Applying General Access Structure to Metering Schemes

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

[1] Department of Mathematics and Informatics,
Veliko Tarnovo Uiversity,
5000 Veliko Tarnovo, Bulgaria
vnikov@mail.com

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

Abstract. In order to decide on advertisement fees for web servers, Naor and Pinkas [15] introduced metering schemes secure against coalition of corrupt servers and clients. In their schemes any server is able to construct a proof to be sent to an audit agency if and only if it has been visited by at least a certain number of clients. After that in series of papers Masucci et. al. [1–3, 13, 14] generalized the idea of Naor and Pinkas proposing first metering scheme with pricing and dynamic multi-threshold metering schemes and later applying general access structures and a linear algebraic approach to metering schemes.

In this paper we are interested in the efficiency of applying general access structure and linear algebraic approach to metering schemes. We propose a new model considering general access structures for clients, corrupted clients and servers. Then we bind the access structures for clients and corrupted clients into one. We propose a new metering scheme, which is more efficient on the communication complexity and memory storage compared with the scheme proposed in [3].

Advertising is one of the approaches for making money on the Internet. In order for advertising to be effective the advertisers must have a way to measure the exposure of their ads. For this purpose the so called metering schemes, which should be secure against coalition of corrupted servers and clients, are used. In the paper we propose a model for metering schemes with fully general access structure which is simpler and more efficient than the known ones and we prove that it satisfies stronger security requirements.

⋆ The author was partially supported by NATO research fellowship and Concerted Research Action GOA-MEFISTO-666 of the Flemish Government.

1 Introduction

A metering scheme is a protocol to measure the interaction between clients and servers on a net. The time is divided in *time frames* and the audit agency is interested in counting the number of visits received by each server in any time frame. Metering schemes are useful in order to decide the amount of money to be paid to web servers hosting ads, as well as measurement of the use of coupons [15]. Franklin and Malkhi [10] were the first to consider the metering problem in rigorous approach. Their solutions only offer "lightweight security", which can not be applied if there are strong commercial interests to falsify the metering result. Naor and Pinkas [15], consequently, introduced metering schemes secure against fraud attempts by servers and clients. In their scheme any server which has been visited by any set of r or more clients in a time frame, where r is a fixed threshold, is able to compute a proof, whereas any server receiving visits from less than r clients has no information about the proof. In this threshold case scenario for both clients and servers, threshold refers to the maximum number of colluding players (server, clients). In order to have more flexible payment system Masucci and Stinson [1, 13] introduced metering scheme with pricing. To be able to measure the number of visits in any granularity Blundo et. al. in [2] introduced dynamic multi-threshold metering schemes which are the metering schemes with associated threshold to any server for any time frame. Masucci and Stinson in [14] made the next step considering the general access structure for the clients and threshold scheme for servers, where the access structure is the family of all subsets of clients enabling a server to compute its proof. They proved also a lower bound on the communication complexity of metering schemes realizing such structures. A linear algebraic approach (i.e. applicable for any general monotone access structure) to metering schemes is presented in [3] by Blundo et. al. Namely, given any access structure for the clients, they presented a method to construct a metering scheme realizing it from any linear secret sharing scheme with the same access structure. Besides, they proved some properties about the relationship between metering schemes and secret sharing schemes. Some new bounds on the information distributed to clients and servers in a metering scheme are presented too. The main difference between the scheme in [3] and the scheme in [14] is that the second one is not optimal with respect to the communication complexity.

We will consider only the metering schemes that provide information theoretic security. For computational secure metering scheme based on the

Decisional Diffe-Hellman Assumption one can see [15]. Since we want to protect against general adversary structures, we need to start from general *Linear Secret Sharing Schemes* (rather than from Shamir's scheme). It is well known that LSSS's are in natural $1-1$ correspondence with *Monotone Span Program* MSP, introduced by Karchmer and Wigderson [12]). MSP's can be viewed as a linear algebra model for computing monotone (access) function. Moreover, such an MSP always exists because MSP's can compute any monotone function. Threshold-based secret sharing and metering make sense only in environment where one assumes that trust is "uniformly distributed" over the players (clients and servers): any player subset of a certain cardinality is equally likely (or unlikely) to cheat. In many natural scenarios this assumption does not model reality well, and moreover, in more realistic model no threshold solution will work. Why we need to introduce a general access structure on the set of servers? In the model introdused by Naor and Pinkas the audit agency deals with servers, but in fact the servers are owned by companies, where each company posses different number of servers. In this scenario the uniformly distributed trust on the set of servers does not model the reality well.

