mathcal{Q}^3). The intuition behind this new definition is the following. In [3] two scenarios (ways) to build MPC are proposed:

- 1) For a given \mathcal{Q}^2 (\mathcal{Q}^3) access structure Γ_1 find (directly construct) a (strongly) multiplicative MSP computing Γ_1 .
- 2) For an MSP \mathcal{M}_1 computing the \mathcal{Q}^2 (\mathcal{Q}^3) access structure Γ_1 , construct a new (strongly) multiplicative MSP \mathcal{M}'_1 computing the same access structure.

It is shown in [3] that in the multiplicative case, for any MSP \mathcal{M}_1 one can efficiently construct multiplicative MSP \mathcal{M}'_1 computing the same access structure. Hence scenario 2) applies in that case. But for the strongly multiplicative case there is no efficient solution neither for scenario 1) nor for 2).

On the other hand we consider more grained mixed adversaries with \mathcal{Q}^2 , (\mathcal{Q}^3) adversary structure. The adversary is called (Δ_1, Δ_A) -adversary if Δ_1 is its privacy structure and $\Delta_A \subseteq \Delta_1$ is its adversary structure. In our adversary model we have adversary with two privacy structures Δ_1 , Δ_2 and with one adversary structure $\Delta_A \subseteq \Delta_1$, $\Delta_A \subseteq \Delta_2$, let us call it $(\Delta_1, \Delta_2, \Delta_A)$ -adversary. In our MPC model there are also two scenarios.

- A) Find conditions for the MSPs \mathcal{M}_1 and \mathcal{M}_2 , computing Γ_1 and Γ_2 respectively, such that the access structure Γ (Γ is the (strongly) multiplicative resulting access structure) fulfills certain conditions.
- B) For a MSP \mathcal{M}_1 computing Γ_1 , find second MSP \mathcal{M}_2 , computing Γ_2 , such that the resulting access structure Γ fulfills certain conditions.

We will discuss later which conditions Γ should fulfil, in order to obtain secure MPC. Note that our main goal is to investigate the properties of the access structure Γ and the MSP \mathcal{M} and how these properties depend on the initial MSPs, while the approaches in [3] are focused on the constructions. A partial answer for scenario A) is given in Proposition 1,

stating that for the resulting access structure Γ we have $\Gamma \subseteq \Gamma_1 \uplus \Gamma_2$. Unfortunately, we still do not know when the equality holds. But solving this problem we will yield an efficient solution for the strongly multiplicative case. Our second main result Theorem 1 shows that the access structure Γ computed by the resulting MSP \mathcal{M} of MSPs \mathcal{M}_1 and \mathcal{M}_1^\perp is in fact the whole set of players P . Theorem 1 implies that only all players together can compute the product of the secrets, hence \mathcal{M} is the multiplicative resulting MSP, but not the strongly multiplicative resulting MSP. Hence for the multiplicative case scenario B) holds, for any \mathcal{M}_1 with its dual $\mathcal{M}_1^\perp = \mathcal{M}_2$. Unfortunately this result also means that the construction proposed by Cramer *et al.* in [3] is not applicable in the strongly multiplicative case, i.e. even if we apply it for the Q^3 access structure. Let us define $\nu_{\mathbb{F}}(f)$ to be the size of the smallest multiplicative resulting MSP over \mathbb{F} computing f and respectively $\nu_{\mathbb{F}}^*(f)$ to be the size of the smallest strongly multiplicative resulting MSP. In fact by the definition of the operation \diamond (see Definition 5) this size depends on the sizes of the two initial MSPs, thus it is more accurate to denote it by $\nu_{\mathbb{F}}(f_1, f_2)$ ($\nu_{\mathbb{F}}^*(f_1, f_2)$). Denote by f^* the function which is the dual of f . The third main result, Theorem 2, shows that $msp_{\mathbb{F}}(f) = \nu_{\mathbb{F}}(f, f^*) \leq \nu_{\mathbb{F}}(f, f) = \mu_{\mathbb{F}}(f)$ and $\nu_{\mathbb{F}}^*(f, \bar{f}) \leq \nu_{\mathbb{F}}^*(f, f) = \mu_{\mathbb{F}}^*(f)$. The relations mean that when we use a (strongly) multiplicative MSP to compute the multiplicative resulting MSP the efficiency is the same. However, if we have an MSP without the (strongly) multiplicative property the usage of specific pair of MSPs (e.g. the given one and its dual in the multiplicative case) we gain better efficiency. The knowledge of the access structure Γ allows us to find which recombination vector corresponds to each qualified group. In the adversary model we consider, for a given adversary \mathcal{A} with adversary structure $\Delta_{\mathcal{A}}$ the requirement is for every set $B \in \Delta_{\mathcal{A}}$ to have $B \notin \Gamma_1$, $B \notin \Gamma_2$ but $B^c \in \Gamma$. Recently Maurer [11] gave necessary and sufficient conditions for the existence of secure MPC in the mixed adversary model. Since Maurer considers general SSS, it was not clear whether using only LSSS these conditions still hold. In our model these conditions correspond to the conditions Γ should fulfill. And as we prove in Theorems 3 and 4 in both settings (unconditional information-theoretic and computational) for secure general MPC we have similar to those of Maurer conditions.

