# Secure Multi-Party Computation from any Linear Secret Sharing Scheme

Ventzislav Nikov[1], Svetla Nikova[2], and Bart Preneel[2]

[1] Department of Mathematics and Computing Science
Eindhoven University of Technology
P.O. Box 513, 5600 MB, Eindhoven, the Netherlands
vnikov@mail.com
[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 present a general treatment of non-cryptographic (i.e. information-theoretically secure) multi-party computation, based on underlying linear secret sharing scheme. This general approach gives pure linear-algebra conditions on the linear mappings describing the scheme. The approach establishing the minimal conditions for security, can lead to design of more efficient Multi-Party Computation (MPC) schemes for general adversary structures. Our first goal is to study the Monotone Span Programs (MSP), which is the result of local multiplication of shares distributed by two given MSPs as well as the access structure that this result MSP computes. Second, we expand the definition for multiplicative MSP from [4] and prove that when we use dual MSPs only all players together can compute the product. The knowledge of the result MSP and the access structure it computes allows us to build an analog of the Genaro et al. algebraic simplification protocol [9]. Using this fact and the homomorphic commitments an efficient general MPC protocol in the computational model for general adversary structures can be build, as described in [9, 4].

**Keywords**: general multi-party computation, verifiable secret sharing, linear secret sharing, monotone span programs, information-theoretic security, general adversaries.

## 1 Introduction

The concept of *secret sharing* was introduced by Shamir [17] 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*, and the groups who should not be able to obtain any information about the secret *forbidden*. The collection of all qualified groups is denoted by $\Gamma$, and the collection of all forbidden groups is denoted by $\Delta$. In fact $\Gamma$ is *monotone increasing* and $\Delta$ is *monotone decreasing*. The tuple $(\Gamma, \Delta)$ is called *access structure* if $\Gamma \cap \Delta = \emptyset$. Denote by $P$ the set of participants in the scheme. If $\Gamma \cup \Delta = 2^P$, i.e. $\Gamma = \Delta^c$ is complement of $\Delta$, then we say that $(\Gamma, \Delta)$ is *complete* and we denote it only by $\Gamma$. Otherwise we say that $(\Gamma, \Delta)$ is *incomplete*. By $\Gamma^-$ we denote the collection of *minimal sets* of $\Gamma$ and by $\Delta^+$ we denote the collection of *maximal sets* of $\Delta$. It is obvious that $(\Gamma^-, \Delta^+)$ generates $(\Gamma, \Delta)$. We will consider general monotone access structure $(\Gamma, \Delta)$, which describes subsets of participants that are qualified to recover the secret $s \in \mathbb{F}$ ($\mathbb{F}$ - finite field) in the set of possible secret values.

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 [8, 14] for recent results. Passive corruption means that the adversary obtains the complete information held by the corrupted players, but the players execute the protocol correctly. Active corruption means that the adversary takes full control of the corrupted players. Active corruption is strictly stronger than passive corruption. The adversary is characterized by a *privacy structure* $\Delta$ and an *adversary structure* $\Delta_A \subseteq \Delta$. Denote the complement $\Gamma_A = \Delta_A^c$. In [8, 16] this set is called *honest* (or *good*) players structure, which in fact appears to be wrong notation. Actually its dual access structure $\Gamma_A^\perp$ should be called honest (or good) players structure.

Both passive and active adversary may be *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 $\Delta_A$.

Most proposed SSS are *linear*, but the concept of an LSSS was first considered in its full generality by Karchmer and Wigderson in [12], 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. Note that the size of $\mathcal{M}$ is also the size of the corresponding LSSS. Now we will consider any access structure, as long as it admits a linear secret sharing scheme.

Since an LSSS neither guarantees reconstructability when some shares are false, nor verifiability of a shared value a stronger primitive were introduced *verifiable secret sharing* (VSS) [6, 1]. Secure *multi-party computation* (MPC) can be defined as the problem of $n$ players to compute an agreed function of their inputs in a secure way, where security 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, interesting in its own right, is 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. We will consider the standard *synchronous model* with a *broadcast channel*.