In this paper we first distinguish between three types of general access structures: for clients, corrupted clients and servers. The access structure for clients consists of qualified and forbidden set of clients, i.e. sets which allow or disallow the server visited by them in given time frame to compute its proof. The corrupted clients access structure gives us a possible distribution for the corrupted clients. These two access structures are bound into one access structure in Lemma 2. A general access structure is considered for the set of servers. In the previous papers all authors considered only the threshold case for them. We propose simpler and more efficient on communication complexity and memory storage metering scheme compared to the scheme proposed by Blundo et. al. [3]. The difference appears in the broadcast public information to clients and servers, which is less in our scheme. As Naor and Pikas [15] pointed out it will be nice to detect illegal behaviour of clients, i.e. verifying the shares received from clients. This issue is not considered in the paper, since it is ignored in the papers [1–3, 13, 14] too.

The paper is organized as follows: In Section 2 we present the notations used to describe the metering schemes. In Section 3 we study the relationship between metering schemes and general access structures for clients, corrupt clients and servers. In Section 4 we first present a linear secret sharing scheme and linear algebraic approach to generalized access structures. Then this approach is used to design a metering scheme. Finally, we examine our scheme for efficiency and correctness.

2 Preliminaries

A *secret sharing scheme* (SSS) allows to share a secret among several participants, such that only qualified subset of them can recover the secret pooling together their information. Subsets of participants that are not enabled to recover the secret have absolutely no information about it. The secret sharing were proposed independently by Shamir [17] and Blakley [4]. The first secret sharing schemes have been (r, k)-*threshold schemes*, were only groups of more than a certain number of participants r (where $r \leq k$ and k is the number of all players) can reconstruct the secret. Brickell [6] points out how the linear algebraic view leads to a natural extension to a wider class of secret sharing schemes that are not necessarily of threshold type. This have later been generalized to all possible so-called monotone access structures by Karchmer and Wigderson [12] based on a linear algebraic computational device called Monotone Span Program.

As usual we call the groups which 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 Γ, and the collection of all forbidden groups is denoted by Δ. In fact, Γ is *monotone increasing* and Δ is *monotone decreasing*. The tuple (Γ, Δ) is called an *access structure* if $\Gamma \cap \Delta = \emptyset$. If $\Gamma \cup \Delta = 2^P$, where P is the set of participants, then we say that (Γ, Δ) is *complete* and we denote it by Γ. Otherwise, we say that (Γ, Δ) is *incomplete*. By Γ^- we denote the collection of *minimal sets* of Γ and by Δ^+ we denote the collection of *maximal sets* of Δ. It is obvious that the (Γ^-, Δ^+) generate the (Γ, Δ). Let K be a finite field. We will consider a general monotone access structure (Γ, Δ), which describes subsets of participants that are qualified to recover the secret $s \in K$ in the set of possible secret values.

A (k, r)-Vandermonde matrix (over K) with $r \leq k$, is a matrix which i-th row is of the form $(1, \alpha_i, ..., \alpha_i^{r-1})$, where $\alpha_1, ..., \alpha_k \in K$. For an arbitrary matrix M over K, with m rows labeled by $1, \ldots, m$ and for arbitrary non-empty subset A of $\{1, \ldots, m\}$, let M_A denote the matrix obtained by keeping only those rows i with $i \in A$. If $\{i\} = A$ we write M_i. Consider the set of row-vectors $v_{i_1}, \ldots, v_{i_k}$ and let $A = \{i_1, \ldots, i_k\}$ be the set of indices, then we denote by v_A the matrix consisting of rows $v_{i_1}, \ldots, v_{i_k}$. Instead of $\langle \varepsilon, v_i \rangle$ for $i \in A$ we will write $\langle \varepsilon, v_A \rangle$. Let M_A^T

denote the transpose of M_A, and let $Im(M_A^T)$ denote the K-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.

It is well known that any square Vandermonde matrix has non-zero determinant. If M is a (k,r)-Vandermonde matrix over K and A is non-empty subset of $\{1,...,k\}$, then the rank of M_A is maximal (i.e. is equal to r, or equivalently, $Im(M_A^T) = K^r$) if and only if $|A| \geq r$. Moreover: Let ε denote the column vector $(1,0,...,0) \in K^r$. If $|A| < r$, then $\varepsilon \notin Im(M_A^T)$, i.e. there is no $\lambda \in K^{|A|}$ such that $M_A^T \lambda = \varepsilon$.

Let us define the standard scalar product $\langle x, y \rangle$ and $x \perp y$, when $\langle x, y \rangle = 0$. For a K-linear subspace V of K^l, $V^\perp$ denotes the collection of elements of K^l, that are orthogonal to all of V (the orthogonal complement), which is again a K-linear subspace. For all subspaces V of K^l we have $V = (V^\perp)^\perp$, $(Im(M_A^T))^\perp = Ker(M_A)$ or $Im(M_A^T) = (Ker(M_A))^\perp$, $\langle x, M_A^T y \rangle = \langle M_A x, y \rangle$. Hence from $Im(M_A^T) = (Ker(M_A))^\perp$ follows the lemma.