3 Enhanced Construction

3.1 The Diamond \diamond Construction and its Properties

A natural construction for the resulting MSP is the well known Kronecker product (construction \otimes) of matrices. The problem with this construction

is that we do not know whom each new row belongs to, since we multiply a row owned by one player to a row owned by another player, hence local computation is not applicable. To avoid the inherent problem of the construction \otimes , we study the *diamond* \diamond construction.

Some useful properties of the matrices $M = M1 \otimes M2$ and $M = M1 \diamond M2$ are given in an earlier version of this paper [13].

Consider the vector \mathbf{x} . Let us collect the coordinates in \mathbf{x} , which belong to the player t in a sub-vector \bar{x}_t or $\mathbf{x} = (\bar{x}_1, \dots, \bar{x}_n)$. Hence $\bar{x}_t \in \mathbb{F}^{|\varphi(t)|}$. The operation diamond \diamond for vectors could be defined as: $\mathbf{x} \diamond \mathbf{y} = (\bar{x}_1 \otimes \bar{y}_1, \dots, \bar{x}_n \otimes \bar{y}_n)$. We define an operation diamond for matrices and we denote the new matrix by $M = M1 \diamond M2$. We construct the new matrix M as follows. Denote by $M1_t$ the matrix formed by rows of $M1$ owned by player t and correspondingly by $M2_t$ the matrix formed by rows of $M2$ owned by player t . Thus the construction diamond \diamond for $M = M1 \diamond M2$ is the concatenation of matrices $M1_t \otimes M2_t$ for $t = 1, \dots, n$. First we show that the construction is symmetric regarding to the MSPs \mathcal{M}_1 and \mathcal{M}_2 .

Lemma 1. *The MSPs $\mathcal{M} = \mathcal{M}_1 \diamond \mathcal{M}_2$ and $\widetilde{\mathcal{M}} = \mathcal{M}_2 \diamond \mathcal{M}_1$ compute the same access structure Γ .*

Lemma 2. *Let $M1$ be an $m_1 \times d_1$ matrix, and $M2$ be an $m_2 \times d_2$ matrix. Construct the matrix M following the construction \diamond (i.e., $M = M1 \diamond M2$ is $m \times d_1 d_2$ matrix), then for arbitrary column vectors $\lambda_1 \in \mathbb{F}^{d_1}$, $\lambda_2 \in \mathbb{F}^{d_2}$ the following equality holds: $(M1 \diamond M2) (\lambda_1 \otimes \lambda_2) = (M1 \lambda_1) \diamond (M2 \lambda_2)$.*

Note that the construction diamond \diamond confirms our intuitive expectations that the players could locally compute their new shares, as shown in the following lemma.

Lemma 3. *Let us denote by $\mathbf{s1} = M1 (s_1, \mathbf{a})$ and $\mathbf{s2} = M2 (s_2, \mathbf{b})$ the shares distributed by MSPs \mathcal{M}_1 and \mathcal{M}_2 , for the secrets s_1 and s_2 resp. Then MSP \mathcal{M} actually distributes shares $\mathbf{s} = \mathbf{s1} \diamond \mathbf{s2}$ for the secret $s_1 s_2$.*

Note that we have $\mathbf{s} = (M1 \diamond M2) ((s_1, \mathbf{a}) \otimes (s_2, \mathbf{b}))$ and that the vector $(s_1, \mathbf{a}) \otimes (s_2, \mathbf{b})$ is no longer random. Now we are in position to prove our first main result using the operation \uplus and the construction \diamond we have introduced.

