Multi-Party Computation from any Linear Secret Sharing Scheme Secure against Adaptive Adversary: The Zero-Error Case

Ventzislav Nikov
Department of Mathematics and Computing Science,
Eindhoven University of Technology
P.O. Box 513, 5600 MB, Eindhoven, the Netherlands
v.nikov@tue.nl
Svetla Nikova, Bart Preneel
Department Electrical Engineering, ESAT/COSIC, K. U. Leuven,
Kasteelpark Arenberg 10, B-3001 Heverlee-Leuven, Belgium
svetla.nikova, bart.preneel@esat.kuleuven.ac.be

Abstract

We use a general treatment of 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), which 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. multiplying two different general access structures and we prove some properties of the resulting MSP $\mathcal{M}$. Next we expand the definition of multiplicative MSPs and we prove that when one uses dual MSPs only all players together can compute the product, i.e., the construction proposed by Cramer et al. gives only multiplicative MPC. Third, we propose a solution for the strongly multiplicative MPC (in presence of adversary). The knowledge of the resulting MSP and the access structure it computes allows us to build an analog of the algebraic simplification protocol of Gennaro et al. We show how to achieve in the computational model MPC secure against adaptive adversary in the zero-error case, through the application of homomorphic commitments. There is an open problem how efficiently we can determine Γ the access structure of the resulting MSP $\mathcal{M}$. This open problem reflects negatively on the efficiency of the proposed solution.

1 Introduction

The concept of *secret sharing* was introduced by Shamir 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 subsets have no information about it.
We call the groups who are allowed to reconstruct the secret *qualified* (denoted by Γ), and the groups who should not be able to obtain any information about the secret *forbidden* (denoted by Δ). The tuple (Γ, Δ) is called an *access structure* if $\Gamma \cap \Delta = \emptyset$. Denote by P the set of

participants in the scheme. If $\Gamma = \Delta^c$ is the complement of Δ, then we say that (Γ, Δ) is *complete* and we denote it only by Γ.
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 and Maurer [6] for recent results. The adversary is characterized by a *privacy structure* Δ and an *adversary structure* $\Delta_A \subseteq \Delta$. Denote the complement $\Gamma_A = \Delta_A^c$. Fehr and Maurer [6] and Nikov et al. [10] called this set *honest* (or *good*) players structure, which in fact appears to be misleading term. Actually its dual access structure $\Gamma_A^\perp$ should be called the honest (or good) players structure, since for any set G of good players the complement G^c is the set of corrupted players from Δ_A. Both passive and active adversaries may be either *static*, meaning that the set of corrupted 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.
Most proposed Secret Sharing Schemes (SSS) are *linear*, but the concept of a Linear Secret Sharing Scheme (LSSS) was first considered in its full generality by Karchmer and Wigderson, who introduced the equivalent notion of *Monotone Span Program* (MSP), which we describe later. Each linear SSS can be viewed as derived from a monotone span program $\mathcal{M}$ computing its access structure. On the other hand, each monotone span program gives rise to an LSSS. Hence, one can identify an LSSS with its underlying monotone span program. Such an MSP always exists, because MSPs can compute any monotone function. Now we will consider any complete access structure Γ, which describes subsets of participants that are qualified to recover the secret $s \in \mathbb{F}$ ($\mathbb{F}$ here is a finite field) in the set of possible secret values, as long as Γ admits a linear secret sharing scheme.
Since an LSSS neither guarantees reconstructability when some shares are incorrect, nor verifiability of a shared value a stronger primitive *verifiable secret sharing* (VSS) has been introduced in [5, 1]. In 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 cheaters' help. 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. A key tool for secure MPC, is VSS. We will consider the standard *synchronous model* with a *broadcast channel*.
The paper is organized as follows: In the next Section 3 we propose the main construction diamond $\diamond$ and investigate its properties. Then in Section 4 an algebraic simplification for multiplication is described. In the last section conditions for existence of MPC secure against adaptive adversary are considered.

2 Preliminaries

2.1 Related Work

The basic notation and linear algebra techniques that we will use from now on are summarized in the Appendix. The following operation (called element-wise union) for monotone decreasing sets was introduced in [10, 6].

Definition 2.1 *[10, 6] 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: $\Gamma_1 \uplus \Gamma_2 = \{A = A_1 \cup A_2; A_1 \notin \Gamma_1, A_2 \notin \Gamma_2\}^c$.*

Definition 2.2 *[2, 4] 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 $\mathbb{F}$, $\psi : \{1, \ldots, m\} \to \{1, \ldots, n\}$ is a surjective function and ε is a fixed vector, called target vector, e.g. , column vector $(1, 0, ..., 0) \in \mathbb{F}^d$. The size of $\mathcal{M}$ is the number m of rows.*