Lemma 1. *[7] The vector $\varepsilon \notin Im(M_A^T)$ if and only if there exists $z \in K^l$ such that $M_A z = 0$ and $z_1 = 1$.*

3 Metering schemes for General Access Structures

Consider the following scenario: there are n clients, k servers and an audit agency $\mathcal{A}$ which is interested in counting the client visits to the servers in τ different time frames. For any $i = 1, \ldots, n$ and $j = 1, \ldots, k$, we denote by $\mathcal{C}_i$ the $i-$th client and by S_j the $j-$th server.

We consider an *access structure* (Γ, Δ) of qualified and forbidden groups for the set of clients $\{\mathcal{C}_1, \ldots, \mathcal{C}_n\}$.

In a metering scheme realizing the client access structure (Γ, Δ) any server which has been visited by at least a qualified subset of clients in Γ in fixed time frame is able to provide the audit agency with a proof for the visits it has received.

A second access structure (complete) Γ_S can be considered for the set of servers $\{S_1, \ldots, S_k\}$. We call the set of subsets of servers *corrupt* if they are not in Γ_S. We denote also the set of possible subsets of *corrupt clients* by Δ_C, which is in fact monotone decreasing. It is obvious requirement that $\Gamma \cap \Delta_C = \emptyset$.

A corrupt server can be assisted by corrupt clients and other corrupt servers in computing its proof without receiving visits from qualified subsets. A corrupt client can donate to a corrupt server the whole private information received by the audit agency during the initialization phase.

A corrupt server can donate to another corrupt server the private information received from clients in previous time frames and in the actual time frame.

Several phases can be defined in the Metering scheme. We will follow the settings of the scheme in [3]:

There is an **initialization phase** in which the audit agency $\mathcal{A}$ chooses the access structures, computes the corresponding matrices and make them public and distributes some information to any client $\mathcal{C}_i$ through a private channel. For any $i = 1, \ldots, n$ we denote by $v^{(t)}_{\varphi(i)}$ the shares that the audit agency $\mathcal{A}$ gives to the client $\mathcal{C}_i$ for time frames $t = 1, \ldots, \tau$.

A **regular operation** consists of a client visit to a server during a time frame. During such a visit the client gives to the visited server a piece of information which depends on the private information, on the identity of the server and on the time frame during which the client visits the server. For any $i = 1, \ldots, n$; $j = 1, \ldots, k$ and $t = 1, \ldots, \tau$, we denote by $c^{(t)}_{\varphi(i),\widetilde{\varphi}(j)}$ the information that the client $\mathcal{C}_i$ sends to the server S_j when visiting him in time frame t.

During the **proof computation phase** any server S_j which has been visited by at least a subset of qualified clients in time frame t is able to compute its proof. For any $j = 1, \ldots, k$ and $t = 1, \ldots, \tau$ we denote by $p^{(t)}_{\widetilde{\varphi}(j)}$ the proof computed by the server S_j at time t when it has been visited by qualified set of clients.

During the **proof verification phase** the audit agency $\mathcal{A}$ verifies the proofs received by servers and decides on the amount of money to be paid to servers. If the proof received from a server at the end of a time frame is correct, then $\mathcal{A}$ pays the server for its services.

Definition 1. *[3] An (n, k, τ) metering scheme realizing the access structures (Γ, Δ), Γ_S and corrupt set of clients Δ_C is a protocol to measure the interaction between n clients $\mathcal{C}_1, \ldots, \mathcal{C}_n$ with access structure (Γ, Δ) and k server $S_1, \ldots, S_k$ with access structure Γ_S during τ time frames in such a way that the following properties are satisfied:*

1. For any time frame t any client is able to compute the information needed to visit any server.

2. For any time frame t any server S_j which has been visited by a qualified subset of clients $G \in \Gamma$ in time frame t can compute its proof for t.

3. Let B_2 be a coalition of corrupt servers, i.e. $B_2 \notin \Gamma_S$ and let B_1 be a coalition of corrupt clients, i.e. $B_1 \in \Delta_C$. Assume that in some time frame t each server in the coalition has been visited by a subset of *forbidden clients B_3, i.e. $B_3 \in \Delta$. Then the servers in the coalition B_2 have no information about their proofs for time frame t, even if they are helped by the corrupt clients in B_1.*

In [16] we introduced an operation for the access structure, which generalize the notion of $Q^2(Q^3)$ adversary structure introduced by Hirt and Maurer [11]. Now we will expand this definition.

