

# Robust Metering Schemes for General Access Structures

Ventzislav Nikov<sup>1</sup>, Svetla Nikova<sup>2</sup> \*, and Bart Preneel<sup>2</sup>

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

<sup>2</sup> 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.** In order to decide on advertisement fees for web servers, Naor and Pinkas introduced (threshold) metering schemes secure against coalitions of corrupt servers and clients. They show that one should be able to detect illegal behavior of clients, i.e., one needs to verify the shares received from clients. Most metering schemes do not offer this feature. But Ogata and Kurosawa pointed out a minor flaw in the extension protocol by Naor and Pinkas providing detection of such illegal behavior and propose a correction. In this paper we extend the linear algebra approach from Nikov et al. in order to build *robust* unconditionally secure general metering schemes. As a tool to achieve this goal we introduce *doubly-labelled* matrices and an operation on such matrices. Certain properties of this operation are proven.

## 1 Introduction

A metering scheme is a protocol to measure the interaction between clients and servers in a network. The time is divided into *time frames* and an audit agency counts the number of visits received by each server in any time frame. Metering schemes are useful in many applications, for instance to decide on the amount of money to be paid to web servers hosting advertisements, or for network accounting and electronic coupon management (see Naor and Pinkas [12]). Franklin and Malkhi [6] were the first to consider a rigorous approach to the metering problem. Their solutions offer “lightweight security”, meaning that they are suitable if there are no strong commercial interests to falsify the metering result. Subsequently, Naor and Pinkas [12], introduced metering schemes secure against fraud attempts by servers and clients. In their scheme any server which has been visited by any set of  $k + 1$  or more clients in a time frame, where  $k$  is a fixed threshold, is able to compute a proof, whereas any server receiving visits from less than  $k + 1$

---

\* The work described in this paper has been supported in part by the European Commission through the IST Programme under Contract IST-2002-507932 ECRYPT, IWT STWW project on Anonymity and Privacy in Electronic Services and Concerted Research Action GOA-MEFISTO-666 of the Flemish Government.

clients has no information about the proof. By proof we mean a value computed by the server that substantiates the visits of a qualified set of clients. In this threshold case scenario for both clients and servers, the threshold refers to the maximum number of colluding players (server, clients). In order to have a more flexible payment system Masuchi et al. have introduced metering schemes with pricing [1, 8]. Blundo et al. in [2] have introduced dynamic multi-threshold metering schemes which are metering schemes with an associated threshold for any server and for any time frame. These schemes allow metering with any granularity. In [7], Masucci and Stinson consider general access structures for the clients and a threshold scheme for the servers, where the access structure is the family of all subsets of clients enabling a server to compute its proof. A linear algebra approach (which is applicable for any general monotone access structure) to metering schemes is presented in [3] by Blundo et al. More specifically, given any access structure for the clients and threshold access structure for the servers, the authors propose a method to construct a metering scheme by means of linear secret sharing schemes. Besides, they prove some properties of the relationship between metering schemes and secret sharing schemes. The main difference between the scheme in [3] and the scheme in [7] is that the second one is not optimal with respect to the communication complexity. In [9] a general access structure on the set of servers is considered and a stronger model is proposed. The authors also describe a simpler, more efficient and more general scheme than the scheme from [3]. Another difference between the scheme in [3] and [9] is that in the first one only the threshold case for the set of servers is considered and that in the second one a Monotone Span Program based linear algebra approach is used. As Naor and Pinkas [12] pointed out one should be able to detect illegal behavior of clients by verifying the shares received from clients. This issue is not considered in [1–3, 7–9], but in [13] a minor flaw in the extension protocols [12] providing detection of such illegal behavior was pointed out and a correction was proposed.

