Multi-party Computation from Any Linear Secret Sharing Scheme Unconditionally Secure against Adaptive Adversary: The Zero-Error Case

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
`v.nikov@tue.nl`
[2] Department Electrical Engineering, ESAT/COSIC,
Katholieke Universiteit Leuven, Kasteelpark Arenberg 10,
B-3001 Heverlee-Leuven, Belgium
`{svetla.nikova,bart.preneel}@esat.kuleuven.ac.be`

Abstract. We consider a generalized adaptive and active adversary model for unconditionally secure Multi-Party Computation (MPC) in the zero error case.

Cramer *et al.* proposed a generic approach to build a *multiplicative* Monotone Span Programs (MSP) – the special property of a Linear Secret Sharing Schemes (LSSS) that is needed to perform a multiplication of shared values. They give an efficient generic construction to build verifiability into every LSSS and to obtain from any LSSS a multiplicative LSSS for the same access structure. But the multiplicative property guarantees security against passive adversary only. For an active adversary a strong multiplicative property is required. Unfortunately there is no known efficient construction to obtain a strongly multiplicative LSSS yet.

Recently Nikov *et al.* have expanded the construction of Cramer *et al.* using a different approach. Multiplying two different MSP M_1 and M_2 computing the access structures Γ_1 and Γ_2 a new MSP M called "resulting" is obtained. M computes a new access structure $\Gamma \subset \Gamma_1$ $(or\, \Gamma_2)$. The goal of this construction is to enable the investigation of how the properties that Γ should fulfil are linked to the initial access structures Γ_1 and Γ_2. It is proved that Γ_2 should be a dual access structure of Γ_1 in order to have a multiplicative resulting MSP. But there are still not known requirements for initial access structures in order to obtain strongly multiplicative resulting MSP. Nikov *et al.* proved that to have unconditionally secure MPC the following minimal conditions for the resulting access structure should be satisfied $(\Gamma_A \uplus \Gamma_A)^{\perp} \subseteq \Gamma$.

In this paper we assume that the resulting MSP could be constructed such that the corresponding access structure Γ satisfies the required

---

properties. Our goal is to study the requirements that Γ should fulfil in order to have an MPC unconditionally secure against adaptive and active adversary in the zero error case. First, we prove that Γ could satisfy weaker conditions than those in Nikov *et al.*, namely $\Gamma_A^\perp \subseteq \Gamma$. Second, we propose a commitment "degree reduction" protocol which allows the players to "reduce" one access structure, e.g. Γ, to another access structure Γ_3. This reduction protocol appears to be a generalization of the reduction protocol of Cramer *et al.* in the sense that we can choose to reduce Γ to the initial access structures Γ_1 or Γ_2, or to a new one Γ_3. This protocol is also more efficient, since it requires less Verifiable Secret Sharing Schemes to be used.

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

1   Introduction

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 the *verifiable secret sharing* (VSS) [6,1]. 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.

In [18] Shamir introduced the concept of *secret sharing* 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. The groups who are allowed to reconstruct the secret are called *qualified*, and the groups who should not be able to obtain any information about the secret are called *forbidden*. The collection of all qualified groups is denoted by Γ, and the collection of all forbidden groups is denoted by Δ. The tuple (Γ, Δ) is called an *access structure* if $\Gamma \cap \Delta = \emptyset$. Denote by $P = \{P_1, \dots, P_n\}$ the set of participants in the scheme and by $\mathcal{P}(P)$ the set of all subsets of P. If $\Gamma \cup \Delta = \mathcal{P}(P)$, i.e., $\Gamma = \Delta^c$ is the complement of Δ, then (Γ, Δ) is *complete* and it is denoted simply by Γ. When Γ is complete the SSS is called perfect.

Usually the cheating is represented as an *adversary* who may corrupt some subset of the players. One can distinguish between *passive* and *active* corruption, see Fehr and Maurer, [8] for recent results. Passive corruption means that the adversary obtains the complete information held by the corrupt players, but the players execute the protocol correctly. Active corruption means that the adversary takes full control of the corrupt players. Active corruption is strictly stronger

than passive corruption. The adversary is characterized by a *privacy structure* Δ and an *adversary structure* $\Delta_A \subseteq \Delta$. Denote the complement $\Gamma_A = \Delta_A^c$ and call its dual access structure $\Gamma_A^\perp$ the *honest* (or *good*) players structure. Both passive and active adversaries may be *static*, meaning that the set of corrupt players is chosen once and for all before the protocol starts, or *adaptive* meaning that the adversary can at any time during the protocol choose to corrupt a new player based on all the information he has at the time, as long as the total set is in Δ_A. 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 in [13], 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. Since an LSSS neither guarantees reconstructability when some shares are incorrect, nor verifiability of a shared value the stronger primitive – Verifiable Secret Sharing has been introduced.

We will consider any complete general monotone 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 it admits a linear secret sharing scheme. We will consider also the standard *synchronous model* with a *broadcast channel*.

1.1   Related Work

This subsection contains some basic definitions, notations and results. For an arbitrary matrix M over $\mathbb{F}$, with m rows labelled by $1, \dots, m$ let M_A denote the matrix obtained by keeping only those rows i with $i \in A$, where A is an arbitrary non-empty subset of $\{1, \dots, m\}$. 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 $v = (v_1, \dots, v_{t_1}) \in \mathbb{F}^{t_1}$ and $w = (w_1, \dots, w_{t_2}) \in \mathbb{F}^{t_2}$ 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., $v \otimes w = (v_1 w, \dots, v_{t_1} w) \in \mathbb{F}^{t_1 t_2}$. The Kronecker product of matrices is defined as tensor vector multiplication of each row from the first matrix to each row from the second matrix.