## 2 Preliminaries

### 2.1 Notations

For an arbitrary matrix $M$ over $\mathbb{F}$, with $m$ rows labeled by $1, \ldots, m$ let $M_A$ denotes 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 $M_i$. Consider the set of row-vectors $\mathbf{v}_{i_1}, \ldots, \mathbf{v}_{i_k}$ and let $A = \{i_1, \ldots, i_k\}$ be the set of indices, then we denote by $\mathbf{v}_A$ the matrix consisting of rows $\mathbf{v}_{i_1}, \ldots, \mathbf{v}_{i_k}$. Instead of $\langle \varepsilon, \mathbf{v}_i \rangle$ for $i \in A$ we will write $\langle \varepsilon, \mathbf{v}_A \rangle$. 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 us define the standard inner product $\langle x, y \rangle$ and $x \perp y$, when $\langle x, y \rangle = 0$. For a $\mathbb{F}$-linear subspace $V$ of $\mathbb{F}^t$, $V^\perp$ denotes the collection of elements of $\mathbb{F}^t$, that are orthogonal to all of $V$ (the orthogonal complement), which is again a $\mathbb{F}$-linear subspace. For all subspaces $V$ of $\mathbb{F}^t$ we have $V = (V^\perp)^\perp$, $(Im(M_N^T))^\perp = Ker(M_N)$ or $Im(M_N^T) = (Ker(M_N))^\perp$, $\langle x, M_N^T y \rangle = \langle M_N x, y \rangle$.
Let $v = (v_1, \ldots, v_{t_1}) \in \mathbb{F}^{t_1}$ and $w = (w_1, \ldots, w_{t_2}) \in \mathbb{F}^{t_2}$ are 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. $v \otimes w = (v_1 w, \ldots, v_{t_1} w) \in \mathbb{F}^{t_1 t_2}$; The tensor matrix product $v \bar{\otimes} w$ is defined as a matrix $M \in \mathbb{F}^{t_1 \times t_2}$ with rows $v_1 w, \ldots, v_{t_1} w$ or equivalent with columns $w_1 v, \ldots, w_{t_2} v$.
Denote by $M_i$ the $i$-th row of $M$; by $M_{(i)}$ the $i$-th column of $M$; by $M_{(i,j)}$ the element in the $i$-th row and in the $j$-th column of $M$.
Let $M_{(k)} \in \mathbb{F}^{m_1}$, for $k = 1, \ldots, d$, are columns in $m_1 \times d$ matrix $M$, sometimes we will denote the matrix $M$ by $(M_{(1)}, \ldots, M_{(d)})$ as well. Also let $v \in \mathbb{F}^{m_2}$ be an arbitrary column vector. Define $v \otimes M$ to be the matrix with columns $v \otimes M_{(k)}$, for $k = 1, \ldots, d$. Analogously define $M \otimes v$ to be the matrix with columns $M_{(k)} \otimes v$, for $k = 1, \ldots, d$.

### 2.2 Related Work

**Definition 1.** *[5] The dual $\Gamma^\perp$ of a monotone access structure $\Gamma$ defined on $P$ is the collection of sets $A \subseteq P$ such that $A^c \notin \Gamma$.*

**Definition 2.** *[7] For an access structure $(\Gamma, \Delta)$ core$\Gamma$ 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 3.** *[7] We will say that an access structure is* **connected** *if core$\Gamma$ = $P$, recall that $P$ is the set of all players.*

The following operation (called element-wise union) for monotone decreasing sets was introduced in [16, 8].

**Definition 4.** *[16, 8] We define the operation $\uplus$ for any monotone* **decreasing** *sets $\Delta_1, \Delta_2$ as follows: $\Delta_1 \uplus \Delta_2 = \{A = A_1 \cup A_2; A_1 \in \Delta_1, A_2 \in \Delta_2\}$.*

**Definition 5.** *[16] We define the operation $\uplus$ for any monotone* **increasing** *sets $\Gamma_1, \Gamma_2$ 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 6.** *[4, 2] 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 $\varepsilon$ 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 of rows $(m)$.*

Thus, $\psi$ labels each row with a number from $[1, \ldots, m]$ corresponding to a fixed player, hence we can think of each player as being the "owner" of one or more rows. For every player we consider a function $\varphi$ which gives the set of rows owned by the player, i.e. $\varphi$ is (in some sense) inverse of $\psi$.
MSP is said to compute an access structure $\Gamma$ when $\varepsilon \in Im(M_G^T)$ if and only if $\psi(G)$ is a member of $\Gamma$. So, 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 $\rho$.