Definition 2. *For the access structure (Γ, Δ) and monotone decreasing set Δ_C we define the operation $*$ as follows: $\Delta * \Delta_C = \{A = A_1 \cup A_2; A_1 \in \Delta, A_2 \in \Delta_C\}$.*

In order to build an (n, k, τ) metering scheme realizing the access structures (Γ, Δ), Γ_S and corrupt set of clients Δ_C, we consider the tuple $(\Gamma, \Delta * \Delta_C)$. It is obvious that $\Delta * \Delta_C$ is monotone decreasing.

Lemma 2. *An (n, k, τ) metering scheme realizing the access structures (Γ, Δ), Γ_S and corrupt set of clients Δ_C exists, if and only if $(\Gamma, \Delta * \Delta_C)$ is an access structure (i.e. $\Gamma \cap \Delta * \Delta_C = \emptyset$).*

In the next section we will present a metering scheme satisfying the conditions of Lemma 2. Such schemes we will call an (n, k, τ) metering scheme realizing the access structures $(\Gamma, \Delta * \Delta_C)$ and Γ_S.

4 Linear SSS and Metering Schemes

4.1 LSSS and MSP

A SSS is linear if the dealer and the participants use only linear operations to compute the shares and the secret. Each *linear SSS* (LSSS) can be viewed as derived from a monotone span program 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. 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.

Definition 3. *[12, 7] The quadruple $\mathcal{M} = (K, M, \varepsilon, \psi)$ is called monotone span program, where K is a finite field, M is a matrix (with m rows and $d \leq m$ columns) over K, $\psi : \{1, \ldots, m\} \to \{1, \ldots, n\}$ is a surjective function and ε is a fixed vector, called target vector, e.g. column vector $(1, 0, ..., 0) \in K^d$. The size of $\mathcal{M}$ is the number of rows m.*

Here ψ labels each row with a number from $[1, \ldots, m]$ corresponding to a fixed player, so we can think of each player as being the "owner" of one or more rows. And for every player we consider a function φ which gives the set of rows owned by the player. In some sense φ is inverse of ψ. It is well known that the number of columns d can be chosen to be smaller than the number of rows m, without changing the access structure that is computed by a MSP.

Theorem 1. *[5, 9] MSP is said to compute an access structure (Γ, Δ) if and only if*
a) $\varepsilon \in Im(M_G^T)$ when G is a member of Γ.
b) $\varepsilon \notin Im(M_G^T)$ when G is a member of Δ.

4.2 The scheme realizing Metering Scheme for General Access Structure

Let M be the matrix obtained from MSP (Definition 3) computing the $(\Gamma, \Delta * \Delta_C)$ access structure.

Conjecture: *For any generalized complete access structure Γ there exists a "special" matrix N with the following property:*
(i) $G \notin \Gamma$ if and only if the rows in N_G are linearly independent.

Note that if Γ is a threshold (r, k) access structure with (k, r)-Vandermonde matrix the requirement *(i)* is satisfied. Also in some cases the matrix N can be derived from the matrix M by removing the first column in M, but this can not be used as a rule.

For the access structure Γ_S we consider such kind of "special" matrix N as in the conjecture above. Analogously to the MSP we will denote by $\widetilde{\psi}$ the surjective function which label each row of N with a corresponding player, and $\widetilde{\varphi}$ the "inverse" of $\widetilde{\psi}$.

Initialization: The audit agency $\mathcal{A}$ chooses access structures $(\Gamma, \Delta * \Delta_C)$ and Γ_S. Using MSP these access structures are bound with matrices M and N. Let M be with m rows and d columns and N be with $\widetilde{m}$ rows and $\widetilde{d}$ colomns. These matrices are made public.
Next $\mathcal{A}$ chooses τ random $d \times \widetilde{d}$ matrices $R^{(t)}$. We can consider them as one "big" $d\tau \times \widetilde{d}$ matrix R, which is kept secret.
So, $\mathcal{A}$ gives to each client $\mathcal{C}_i$ row vectors $v_{\varphi(i)}^{(t)} = M_{\varphi(i)} R^{(t)}$ for $t = 1, \ldots, \tau$. These are the shares of client $\mathcal{C}_i$ in time frame t.

Regular Operation: When client $\mathcal{C}_i$ visits a server S_j during a time frame t, $\mathcal{C}_i$ computes the values $c_{\varphi(i),\widetilde{\varphi}(j)}^{(t)} = N_{\widetilde{\varphi}(j)} (v_{\varphi(i)}^{(t)})^T$ and sends them to the server S_j.