Proposition 1. *Let Γ_1 and Γ_2 be the access structures computed by the MSPs \mathcal{M}_1 and \mathcal{M}_2 . Let the MSP \mathcal{M} be the strongly multiplicative result of MSPs \mathcal{M}_1 and \mathcal{M}_2 , and let the access structure Γ be computed by the MSP \mathcal{M} . Then $\Gamma \subseteq \Gamma_1 \uplus \Gamma_2$. (Notice that Γ may be trivial, e.g. \emptyset .)*

Proof: Let $A1 \notin \Gamma_1$. Hence there exists a vector $\mathbf{k} \in \text{Ker}(M1_{A1})$ such that $\mathbf{k}_1 = 1$. Analogously, let $A2 \notin \Gamma_2$. Hence there exists a vector $\mathbf{r} \in \text{Ker}(M2_{A2})$ such that $\mathbf{r}_1 = 1$. Notice that $\mathbf{k} \in \mathbb{F}^{d_1}$ and $\mathbf{r} \in \mathbb{F}^{d_2}$. Let $A = A1 \cup A2$, so we have $A \notin \Gamma_1 \uplus \Gamma_2$. Form a new vector $\mathbf{k} \otimes \mathbf{r} \in \mathbb{F}^{d_1 d_2}$. It is easy to check that the vector $\mathbf{k} \otimes \mathbf{r} \in \text{Ker}(M_A)$ and $(\mathbf{k} \otimes \mathbf{r})_1 = 1$. Hence $A \notin \Gamma$, thus $\Gamma \subseteq \Gamma_1 \uplus \Gamma_2$. \square

3.2 Properties of the Resulting MSP

An interesting open question is when the “equality” holds? One can see from the examples given in [13] that “equality” does not always hold. Consider for example the threshold case. Denote by $T_{s,n}$ the s -out-of- n threshold access structure, then it is easy to verify that $T_{l,n} \uplus T_{s,n} = T_{l+s-1,n}$. On the other hand each player t holds vectors $\mathbf{w} = (1, \alpha_t, \dots, \alpha_t^{s-1})$ and $\mathbf{v} = (1, \alpha_t, \dots, \alpha_t^{l-1})$ from MSPs computing $T_{s,n}$ and $T_{l,n}$ correspondingly. Thus the construction proposed above gives $\mathbf{v} \otimes \mathbf{w} = (1, \alpha_t, \dots, \alpha_t^{s-1}, \alpha_t, \alpha_t^2, \dots, \alpha_t^s, \dots, \alpha_t^{l-1}, \dots, \alpha_t^{s+l-2})$.

Using the fact that without changing the access structure we can always replace the 2nd up to the last column of M by any set of vectors that generates the same space we obtain that $\mathbf{v} \otimes \mathbf{w}$ is equivalent to $(1, \alpha_t, \dots, \alpha_t^{s+l-2})$. But this is exactly the row owned by the player t in MSP computing $T_{l+s-1,n}$. This means that in the threshold case we have equality in Proposition 1. That is why we believe that for an MSP \mathcal{M}_1 there should exist another MSP \mathcal{M}_2 such that for their strongly multiplicative resulting MSP \mathcal{M} , computing the access structure Γ , we have $\Gamma = \Gamma_1 \uplus \Gamma_2$. The first step in this direction is [3, Theorem 7], where \mathcal{M}_1 and \mathcal{M}_2 are dual, i.e., $\Gamma_2^\perp = \Gamma_1$. Cramer *et al.* have proved in [3, Theorem 7] that $\varepsilon = \varepsilon 1 \diamond \varepsilon 1$ belongs to the linear span of the rows of $M = M1 \diamond M1^\perp$, when the matrices $M1$ and $M1^\perp$ satisfy the condition $M1^T M1^\perp = \overline{E}$. Here $\overline{E} = \varepsilon 1 \varepsilon 1^T$ is the matrix with zeros everywhere, except in its upper-left corner where the entry is 1. It is known [3] how to derive the matrix $M1^\perp$ from $M1$ such that they satisfy $M1^T M1^\perp = \overline{E}$.