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

The following operation (called element-wise union) for monotone decreasing (increasing) sets was introduced in [15,8].

Definition 2. *For monotone decreasing sets Δ_1, Δ_2 and for monotone increasing sets Γ_1, Γ_2, all defined for the same set of participants, the* element-wise union *operation $*$ is defined by:*

$$\Delta_1 * \Delta_2 = \{A_1 \cup A_2; A_1 \in \Delta_1, A_2 \in \Delta_2\},$$
$$resp. \qquad \Gamma_1 * \Gamma_2 = \{A_1 \cup A_2; A_1 \notin \Gamma_1, A_2 \notin \Gamma_2\}^c.$$

Throughout the paper we will consider presence of adaptive adversary. Let Q^2, resp. Q^3 be the conditions on an adversary structure that *no two*, resp. *no three* of the sets in the structure cover the full players set P. The adversary that we tolerate is at least a Q^2 (resp. Q^3) adversary in the passive (resp. active) scenario (see [12,4]). Since the condition Q^2 is equivalent to $\Delta_A \cap \Gamma_A^\perp = \emptyset$ (i.e., $\Gamma_A^\perp \subseteq \Gamma_A$), the honest players structure has no intersection with the adversary structure.

Recently Maurer [14] proved that general perfect information-theoretically secure 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 $\Gamma_A^\perp \subseteq \Gamma_1 \uplus \Gamma_1$. Maurer consider the case, when the secrets are shared using only one MSP. 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 information-theoretically secure VSS has been presented by Fehr and Maurer in [8]. They prove that the 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 $(\Gamma_A \uplus \Gamma_A)^\perp \subseteq \Gamma$.

As mentioned earlier, MSPs are essentially equivalent to LSSS's (see e.g. [13]). It turns out to be convenient to describe our protocols in terms of MSPs as we will do for the rest of the paper. A formal definition for an MSP follows.

Definition 3. *[3,4] 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 (in some sense) inverse of ψ.

An MSP is said to compute a (complete) access structure Γ when $\varepsilon \in Im(M_{\varphi(G)}^T)$ if and only if G is a member of Γ. Hence, the players can reconstruct the secret precisely if the rows they own contain in their linear span the target vector of $\mathcal{M}$, and otherwise they get no information about the secret, i.e., there exists a so called *recombination vector* $\mathbf{r}$ such that $\langle \mathbf{r}, M_{\varphi(G)}(s, \rho) \rangle = s$ and $M_{\varphi(G)}^T \mathbf{r} = \varepsilon$ 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. It is well known that because of the linearity the LSSS provides it is easy to add secrets securely. Therefore to achieve general

MPC, it suffices to implement multiplication of shared secrets. That is, we need a protocol where each player initially holds shared secrets s and s', and ends up holding a share of the product ss'. Several such protocols are known for the threshold case [1,2,10,11] and for general access structure [3,4,17].

We follow the approach proposed by Cramer *et al.* in [3,4] to build an MPC from any LSSS, provided that the LSSS is what is called *(strongly) multiplicative*. Loosely speaking, an LSSS is (strongly) multiplicative if each player P_i can compute from his shares (of secrets s and s') a value c_i, such that the product ss' can be obtained using all values (only values from honest players).

In a recent paper by Nikov *et al.* [17] the $\diamond$ construction for multiplying two MSPs has been proposed. 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 be the "inverse" functions of ψ_1 and ψ_2. Consider the vector x. The coordinates in x, which belong to the player t are collected in a sub-vector x_t or $x = (\bar{x}_1, \dots, \bar{x}_n)$. First the operation $\diamond$ for vectors is defined as follows:

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

Denote by $(M_1)_\mathbf{t}$ the matrix formed by rows of M_1 owned by the player t and correspondingly by $(M_2)_\mathbf{t}$ the matrix formed by rows of M_2 owned by the same player. Hence M_1 can be presented as a concatenation of the matrices $(M_1)_\mathbf{t}$ for $t = 1, \dots, n$. Then the operation $\diamond$ for matrices is defined as the concatenation of matrices $(M_1)_\mathbf{t} \otimes (M_2)_\mathbf{t}$ for $t = 1, \dots, n$, i.e.,

$$M = M_1 \diamond M_2 = \begin{pmatrix} (M_1)_\mathbf{1} \otimes (M_2)_\mathbf{1} \\ \dots \\ (M_1)_\mathbf{n} \otimes (M_2)_\mathbf{n} \end{pmatrix}.$$

Finally, the operation $\diamond$ for two MSP could be defined as:

Definition 4. *[17] 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 $\psi_1(i) = \psi_2(j) = r$ and the size of $\mathcal{M}$ is $m = \sum_i |\varphi_1(i)||\varphi_2(i)| = \sum_i |\varphi(i)|$. Given two MSPs $\mathcal{M}_1$ and $\mathcal{M}_2$, the MSP $\mathcal{M}$ is called their* **multiplicative resulting MSP** *and denoted by $\mathcal{M} = \mathcal{M}_1 \diamond \mathcal{M}_2$ if there exists an 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*

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

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 case $\Gamma = \{P\}$.

It has been proved in [17] that $\varGamma \subseteq \varGamma_1 \uplus \varGamma_2$. In the model of MPC proposed in [17] the secrets are shared using VSS and two MSP $\mathcal{M}_1$ and $\mathcal{M}_2$. Hence the adaptive adversary has two privacy structures $\varDelta_1$, $\varDelta_2$ and one adversary structure $\varDelta_A \subseteq \varDelta_1$, $\varDelta_A \subseteq \varDelta_2$. Such an adversary is denoted by $(\varDelta_1, \varDelta_2, \varDelta_A)$-adversary.