As ψ labels each row with a number from $[1, \ldots, m]$ 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.
An MSP is said to compute a (complete) access structure Γ when $\varepsilon \in Im(M^T_{\varphi(G)})$ 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 $\langle \mathbf{r}, M_G(s, \rho) \rangle = s$ and $M_G^T \mathbf{r} = \varepsilon$ for any secret s and any ρ. It is well known that the vector $\varepsilon \notin Im(M_N^T)$ if and only if there exists a $\mathbf{k} \in \mathbb{F}^d$ such that $M_N \mathbf{k} = 0$ and $\mathbf{k}_1 = 1$.
The main goal of our paper is to study the properties of a construction which builds MPCs from any LSSS. Because of the linearity LSSS 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, 3, 7, 8] and for general access structure [2, 4]. We follow the approach proposed by Cramer et al. in [2, 4] to build an MPC from any LSSS, provided that the LSSS is what they call *(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).
One possible construction for MSP, introduced by Cramer [2], is $M_\otimes$, i.e., a matrix obtained from matrix M by replacing each row v of M with $v \otimes v$. Denote the new MSP by $\mathcal{M}_\otimes = (\mathbb{F}, M_\otimes, \varepsilon \otimes \varepsilon, \psi)$. Hence $\mathcal{M} = (\mathbb{F}, M, \varepsilon, \psi)$ is an MSP with multiplication if and only if $\varepsilon \otimes \varepsilon \in Im(M_\otimes^T)$. It is shown also in [2] that for any MSP $\mathcal{M}$, and for all b and b', the following equality holds $s * s' = (Mb) * (Mb') = M_\otimes(b \otimes b')$. where $s * s'$ is the so-called *star product*, i.e., $s * s' = (s_1, \ldots, s_n) * (s'_1, \ldots, s'_n) = (s_1 s'_1, \ldots, s_n s'_n)$.
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 [2, 4] 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 $\overline{m} = \sum_i m_i^2$ 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. Hence when each player owns exactly one row in $\mathcal{M}$ the operations $\diamond$ and $*$ coincide.
Denote by $\mathcal{M}_A$ the MSP obtained from $\mathcal{M}$ by keeping only the rows owned by players in A, for any players subset A.

Definition 2.3 *[2, 4] A* **multiplicative** *MSP is an MSP $\mathcal{M}$ for which there exists an $\overline{m}$-vector* $\mathbf{r}$ *called a* **recombination vector***, such that for any two secrets s' and s'' and any ρ' and ρ'',* *it holds that*

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

It is said that $\mathcal{M}$ is **strongly multiplicative** *if for any subset A of players that is not qualified by $\mathcal{M}$, $\mathcal{M}_{A^c}$ is multiplicative.*

Throughout the paper we will consider presence of adaptive adversary. Since the adversary we can tolerate is at least a Q^2 adversary and since the condition Q^2 is equivalent to $\Delta_A \cap \Gamma_A^\perp = \emptyset$ (and to $\Gamma_A^\perp \subseteq \Gamma_A$), we have that the honest players structure has no intersection with the adversary structure.
Recently Maurer [9] has proved that general perfect information-theoretically MPC secure against a (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 $\Gamma_A^\perp \subseteq \Gamma_1 \uplus \Gamma_1$. Notice that thanks to the local computation model for MPC the interaction between players is reduced, and in this way we may think of the MPC as a kind of VSS.
A recent result, which gives necessary and sufficient conditions for the existence of VSS has been proved by Fehr and Maurer in [6]: the robustness, strong robustness and very strong robustness conditions for VSS are fulfilled if and only if $P \notin \Delta \uplus \Delta_A \uplus \Delta_A$ or equivalently if and only if $(\Gamma_A \uplus \Gamma_A)^\perp \subseteq \Gamma$.

2.2 Our Results

We focus on the general treatment of non-cryptographic (i.e., information-theoretically secure) multi-party computation, based on an underlying linear secret sharing scheme. Our research relies mainly on the definitions and results by Cramer et al. in [4] about General Secure Multi-Party Computation.
First we expand the construction proposed by Cramer et al. in [2, 4]. Let Γ_1 and Γ_2 be access structures, computed by MSPs $\mathcal{M}_1 = (\mathbb{F}, M_1, \varepsilon_1, \psi_1)$ and $\mathcal{M}_2 = (\mathbb{F}, M_2, \varepsilon_2, \psi_2)$. Let also M_1 be an $m_1 \times d_1$ matrix, M_2 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 $\overline{m} = \sum_i |\varphi_1(i)||\varphi_2(i)|$ entries (notice that $\overline{m} < m_1 m_2$). 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 tensor ($\otimes$) multiplication of the sub-vectors (sub-matrices) belonging to a fixed player, see (1) and (2).
Following this new model we expand the definition for a multiplicative MSP.

Definition 2.4 *Define MSP $\mathcal{M}$ to be $(\mathbb{F}, M = M_1 \diamond M_2, \varepsilon = \varepsilon_1 \diamond \varepsilon_2, \psi)$, 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 $\overline{m}$-vector $\mathbf{r}$ called a recombination vector, such that for any two secrets s' and s'' and any ρ' and ρ'', it holds that*

$$s's'' = \langle \mathbf{r}, M_1(s',\rho') \diamond M_2(s'',\rho'') \rangle = \langle \mathbf{r}, M((s',\rho') \otimes (s'',\rho'')) \rangle .$$

This means that one can construct a multiplicative resulting MSP that computes the product of the secrets shared by MSPs $\mathcal{M}_1$ and $\mathcal{M}_2$.

Definition 2.5 *Given two MSPs $\mathcal{M}_1$ and $\mathcal{M}_2$, the MSP $\mathcal{M}$ is called their* **strongly multiplicative resulting MSP** *if the access structure Γ computed by $\mathcal{M}$ is such that for any players' subset $A \in \Gamma$, $\mathcal{M}_A$ is the multiplicative resulting MSP of $(\mathcal{M}_1)_A$ and $(\mathcal{M}_2)_A$.*

The last definition means that one can construct a strongly multiplicative resulting MSP, computing the product of the secrets shared by MSPs $\mathcal{M}_1$ and $\mathcal{M}_2$, with some access structure Γ. The difference between the multiplicative resulting MSP and the strongly multiplicative resulting MSP is that in the first one $\Gamma = \{P\}$.