One of the key results in [3] is a method to construct, from any MSP \mathcal{M}_1 with Q^2 access structure Γ_1 , a multiplicative MSP \mathcal{M}' with the same access structure and with twice bigger size (hence with twice bigger complexity). Unfortunately no similar result is known for the strongly multiplicative case. It is natural to ask what happens if \mathcal{M}_1 computes the Q^3 access structure Γ_1 instead. We are ready to prove our second main result, which gives an answer to this question.

Theorem 1. *Let Γ_1 and Γ_1^\perp be the connected access structures computed by the MSPs \mathcal{M}_1 and \mathcal{M}_1^\perp and $M^T M^\perp = \overline{E}$ holds. Let the MSP \mathcal{M} be the strongly multiplicative result of MSPs \mathcal{M}_1 and \mathcal{M}_1^\perp , and let the access structure Γ be computed by the MSP \mathcal{M} . Then $\Gamma = \Gamma_1 \uplus \Gamma_1^\perp = \{P\}$.*

Proof: It is known that $\{P\} \in \Gamma$. On the other hand from Proposition 1 we have $\Gamma \subseteq \Gamma_1 \uplus \Gamma_1^\perp$, thus it is sufficient to prove that there is no other sets in $\Gamma_1 \uplus \Gamma_1^\perp$ except $\{P\}$.

For any set $A \in \Delta_1^+$ and any player $i \in P, i \notin A$ we have $(A \cup i) \in \Gamma_1$. Set $B^c = A \cup i$ and hence $B = P \setminus B^c \in \Delta_1^\perp$. Therefore $A \cup B = (P \setminus i) \in (\Delta_1 \uplus \Delta_1^\perp)$. Let us assume that there exists a player j such that $(P \setminus j) \notin (\Delta_1 \uplus \Delta_1^\perp)$. So, $j \in A$ for every set $A \in \Delta_1^+$, because otherwise using the construction given above we arrive at a contradiction. Hence the access structure Γ_1 has the star topology for the forbidden sets, i.e., there exists a player j such that for any set $A \in \Delta^+, j \in A$. Hence Γ_1 is not connected – contradiction and we are done. \square

As an example let us consider again the threshold case. Taking into account that $(T_{l,n})^\perp = T_{n-l+1,n}$, we have $T_{l,n} \uplus (T_{l,n})^\perp = T_{n,n} = \{P\}$, which is in accordance with Theorem 1.

3.3 Relations with Multiplicative MSPs

Lemma 4. *Let \mathcal{M} be a multiplicative MSP computing Γ and satisfying $M^T DM = \overline{E}$ for some block-diagonal matrix D (Definition 4). Define MSP $\overline{\mathcal{M}}$ computing $\overline{\Gamma}$ by $\overline{M} = DM$. Then $\Gamma^\perp \subseteq \overline{\Gamma} \subseteq \Gamma$ holds.*

Note that as a consequence we obtain that $M^T DM = \overline{E}$ imply $\Gamma^\perp \subseteq \Gamma$, i.e. the \mathcal{Q}^2 property. Let \mathcal{M} be multiplicative MSP and D be a block-diagonal matrix satisfying the condition from Definition 4. Then for any invertible block-diagonal matrix \tilde{D} the matrices $\overline{M} = \tilde{D}M$ and $\overline{D} = (\tilde{D}^{-1})^T D \tilde{D}^{-1}$ satisfy also the condition from Definition 4.

Corollary 1. *For any self-dual access structure Γ there exist MSPs \mathcal{M} and \mathcal{M}^\perp and block-diagonal matrix D such that the following relations hold $M^T M^\perp = \overline{E}$ and $DM = M^\perp$.*

Theorem 2. *For any (strongly) multiplicative MSP \mathcal{M} computing $\Gamma(f)$ and its dual MSP \mathcal{M}^\perp computing $\Gamma^\perp(f^*)$ we have*

$$msp_{\mathbb{F}}(f) = \nu_{\mathbb{F}}(f, f^*) \leq \nu_{\mathbb{F}}(f, f) = \mu_{\mathbb{F}}(f) \quad \text{and} \quad \nu_{\mathbb{F}}^*(f, \bar{f}) \leq \nu_{\mathbb{F}}^*(f, f) = \mu_{\mathbb{F}}^*(f).$$