In the computational model for MPC the authors in [17] propose the so called "algebraic simplification for multiplication" protocol which uses homomorphic commitments in the strongly multiplicative case of general MPC. In fact, the "algebraic simplification for multiplication" protocol allows the players to "reduce" one access structure $\varGamma$ to another access structure $\varGamma_3$, provided that the VSS conditions for $\varGamma_3$ hold. As it is proved in [17] to build a MPC protocol secure against an adaptive adversary in the computational model it is sufficient the MSPs $\mathcal{M}_1$, $\mathcal{M}_2$, $\mathcal{M}_3$ to satisfy the VSS conditions, i.e., $\varGamma_A^\perp \subseteq \varGamma_i$ for $i = 1, 2, 3$; $\mathcal{M}$ to be resulting MSP of $\mathcal{M}_1$ and $\mathcal{M}_2$, i.e., $\varGamma \subseteq \varGamma_1 \uplus \varGamma_2$ and $\varGamma$ to satisfy the strong multiplicative property, i.e., $\varGamma_A^\perp \subseteq \varGamma$. On the other hand the lack of "algebraic simplification for multiplication" protocol in the information-theoretic scenario impose stronger conditions for the strongly multiplicative case of general MPC. It is proved in [17] that it is sufficient for the MSPs $\mathcal{M}_1$ and $\mathcal{M}_2$ to satisfy the VSS conditions from [8], i.e., $(\varGamma_A \uplus \varGamma_A)^\perp \subseteq \varGamma_i$ for $i = 1, 2$; $\mathcal{M}$ to be resulting MSP of $\mathcal{M}_1$ and $\mathcal{M}_2$, i.e., $\varGamma \subseteq \varGamma_1 \uplus \varGamma_2$ and $\varGamma$ to satisfy the following property,

$$(\varGamma_A \uplus \varGamma_A)^\perp \subseteq \varGamma. \tag{1}$$

1.2   Results of This Paper

The condition (1) is sufficient to multiply securely two secrets, but it is insufficient to perform general MPC, since with each multiplication the access structure $\varGamma$ becomes "smaller" and "smaller". Hence besides multiplying securely we need a "degree reduction" protocol to "reduce" the access structure $\varGamma$ to another access structure e.g. $\varGamma_3$. The solution that we propose is parallel to the one in the threshold case, where after multiplication we have threshold $2t$ and reduce it to threshold t as Ben-Or *et al.* show in [1].

In this paper we build an information-theoretically secure simplification protocol for multiplication, which is an important step in order to be achieved general secure MPC. The main hurdle to overcome in the "degree reduction" protocol is the additional check which ensures the commitment to the re-shared shares. The clue in this additional check is the change of the basis (see Section 3.3). Our main result follows:

Theorem 1. *Suppose that for the MSPs $\mathcal{M}_1$ and $\mathcal{M}_2$ there exist MSPs $\mathcal{M}_3$ and $\mathcal{M}_4$ such that $\mathcal{M}_1 \diamond \mathcal{M}_2 = \mathcal{M} = \mathcal{M}_3 \diamond \mathcal{M}_4$. Then the sufficient condition for existence of general perfect information-theoretically secure MPC secure against $(\varDelta_1, \varDelta_2, \varDelta_A)$-adversary is*

$$\varGamma_A^\perp \subseteq \varGamma \subseteq \varGamma_1 \uplus \varGamma_2, \quad (\varGamma_A \uplus \varGamma_A)^\perp \subseteq \varGamma_i \;\; for \;\; i = 1, 2, 3,$$

where Γ is the access structure computed by the strongly multiplicative resulting MSP $\mathcal{M}$ from MSPs $\mathcal{M}_1$ and $\mathcal{M}_2$ and/or from MSPs $\mathcal{M}_3$ and $\mathcal{M}_4$.

We will call the access structure Γ_3 (the MSP $\mathcal{M}_3$, resp.) "reduced". It is easy to see that such MSPs $\mathcal{M}_3$ and $\mathcal{M}_4$ always exist, e.g. $\mathcal{M}_1 = \mathcal{M}_3$ and $\mathcal{M}_2 = \mathcal{M}_4$. In the threshold case there exist several pairs of MSPs that satisfy the assumption of Theorem 1.

Note also that the Maurer's [14] necessary and sufficient condition $P \notin \Delta_1 \uplus \Delta_1 \uplus \Delta_A$ is satisfied (in case $\Gamma_1 = \Gamma_2$), on the other hand this conditions does not guarantee that $\Gamma_A^\perp \subseteq \Gamma$, when $\Gamma \neq \Gamma_1 \uplus \Gamma_2$, i.e., $\Gamma \subset \Gamma_1 \uplus \Gamma_2$.

The picture in the general access structure appears to be analogous to this in the threshold case [7,9]. Remarkably the conditions in the information-theoretic settings are "similar" to the conditions in the cryptographic settings (see the result of Nikov *et al.* for the computational model). Note that it is not required anymore Γ to satisfy the VSS conditions.

If we compare with the protocol in [4] we can see that now the player who re-shares his share do not need to commit to every single entry in the used vector. Hence the number of the used VSS is reduced. Also note that this protocol does not depend on the model considered here (Nikov *et al.*), it could be applied also for the model of Cramer *et al.*

The paper is organized as follows: In Section 2 the information-theoretically secure VSS, randomization and re-sharing protocols are presented. In Section 3 we introduce some terminology and concepts, we state the results and explain the role they play in comparison with earlier results.

2   Background

2.1   VSS – Share Phase

Let the dealer $\mathcal{D}$ shares the secret s to the players P_i using the VSS protocol, as described by Cramer *et al.* in [4], and let $\mathcal{M}$ be an MSP with matrix M $(m \times d)$.