Let Γ_1 and Γ_2 be the access structures computed by the MSPs $\mathcal{M}_1$ and $\mathcal{M}_2$, and such that satisfy the VSS conditions given in [6]. 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 $\mathcal{M}$. Our first goal will be to investigate the properties that the access structure Γ and the MSP $\mathcal{M}$ posses. We will prove in Proposition 3.4 that for the resulting MSP $\mathcal{M}$ we have $\Gamma \subseteq \Gamma_1 \uplus \Gamma_2$. (Notice that Γ may be equal to $\emptyset$.) Note that, if we know the requirements for the access structures Γ_1 and Γ_2 such that the equality in Proposition 3.4 holds, then in fact we will have an efficient solution for the strongly multiplicative case.

Our second main result Theorem 3.5 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 3.5 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. Therefore the construction proposed by Cramer et al. in [4] is not applicable in the strongly multiplicative case.

The use of strongly multiplicative LSSS allows us to think about the MPC as a kind of VSS, since no interaction between the players is needed to compute the product of two secrets. 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 will prove 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.

The adversary is called (Δ_1, Δ_A)-adversary if Δ_1 is his privacy structure and $\Delta_A \subseteq \Delta_1$ is his adversary structure. In our adaptive adversary model we have adversary with two privacy structures Δ_1, Δ_2 and with one adversary structure $\Delta_A \subseteq \Delta_1$, $\Delta_A \subseteq \Delta_2$. Finally, we propose solutions in both information-theoretic and computational models for the strongly multiplicative MPC.

In the information-theoretically secure general MPC model it is sufficient Γ to satisfy the VSS conditions form [6] and Γ to be the strongly multiplicative result of MSPs computing Γ_1 (MSP $\mathcal{M}_1$) and Γ_2 (MSP $\mathcal{M}_2$). Combining these conditions we prove our third main result Theorem 5.1, which gives that sufficient conditions for existence of general perfect information-theoretically secure MPC, secure against $(\Delta_1, \Delta_2, \Delta_A)$-adversary is $(\Gamma_A \uplus \Gamma_A)^{\perp} \subseteq \Gamma \subseteq \Gamma_1 \uplus \Gamma_2$.

In the computational model for secure general MPC we use the algebraic simplification for multiplication protocol, presented in Section 4, and the homomorphic commitments [7, 4] to "reduce" the access structure Γ to any access structure Γ_3, provided the VSS conditions for Γ_3 holds. As

we will prove in our fourth main result Theorem 5.2 here we need weaker conditions for Γ than in the information-theoretic model. In other words, if trapdoor one-way permutation exists, then the sufficient conditions for existence of general perfect secure MPC in the cryptographic scenario, secure against $(\Delta_1, \Delta_2, \Delta_A)$-adversary is $\Gamma_A^\perp \subseteq \Gamma \subseteq \Gamma_1 \uplus \Gamma_2$.
As a consequence of the considered model we obtain a requirement for the access structure Γ i.e. $\Gamma_A^\perp \subseteq \Gamma$, which turns out to be equivalent to the Cramer et al. definition of the strongly multiplicative MSP.

3 Main Results

3.1 The Diamond $\diamond$ Construction

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 row belongs to and that the local computation case is not applicable. In the appendix we give some useful properties of the matrix $M = M_1 \otimes M_2$. To avoid the inherent problem of the construction $\otimes$, we study the *diamond* $\diamond$ construction.
Consider the vector x. Let us collect the coordinates in x, which belong to the player t in a sub-vector x_t or $x = (\bar{x}_1, \ldots, \bar{x}_n)$. Hence $\bar{x}_t \in \mathbb{F}^{|\varphi(t)|}$. Thus we have obviously $\langle x, y\rangle = \langle(\bar{x}_1, \ldots, \bar{x}_n), (\bar{y}_1, \ldots, \bar{y}_n)\rangle = \sum_t \langle \bar{x}_t, \bar{y}_t\rangle$. Also notice that the operation diamond $\diamond$ for vectors could be defined as:

$$x \diamond y = (\bar{x}_1 \otimes \bar{y}_1, \ldots, \bar{x}_n \otimes \bar{y}_n) \, . \tag{1}$$