**Lemma 1.** *The vector $\varepsilon \notin Im(M_N^T)$ if and only if there exists $\mathbf{k} \in \mathbb{F}^d$ such that $M_N \mathbf{k} = 0$ and $\mathbf{k}_1 = 1$.*

The main goal in our paper is to provide an efficient construction which builds MPC from any LSSS. It is well known that because of the linearity LSSS provide it is easy to add secrets securely. It is enough only for each player to add up the shares he holds.
Therefore, to do general MPC, it will suffice 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 [10, 1, 3, 9] and for general access structure [4, 2].
We follow the approach proposed by Cramer et al. in [4, 2] to build an MPC for any LSSS, provided that the LSSS is what they call *multiplicative*. Loosely speaking, an LSSS is 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 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 a 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 $\Gamma$ be an access structure, computed by MSP $\mathcal{M} = (\mathbb{F}, M, \varepsilon, \psi)$ Given two $m$-vectors $\mathbf{x}$ and $\mathbf{y}$, Cramer et. al. in [4, 2] denote $\mathbf{x} \diamond \mathbf{y}$ to be the vector containing all entries of 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 $\mathbf{x}$, $\mathbf{y}$ 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. So 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 7.** *[4, 2] 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 $\rho'$ and $\rho''$, 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 players subset $A$ that is qualified by $\mathcal{M}$, $\mathcal{M}_A$ is multiplicative.*

### 2.3 Our Results

We focus on general treatment of non-cryptographic (i.e. information-theoretically secure) multi-party computation, based on underlying linear secret sharing scheme. Our research is based mainly on the definitions and results by Cramer et. al. in [4] about General Secure Multi-Party Computation.
First we slightly expand the construction proposed by Cramer et al. in [2, 4]. Let $\Gamma_1$ and $\Gamma_2$ are 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 $m_1 \times d_1$ matrix, $M_2$ be $m_2 \times d_2$ matrix and $\varphi_1$, $\varphi_2$ are the "inverse" functions of $\psi_1$ and $\psi_2$. Given a $m_1$-vector $\mathbf{x}$ and a $m_2$-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 $m < m_1 m_2$. So, if $\mathbf{x}$, $\mathbf{y}$ 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. Correspondingly to this new model we expand the definition for the multiplicative MSP.

**Definition 8.** *Given two MSPs $\mathcal{M}_1$ and $\mathcal{M}_2$, the MSP $\mathcal{M}$ is called their* **multiplicative result 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 $\rho'$ and $\rho''$, it holds that*

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

It means that one can construct a multiplicative result MSP computing the product of the secrets shared by MSPs $\mathcal{M}_1$ and $\mathcal{M}_2$.
Recall that by $\mathcal{M}_A$ we denote the MSP obtained from $\mathcal{M}$ by keeping only the rows owned by players in $A$ for any players subset $A$.

**Definition 9.** *Given two MSPs $\mathcal{M}_1$ and $\mathcal{M}_2$, the MSP $\mathcal{M}$ is called their strongly multiplicative result MSP if there exists an access structure $\Gamma$ computed by $\mathcal{M}$ such that for any players subset $A \in \Gamma$, ($\mathcal{M}_A$ is the multiplicative result MSP of $(\mathcal{M}_1)_A$ and $(\mathcal{M}_2)_A$.*

The last definition means that one can construct a strongly multiplicative result MSP, computing the product of the secrets shared by MSPs $\mathcal{M}_1$ and $\mathcal{M}_2$, with some access structure $\Gamma$. The difference between multiplicative result MSP and strongly multiplicative result MSP is that in the first one $\Gamma = \{P\}$ whereas in the second $\{P\} \neq \Gamma$.
Now let us consider the access structure $\Gamma$ computed by the MSP $\mathcal{M} = (\mathbb{F}, M = M_1 \diamond M_2, \varepsilon = \varepsilon_1 \diamond \varepsilon_2, \psi)$, where $\psi(i, j) = r$ if and only if $\psi_1(i) = \psi_2(j) = r$.
Our first goal will be to investigate the properties that the access structure $\Gamma$ and the MSP $\mathcal{M}$ posses. We will prove that the MSP $\mathcal{M}$ is strongly multiplicative result of MSPs $\mathcal{M}_1$ and $\mathcal{M}_2$.