1. The Dealer $\mathcal{D}$ chooses a symmetric $d \times d$ matrix R subject to s (the secret) in its upper left corner.
2. The Dealer $\mathcal{D}$ gives to the participant P_i shares $v_{\varphi(i)} = M_{\varphi(i)}R$ ($v_{\varphi(i)}$ is $|\varphi(i)| \times d$ matrix), where the "true part" (which will be used in the reconstruction) of the shares is $v_{\varphi(i)}\varepsilon$.
3. The players P_i and P_j perform a pairwise-check as follows:

$$M_{\varphi(j)}v_{\varphi(i)}^T = M_{\varphi(j)}RM_{\varphi(i)}^T = v_{\varphi(j)}M_{\varphi(i)}^T.$$

2.2   VSS – Reconstruction Phase

For any group of players $G \in \Gamma$ there exists a recombination vector $\lambda_{\varphi(G)}$, such that they can reconstruct together the secret s as follows:

$$(v_{\varphi(G)}\varepsilon)\lambda_{\varphi(G)}^T = \langle \lambda_{\varphi(G)}, v_{\varphi(G)}\varepsilon \rangle = \sum_{i \in G} \lambda_{\varphi(i)}(v_{\varphi(i)}\varepsilon) = s.$$

2.3 Information-Theoretic Homomorphic Commitments and Re-share Phase

In the re-share phase each player P_i plays the role of the dealer sharing the true part of his shares among the participants using VSS with the same MSP $\mathcal{M}$.

1. Any player P_i re-shares his true part of the share $v_{\varphi(i)}\varepsilon$, i.e., for any $i_1 \in \varphi(i)$ he chooses a symmetric $d \times d$ matrix $R^{(i_1)}$ such that its first row (column) is v_{i_1} and the value in its upper left corner is $v_{i_1}\varepsilon$.
2. P_i sends to P_j temporary shares $y_{i_1,\varphi(j)} = M_{\varphi(j)}R^{(i_1)}$, whose true part is $y_{i_1,\varphi(j)}\varepsilon$.
3. The players P_k and P_j perform the usual commitment verification (VSS pairwise-check):

$$M_{\varphi(j)}y_{i_1,\varphi(k)}^T = M_{\varphi(j)}R^{(i_1)}M_{\varphi(k)}^T = y_{i_1,\varphi(j)}M_{\varphi(k)}^T.$$

4. In addition P_j checks his true part of the share

$$y_{i_1,\varphi(j)}\varepsilon = M_{\varphi(j)}R^{(i_1)}\varepsilon = M_{\varphi(j)}v_{i_1}^T = v_{\varphi(j)}M_{i_1}^T.$$

   The last equality is the pair-wise check in VSS (step 3 in the Share phase). Note that this additional check ensures that the player P_i really re-shares his share, i.e., he is honest.
5. As usual for any group of players $\widetilde{G} \in \Gamma$ there exists a recombination vector $\widetilde{\lambda}_{\varphi(\widetilde{G})}$ such that they can together reconstruct the true part of the initial share – $v_{i_1}\varepsilon$.

$$(y_{i_1,\varphi(\widetilde{G})}\varepsilon)\widetilde{\lambda}_{\varphi(\widetilde{G})}^T = \langle\widetilde{\lambda}_{\varphi(\widetilde{G})}, y_{i_1,\varphi(\widetilde{G})}\varepsilon\rangle = \sum_{j\in\widetilde{G}}\widetilde{\lambda}_{\varphi(j)}(y_{i_1,\varphi(j)}\varepsilon) = v_{i_1}\varepsilon.$$

6. Denote the list of good players by $\mathcal{L} \in \Gamma$. Then P_j, using the corresponding recombination vector $\lambda_{\varphi(\mathcal{L})}$, computes

$$z_{\varphi(j)} = \sum_{i\in\mathcal{L}}\lambda_{\varphi(i)}y_{\varphi(i),\varphi(j)}.$$

The new shares (of the same secret s) are $z_{\varphi(j)}$ and they satisfy all the necessary properties as follows:

- The pair-wise check holds:

$$M_{\varphi(k)}z_{\varphi(j)}^T = \sum_{i\in\mathcal{L}}\lambda_{\varphi(i)}M_{\varphi(k)}y_{\varphi(i),\varphi(j)}^T$$

$$= (\sum_{i\in\mathcal{L}}\lambda_{\varphi(i)}y_{\varphi(i),\varphi(k)})M_{\varphi(j)}^T = z_{\varphi(k)}M_{\varphi(j)}^T.$$

- The players in any group $\widetilde{G} \in \Gamma$ can reconstruct the secret s together.

$$(z_{\varphi(\widetilde{G})}\varepsilon)\widetilde{\lambda}^T_{\varphi(\widetilde{G})} = \langle \widetilde{\lambda}_{\varphi(\widetilde{G})}, z_{\varphi(\widetilde{G})}\varepsilon \rangle = \sum_{j \in \widetilde{G}} \widetilde{\lambda}_{\varphi(j)}(z_{\varphi(j)}\varepsilon)$$

$$= \sum_{j \in \widetilde{G}} \widetilde{\lambda}_{\varphi(j)}(\sum_{i \in \mathcal{L}} \lambda_{\varphi(i)}(y_{\varphi(i),\varphi(j)}\varepsilon))$$

$$= \sum_{i \in \mathcal{L}} \lambda_{\varphi(i)}(\sum_{j \in \widetilde{G}} \widetilde{\lambda}_{\varphi(j)}(y_{\varphi(i),\varphi(j)}\varepsilon)) = \sum_{i \in \mathcal{L}} \lambda_{\varphi(i)}(v_{\varphi(i)}\varepsilon) = s.$$

2.4   The Randomization Phase

We can use the Renewal phase from [16] as a randomization protocol.