We define an operation diamond for the matrices and construct a new matrix M as follows. We will denote it by $M = M_1 \diamond M_2$.
For each participant t consider the rows he owns in both matrices. Then for each row $(M_1)_i$ of M_1, such that $\psi_1(i) = t$ and for each row $(M_2)_j$ of M_2, such that $\psi_2(j) = t$, calculate a new row $(M_1)_i \otimes (M_2)_j$ of M, and write $\psi(i, j) = t$. Hence m is defined as $m = \sum_{t\in P} |\varphi_1(t)||\varphi_2(t)|$, and M is an $m \times d_1 d_2$ matrix.
Remarks on the Construction: We assume, without restriction for the MSP, that its rows are ordered as follows: first we have $|\varphi(1)|$ rows that belong to the player 1, next $|\varphi(2)|$ rows belonging to the player 2, etc. Then the construction shows that each row $(M_1)_i$ of M_1, such that $\psi_1(i) = t$ is tensor multiplied to each row $(M_2)_j$ of M_2, such that $\psi_2(j) = t$. In other words for any sub-matrix, which belongs to a fixed player we apply the construction $\otimes$.
On the other hand for the columns in M we have the following result: the first column of M_1 is $\diamond$ multiplied to each column of M_2, next the second column of M_1 is $\diamond$ multiplied to each column of M_2, and so on. Thus the process is analogous to the case of $\otimes$ construction, with the difference that the operation $\otimes$ is replaced by $\diamond$.
To make the explanations clearer let us denote by $(M_1)_\mathbf{t}$ the matrix formed by rows of M_1 owned by player t and correspondingly by $(M_2)_\mathbf{t}$ the matrix formed by rows of M_2 owned by player t. Then $(M_1)_\mathbf{t}$ is a $|\varphi_1(t)| \times d_1$ matrix and $(M_2)_\mathbf{t}$ is a $|\varphi_2(t)| \times d_2$ matrix. Hence we can present M_1 as a concatenation of the matrices $(M_1)_\mathbf{t}$ for $t = 1, \ldots, n$ and analogously we can present M_2 as a concatenation of the matrices $(M_2)_\mathbf{t}$ for $t = 1, \ldots, n$. Now from the construction diamond $\diamond$ follows that the matrix $M = M_1 \diamond M_2$ is the concatenation of matrices $(M_1)_\mathbf{t} \otimes (M_2)_\mathbf{t}$ for

$t = 1, \ldots, n$. i.e.,

$$M_1 = \begin{pmatrix} (M_1)_{\mathbf{1}} \\ \ldots \\ (M_1)_{\mathbf{n}} \end{pmatrix}, \quad M_2 = \begin{pmatrix} (M_2)_{\mathbf{1}} \\ \ldots \\ (M_2)_{\mathbf{n}} \end{pmatrix}, \text{ and } \ M = \begin{pmatrix} (M_1)_{\mathbf{1}} \otimes (M_2)_{\mathbf{1}} \\ \ldots \\ (M_1)_{\mathbf{n}} \otimes (M_2)_{\mathbf{n}} \end{pmatrix}. \tag{2}$$

3.2 Properties of the Diamond $\diamond$ Construction

We present some useful properties of the new construction diamond $\diamond$ as well as some properties of the Kronecker product in the Appendix. Here, first we show that the construction is symmetric regarding to the MSPs $\mathcal{M}_1$ and $\mathcal{M}_2$.

Lemma 3.1 *The MSPs $\mathcal{M} = \mathcal{M}_1 \diamond \mathcal{M}_2$ and $\widetilde{\mathcal{M}} = \mathcal{M}_2 \diamond \mathcal{M}_1$ actually compute the same access structure Γ.*

Lemma 3.2 *Let M_1 be an $m_1 \times d_1$ matrix, and M_2 be an $m_2 \times d_2$ matrix. Construct the matrix M following the construction $\diamond$ (i.e., $M = M_1 \diamond M_2$ 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*

$$M(\lambda_1 \otimes \lambda_2) = (M_1 \diamond M_2)(\lambda_1 \otimes \lambda_2) = (M_1\lambda_1) \diamond (M_2\lambda_2)\,.$$

Note that the construction diamond $\diamond$ and Lemma 3.2 confirm our intuitive expectations, as shown in the following lemma.

Lemma 3.3 *Let us denote by $S_1 = M_1(s_1, a)$ and $S_2 = M_2(s_2, b)$ the shares distributed by MSPs $\mathcal{M}_1$ and $\mathcal{M}_2$, for the secrets s_1 and s_2 respectively. Then MSP $\mathcal{M}$ actually distributes shares $S = S_1 \diamond S_2$ for the secret $s_1 s_2$.*

Note that we have $S = (M_1 \diamond M_2)((s_1, a) \otimes (s_2, b))$ and that the vector $(s_1, a) \otimes (s_2, b)$ is not a random any more.
Now we are in position to prove our first main proposition.

Proposition 3.4 *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 equal to $\emptyset$.)*

Proof: Let $A_1 \notin \Gamma_1$. Hence there exists a vector $k \in Ker((M_1)_{A_1})$ such that $k_1 = 1$. Analogously, let $A_2 \notin \Gamma_2$. Hence there exists a vector $r \in Ker((M_2)_{A_2})$ such that $r_1 = 1$. Notice that $k \in \mathbb{F}^{d_1}$ and $r \in \mathbb{F}^{d_2}$. Let $A = A_1 \cup A_2$, so we have $A \notin \Gamma_1 \uplus \Gamma_2$. Form a new vector $k \otimes r \in \mathbb{F}^{d_1 d_2}$. Now using Lemma 6.5 it follows that the vector $k \otimes r \in Ker(M_A)$ and $(k \otimes r)_1 = 1$. Hence $A \notin \Gamma$, thus $\Gamma \subseteq \Gamma_1 \uplus \Gamma_2$. $\square$

3.3 Properties of the Resulting MSP

An interesting open question is when the "equality" holds? One can see from the examples given in the appendix that "equality" does not always hold.
Note that $\lambda = \lambda_1 \diamond \lambda_2$ may not be the recombination vector for $M = M_1 \diamond M_2$. For each $B \in \Gamma_1 \uplus \Gamma_2$ we have that $B \in \Gamma_1$ and $B \in \Gamma_2$, hence there exist recombination vectors λ_1 and λ_2 such that $M_1^T \lambda_1 = \sum_{t=1}^n (M_1)_{\mathbf{t}}^T (\bar{\lambda}_1)_{\mathbf{t}} = \varepsilon_1$ and $M_2^T \lambda_2 = \sum_{t=1}^n (M_2)_{\mathbf{t}}^T (\bar{\lambda}_2)_{\mathbf{t}} = \varepsilon_2$.