Proof: For the sake of simplicity we will prove only the multiplicative case, since the strongly multiplicative case is a straightforward consequence. Let M be an $m \times d$ matrix, thus D is an $m \times m$ matrix. Let's

compute $M^T DM$ denoting by $d_{i,j}$ the element in i -th row and j -th column in D . Thus $M^T DM = \sum_{i,j=1}^m d_{i,j} M_i \bar{\otimes} M_j$, where $\bar{\otimes}$ is the tensor matrix product. But $\sum_{i,j=1}^m d_{i,j} M_i \bar{\otimes} M_j = \bar{E}$ is equivalent to $\sum_{i,j=1}^m d_{i,j} M_i \otimes M_j = \varepsilon$, since the only difference is the way the tensor product is presented in matrix or in vector form. Thus the condition $M^T DM = \bar{E}$ for some block-diagonal matrix D is equivalent to the condition of the existence of a recombination vector \mathbf{r} for the resulting matrix $M \diamond M$. In fact the block-diagonal matrix D is the recombination vector \mathbf{r} written in matrix block form. Thus we prove the right equality, namely $\nu_{\mathbb{F}}(f, f) = \mu_{\mathbb{F}}(f)$.

Revisiting the construction for multiplicative MSP given in [3], we notice that the matrix \widetilde{M} from the multiplicative MSP consists of two separate parts (matrices) M and M^\perp . Thus sharing a secret s by \widetilde{M} with random vector of the form (\mathbf{a}, \mathbf{b}) , where $\mathbf{a}, \mathbf{b} \in \mathbb{F}^{d-1}$ we have $\widetilde{\mathbf{s}} = \widetilde{M}(s, \mathbf{a}, \mathbf{b})$. Define $\mathbf{s} = M(s, \mathbf{a})$ and $\mathbf{s}^\perp = M^\perp(s, \mathbf{b})$. Notice that $\widetilde{\mathbf{s}} = (\mathbf{s}, \mathbf{s}^\perp)$. Therefore using the construction of Cramer *et al.* for multiplicative MSP we have two shares of the secret s : one corresponding to M and one corresponding to its dual M^\perp . Now considering a multiplication gate we have as input two secrets s_1 and s_2 sharing them with \widetilde{M} gives us shares for s_i ($i = 1, 2$) for both M and M^\perp . On the other hand using the resulting MSP of M and M^\perp we need only shares of s_1 shared by M and s_2 shared by M^\perp . Thus we need twice less shares to be distributed. Therefore we have $m_{sp_{\mathbb{F}}}(f) = m_{sp_{\mathbb{F}}}(f^*) = \nu_{\mathbb{F}}(f, f^*)$ and that is the best possible, since we always need to share the two inputs to a given multiplication gate. \square

The fact that we use two different MSP to share the inputs in every multiplication gate make the computation of a given arithmetic circuit more complicated compared to the case when all inputs are shared just by one MSP. Let consider some examples:

- If the function we want to compute is $s_1 s_2$, then as we proved we need to share s_1 by M and s_2 by M^\perp and we are twice more efficient, note that this is the best possible improvement.
- On the other hand if the function we want to compute is s^2 , then in fact sharing s by M and M^\perp gives us the same as sharing it by \widetilde{M} thus the efficiency here is the same.
- Another indicative example is when the function we want to compute is $s_1 s_2 + s_2 s_3 + s_3 s_1$, then we share s_1 by M and s_2 by M^\perp , but then we are forced to share s_3 by both M and M^\perp , (i.e. by \widetilde{M}). Thus we are $\frac{3}{2}$ times more efficient.

Since the function we want to compute is public it is required in our model to figure out in advance for each multiplication gate which MSP

we will use $(M, M^\perp$ or $\widetilde{M})$. We can compare this to coloring a graph with two colors (for M and M^\perp), but some nodes could be colored by both colors. Thus the following question arises: classify the functions by the criterion whether the inputs and all nodes could be “colored” only by the two colors, i.e. there are no nodes colored by both colors.