3   Reduction Protocol

3.1   The Set-up

Let Γ_1 and Γ_2 be access structures, computed by MSPs $\mathcal{M}_1 = (\mathbb{F}, M_1, \varepsilon_1, \psi_1)$ and $\mathcal{M}_2 = (\mathbb{F}, M_2, \varepsilon_2, \psi_2)$, respectively. Let also M_1 be $m_1 \times d_1$ matrix, M_2 be $m_2 \times d_2$ matrix and φ_1, φ_2 be the "inverse" functions of ψ_1 and ψ_2.

Let $\mathcal{M} = \mathcal{M}_1 \diamond \mathcal{M}_2$ be the multiplicative resulting MSP, i.e., $\mathcal{M} = (\mathbb{F}, M = M_1 \diamond M_2, \varepsilon = \varepsilon_1 \diamond \varepsilon_2, \psi)$, where $\psi(i, j) = r$ if and only if $\psi_1(i) = \psi_2(j) = r$. Hence M is $m \times d_1 d_2$ matrix, where $m = \sum_i |\varphi_1(i)||\varphi_2(i)| = \sum_i |\varphi(i)|$. Let us consider the access structure Γ computed by the MSP $\mathcal{M}$.

Let the first secret s_1 is shared using VSS by MSP $\mathcal{M}_1$ with symmetric $d_1 \times d_1$ matrix $R^{(1)}$, i.e., $v_{\varphi_1(i)} = (M_1)_{\varphi_1(i)} R^{(1)}$ be the shares of P_i ($v_{\varphi_1(i)}$ is $|\varphi_1(i)| \times d_1$ matrix). The "true part" of the shares are the first coordinates of each share, i.e., $v_{\varphi_1(i)}\varepsilon_1$.

Analogously, let the second secret s_2 is shared by MSP $\mathcal{M}_2$ with symmetric $d_2 \times d_2$ matrix $R^{(2)}$, i.e., $w_{\varphi_2(i)} = (M_2)_{\varphi_2(i)} R^{(2)}$ be the shares of P_i. ($w_{\varphi_2(i)}$ is $|\varphi_2(i)| \times d_2$ matrix). The "true part" of the shares are the first coordinates of each share, i.e., $w_{\varphi_2(i)}\varepsilon_2$.

3.2   Local Computation Phase

Denote by $R = R^{(1)} \otimes R^{(2)}$ a $d_1 d_2 \times d_1 d_2$ symmetric matrix. Note that the value in the upper left corner of R is the product $s_1 s_2$. Let us choose the indices $i_1 \in \varphi_1(i)$, $i_2 \in \varphi_2(i)$, $j_1 \in \varphi_1(j)$ and $j_2 \in \varphi_2(j)$.

If the player P_i locally computes $\otimes$ product of his shares he obtains his new shares $v_{\varphi_1(i)} \otimes w_{\varphi_2(i)}$ (which are an $|\varphi(i)| \times d_1 d_2$ matrix).

This shares correspond to an MSP $\mathcal{M}$ and the random matrix R as defined above, i.e., $((M_1)_{i_1} \otimes (M_2)_{i_2})R = v_{i_1} \otimes w_{i_2}$.

The pair-wise check for the new shares also holds:

$$((M_1)_{i_1} \otimes (M_2)_{i_2})(v_{j_1} \otimes w_{j_2})^T = ((M_1)_{i_1} v_{j_1}^T)((M_2)_{i_2} w_{j_2}^T) =$$
$$(v_{i_1}(M_1)_{j_1}^T)(w_{i_2}(M_2)_{j_2}^T) = (v_{i_1} \otimes w_{i_2})((M_1)_{j_1} \otimes (M_2)_{j_2})^T.$$

Note that the new "true part" of the shares is the product

$$(v_{\varphi 1(i)} \otimes w_{\varphi 2(i)})\varepsilon = (v_{\varphi 1(i)}\varepsilon_1) \otimes (w_{\varphi 2(i)}\varepsilon_2).$$

In the new MSP $\mathcal{M}$ for any group of players $G \in \Gamma$ there exists a recombination vector $\lambda_{\varphi(G)}$ such that they can reconstruct together the product of the secrets $- s_1 s_2$.

$$((v_{\varphi 1(G)} \otimes w_{\varphi 2(G)})\varepsilon)\lambda_{\varphi(G)}^T = \langle \lambda_{\varphi(G)}, (v_{\varphi 1(G)} \otimes w_{\varphi 2(G)})\varepsilon \rangle$$
$$= \sum_{j \in G} \lambda_{\varphi(j)}((v_{\varphi 1(j)} \otimes w_{\varphi 2(j)})\varepsilon) = s_1 s_2.$$

3.3   Decomposition – Change of the Basis

Let d_3 and d_4 are integers such that $d_1 d_2 = d_3 d_4$ and, as usual, $\varepsilon_3 \in \mathbb{F}^{d_3}$ be the unit column vector. Denote by $e_i = (0, \ldots, 0, 1, 0, \ldots, 0) \in \mathbb{F}^{d_4}$ the unit row vectors, for $i = 1, \ldots, d_4$.