On the other hand we have $\varepsilon_1 \diamond \varepsilon_2 = \varepsilon$ and each column in M is equal to a column of M_1 $\diamond$ a column of M_2. Unfortunately λ may not satisfy the condition (applying Lemma 6.11) $M^T\lambda = M^T(\lambda_1 \diamond \lambda_2) = \sum_{t=1}^{n}((M_1)_{\mathbf{t}}^T(\bar{\lambda}_1)_{\mathbf{t}}) \otimes ((M_2)_{\mathbf{t}}^T(\bar{\lambda}_2)_{\mathbf{t}}) = \varepsilon$.
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 $w = (1, \alpha_t, \ldots, \alpha_t^{s-1})$ and $v = (1, \alpha_t, \ldots, \alpha_t^{l-1})$ from MSPs computing $T_{s,n}$ and $T_{l,n}$ correspondingly. Thus the construction proposed above gives

$$v \otimes w = (1, \alpha_t, \ldots, \alpha_t^{s-1}, \alpha_t, \alpha_t^2, \ldots, \alpha_t^s, \ldots\ldots\ldots, \alpha_t^{l-1}, \ldots, \alpha_t^{s+l-2}) .$$

It is well known that the number of columns (here $d = sl - 1$) can be increased without changing the access structure computed by an MSP. The space generated by the 2nd up to the d-th column of M does not contain even a non-zero multiple of the first column. Without changing the access structure that is computed, we can always replace the 2nd up to the d-th column of M by any set of vectors that generates the same space.
Hence $v \otimes w$ is equivalent to $(1, \alpha_t, \ldots, \alpha_t^{s+l-2})$, which 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 3.4. This example shows something more: it is very important to choose the MSPs $\mathcal{M}_1$ and $\mathcal{M}_2$ correctly.
Let the player t holds vectors $w = (1, \alpha_t, \ldots, \alpha_t^{s-1})$ and $v = (1, \beta_t, \ldots, \beta_t^{l-1})$ from MSPs computing $T_{s,n}$ and $T_{l,n}$, and $\alpha_t \neq \beta_t$. Let also MSP $\mathcal{M} = \mathcal{M}_1 \diamond \mathcal{M}_2$ computes Γ. Since $\alpha_t \neq \beta_t$ it is easy to check that Γ is not $T_{l+s-1,n}$ as should be expected from the example above.
Actually the importance of the choice of the MSPs $\mathcal{M}_1$ and $\mathcal{M}_2$ could be illustrated also with the addition of shared secrets. Recall that in the case of addition each player adds up the shares he holds. It means that we use the same MSP (i.e., $\mathcal{M}_1 = \mathcal{M}_2$) to share two secrets the sum of which we want to calculate. Now if we take $\mathcal{M}_1 \neq \mathcal{M}_2$ and share two secrets by $\mathcal{M}_1$ and $\mathcal{M}_2$ simple additions of the shares each player holds are not enough. This observation leads us to the conclusion that (may be) for an MSP $\mathcal{M}_1$ there exists 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 [4, Theorem 7], where $\mathcal{M}_1$ and $\mathcal{M}_2$ are dual, i.e., $\Gamma_2^\perp = \Gamma_1$ and in this case we have $\psi_1 = \psi_2$, $\varepsilon_1 = \varepsilon_2$ and $\varphi_1 = \varphi_2$. Cramer et al. proved in [4, Theorem 7] that $\varepsilon = \varepsilon_1 \diamond \varepsilon_1$ belongs to the linear span of the rows of $M = M_1 \diamond M_1^\perp$, when the matrices M_1 and $M_1^\perp$ satisfy the condition $M_1^T M_1^\perp = \overline{E}$. Here $\overline{E}$ is the matrix that is zero everywhere, except in its upper-left corner where the entry is 1. It is known how to derive the matrix $M_1^\perp$ from matrix M_1 such that they satisfy the equation above.
We are ready to prove our second main result.

Theorem 3.5 *Let Γ_1 and $\Gamma_1^\perp$ be the connected access structures computed by the MSPs $\mathcal{M}_1$ and $\mathcal{M}_1^\perp$. 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 3.4 we have $\Gamma \subseteq \Gamma_1 \uplus \Gamma_1^\perp$, thus it is sufficient to prove that $\Gamma_1 \uplus \Gamma_1^\perp \subseteq \{P\}$.
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 which proves the statement of the theorem. $\square$

As 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 3.5.

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

Now it is easy to describe an analog of the algebraic simplification protocol by Gennaro et al. in [7]. From Lemma 3.3 we have $S_1 = M_1(s_1, a)$ and $S_2 = M_2(s_2, b)$ so $S = S_1 \diamond S_2 = (M_1 \diamond M_2)((s_1, a) \otimes (s_2, b)) = M(s_1 s_2, \rho)$. For any set $A \in \Gamma$ there exists a recombination vector λ such that $M_A^T\ \lambda = \varepsilon$ or in other words $\langle \lambda, S_A \rangle = s_1 s_2$, where as usual $S_A = M_A(s_1 s_2, \rho)$.