**Theorem 1.** *Let $\Gamma_1$ and $\Gamma_2$ are the access structures computed by the MSPs $\mathcal{M}_1$ and $\mathcal{M}_2$. Let the MSP $\mathcal{M}$ is the strongly multiplicative result of MSPs $\mathcal{M}_1$ and $\mathcal{M}_2$, and let the access structure $\Gamma$ is computed by the MSP $\mathcal{M}$. Then $\Gamma \subseteq \Gamma_1 \uplus \Gamma_2$. (Notice also that it is possible $\Gamma$ to be $\emptyset$.)*

Now let us consider again the Definition 7, and for $A \in \Gamma_1$ consider the MSP $(\mathcal{M}_1)_A$. Let $\mathcal{M}_1 = \mathcal{M}_2$ are MSPs computing the access structure $\Gamma_1$. Applying Theorem 1 for $\mathcal{M}_1$ it follows that not for any set $A \in \Gamma_1$ the MSP $(\mathcal{M}_1)_A$ is multiplicative, in fact only the sets in $(\Gamma_1 \uplus \Gamma_1) \subset \Gamma_1$ satisfy the definition. Hence the strongly multiplicative property as defined in Definition 7 never holds. That is why it is necessary to give the Definitions 8 and 9.
Our second main theorem shows that the access structure $\Gamma$, computed by the strongly multiplicative result MSP $\mathcal{M}$ of MSPs $\mathcal{M}_1$ and $\mathcal{M}_1^{\perp}$, is in fact the whole set of players $P$.

**Theorem 2.** *Let $\Gamma_1$ and $\Gamma_1^{\perp}$ are the connected access structures computed by the MSPs $\mathcal{M}_1$ and $\mathcal{M}_1^{\perp}$. Let the MSP $\mathcal{M}$ is the strongly multiplicative result of MSPs $\mathcal{M}_1$ and $\mathcal{M}_1^{\perp}$, and let the access structure $\Gamma$ is computed by the MSP $\mathcal{M}$. Then $\Gamma = \Gamma_1 \uplus \Gamma_1^{\perp} = \{P\}$.*

Theorem 2 imply that only all players together can compute the product of the secrets, hence $\mathcal{M}$ is the multiplicative result MSP, but not strongly multiplicative result MSP.
The use of strongly multiplicative LSSS allows us to think about the MPC as a kind of VSS, since no interactions between the players is needed to compute the product of two secrets. Unfortunately in the general case the picture coincide 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 Genaro et al. [9] achieve both tasks in a single step, which they call an algebraic simplification for multiplication protocol. As we will prove in case of general access structure 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 $\Gamma$ and the shares are computed not using a random vector. On the other hand the knowledge of the access structure $\Gamma$ allows us to build an analog of the algebraic simplification protocol of Genaro et al.
Until now the adversary was static, from now on let us consider the adaptive adversary. We will say that the adversary is $(\Delta_1, \Delta_A)$-adversary if $\Delta_1$ 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 $\Delta_1$, $\Delta_2$ and with one adversary structure $\Delta_A \subseteq \Delta_1$, $\Delta_A \subseteq \Delta_2$.
In the considered model of MPC we take into account that it is sufficient to exist a VSS with an access structure $\Gamma$ such that: $\Gamma$ tolerate an adversary structure $\Delta_A$ and $\Gamma$ is strongly multiplicative result of MSPs computing $\Gamma_1$ and $\Gamma_2$. Combining Theorem 1 and Corollary 3 our third main result follows.