Proof Computation: Assume that the server S_j has been visited from a qualified set $G \in \Gamma$ of clients during a time frame t. So, it computes λ (Theorem 1) s.t. $M^T_{\varphi(G)}\lambda = \varepsilon$. With λ it computes $p^{(t)}_{\widetilde{\varphi}(j)} = \langle c^{(t)}_{\varphi(G),\widetilde{\varphi}(j)}, \lambda^T\rangle$ which are the desired proofs and sends them to $\mathcal{A}$.

Proof Verification: When the audit agency $\mathcal{A}$ receives these values $p^{(t)}_{\widetilde{\varphi}(j)}$ it can easily verify if this is the correct proof for the server S_j for time t. $\mathcal{A}$ calculates the value $\widetilde{p}^{(t)}_{\widetilde{\varphi}(j)} = \langle N_{\varphi(j)}, (R^{(t)})_1\rangle$, where by $(R^{(t)})_1$ we denote the first row of matrix $R^{(t)}$ and compares whether $p^{(t)}_{\widetilde{\varphi}(j)} = \widetilde{p}^{(t)}_{\widetilde{\varphi}(j)}$. We will prove that if the server S_j has been visited from a qualified set $G \in \Gamma$ of clients during a time frame t the equality should hold.

$$\begin{aligned}
p^{(t)}_{\widetilde{\varphi}(j)} &= \langle c^{(t)}_{\varphi(G),\widetilde{\varphi}(j)}, \lambda^T\rangle = \langle N_{\widetilde{\varphi}(j)}(v^{(t)}_{\varphi(G)})^T, \lambda^T\rangle \\
&= \langle N_{\widetilde{\varphi}(j)}(M_{\varphi(G)}R^{(t)})^T, \lambda^T\rangle = \langle N_{\widetilde{\varphi}(j)}(R^{(t)})^T M^T_{\varphi(G)}, \lambda^T\rangle \\
&= \langle N_{\widetilde{\varphi}(j)}(R^{(t)})^T, \lambda^T M_{\varphi(G)}\rangle = \langle N_{\widetilde{\varphi}(j)}(R^{(t)})^T, \varepsilon^T\rangle \\
&= \langle N_{\widetilde{\varphi}(j)}, \varepsilon^T R^{(t)}\rangle = \langle N_{\widetilde{\varphi}(j)}, (R^{(t)})_1\rangle \\
&= \widetilde{p}^{(t)}_{\widetilde{\varphi}(j)}
\end{aligned}$$

4.3 Analysis of the Scheme

It is obvious that *Property 1* and *Property 2* of Definition 1 are satisfied. Now we prove that *Property 3* is satisfied. We consider the worst possible case, in which a subset of clients $D \in \Delta * \Delta_C$ helps a coalition of corrupt servers $B_2 \notin \Gamma_S$ in computing their proofs for time frame τ. The total information known to the coalition of corrupt servers is constituted by the maximum information collected in time frames $1, \ldots, \tau - 1$. That is, we assume that each server in the coalition has been visited by all clients $\mathcal{C}_1, \ldots, \mathcal{C}_n$ in these time frames plus the information received in time frame τ.

Since the audit agency $\mathcal{A}$ chooses the matrices $R^{(t)}$ randomly and keep them secret the clients have different shares for different time frames, so the information they give visiting the server S_j is different. Hence all collected information for previous visit is not consistent with the current information and the coalition of corrupt serves can not use it.

Let us consider the value $p^{(t)}_{\widetilde{\varphi}(j)} = \langle c^{(t)}_{\varphi(G),\widetilde{\varphi}(j)}, \lambda^T\rangle$. Assume that the group of clients $D \in \Delta * \Delta_C$ helps S_j to compute his proof. It is easy to prove (see [7, 8] or [16, Theorem 2]) that from the point of view of the clients in D, the information $c^{(t)}_{\varphi(D),\widetilde{\varphi}(j)}$ can be consistent with any secret matrix $\widetilde{R}^{(t)}$. So, the clients in D have no information about the secret matrix $R^{(t)}$ and hence for the information $c^{(t)}_{\varphi(G),\widetilde{\varphi}(j)}$ for some $G \in \Gamma$.

Finally, let us consider the value $\widetilde{p}^{(t)}_{\widetilde{\varphi}(j)} = \langle N_{\widetilde{\varphi}(j)}, (R^{(t)})_1 \rangle$. The coalition $B_2 \notin \Gamma_s$ can try to guess $(R^{(t)})_1$ or if there is a linear dependance between the row-vectors $N_{\widetilde{\varphi}(j)}$ for $j \in B_2$ to compute $\widetilde{p}^{(t)}_{\widetilde{\varphi}(j)}$ provided that they already know all values $\widetilde{p}^{(t)}_{\widetilde{\varphi}(j_1)}$ for $j_1 \in B_2 \setminus \{j\}$.