Let us choose a new access structure Γ_3 with MSP $\mathcal{M}_3$ (it is possible for example $\Gamma_1 = \Gamma_2 = \Gamma_3$) and vectors $h(i)$ for $i = 1, \dots, m$ such that the first coordinate of $h(i)$ is S_i, i.e., $\langle h(i), \widetilde{\varepsilon} \rangle = S_i$. We use vectors $h(i)$ to re-share the shares S_i. Denote by H the matrix consisting of columns $h(i)$. It is easy to see that $\langle H_A \lambda, \widetilde{\varepsilon} \rangle = s_1 s_2$, since

$$\langle H_A \lambda, \widetilde{\varepsilon} \rangle = \sum_{i \in A} \lambda_i\ \langle h(i), \widetilde{\varepsilon} \rangle = \sum_{i \in A} \lambda_i\ S_i = \langle \lambda, S_A \rangle = s_1 s_2\,.$$

Re-sharing the vectors $h(i)$ with $\mathcal{M}_3$ we have $M_3 h(i) = TS(i)$ which are temporary shares for the secret S_i. Note that for any $B \in \Gamma_3$ there exists a recombination vector $\widetilde{\lambda}$ such that $(M_3)_B^T \widetilde{\lambda} = \widetilde{\varepsilon}$ or in other words $\langle \widetilde{\lambda}, TS(i)_B \rangle = S_i$, where as usual $TS(i)_B = (M_3)_B h(i)$. Let the matrix G consists of columns $TS(i)$, hence $G = M_3 H$. Notice that this matrix corresponds to the temporary shares of all $h(i)$'s. And finally denote by $NS = G\ \lambda = \sum \lambda_j\ TS(j)$.

Note that $NS_j = G_j\ \lambda = G_{j,A}\ \lambda_A$ is the new share of the player j to the secret $s_1 s_2$ distributed by MSP $\mathcal{M}_3$ and it is obtained using only the temporary shares of the players from $A \in \Gamma$. Indeed for $j \in B$ we have $NS_B = G_{B,A} \lambda_A$ and

$$\begin{aligned} \langle NS_B,\ \widetilde{\lambda} \rangle &= \langle G_{B,A}\ \lambda_A, \widetilde{\lambda} \rangle = \sum_{i \in A} \langle TS(i)_B\ \lambda_i,\ \widetilde{\lambda} \rangle = \sum_{i \in A} \lambda_i \langle (M_3)_B h(i),\ \widetilde{\lambda} \rangle \\ &= \sum_{i \in A} \lambda_i \langle h(i), (M_3)_B^T\ \widetilde{\lambda} \rangle = \sum_{i \in A} \lambda_i \langle h(i),\ \widetilde{\varepsilon} \rangle = \langle H_A\ \lambda,\ \widetilde{\varepsilon} \rangle = s_1 s_2\,. \end{aligned}$$

Thus the simplified multiplication protocol is as follows:

1) Each player i multiply locally his shares (for simplicity let they own one share from each of the access structures) $(S_1)_i$ and $(S_2)_i$.

2) The player i chooses a random vector $h(i)$ such that its first coordinate is the product, (i.e., $(S_1)_i (S_2)_i = S_i$.)

3) Using the vector $h(i)$ and $\mathcal{M}_3$ he re-shares (using VSS) the product.

4) Every player k receives from player i a temporary share $TS(i)_k$.

5) For some set of "good" players $A \in \Gamma$ with recombination vector λ_A, each player k calculates his new-share NS_k as $NS_k = \sum_{i \in A} TS(i)_k\ \lambda_i$.

6) The new-shares have the property that any set of "good" players $B \in \Gamma_3$ could restore the secret $s_1 s_2$.

5 Adaptive Adversary: The Zero-Error Case

Recall that in our adaptive adversary model we have adversary with two privacy structures Δ_1, Δ_2 and with one adversary structure $\Delta_A \subseteq \Delta_1$, $\Delta_A \subseteq \Delta_2$. For commitments based on MSPs one can construct error-free commitment protocol, provided that the MSP we have is strongly multiplicative.
In order to build a MPC protocol secure against active adversary in the non-computational model it is sufficient for the MSPs $\mathcal{M}_1$, $\mathcal{M}_2$ and $\mathcal{M}$ to satisfy the VSS conditions from [6] and Γ to be the strongly multiplicative result of MSPs computing Γ_1 and Γ_2. Combining Proposition 3.4 and the VSS conditions of Fehr and Maurer our third main result follows.

Theorem 5.1 *The sufficient conditions for existence of general perfect information-theoretically secure MPC, secure against* $(\Delta_1, \Delta_2, \Delta_A)$*-adversary are*

$$(\Gamma_A \uplus \Gamma_A)^{\perp} \subseteq \Gamma \subseteq \Gamma_1 \uplus \Gamma_2,$$

where Γ *is the access structure computed by the strongly multiplicative resulting MSP* $\mathcal{M} = \mathcal{M}_1 \diamond \mathcal{M}_2$*.*

Note that from Theorem 5.1 it follows that we have $P \notin \Delta_1 \uplus \Delta_2 \uplus \Delta_A \uplus \Delta_A$, which is weaker condition than the condition of Maurer [9].
In order to build a MPC protocol secure against 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 that Γ be the strongly multiplicative result of MSPs computing Γ_1 and Γ_2. Note that we do not need anymore $\mathcal{M}$ to satisfy the VSS conditions, since the algebraic simplification for multiplication protocol presented in the previous section and the homomorphic commitments [7, 4] allow us to detect cheaters and to "reduce" the access structure Γ to any access structure Γ_3, which we will call "reduced". Hence we obtain our fourth main result.