**Theorem 3.** *The sufficient condition for existence of general perfect information-theoretically secure MPC secure against $(\Delta_1, \Delta_2, \Delta_A)$-adversary is $(\Gamma_A * \Gamma_A)^{\perp} \subseteq \Gamma \subseteq \Gamma_1 \uplus \Gamma_2$, where $\Gamma$ is the access structure computed by the strongly multiplicative result MSP $\mathcal{M}$ from MSPs $\mathcal{M}_1$ and $\mathcal{M}_2$.*

The paper is organized as follows: In the next section we investigate one natural construction $\otimes$ for the MSP. Then we propose our main construction $\diamond$. In Section 5 an algebraic simplification for multiplication is described. In the last section conditions for existence of MPC secure against adaptive adversary are considered.

## 3 Construction for $\otimes$

In this section we will study the properties of one natural construction for the MSP.
First we prove some useful technical lemmas.

**Lemma 2.** *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 3.** *Let $x, a \in \mathbb{F}^m$ and $y, b \in \mathbb{F}^n$ are arbitrary vectors, then the following equality holds*

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

Now we construct a new matrix $M$ as follows, and denote it by $M = M_1 \otimes M_2$.
**Construction for $\otimes$:** For each row $(M_1)_i$ of $M_1$, and for each row $(M_2)_j$ of $M_2$, compute a new row $(M_1)_i \otimes (M_2)_j$ of $M$.
The construction above, says that the first row of $M_1$ is $\otimes$ multiplied to each row of $M_2$, next the second row of $M_1$ is $\otimes$ multiplied to each row of $M_2$, and so on. But the construction is symmetric, hence we have the same operation for the columns as we applied to the rows. If $(M_1)_{(s)}$, for $s = 1, \ldots, d_1$ are columns in $M_1$ then we can represent the above construction as $M = ((M_1)_{(1)} \otimes M_2 | \ldots | (M_1)_{(d_1)} \otimes M_2)$ using the notations from Lemma 2.
We will give some properties of the matrix $M = M_1 \otimes M_2$.

**Lemma 4.** *The construction for $\otimes$ is symmetric regarding the rows and columns, i.e.*

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

We can generalize the statement in Lemma 2 as follows:

**Lemma 5.** *Let $M_1$ be $m_1 \times d_1$ matrix, and $M_2$ be $m_2 \times d_2$ matrix. And let $M = M_1 \otimes M_2$ (i.e. $M$ is $m_1 m_2 \times d_1 d_2$ 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*

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

Now using the previous lemma it is easy to prove that $\varepsilon = \varepsilon_1 \otimes \varepsilon_2$ belongs to the linear span of the rows of $M$.

**Corollary 1.** *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*

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

Note that the construction $\otimes$ appears to be well the known Kronecker product of matrices see [15]. The problem with this construction is that we do not know whom each row belongs to.

## 4 Construction for $\diamond$

To avoid the inherent problem of the construction $\otimes$, in this section we consider the $\diamond$ construction.
Let us denote for arbitrary vector $x = (\bar{x}_1, \ldots, \bar{x}_n)$, where $\bar{x}_t$ is the sub-vector corresponding to player $t$ (i.e. the coordinates in $x$ which belong to the player $t$

are collected in a sub-vector denoted by $\bar{x}_t$). 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

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

We are now in position to state a property analogous to that in Lemma 3 for the operation $\diamond$.

**Lemma 6.** *Let $x, a \in \mathbb{F}^{d_1}$ and $y, b \in \mathbb{F}^{d_2}$ are arbitrary vectors, then the following equality holds.*

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

Now we construct a new matrix $M$ as it is described bellow. We will denote it by $M = M_1 \diamond M_2$.

**Construction for** $\diamond$ (the main construction): 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 new row $(M_1)_i \otimes (M_2)_j$ of $M$, also denote $\psi(i, j) = t$ in this case. Thus define $m = \sum_{t\in P} |\varphi_1(t)||\varphi_2(t)|$, and $M$ is $m \times d_1 d_2$ 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 belong to the player 2 etc. Then the construction above 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 things 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 $|\varphi_1(t)| \times d_1$ matrix and $(M_2)_{\mathbf{t}}$ is $|\varphi_2(t)| \times d_2$ 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 concatenation of the matrices $(M_2)_{\mathbf{t}}$ for $t = 1, \ldots, n$. Now from the construction $\diamond$ follows that the matrix $M = M_1 \diamond M_2$ is concatenation of matrices $(M_1)_{\mathbf{t}} \otimes (M_2)_{\mathbf{t}}$ for $t = 1, \ldots, n$. i.e.