Let us consider the second possibility for server S_j which is visited only by clients $D \in \Delta * \Delta_C$. In fact, we can prove a stronger requirement in addition to the requirements in Definition 1.

Definition 4. *An (n, k, τ) metering scheme realizing the access structures (Γ, Δ), Γ_S and corrupt set of clients Δ_C is a protocol to measure the interaction between n clients $\mathcal{C}_1, \ldots, \mathcal{C}_n$ with access structure (Γ, Δ) and k server $S_1, \ldots, S_k$ with access structure Γ_S during τ time frames in such a way that the following properties are satisfied:*

1. - 3. As in Definition 1

4. Let B_2 be a coalition of corrupt servers, i.e. $B_2 \notin \Gamma_S$ and let B_1 be a coalition of corrupt clients, i.e. $B_1 \in \Delta_C$. Assume that in some time frame t the fixed server in the coalition (e.g. S_j and $j \in B_2$) has been visited by a subset of forbidden clients B_3, i.e. $B_3 \in \Delta$. Assume that in the same time frame t each other server in the coalition B_2 has been visited by a subset of qualified clients B_4, i.e. $B_4 \in \Gamma$. Then the servers in the coalition $B_2 \setminus \{j\}$ are able to compute their proofs for time frame t, but they are unable to "help" the server S_j with the computation of his proofs, even if they are helped by the corrupt clients in B_1.

Even if all the servers in the corrupted coalition B_2, except S_j, have been visited by a qualified subset of clients B_4 at that time frame (i.e. they are able to compute their proofs), S_j can not compute its proofs by finding a linear combination of their proofs $p^{(t)}_{\widetilde{\varphi}(j_1)}$ for $j_1 \in B_2 \setminus \{j\}$. This is true since B_2 is not in Γ_S and by requirement *(i)* of the Conjecture there is no linear combination between the row vectors $N_{\widetilde{\varphi}(j_1)}$ for $j_1 \in B_2 \setminus \{j\}$ and $N_{\widetilde{\varphi}(j)}$. Hence the *Property 4* of Definition 4 also holds.

4.4 Efficiency of the Scheme.

Let $|K| = q$ and denote by $dimE_i$ the dimension of the vector space generated by the vectors $M_{\varphi(i)}$ of client $\mathcal{C}_i$ over K, i.e. $dimE_i = |\varphi(i)|$. We denote by E_0 the set of secrets and by $dimE_0$ the dimension of E_0. It is well known that the information rate of a LSSS is $\rho = dimE_0/(max_{1\leq i\leq n} dimE_i)$

and this rate is optimal (e.g. $\rho = 1$) in the threshold case. Assuming that the matrix M (built by means of MSP) has a maximum possible information rate for the given access structure Γ. To be able to compare our result with the result of [3] we need to consider Γ_S to be a threshold (r, k) access structure. In this case the matrix N is (k, r)-Vandermonde matrix (i.e. $\widetilde{m} = k$, $\widetilde{d} = r$ and $\widetilde{\psi}$, $\widetilde{\varphi}$ are bijections).

In [3] the audit agency broadcasts two types of public information one is the linear mapping M_χ that enables the clients in $\chi \in \Gamma$ to compute the secret. The second is the linear mapping Π_j^t, i.e. the numbers $\lambda_{j,i}^t$ for $j = 1, \ldots, k$; $i = 1, \ldots, r\tau$; and $t = 1, \ldots, \tau$.

The amount of information that a client $\mathcal{C}_i$ receives from the audit agency during the initialization phase (i.e. the shares of the client) is equal to $r\ \tau\ log(q)\ dimE_i$, which is the same as in [3].

The amount of information that a client sends to a server during a visit is equal to $log(q)\ dimE_i$, which is again the same as in [3].

In our scheme the public information broadcast by audit agency $\mathcal{A}$ given by the matrices M and N is equal to $d\ log(q) \sum_{i=1}^{n} dimE_i = m\ d\ log(q)$ and $k\ r\ log(q)$, respectively. Note also that the clients need to know only the matrix N, and the servers need to know only the matrix M, in order to perform their duties.

On the other hand the amount of broadcast information in [3] is the linear mapping M_χ, which corresponds to our matrix M, and the numbers $\lambda_{j,i}^t$ from the second linear map Π_j^t. Hence the amount of information for the second mapping is $\tau^2\ k\ r\ log(q)$. Note also that both the clients and the servers need to know these numbers $\lambda_{j,i}^t$.

Therefore our scheme is more efficient on the communication complexity comparing with the scheme proposed in [3], since it broadcasts less ($k\ r\ log(q)$ v.s. $\tau^2\ k\ r\ log(q)$) public information to clients and servers. Another consequences is that the memory storage required in our scheme is less comparing with the Blundo et. al. [3] scheme.