Theorem 5.2 *If trapdoor one-way permutation exists, then the sufficient conditions for existence of general perfect secure MPC in the cryptographic scenario, secure against* $(\Delta_1, \Delta_2, \Delta_A)$*-adversary are*

$$\Gamma_A^{\perp} \subseteq \Gamma \subseteq \Gamma_1 \uplus \Gamma_2, \quad \Gamma_A^{\perp} \subseteq \Gamma_3,$$

where Γ *is the access structure computed by the strongly multiplicative resulting MSP* $\mathcal{M} = \mathcal{M}_1 \diamond \mathcal{M}_2$ *and* Γ_3 *is the "reduced" access structure.*

References

[1] M. Ben-Or, S. Goldwasser and A. Wigderson, Completeness theorems for Non- Cryptographic Fault-Tolerant Distributed Computation, *ACM STOC 1988*, 1988, pp. 1-10.

[2] R. Cramer, Introduction to Secure Computation, *Lectures on Data Security - Modern Cryptology in Theory and Practice*, *LNCS* 1561, 1999, pp. 16-62.

[3] D. Chaum, C. Crepeau and I. Damgard, Multi-Party Unconditionally Secure Protocols, *Proc. ACM STOC 1988*, 1988, pp. 11-19.

[4] R. Cramer, I. Damgard and U. Maurer, General Secure Multi-Party Computation from any linear secret sharing scheme, *EUROCRYPT 2000*, LNCS, Springer-Verlag, vol. 1807, pp. 316-334.

[5] B. Chor, S. Goldwasser, S. Micali and B. Awerbuch, Verifiable secret sharing and achieving simultaneity in the presence of faults, *Proc. of the IEEE 26th Annual Symp. on Foundations of Computer Science*, 1985, pp. 383-395.

[6] S. Fehr, U. Maurer, Linear VSS and Distributed Commitments Based on Secret Sharing and Pairwise Checks, *Proc. CRYPTO 2002*, Springer Verlag 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, *ACM PODC'98*, 1998.

[8] O. Goldreich, S. Micali and A. Wigderson, How to Play Any Mental Game or a Completeness Theorem for Protocols with Honest Majority, *ACM STOC'87*, 1987, pp. 218-229.

[9] U. Maurer, Secure Multi-Party Computation Made Simple, *3rd Conference on Security in Communication Networks*, September 12-13, 2002, Amalfi, Italy, to appear in LNCS, Springer-Verlag, 2002.

[10] 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.

6 Appendix

6.1 Notation

For an arbitrary matrix M over $\mathbb{F}$, with m rows labelled by $1, \ldots, m$ let M_A denote the matrix obtained by keeping only those rows i with $i \in A$, where A is an arbitrary non-empty subset of $\{1, \ldots, m\}$. If $\{i\} = A$ we write M_i. 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 $v = (v_1, \ldots, v_{t_1}) \in \mathbb{F}^{t_1}$ and $w = (w_1, \ldots, w_{t_2}) \in \mathbb{F}^{t_2}$ be two vectors. The tensor vector product $v \otimes w$ is defined as a vector in $\mathbb{F}^{t_1 t_2}$ such that the j-coordinate in v (denoted by v_j) is replaced by $v_j w$, i.e., $v \otimes w = (v_1 w, \ldots, v_{t_1} w) \in \mathbb{F}^{t_1 t_2}$. Define $v \otimes M$ to be the matrix with columns $v \otimes$ k-th column of M, for $k = 1, \ldots, d$. Analogously define $M \otimes v$ to be the matrix with columns k-th column of M $\otimes v$, for $k = 1, \ldots, d$.

Definition 6.1 *The dual $\Gamma^{\perp}$ of a monotone access structure Γ defined on P is the collection of sets $A \subseteq P$ such that $A^c \notin \Gamma$.*

Definition 6.2 *For an access structure* (Γ, Δ) *core*Γ *is defined to be the set of players which are in some minimal authorized set, that is*

$$core\Gamma = \cup_{A \in [\Gamma]^-} A\,.$$

Definition 6.3 *An access structure* (Γ, Δ) *is connected if core*$\Gamma = P$*, when* P *is the set of all players.*

6.2 Technical Lemmas

Some useful technical lemmas.

Lemma 6.4 *Let* $w \in \mathbb{F}^d$ *and* $v \in \mathbb{F}^{m_2}$ *be arbitrary column vectors and* M *be a* $m_1 \times d$ *matrix. Then the following equations hold*

$$(M \otimes v)w = (Mw) \otimes v, \quad (v \otimes M)w = v \otimes (Mw).$$

Lemma 6.5 *Let* $x, a \in \mathbb{F}^m$ *and* $y, b \in \mathbb{F}^n$ *are arbitrary vectors, then the following equality holds*

$$\langle x \otimes y, a \otimes b \rangle = \langle x, a \rangle \langle y, b \rangle.$$

Lemma 6.6 *The construction for* $\otimes$ *is symmetric with respect to the rows and columns, i.e.,*

$$(M_1 \otimes M_2)^T = M_1^T \otimes M_2^T\,.$$

Lemma 6.7 *Let* M_1 *be an* $m_1 \times d_1$ *matrix, and* M_2 *be an* $m_2 \times d_2$ *matrix. And let* $M = M_1 \otimes M_2$ *(i.e.,* M *is an* $m_1 m_2 \times d_1 d_2$ *matrix), then for arbitrary column vectors* $\lambda_1 \in \mathbb{F}^{d_1}$ *and* $\lambda_2 \in \mathbb{F}^{d_2}$ *the following equality holds*

$$M(\lambda_1 \otimes \lambda_2) = (M_1 \otimes M_2)(\lambda_1 \otimes \lambda_2) = (M_1 \lambda_1) \otimes (M_2 \lambda_2)\,.$$

Using the Lemma 6.7 it is easy to see that $\varepsilon = \varepsilon_1 \otimes \varepsilon_2$ belongs to the linear span of the rows of M.