4 Adaptive, Active Adversary: The Zero-Error Case

Recall that in our adversary model we have adversary with two privacy structures Δ_1, Δ_2 and with one adversary structure $\Delta_A \subseteq \Delta_1, \Delta_A \subseteq \Delta_2$. To build secure MPC protocol we employ the error-free commitment protocols [3], provided that the MSP we have is strongly multiplicative.

The use of strongly multiplicative LSSS allows to compute the product of two secrets without interaction between the players. Unfortunately in the general case the picture coincides with the threshold case. As Ben-Or *et al.* note in their seminal paper [1] the new shares computed after local multiplication correspond to a higher (double) degree polynomial which is not random. To overcome this problem they introduced a degree reduction and randomization protocols. Later Gennaro *et al.* [7] achieve both tasks in a single step, which they call an algebraic simplification for the multiplication protocol. As we noticed in the case of general access structures we have the same problem as described by Ben-Or *et al.* The new shares computed after local multiplication correspond to a much “smaller” access structure Γ and the shares are computed using a non-random vector. On the other hand the knowledge of the access structure Γ allows us to build an analog of the algebraic simplification protocol of Gennaro *et al.* [7], which we will describe in the next subsection.

4.1 Algebraic Simplification for the Multiplication Protocol on a General Access Structure

Let the two secrets s_1 and s_2 are shared using the MSPs \mathcal{M}_1 and \mathcal{M}_2 (computing Γ_1 and Γ_2 respectively). Denote their resulting MSP as usual by \mathcal{M} with access structure Γ . Let us choose another MSP \mathcal{M}_3 computing Γ_3 to which we want to reduce Γ . Then the simplified multiplication protocol is as follows:

1. Each player i multiplies locally his shares (for simplicity let they own one share from each of the access structures and denote them by) $\mathbf{s1}_i$ and $\mathbf{s2}_i$.

2. Then the player i chooses a random vector $\mathbf{h}(\mathbf{i})$ such that its first coordinate is the product, (i.e., $\mathbf{s}\mathbf{1}_i \mathbf{s}\mathbf{2}_i = \mathbf{s}_i$.)
3. With the MSP \mathcal{M}_3 and applying the VSS protocol the i -th player re-shares its product \mathbf{s}_i , i.e. using vector $\mathbf{h}(\mathbf{i})$.
4. In this way every player k receives from player i a temporary share, denoted by $\mathbf{ts}(\mathbf{i})_k$.
5. For some set of “good” players $A \in \Gamma$ with recombination vector λ , each player k calculates its new-share $\mathbf{ns}_k = \sum_{i \in A} \mathbf{ts}(\mathbf{i})_k \lambda_i$.
6. Finally the new-shares have the property that any set of “good” players $B \in \Gamma_3$ could restore the secret $s_1 s_2$.

For a proof that this protocol is correct and secure we refer to [13].

4.2 Information-Theoretic Settings

In order to build an MPC protocol secure against active, adaptive adversary in the non-computational model it is sufficient for the MSPs $\mathcal{M}_1, \mathcal{M}_2$ and \mathcal{M}_3 to satisfy the VSS conditions from [6] and Γ to be the strongly multiplicative result of MSPs computing Γ_1 and Γ_2 . Using the algebraic simplification protocol, and the homomorphic commitments (information-theoretic secure VSS) [3] we could “reduce” the access structure Γ to any access structure Γ_3 , which we call “reduced”, provided Γ_3 satisfies the VSS conditions. Hence combining Proposition 1 and the VSS conditions of Fehr and Maurer our fourth main result follows.

Theorem 3. *Let Γ be the access structure computed by the strongly multiplicative resulting MSP $\mathcal{M} = \mathcal{M}_1 \diamond \mathcal{M}_2$ and Γ_3 be the “reduced” access structure. Then the sufficient conditions for the existence of general unconditional information-theoretically secure MPC, secure against $(\Delta_1, \Delta_2, \Delta_A)$ -adversary are:*

$$\Gamma_A^\perp \subseteq \Gamma \subseteq \Gamma_1 \uplus \Gamma_2, \quad (\Gamma_A \uplus \Gamma_A)^\perp \subseteq \Gamma_i, \quad \text{for } i = 1, 2, 3.$$

Note that from Theorem 3 it follows that we have $P \notin \Delta_1 \uplus \Delta_2 \uplus \Delta_A$, which corresponds to the condition of Maurer [11].