5 Conclusions and Open Problem

In the paper we propose a model for metering schemes with fully general access structure – for clients, corrupted clients and servers. In the literature till now was considered general access structure for clients and threshold access structure for servers. The scheme is simpler, with more efficient communication complexity and memory storage than the known ones and we prove that it satisfies stronger security requirements.

There is still an open problem: to be proved the existence of a "special" matrix N for any access structure. It is well known [8] that every non-zero vector can be used as a target vector in the MSP. So, the question is whether we can build a matrix with a zero target vector. We can restate the conjecture as:

Conjecture′: A "special" matrix N is said to compute a generalized access structure (Γ, Δ) if and only if
a) $Ker(N_G^T) \neq \emptyset$ when G is a member of Γ.
b) $Ker(N_G^T) = \emptyset$ when G is a member of Δ.

References

1. **C. Blundo, A. De Bonis, B. Masucci**, Metering Schemes with Pricing, *Proc. of DISC'2000*, Toledo, October 4-6, 2000, *LNCS* 1914, pp.194-208, Springer Verlag, 2000.
2. **C. Blundo, A. De Bonis, B. Masucci, D. R. Stinson**, Dynamic Multi-Threshold Metering Schemes, *in Proc. of Seventh Annual Workshop on Selected Areas in Cryptography SAC 2000*, Waterloo, Canada, Aug 14 - 15, 2000, LNCS 2012, pp.130-144, Springer Verlag, 2001.
3. **C. Blundo, S. Martin, B. Masucci, C. Padro**, A Linear Algebraic Approach to Metering Schemes, *Cryptology ePrint Archive: Report 2001/087*.
4. **G. R. Blakley**, Safeguarding Cryptographic Keys, *AFIPS Conference Proc.* 48, 1979, pp.313-317.
5. **G. R. Blakley, G. A. Kabatianskii**, Linear Algebra Approach to Secret Sharing Schemes, Springer Verlag LNCS 829, 1994, pp.33-40.
6. **E. F. Brickell**, Some Ideal Secret Sharing Schemes, *J. of Comb. Math. and Comb. Computing* 9, 1989, pp.105-113.
7. **R. Cramer**, Introduction to Secure Computation. In Lectures on Data Security - Modern Cryptology in Theory and Practice, Ivan Damgaard (Ed.), Springer LNCS Tutorial, vol.1561, March 1999, pp.16-62.
8. **R. Cramer and S. Fehr** Optimal Black-Box Secret Sharing over Arbitrary Abelian Groups, to appear in *Proc. CRYPTO '02*.
9. **M. van Dijk**, A Linear Construction of Secret Sharing Schemes, *DCC* 12, 1997, pp.161-201.
10. **M. K. Franklin, D. Malkhi**, Auditable Metering with Lightweight Security, Finacial Cryptography'97, Febr. 1997, LNCS 1318, R.Hirschfeld (Ed.), 1997, pp.151-160.
11. **M. Hirt, U. Maurer**, Player Simulation and General Adversary Structures in Perfect Multiparty 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. **B. Masucci, D. R. Stinson**, Efficient Metering Schemes with Pricing, *IEEE Transactions on Information Theory*, Vol. 47, No. 7, November 2001, pp.2835-2844.
14. **B. Masucci, D. R. Stinson**, Metering Schemes for General Access Structures, *in Proc. of 6th European Symposium on Research in Computer Security ESORICS*

2000, Toulouse, France, October 4 - 6, 2000, LNCS 1895, pp.72-87, Springer Verlag, 2000.

15. **M. Naor, B. Pinkas**, Secure and Efficient Metering, *in Proc. of EUROCRYPT '98*, LNCS 1403, pp.576-590, 1998.
16. **V. Nikov, S. Nikova, B. Preneel, J. Vandewalle**, Applying General Access Structure to Proactive Secret Sharing Schemes, 23rd Symposium on Information Theory in the BENELUX, May 29-31, 2002, Louvain-la-Neuve, Belgium, pp.197-206.
17. **A. Shamir**, How to Share a Secret, *Communications of the ACM* 22, 1979, pp.612-613.