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

First we will show that the construction is symmetric regarding to the MSPs $\mathcal{M}_1$ and $\mathcal{M}_2$.

**Lemma 7.** *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 $\Gamma$.*

A lemma analogous to Lemma 2 immediately follows from the construction above.

**Lemma 8.** *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*

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

We will present also some useful properties of the new construction.

**Lemma 9.** *Let $M_1$ be $m_1 \times d_1$ matrix, and $M_2$ be $m_2 \times d_2$ 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*

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

**Lemma 10.** *Let $M_1$ be $m_1 \times d_1$ matrix, and $M_2$ be $m_2 \times d_2$ 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*

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

In fact the construction $\diamond$ and Lemma 9 confirm our intuitive expectations, as it is shown in the following lemma.

**Lemma 11.** *Let us denote by $\overline{Share}_1 = M_1(s_1, a)$ and $\overline{Share}_2 = M_2(s_2, b)$ the shares distributed by MSPs $\mathcal{M}_1$ and $\mathcal{M}_2$, for the secrets $s_1$ and $s_2$ correspondingly. Thus MSP $\mathcal{M}$ actually distributes shares $Share = \overline{Share}_1 \diamond \overline{Share}_2$ for secret $s_1 s_2$.*

*Note* that we have $Share = (M_1 \diamond M_2)((s_1, a) \otimes (s_2, b))$ 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 main theorem.
**Proof of Theorem 1:** 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 3 it follows that the vector $k \otimes r \in Ker(M_A)$ and $(k \otimes r)_1 = 1$. Hence $A \notin \Gamma$, thus $\Gamma \subseteq \Gamma_1 \uplus \Gamma_2$. □
Interesting 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 $\lambda_1$ and $\lambda_2$ such that $(M_1)_B^T\lambda_1 = \varepsilon_1$ and $(M_2)_B^T\lambda_2 = \varepsilon_2$. On the other hand we have $\varepsilon_1 \diamond \varepsilon_2 = \varepsilon$ and each column in $M$ is equal to column

of $M_1 \diamond$ column of $M_2$. Unfortunately $\lambda$ may not satisfy the condition $M_B^T \lambda = \varepsilon$. Applying Lemma 10 for $\lambda_1$ and $\lambda_2$ such that

$$M_1^T \lambda_1 = \sum_{t=1}^{n} (M_1)_{\mathbf{t}}^T (\bar{\lambda}_1)_{\mathbf{t}} = \varepsilon_1 \text{ and } \; M_2^T \lambda_2 = \sum_{t=1}^{n} (M_2)_{\mathbf{t}}^T (\bar{\lambda}_2)_{\mathbf{t}} = \varepsilon_2$$

we have

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

Let us consider as 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

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

In [5] the authors prove that the number of columns (here $d = sl - 1$) can be increased without changing the access structure computed by a MSP. The space generated by the 2-nd 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 2-nd 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 player $t$ in MSP computing $T_{l+s-1,n}$. This means that in the threshold case we have equality in Theorem 1. This example shows something more, how important is to choose correctly the MSPs $\mathcal{M}_1$ and $\mathcal{M}_2$.
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 the MSP $\mathcal{M} = \mathcal{M}_1 \diamond \mathcal{M}_2$ computes $\Gamma$. Since $\alpha_t \neq \beta_t$ it is easy to check that $\Gamma$ is not $T_{l+s-1,n}$ as it 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 which sum 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 result MSP $\mathcal{M}$, computing the access structure $\Gamma$, we have $\Gamma = \Gamma_1 \uplus \Gamma_2$.
In fact, the first step in this direction is [4, Theorem 7], where $\mathcal{M}_1$ and $\mathcal{M}_2$ are dual i.e. $\Gamma_2^{\perp} = \Gamma_1$. Notice that 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. Recently in [5] a way of deriving the matrix $M_1^{\perp}$ from matrix $M_1$ were proposed (see Lemma 2) such that they satisfy the equation above.