In this paper we will extend the linear algebra approach from [9] in order to build *robust* metering schemes. We introduce a new notion *doubly-labelled* matrices and an operation on the matrices. Certain properties of this operation are given. The paper is organized as follows. In Sect. 2 we introduce linear Secret Sharing Schemes (SSS) and Multiplicative linear SSSs. In Sect. 3 *doubly-labelled* matrices are defined and certain properties of these matrices are proven. Sect. 4 focuses on the model of Metering Schemes. Sect. 5 discusses the known (threshold) solutions for Robust Metering. Section 6 shifts to the general case; two solutions are proposed and proved to be secure. Conclusions are presented in Sect. 7.

## 2 Preliminaries

### 2.1 Linear Secret Sharing Schemes

Denote the *participants* of the scheme by  $P_i$ ,  $1 \leq i \leq n$ , and the set of all *players* by  $\mathcal{P} = \{P_1, \dots, P_n\}$ . Denote the *dealer* of the scheme by  $\mathcal{D}$ . The role of the dealer is to share a secret  $s$  to all participants in the scheme. The simplest

access structure  $\Gamma$  is called  $(k, n)$ -threshold: all subsets of players  $\mathcal{P}$  with at least  $k + 1$  participants are *qualified* to reconstruct the secret and any subset of up to  $k$  players are *forbidden* of doing it. Accordingly we will call a Secret Sharing Scheme (SSS)  $(k, n)$ -threshold if the access structure  $\Gamma$  associated with it is  $(k, n)$ -threshold. It is well known that all threshold SSS protocols can be generalized for general access structures using Monotone Span Programs (see Cramer et al. [4]). Denote the set of all subsets of  $\mathcal{P}$  (i.e. the power set of  $\mathcal{P}$ ) by  $P(\mathcal{P})$ . The set of qualified groups is denoted by  $\Gamma$  and the set of forbidden groups by  $\Delta$ . The set  $\Gamma$  is called *monotone increasing* if for each set  $A$  in  $\Gamma$  each set containing  $A$  is also in  $\Gamma$ . Similarly,  $\Delta$  is called *monotone decreasing*, if for each set  $B$  in  $\Delta$  each subset of  $B$  is also in  $\Delta$ . The tuple  $(\Gamma, \Delta)$  is called an *access structure* if  $\Gamma \cap \Delta = \emptyset$ . If the union of  $\Gamma$  and  $\Delta$  is equal to  $P(\mathcal{P})$  (so,  $\Gamma$  is equal to  $\Delta^c$ , the complement of  $\Delta$ ), then we say that access structure  $(\Gamma, \Delta)$  is *complete* and we denote it just by  $\Gamma$ . The adversary is characterized by a particular subset  $\Delta_A$  of  $\Delta$ , which is itself monotone decreasing structure. The set  $\Delta_A$  ( $\Delta_A \subseteq \Delta$ ) is called an *adversary structure* while the set  $\Delta$  is called a *privacy structure*. The players which belong to  $\Delta$  are also called *curious* and the players which belong to  $\Delta_A$  are called *corrupt*. An  $(\Delta, \Delta_A)$ -adversary is an adversary who can (adaptively) corrupt some players passively and some players actively, as long as the set  $A$  of actively corrupt players and the set  $B$  of passively corrupt players satisfy both  $A \in \Delta_A$  and  $(A \cup B) \in \Delta$ . Now we give a formal definition of a Monotone Span Program.

**Definition 1.** A Monotone Span Program (MSP)  $\mathcal{M}$  is a quadruple  $(\mathbb{F}, M, \varepsilon, \psi)$ , where  $\mathbb{F}$  is a finite field,  $M$  is a matrix (with  $m$  rows and  $d \leq m$  columns) over  $\mathbb{F}$ ,  $\psi : \{1, \dots, m\} \rightarrow \{1, \dots, n\}$  is a surjective function and  $\varepsilon = (1, 0, \dots, 0)^T \in \mathbb{F}^d$  is called the target vector.