It is easy to see that there exist uniquely defined vectors $x_{j_1,j_2}^{(i)}, \widetilde{x}_{j_1,j_2}^{(i)} \in \mathbb{F}^{d_3}$ for $i = 1, \ldots, d_4$, such that the following equalities hold

$$v_{j_1} \otimes w_{j_2} = \sum_{i=1}^{d_4} x_{j_1,j_2}^{(i)} \otimes e_i; \qquad v_{j_1} \otimes w_{j_2} = \sum_{i=1}^{d_4} e_i \otimes \widetilde{x}_{j_1,j_2}^{(i)}. \qquad (2)$$

Note that $(v_{j_1} \otimes w_{j_2})\varepsilon = x_{j_1,j_2}^{(1)}\varepsilon_3 = \widetilde{x}_{j_1,j_2}^{(1)}\varepsilon_3.$

3.4   Degree Reduction Phase

Let Γ_3 be an access structure, computed by the MSP $\mathcal{M}_3 = (\mathbb{F}, M_3, \varepsilon_3, \psi_3)$. Let also M_3 be $m_3 \times d_3$ matrix and φ_3 be the "inverse" functions of ψ_3.

Any player P_j re-shares the first coordinate of the vector $x_{j_1,j_2}^{(i)}$, i.e., $x_{j_1,j_2}^{(i)}\varepsilon_3$ for $i = 1, \ldots, d_4$ using VSS Share protocol. Let us denote the different copies of VSSs by $VSS(i)$. For each VSS the player uses a symmetric $d_3 \times d_3$ matrix $R_{j_1,j_2}^{(i)}$, such that its first row (column) is $x_{j_1,j_2}^{(i)}$. So, the player P_k receives from P_j the following temporary shares:

$$y_{j_1,j_2,\varphi_3(k)}^{(i)} = (M_3)_{\varphi_3(k)} R_{j_1,j_2}^{(i)}$$

As in Subsection 2.3 the player P_k verifies the commitments of P_j using usual pair-wise check for each $VSS(i)$.

3.5   Additional Check on the Degree Reduction Phase

Now we need to ensure that the player P_j re-shares the correct vectors $x^{(i)}_{j_1,j_2}$ and in particular their true part. Unfortunately we can not apply directly the additional check procedure from step 4. in the re-share protocol, because in the degree reduction phase we use two different access structures.

Let us choose the indices $j_1 \in \varphi_1(j)$, $j_2 \in \varphi_2(j)$, $k_1 \in \varphi_1(k)$, $k_2 \in \varphi_2(k)$, $k_3 \in \varphi_3(k)$ and $k_4 \in \varphi_4(k)$. In order to perform this additional check we assume that there exist matrices M_3 and M_4, such that $M_1 \diamond M_2 = M = M_3 \diamond M_4$. This assumption means that we have $(M_3)_{k_3} \otimes (M_4)_{k_4} = (M_1)_{k_1} \otimes (M_2)_{k_2}$ for some rows k_1, k_2, k_3, k_4 of the corresponding matrices.

We first prove the following three equalities.

$$\langle y^{(i)}_{j_1,j_2,k_3}, \varepsilon_3^T \rangle = \langle (M_3)_{k_3} R^{(i)}_{j_1,j_2}, \varepsilon_3^T \rangle \tag{3}$$
$$= \langle (M_3)_{k_3}, (R^{(i)}_{j_1,j_2})_1 \rangle = \langle (M_3)_{k_3}, x^{(i)}_{j_1,j_2} \rangle,$$

$$\langle (M_3)_{k_3} \otimes (M_4)_{k_4}, x^{(i)}_{j_1,j_2} \otimes e_i \rangle = \langle (M_3)_{k_3}, x^{(i)}_{j_1,j_2} \rangle \langle (M_4)_{k_4}, e_i \rangle, \tag{4}$$

$$\langle (M_1)_{k_1} \otimes (M_2)_{k_2}, v_{j_1} \otimes w_{j_2} \rangle = \langle (M_1)_{k_1}, v_{j_1} \rangle \langle (M_2)_{k_2}, w_{j_2} \rangle \tag{5}$$
$$= ((M_1)_{k_1} v_{j_1}^T)((M_2)_{k_2} w_{j_2}^T) = (v_{k_1}(M_1)_{j_1}^T)(w_{k_2}(M_2)_{j_2}^T)$$
$$= \langle (M_1)_{j_1}, v_{k_1} \rangle \langle (M_2)_{j_2}, w_{k_2} \rangle.$$

Now using (2) together with (3),(4), and (5) we are ready to prove that the player P_k can make an additional check whether P_j re-shared correctly the shares in the degree reduction phase. To perform this check P_k uses his old shares v_{k_1} and w_{k_2} together with the newly received shares $y^{(i)}_{j_1,j_2,k_3}$ from P_j and some public information.

$$\langle (M_1)_{j_1}, v_{k_1} \rangle \langle (M_2)_{j_2}, w_{k_2} \rangle = \sum_{i=1}^{d_4} \langle (M_4)_{k_4}, e_i \rangle \langle y^{(i)}_{j_1,j_2,k_3}, \varepsilon_3^T \rangle.$$

Note that we can simply choose $\mathcal{M}_3 = \mathcal{M}_1$ and $\mathcal{M}_4 = \mathcal{M}_2$, in this case we have $\Gamma_1 = \Gamma_3$.

3.6   The New Shares

Finally, in order to complete the protocol we need to define the new shares. Recall that $j_1 \in \varphi_1(j)$ and $j_2 \in \varphi_2(j)$ if and only if $\{j_1, j_2\} \in \varphi(j)$. That is way we will denote $x^{(i)}_{j_1,j_2}$ and $y^{(i)}_{j_1,j_2,\varphi_3(k)}$ for $j_1 \in \varphi_1(j)$ and $j_2 \in \varphi_2(j)$ also by $x^{(i)}_{\varphi(j)}$ and by $y^{(i)}_{\varphi(j),\varphi_3(k)}$.