Appendix

Toy Example In order to give to the reader a better idea of the protocol, we will consider the following example: Let $K = GF(2)$ and let we have the access structures $\Gamma^- = \{123, 145, 245, 235, 135\}$, $(\Delta * \Delta_c)^+ = \{124, 125, 134, 234, 345\}$ and $\Gamma_S^- = \{12, 23, 34\}$, $\Delta_S^+ = \{14, 13, 24\}$. Let the public matrices M and N and the secret random matrix R (i.e. $\tau = t = 1$) are as follows:

$$M = \left(\begin{array}{c} 0\,0\,0\,1\,1 \\ \hline 0\,0\,1\,0\,1 \\ 0\,0\,0\,0\,1 \\ \hline 1\,0\,1\,1\,1 \\ 1\,1\,1\,1\,1 \\ \hline 0\,1\,0\,0\,0 \\ \hline 1\,1\,0\,1\,1 \\ 0\,0\,1\,1\,0 \end{array}\right) \qquad N = \left(\begin{array}{c} 1\,0\,0 \\ \hline 1\,0\,0 \\ 0\,1\,0 \\ \hline 0\,1\,0 \\ 0\,0\,1 \\ \hline 0\,0\,1 \end{array}\right) \qquad R = \begin{pmatrix} a_1 & b_1 & c_1 \\ a_2 & b_2 & c_2 \\ a_3 & b_3 & c_3 \\ a_4 & b_4 & c_4 \\ a_5 & b_5 & c_5 \end{pmatrix}$$

The agency gives to each client the corresponding row vectors:

$$v_{\varphi(1)} = \left(a_4 + a_5 | b_4 + b_5 | c_4 + c_5 \right), \quad v_{\varphi(2)} = \left(\begin{array}{c|c|c} a_3 + a_5 & b_3 + b_5 & c_3 + c_5 \\ a_5 & b_5 & c_5 \end{array}\right),$$

$$v_{\varphi(3)} = \left(\begin{array}{c|c|c} a_1 + a_3 + a_4 + a_5 & b_1 + b_3 + b_4 + b_5 & c_1 + c_3 + c_4 + c_5 \\ a_1 + a_2 + a_3 + a_4 + a_5 & b_1 + b_2 + b_3 + b_4 + b_5 & c_1 + c_2 + c_3 + c_4 + c_5 \end{array}\right),$$

$$v_{\varphi(4)} = \left(a_2 | b_2 | c_2 \right),$$

$$v_{\varphi(5)} = \left(\begin{array}{c|c|c} a_1 + a_2 + a_4 + a_5 & b_1 + b_2 + b_4 + b_5 & c_1 + c_2 + c_4 + c_5 \\ a_3 + a_4 & b_3 + b_4 & c_3 + c_4 \end{array}\right)$$

Let the set of qualified clients $\mathcal{C}_1, \mathcal{C}_4, \mathcal{C}_5$ visits the server S_3 and the set of forbidden clients $\mathcal{C}_1, \mathcal{C}_2, \mathcal{C}_4$ visits the servers S_1, S_2.

The server S_1 receives the following values from the clients:
$c_{\varphi(1),\widetilde{\varphi}(1)} = (a_4 + a_5)$, $\quad c_{\varphi(2),\widetilde{\varphi}(1)} = \begin{pmatrix} a_3 + a_5 \\ a_5 \end{pmatrix}$, $\quad c_{\varphi(4),\widetilde{\varphi}(1)} = (a_2)$.

Respectively, for the server S_2 the values are as follows:
$c_{\varphi(1),\widetilde{\varphi}(2)} = (a_4 + a_5 | b_4 + b_5)$, $\quad c_{\varphi(2),\widetilde{\varphi}(2)} = \left(\begin{array}{c|c} a_3 + a_5 & b_3 + b_5 \\ a_5 & b_5 \end{array}\right)$,
$c_{\varphi(4),\widetilde{\varphi}(2)} = (a_2 | b_2)$.

And for the server S_3:
$c_{\varphi(1),\widetilde{\varphi}(3)} = (b_4 + b_5 | c_4 + c_5)$, $\quad c_{\varphi(4),\widetilde{\varphi}(3)} = (b_2 | c_2)$,
$c_{\varphi(5),\widetilde{\varphi}(3)} = \left(\begin{array}{c|c} b_1 + b_2 + b_4 + b_5 & c_1 + c_2 + c_4 + c_5 \\ b_3 + b_4 & c_3 + c_4 \end{array}\right)$.

Since the server S_3 is visited by the set of qualified clients, it computes $\lambda = (1, 1, 1, 0)$ such that $M^T_{\varphi(1,4,5)}\lambda = \varepsilon$ and calculates his proof $p_3 = \begin{pmatrix} b_1 \\ c_1 \end{pmatrix}$.

Finally, the audit agency verifies that $\widetilde{p}_3 = \begin{pmatrix} b_1 \\ c_1 \end{pmatrix} = p_3$. Note that if S_1 and S_2 are corrupted servers they can not (even together) calculate their proofs $\widetilde{p}_1 = (a_1)$, $\widetilde{p}_2 = \begin{pmatrix} a_1 \\ b_1 \end{pmatrix}$, respectively. Even more, if one of the corrupted servers is S_3, which is visited by the set of qualified clients, the other bad server (e.g. S_1) is not able to compute its proof (by *Property 4* of Definition 4).