Corollary 6.8 *Let* $\lambda_1 \in \mathbb{F}^{m_1}$ *and* $\lambda_2 \in \mathbb{F}^{m_2}$ *be recombination vectors for* M_1 *and* M_2 *(i.e.,* $M_1^T \lambda_1 = \varepsilon_1$ *and* $M_2^T \lambda_2 = \varepsilon_2$*). Then* $\lambda = \lambda_1 \otimes \lambda_2 \in \mathbb{F}^{m_1 m_2}$ *is the recombination vector for* $M = M_1 \otimes M_2$*, i.e., the following equality holds*

$$M^T \lambda = \varepsilon\,.$$

A property analogous to that in Lemma 6.5 for the operation diamond $\diamond$ holds.

Lemma 6.9 *Let* $x, a \in \mathbb{F}^{d_1}$ *and* $y, b \in \mathbb{F}^{d_2}$ *be arbitrary vectors, then the following equality holds.*

$$\langle x \diamond y, a \diamond b \rangle = \sum_t \langle \bar{x}_t, \bar{a}_t \rangle \langle \bar{y}_t, \bar{b}_t \rangle\,.$$

A lemma analogous to Lemma 6.4 immediately follows from the construction diamond $\diamond$.

Lemma 6.10 *Let* $w \in \mathbb{F}^d$ *and* $v \in \mathbb{F}^m$ *be arbitrary column vectors and* M *be an* $m \times d$ *matrix. Then the following equations hold*

$$(M \diamond v)w = (Mw) \diamond v, \quad (v \diamond M)w = v \diamond (Mw).$$

Lemma 6.11 *Let M_1 be an $m_1 \times d_1$ matrix, and M_2 be an $m_2 \times d_2$ matrix. Construct the matrix M as explained above (i.e., $M = M_1 \diamond M_2$ is $m \times d_1 d_2$ matrix), then for arbitrary column vectors $\lambda_1 \in \mathbb{F}^{m_1}$, $\lambda_2 \in \mathbb{F}^{m_2}$ the following equality holds*

$$M^T(\lambda_1 \diamond \lambda_2) = (M_1 \diamond M_2)^T(\lambda_1 \diamond \lambda_2) = \sum_{t=1}^{n}((M_1)_{\mathbf{t}}^T(\bar{\lambda}_1)_{\mathbf{t}}) \otimes ((M_2)_{\mathbf{t}}^T(\bar{\lambda}_2)_{\mathbf{t}}) \,.$$

6.3 Examples

Example 1

Let $\Gamma_1^- = \{13, 14, 23, 24, 34\}$ and $\mathbb{F} = GF(2)$. It is easy to check that $(\Gamma_1 \uplus \Gamma_1)^- = \{234, 134\}$. On the other hand for the access structure Γ computed by the MSP $M_1 \diamond M_1$ we have $\Gamma = \Gamma_1 \uplus \Gamma_1$. (sum 3th, 5th, 8th and 9th row with the first or the second row).

$$M_1 = \left(\begin{array}{ccc} 0 & 1 & 1 \\ \hline 0 & 1 & 1 \\ \hline 1 & 1 & 0 \\ 0 & 0 & 1 \\ \hline 1 & 1 & 1 \\ 0 & 1 & 0 \end{array}\right) \qquad M_1 \diamond M_1 = \left(\begin{array}{ccc@{\qquad}ccc@{\qquad}ccc} 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 \\ \hline 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 \\ \hline 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ \hline 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{array}\right).$$

Example 2

Let $\Gamma_2^- = \{12, 14, 23, 24, 34\}$ and $\mathbb{F} = GF(2)$. It is easy to check that $(\Gamma_1 \uplus \Gamma_2)^- = \{234\}$. On the other hand for the access structure Γ computed by the MSP $M_1 \diamond M_2$ we have $\Gamma = \{P\} \subset \Gamma_1 \uplus \Gamma_2$ (sum all rows except last three ones, for the set $\{P\}$). For the set $\{234\}$ there is a vector $k = (110|101|011) \in Ker(M_1 \diamond M_2)$, i.e., the set $\{234\} \notin \Gamma$.

$$M_2 = \left(\begin{array}{ccc} 0 & 1 & 1 \\ \hline 1 & 1 & 0 \\ 0 & 0 & 1 \\ \hline 0 & 1 & 1 \\ \hline 1 & 1 & 1 \\ 0 & 1 & 0 \end{array}\right) \qquad M_1 \diamond M_2 = \left(\begin{array}{ccc@{\qquad}ccc@{\qquad}ccc} 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 \\ \hline 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ \hline 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ \hline 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{array}\right).$$