4.3 Computational Settings

In order to build an MPC protocol secure against an active adversary in the computational model it is sufficient for the MSPs $\mathcal{M}_1, \mathcal{M}_2, \mathcal{M}_3$ to satisfy the VSS conditions and for Γ to be the strongly multiplicative result of MSPs computing Γ_1 and Γ_2 . Again using the algebraic simplification protocol, and the homomorphic commitments (computational secure

VSS plus one-way trapdoor permutations) [3, 7] we could “reduce” the access structure Γ to any access structure Γ_3 , provided Γ_3 satisfy VSS conditions. Hence we obtain our next result.

Theorem 4. *Let Γ be the access structure computed by the strongly multiplicative resulting MSP $\mathcal{M} = \mathcal{M}_1 \diamond \mathcal{M}_2$ and Γ_3 be the “reduced” access structure. If a trapdoor one-way permutation exists, then the sufficient conditions for the existence of general unconditional secure MPC in the cryptographic scenario, secure against $(\Delta_1, \Delta_2, \Delta_A)$ -adversary are:*

$$\Gamma_A^\perp \subseteq \Gamma \subseteq \Gamma_1 \uplus \Gamma_2, \quad \Gamma_A^\perp \subseteq \Gamma_i, \quad \text{for } i = 1, 2, 3.$$

Note again the similarity of the conditions for existence of MPC.

Acknowledgements. The authors would like to thank Ronald Cramer and Ivan Damgård for the helpful discussions and comments.

References

1. M. Ben-Or, S. Goldwasser, A. Wigderson, Completeness theorems for Non-Cryptographic Fault-Tolerant Distributed Computation, *STOC 1988*, 1988, pp. 1-10.
2. D. Chaum, C. Crepeau, I. Damgård, Multi-Party Unconditionally Secure Protocols, *STOC 1988*, 1988, pp. 11-19.
3. R. Cramer, I. Damgård, U. Maurer, General Secure Multi-Party Computation from any linear secret sharing scheme, *EUROCRYPT 2000*, LNCS 1807, pp. 316-334.
4. R. Cramer, S. Fehr, Y. Ishai, E. Kushilevitz, Efficient Multi-Party Computation over Rings, *EUROCRYPT 2003*, LNCS 2656, pp. 596-613.
5. B. Chor, S. Goldwasser, S. Micali, B. Awerbuch, Verifiable secret sharing and achieving simultaneity in the presence of faults, *FOCS 1985*, pp. 383-395.
6. S. Fehr, U. Maurer, Linear VSS and Distributed Commitments Based on Secret Sharing and Pairwise Checks, *CRYPTO 2002*, LNCS 2442, pp. 565-580.
7. R. Gennaro, M. Rabin, T. Rabin, Simplified VSS and Fast-Track Multi-party Computations with Applications to Threshold Cryptography, *PODC'98*, pp. 101-111.
8. O. Goldreich, S. Micali, A. Wigderson, How to Play Any Mental Game or a Completeness Theorem for Protocols with Honest Majority, *STOC'87*, pp. 218-229.
9. M. Hirt, U. Maurer, Complete characterization of Adversaries Tolerable in General Multiparty Computations, *PODC'97*, pp. 25-34.
10. M. Karchmer, A. Wigderson. On Span Programs, *Proc. of 8-th Annual Structure in Complexity Theory Conference*, 1993, pp. 102-111.
11. U. Maurer, Secure Multi-Party Computation Made Simple, *3rd Conference on Security in Communication Networks 2002*, LNCS 2576, pp. 14-28, 2003.
12. V. Nikov, S. Nikova, B. Preneel, J. Vandewalle, Applying General Access Structure to Proactive Secret Sharing Schemes, *Proc. of the 23rd Symposium on Information Theory in the Benelux*, May 29-31, 2002, Universite Catolique de Lovain (UCL), Lovain-la-Neuve, Belgium, pp. 197-206, *Cryptology ePrint Archive: Report 2002/141*.
13. V. Nikov, S. Nikova, B. Preneel. Multi-Party Computation from any Linear Secret Sharing Scheme Secure against Adaptive Adversary: The Zero-Error Case, *Cryptology ePrint Archive: Report 2003/006*.
14. A. Shamir. How to share a secret, *Commun. ACM* 22, 1979, pp. 612-613.