**Definition 10.** *We will say that an access structure has* **star topology** *for forbidden sets, if there exists a player $i$ such that for any set $A \in [\Delta]^+$, $i \in A$.*

We are ready to prove our second main result.
**Proof of Theorem 2:** It is known that $\{P\} \in \Gamma$ [4]. On the other hand from Theorem 1 we have $\Gamma \subseteq \Gamma_1 \uplus \Gamma_1^{\perp}$, thus it is enough 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 $B = P \setminus B^c \in \Delta_1^+$. 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 contradiction. Hence the access structure $\Gamma_1$ has star topology for the forbidden sets, i.e. $\Gamma_1$ is not connected - contradiction which proves the statement of the theorem. □
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 2.

## 5 Algebraic Simplification for Multiplication Protocol on General Access Structure

Now it is easy to describe an analog of the algebraic simplification protocol by Genaro et al. in [9]. From Lemma 11 we have $\overline{Share}_1 = M_1(s_1, a)$ and $\overline{Share}_2 = M_2(s_2, b)$ so $Share = \overline{Share}_1 \diamond \overline{Share}_2 = (M_1 \diamond M_2)((s_1, a) \otimes (s_2, b)) = M(s_1 s_2, \rho)$. For any set $A \in \Gamma$ there exists a recombination vector $\lambda$ such that $M_A^T \lambda = \varepsilon$ or in other words $\langle \lambda, Share_A \rangle = s_1 s_2$, where as usual $Share_A = M_A(s_1 s_2, \rho)$.
Let us choose new access structure $\Gamma_3$ with MSP $\mathcal{M}_3$ (it is possible for example $\Gamma_1 = \Gamma_2 = \Gamma_3$) and vectors $h(i)$ for $i = 1, \ldots, \overline{m}$ such that the first coordinate of $h(i)$ is the $Share_i$, i.e. $\langle h(i), \widetilde{\varepsilon} \rangle = Share_i$. We will use vectors $h(i)$ to re-share the shares $Share_i$. 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

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

Re-sharing the vectors $h(i)$ with $\mathcal{M}_3$ we have $M_3 h(i) = TShare(i)$ which are temporary shares for the secret $Share_i$. 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 $\langle \widetilde{\lambda}, TShare(i)_B \rangle = Share_i$, where as usual $TShare(i)_B = (M_3)_B h(i)$. Set the matrix $G$ to consist

of columns $TShare(i)$, hence $G = M_3H$. Notice that this matrix corresponds to the temporary shares of all $h(i)$. And finally denote by $NShare = G\ \lambda = \sum \lambda_j\ TShare(j)$.
Note that $NShare_j = G_j\ \lambda = G_{j,A}\ \lambda_A$ is the new share of the player $j$ to the secret $s_1s_2$ 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 $NShare_B = G_{B,A}\lambda_A$ and

$$\langle NShare_B,\ \widetilde{\lambda}\rangle = \langle G_{B,A}\ \lambda_A, \widetilde{\lambda}\rangle = \sum_{i\in A}\langle TShare(i)_B\ \lambda_i,\ \widetilde{\lambda}\rangle = \sum_{i\in A}\ \lambda_i\langle (M_3)_B h(i),\ \widetilde{\lambda}\rangle$$
$$= \sum_{i\in A}\ \lambda_i\langle h(i), (M_3)_B^T\ \widetilde{\lambda}\rangle = \sum_{i\in A}\ \lambda_i\langle h(i),\ \widetilde{\varepsilon}\rangle = \langle H_A\ \lambda,\ \widetilde{\varepsilon}\rangle = s_1s_2$$

Thus if the Player $i$ distribute his shares using $h(\varphi(i))$ and $\mathcal{M}_3$ as described above to $TShare(\varphi(i))$. Then each Player $k$ could combine the temporary shares he receives $TShare(\varphi(A))_k$ from some "good" set of players $A$ with recombination vector $\lambda_{\varphi(A)}$ to calculate his new-share $NShare_k$
as $NShare_k = \sum_{i\in\varphi(A)} TShare(i)_k\ \lambda_i$. Now any qualified group $B \in \Gamma_3$ could restore the secret $s_1s_2$.