As we mentioned earlier in Section 3.4 for any group of players $\widetilde{G} \in \Gamma_3$ there exists a recombination vector $\widetilde{\lambda}_{\varphi_3(\widetilde{G})}$ such that they can reconstruct together the

first coordinate of the vector $x^{(i)}_{\varphi(j)}$, i.e., $x^{(i)}_{\varphi(j)}\varepsilon_3$, for $i = 1, \ldots, d_4$ (reconstruction phase of $VSS(i)$) as follows:

$$
\begin{aligned}
(y^{(i)}_{\varphi(j),\varphi_3(\widetilde{G})}\varepsilon_3)\widetilde{\lambda}^T_{\varphi_3(\widetilde{G})} &= \langle \widetilde{\lambda}_{\varphi_3(\widetilde{G})}, y^{(i)}_{\varphi(j),\varphi_3(\widetilde{G})}\varepsilon_3 \rangle \\
&= \sum_{k \in \widetilde{G}} \widetilde{\lambda}_{\varphi_3(k)}(y^{(i)}_{\varphi(j),\varphi_3(k)}\varepsilon_3) = x^{(i)}_{\varphi(j)}\varepsilon_3,
\end{aligned} \tag{6}
$$

Note also that for any group of players $G \in \Gamma$ there exists a recombination vector $\lambda_{\varphi(G)}$ such that they can reconstruct together the product of the secrets $s_1 s_2$.

$$
\begin{aligned}
(x^{(1)}_{\varphi(G)}\varepsilon_3)\lambda^T_{\varphi(G)} &= \langle \lambda_{\varphi(G)}, x^{(1)}_{\varphi(G)}\varepsilon_3 \rangle \\
&= \langle \lambda_{\varphi(G)}, (v_{\varphi_1(G)} \otimes w_{\varphi_2(G)})\varepsilon \rangle = s_1 s_2.
\end{aligned} \tag{7}
$$

(Here the last equality from Subsection 3.2 and the note from Subsection 3.3 are used.)

Now we are ready to define the new shares. Denote the list of good players by $\mathcal{L} \in \Gamma$, then P_k computes his new shares as follows:

$$
z_{\varphi_3(k)} = \sum_{j \in \mathcal{L}} \lambda_{\varphi(j)} y^{(1)}_{\varphi(j),\varphi_3(k)}.
$$

For the new shares $z_{\varphi_3(k)}$ the pair-wise check holds:

$$
\begin{aligned}
(M_3)_{\varphi_3(i)} z^T_{\varphi_3(k)} &= \sum_{j \in \mathcal{L}} \lambda_{\varphi(j)} (M_3)_{\varphi_3(i)} (y^{(1)}_{\varphi(j),\varphi_3(k)})^T \\
&= (\sum_{i \in \mathcal{L}} \lambda_{\varphi(j)} y^{(1)}_{\varphi(j),\varphi_3(i)}) (M_3)^T_{\varphi_3(k)} = z_{\varphi_3(i)} (M_3)^T_{\varphi_3(k)}.
\end{aligned}
$$

For any $\widetilde{G} \in \Gamma_3$ the players can reconstruct together the product $s_1 s_2$ using (6) and (7) as follows:

$$
\begin{aligned}
(z_{\varphi_3(\widetilde{G})}\varepsilon_3)\widetilde{\lambda}^T_{\varphi_3(\widetilde{G})} &= \langle \widetilde{\lambda}_{\varphi_3(\widetilde{G})}, z_{\varphi_3(\widetilde{G})}\varepsilon_3 \rangle = \sum_{k \in \widetilde{G}} \widetilde{\lambda}_{\varphi_3(k)}(z_{\varphi_3(k)}\varepsilon_3) \\
&= \sum_{k \in \widetilde{G}} \widetilde{\lambda}_{\varphi_3(k)}(\sum_{j \in \mathcal{L}} \lambda_{\varphi(j)}(y^{(1)}_{\varphi(j),\varphi_3(k)}\varepsilon_3)) \\
&= \sum_{j \in \mathcal{L}} \lambda_{\varphi(j)}(\sum_{k \in \widetilde{G}} \widetilde{\lambda}_{\varphi_3(k)}(y^{(1)}_{\varphi(j),\varphi_3(k)}\varepsilon_3)) \\
&= \sum_{j \in \mathcal{L}} \lambda_{\varphi(j)}(x^{(1)}_{\varphi(j)}\varepsilon_3) = s_1 s_2
\end{aligned}
$$

At the end of the protocol each player P_k possesses new shares $z_{\varphi_3(k)}$ of MSP $\mathcal{M}_3$ (computing the access structure Γ_3) of the product $s_1 s_2$.

Lemma 1. *Suppose that for the MSPs $\mathcal{M}_1$ and $\mathcal{M}_2$ there exist MSPs $\mathcal{M}_3$ and $\mathcal{M}_4$ such that*

$$\mathcal{M}_1 \diamond \mathcal{M}_2 = \mathcal{M} = \mathcal{M}_3 \diamond \mathcal{M}_4.$$

Let Γ be the access structure computed by the strongly multiplicative resulting MSP $\mathcal{M}$ from MSPs $\mathcal{M}_1$ and $\mathcal{M}_2$ and/or from MSPs $\mathcal{M}_3$ and $\mathcal{M}_4$ and let also the access structures Γ and Γ_i for $i = 1, 2, 3$ satisfy the conditions

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

Then the "degree reduction" protocol is information-theoretically secure against $(\Delta_1, \Delta_2, \Delta_A)$-adversary.