**Lemma 12.** *This protocol is a secure multiplication protocol in the presence of passive adversary computationally unbounded.*

## 6 Adaptive Adversary

Until now we have considered schemes only with passive adversary. In this section we will consider presence of adaptive adversary. Since the adversary we can tolerate is at least $Q^2$ adversary (see [11]) 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 [14] proved the following theorem.

**Theorem 4.** *[14] General perfect information-theoretically secure MPC secure against a $(\Delta_1, \Delta_A)$-adversary is possible if and only if $P \notin \Delta_1 \uplus \Delta_1 \uplus \Delta_A$.*

It is easy to rewrite the above theorem into the following form.

**Corollary 2.** *General perfect information-theoretically secure MPC secure against a $(\Delta_1, \Delta_A)$-adversary is possible if and only if $\Gamma_A^\perp \subseteq \Gamma_1 \uplus \Gamma_1$.*

Notice that thanks to the model we use for MPC we reduce the interaction between players, and in this way we may think for the MPC as kind of VSS.
A recent result, which gives necessary and sufficient conditions for existence of VSS was proved in [8].

**Theorem 5.** *[8] The robustness, strong robustness and very strong robustness conditions for VSS are fulfilled if and only if $P \notin \Delta \uplus \Delta_A \uplus \Delta_A$.*

It is easy to rewrite the Fehr and Maurer's result as follows.

**Corollary 3.** *The robustness, strong robustness and very strong robustness conditions for VSS are fulfilled if and only if* $(\Gamma_A \uplus \Gamma_A)^{\perp} \subseteq \Gamma$.

Recall that in our adaptive adversary model we have adversary with two privacy structures $\Delta_1$, $\Delta_2$ and with one adversary structure $\Delta_A \subseteq \Delta_1$, $\Delta_A \subseteq \Delta_2$.
In order to build a MPC protocol secure against active adversary it is sufficient the MSPs $\mathcal{M}_1$, $\mathcal{M}_2$, $\mathcal{M}_3$ to satisfy the VSS conditions and to exist a VSS with an access structure $\Gamma$ such that: $\Gamma$ tolerate an adversary structure $\Delta_A$ and $\Gamma$ is strongly multiplicative result of MSPs computing $\Gamma_1$ and $\Gamma_2$. Combining Theorem 1 and Corollary 3 our third main result follows (Theorem 3). Again the statement in the Theorem 3 can be easily rewritten into the following form.

**Corollary 4.** *General perfect information-theoretically secure MPC secure against* $(\Delta_1, \Delta_2, \Delta_A)$*-adversary is possible if and only if* $P \notin \Delta_1 \uplus \Delta_2 \uplus \Delta_A \uplus \Delta_A$.

## 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. **R. Cramer, S. Fehr**, Optimal Black-Box Secret Sharing over Arbitrary Abelian Groups, *Proc. CRYPTO 2002*, Springer Verlag LNCS 2442, pp. 272-287.
6. **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.
7. **M. van Dijk**, Secret Key Sharing and Secret Key Generation, *Ph.D. Thesis*, 1997, TU Eindhoven.
8. **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.
9. **R. Gennaro, M. Rabin, T. Rabin**, Simplified VSS and Fast-Track Multi-party Computations with Applications to Threshold Cryptography, *ACM PODC'98*, 1998.
10. **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.
11. **M. Hirt, U. Maurer**, Player Simulation and General Adversary Structures in Perfect Multi-party Computation, *J. of Cryptology* 13, 2000, pp. 31-60.

12. **M. Karchmer, A. Wigderson**, On Span Programs, *Proc. of 8-th Annual Structure in Complexity Theory Conference*, San Diego, California, 18-21 May 1993. IEEE Computer Society Press, pp. 102-111.
13. **K. Martin** New secret sharing schemes from old, *J. of Comb. Math. and Combin. Comput.*, 14, 1993, pp. 65-77.
14. **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.
15. **F. J. Mac Williams, N. J. A. Sloane**, The Theory of Error-Correcting Codes, *Elsevier Science*, Amsterdam, 1988.
16. **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.
17. **A. Shamir**, How to share a secret, *Commun. ACM* 22, 1979, pp. 612-613.

# 7 Appendix

**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 $\Gamma$ 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)

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

**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 $\Gamma$ 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 $\{234\} \notin \Gamma$.

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