Due to lack of space we will not give a formal security proof for our protocol. However, to have a feeling why it is secure, note first that in the re-sharing phase every player could verify whether the "true" part of his share is correct or not. Then, as in the protocol from [4], the shares of the players (in our case the "true" part of the shares) have to satisfy a fixed linear relation, which allow every player to complain against incorrect re-sharing.

3.7 Complexity Issues

In this subsection we will follow [4]. Define $msp_{\mathbb{F}}(f)$ to be the size of the smallest MSP over $\mathbb{F}$ computing a monotone boolean function f. Next define $\mu_{\mathbb{F}}(f)$ to be the size of the smallest multiplicative MSP over $\mathbb{F}$ computing f. Similarly, $\mu_{\mathbb{F}}^*(f)$ to be the size of the smallest strongly multiplicative MSP. In other words for a given adversary A with adversary structure Δ_A we require for every set $B \in \Delta_A$ to have $B \notin \Gamma$, but $B^c \in \Gamma$. By definition, we have $msp_{\mathbb{F}}(f) \leq \mu_{\mathbb{F}}(f) \leq \mu_{\mathbb{F}}^*(f)$. In [4] Cramer *et al.* characterized the functions that (strongly) multiplicative MSP's can compute, and proved that the multiplication property for an MSP can be assumed without loss of efficiency. In particular, for the passive (multiplicative) case they proved that $\mu_{\mathbb{F}}(f) \leq 2\ msp_{\mathbb{F}}(f)$ provided that f is Q^2 function. Unfortunately there is no similar result for the strongly multiplicative case. Instead the authors in [4] proved that for an active adversary $\mu_{\mathbb{F}}^*(f)$ is bounded by the so-called "formula complexity".

In the recent paper of Nikov *et al.* [17] a different approach is considered. Recall that in that model given an Q^3 adversary A we are looking for two access structures (resp. monotone boolean functions) Γ_1 and Γ_2 (resp. f_1 and f_2) such that their strongly multiplicative resulting MSP computes Γ (resp. f). Or in other words for a given adversary A with adversary structure Δ_A we require that for every set $B \in \Delta_A$ to have $B \notin \Gamma_1$, $B \notin \Gamma_2$ but $B^c \in \Gamma$. Let us define $\nu_{\mathbb{F}}(f)$ to be the size of the smallest strongly multiplicative resulting MSP over $\mathbb{F}$ computing f. How these two measures $\mu_{\mathbb{F}}^*(f)$ and $\nu_{\mathbb{F}}(f)$ are related as well as whether this new notion give us better measure for the complexity of an MPC is subject of ongoing research.

References

1. M. Ben-Or, S. Goldwasser and A. Wigderson, Completeness Theorems for Non-Cryptographic Fault-Tolerant Distributed Computation, *Proc. ACM STOC'88*, 1988, pp. 1–10.
2. D. Chaum, C. Crepeau and I. Damgard, Multi-Party Unconditionally Secure Protocols, *Proc. ACM STOC'88*, 1988, pp. 11–19.
3. R. Cramer, Introduction to Secure Computation, *Lectures on Data Security – Modern Cryptology in Theory and Practice*, Springer-Verlag LNCS 1561, 1999, pp. 16–62.
4. R. Cramer, I. Damgard and U. Maurer, General Secure Multi-Party Computation from any Linear Secret Sharing Scheme, *Proc. EUROCRYPT 2000*, Springer-Verlag LNCS 1807, 2000, pp. 316–334.
5. R. Cramer, S. Fehr, Optimal Black-Box Secret Sharing over Arbitrary Abelian Groups, *Proc. CRYPTO 2002*, Springer-Verlag LNCS 2442, 2002, 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. I.Damgard, An Error in the Mixed Adversary Protocol by Fitzi, Hirt and Maurer, *Bricks Report*, RS-99-2, 1999.
8. S. Fehr, U. Maurer, Linear VSS and Distributed Commitments Based on Secret Sharing and Pairwise Checks, *Proc. CRYPTO 2002*, Springer Verlag LNCS 2442, 2002, pp. 565–580.
9. M.Fitzi, M.Hirt and U.Maurer, Trading Correctness for Privacy in Unconditional Multi-Party Computation, *Proc. CRYPT0'98*, Springer-Verlag, LNCS 1462, 1998, pp. 121–136.
10. R. Gennaro, M. Rabin, T. Rabin, Simplified VSS and Fast-Track Multi-party Computations with Applications to Threshold Cryptography, *Proc. ACM PODC'98*, 1998.
11. O. Goldreich, S. Micali and A. Wigderson, How to Play Any Mental Game or a Completeness Theorem for Protocols with Honest Majority, *Proc. ACM STOC'87*, 1987, pp. 218–229.
12. M. Hirt, U. Maurer, Player Simulation and General Adversary Structures in Perfect Multi-party Computation, *J. of Cryptology* 13, 2000, pp. 31–60.
13. 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.
14. U. Maurer, Secure Multi-Party Computation Made Simple, *3rd Conference on Security in Communication Networks*, September 12–13, 2002, Amalfi, Italy, Springer-Verlag LNCS 2576, 2003, pp. 14–28.
15. 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.

16. V. Nikov, S. Nikova, B. Preneel, J. Vandewalle, On Distributed Key Distribution Centers and Unconditionally Secure Proactive Verifiable Secret Sharing Schemes based on General Access Structure, *INDOCRYPT* 2002, Springer-Verlag LNCS 2551, 2002, pp. 422–437.
17. V. Nikov, S. Nikova, B. Preneel, Multi-Party Computation from any Linear Secret Sharing Scheme Secure against Adaptive Adversary: The Zero-Error Case, *Cryptology ePrint Archive*: Report 2003/006.
18. A. Shamir, How to share a secret, *Commun. ACM* 22, 1979, pp. 612–613.