As  $\psi$  labels each row with an integer  $i$  from  $[1, \dots, m]$  that corresponds to player  $P_{\psi(i)}$ , we can think of each player as being the “owner” of one or more rows. Also consider a “function”  $\varphi$  from  $[1, \dots, n]$  to  $[1, \dots, m]$  which gives for every player  $P_i$  the set of rows owned by him (denoted by  $\varphi(P_i)$ ). In some sense  $\varphi$  is the “inverse” of  $\psi$ . Let  $M_A$  denote the restriction of  $M$  to the rows  $i$  with  $i \in A$ . An MSP is said to *compute* a (complete) access structure  $\Gamma$  when  $\varepsilon \in \text{im}(M_A^T)$  if and only if  $A$  is a member of  $\Gamma$ . We say that  $A$  is *accepted* by  $\mathcal{M}$  if and only if  $A \in \Gamma$ , otherwise we say  $A$  is *rejected* by  $\mathcal{M}$ . In other words, the players in  $A$  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.

Consider a vector  $\mathbf{v} \in \mathbb{F}^m$ . The coordinates in  $\mathbf{v}$ , which belong to player  $P_i$  are collected in a sub-vector denoted by  $\mathbf{v}^i$ , or in other words  $\mathbf{v} = (\mathbf{v}^1, \dots, \mathbf{v}^n)$  where  $\mathbf{v}^i \in \mathbb{F}^{|\varphi(P_i)|}$ . The *p-support* of vector  $\mathbf{v}$ , denoted by  $\text{sup}_P(\mathbf{v})$ , is defined as the set of coordinates  $i$ ,  $1 \leq i \leq n$  for which  $\mathbf{v}^i \neq \mathbf{0}$ , i.e.  $\text{sup}_P(\mathbf{v}) = \{i : \mathbf{v}^i \neq \mathbf{0}\}$ .

**Definition 2.** [11] An MSP is called  $\Delta$ -non-redundant (denoted by  $\Delta$ -rMSP) when  $v \neq 0 \in \ker(M^T) \iff \text{sup}_P(v) \in \Gamma$  ( $\Gamma = \Delta^c$ ).

For the sake of simplicity we will call an  $\Delta$ -rMSP simply an rMSP.

*Remark 1.* In [14] the authors consider (and prove the existence of) another class of MSPs called “monotone dependency program” (MDP). It can be easily verified that MDPs are even more restricted class of MSPs than rMSP. Thus the conjecture from [9] has a positive answer.

## 2.2 Multiplicative Linear SSSs

Cramer *et al.* proposed in [4] an approach to build a Multi-Party Computation (MPC) protocol from any Linear SSS introducing so-called (*strongly*) *multiplicative* LSSS. The construction for multiplicative MSPs was extended in [10] by proposing the diamond operation  $\diamond$ . Next we provide the definition and some basic properties of this operation.

Let  $\Gamma_1$  and  $\Gamma_2$  be two access structures, computed by MSPs  $\mathcal{M}_1 = (\mathbb{F}, M^{(1)}, \boldsymbol{\varepsilon}^1, \psi_1)$  and  $\mathcal{M}_2 = (\mathbb{F}, M^{(2)}, \boldsymbol{\varepsilon}^2, \psi_2)$ . Let  $M^{(1)}$  be an  $m_1 \times d_1$  matrix,  $M^{(2)}$  be an  $m_2 \times d_2$  matrix and let  $\varphi_1, \varphi_2$  be the “inverse” functions of  $\psi_1$  and  $\psi_2$ . Consider a vector  $\mathbf{x}$ . Let the coordinates in  $\mathbf{x}$ , which belong to the player  $P_j$ , form a sub-vector  $\mathbf{x}^j \in \mathbb{F}^{|\varphi(P_j)|}$  and let  $\mathbf{x} = (\mathbf{x}^1, \dots, \mathbf{x}^n)$ . Given an  $m_1$ -vector  $\mathbf{x}$  and an  $m_2$ -vector  $\mathbf{y}$ ,  $\mathbf{x} \diamond \mathbf{y}$  will denote the vector containing all entries of the form  $x_i y_j$ , where  $\psi_1(i) = \psi_2(j)$ . Thus the *diamond* operation  $\diamond$  for vectors can be defined as follows:

$$\mathbf{x} \diamond \mathbf{y} = (\mathbf{x}^1 \otimes \mathbf{y}^1, \dots, \mathbf{x}^n \otimes \mathbf{y}^n), \quad (1)$$

where  $\otimes$  is the usual tensor vector product. So,  $\mathbf{x} \diamond \mathbf{y}$  has  $m = \sum_{P_u \in \mathcal{P}} |\varphi_1(u)| |\varphi_2(u)|$  entries and notice that  $m < m_1 m_2$ . Let  $M_u^{(1)}$  denote the matrix formed by the rows of  $M^{(1)}$  owned by player  $P_u$ . Correspondingly, let  $M_u^{(2)}$  denote the matrix formed by the rows of  $M^{(2)}$  owned by player  $P_u$ . Then  $M_u^{(1)}$  is an  $|\varphi_1(u)| \times d_1$  matrix and  $M_u^{(2)}$  is an  $|\varphi_2(u)| \times d_2$  matrix. Now the diamond operation  $\diamond$  for matrices can be defined as follows

$$M^{(1)} = \begin{pmatrix} M_1^{(1)} \\ \dots \\ M_n^{(1)} \end{pmatrix}, \quad M^{(2)} = \begin{pmatrix} M_1^{(2)} \\ \dots \\ M_n^{(2)} \end{pmatrix}, \quad \text{and} \\ M^{(1)} \diamond M^{(2)} = \begin{pmatrix} M_1^{(1)} \otimes M_1^{(2)} \\ \dots \\ M_n^{(1)} \otimes M_n^{(2)} \end{pmatrix}. \quad (2)$$

In other words, the diamond operation  $\diamond$  for vectors (and analogously for matrices) is defined as the concatenation of vectors (matrices), which are the tensor ( $\otimes$ ) multiplication of the sub-vectors (sub-matrices) belonging to a fixed player.

*Remarks on the operation:* The process is analogous to the Kronecker product, with the *difference* that the tensor operation  $\otimes$  is replaced by the diamond operation  $\diamond$  for the columns. But note that the “symmetry” which the Kronecker product have between rows and columns is destroyed. This is illustrated by the following lemma.

**Lemma 1.** [10] Let  $M^{(1)}$  be an  $m_1 \times d_1$  matrix,  $M^{(2)}$  be an  $m_2 \times d_2$  matrix,  $N^{(1)}$  be an  $n_1 \times m_1$  matrix and an  $N^{(2)}$  be  $n_2 \times m_2$  matrix. Then

$$(N^{(1)} M^{(1)}) \diamond (N^{(2)} M^{(2)}) = (N^{(1)} \diamond N^{(2)})(M^{(1)} \otimes M^{(2)}).$$

The *diamond* operation  $\diamond$  for MSPs is defined as follows.

**Definition 3.** [10] Let MSPs  $\mathcal{M}_1 = (\mathbb{F}, M^{(1)}, \boldsymbol{\varepsilon}^1, \psi_1)$  and  $\mathcal{M}_2 = (\mathbb{F}, M^{(2)}, \boldsymbol{\varepsilon}^2, \psi_2)$ . Define an MSP  $\mathcal{M}_1 \diamond \mathcal{M}_2 = (\mathbb{F}, M^{(1)} \diamond M^{(2)}, \boldsymbol{\varepsilon}^1 \otimes \boldsymbol{\varepsilon}^2, \psi)$ , where  $\psi(i, j) = r$  if and only if  $\psi_1(i) = \psi_2(j) = r$ .

### 3 Operations on Doubly-Labelled Matrices

Now let us consider a matrix  $A$ , which rows are labelled by a function  $\psi$  and the columns are labelled by a function  $\bar{\psi}$ . Denote the sub-matrices labelled with  $\psi(i)$  and  $\bar{\psi}(j)$  by  $A_{i,j}$ . We call such a matrix *doubly-labelled*. Note that the block-diagonal matrix  $D$  from the Definition of multiplicative MSPs in [5] is in-fact doubly-labelled matrix with  $\psi = \bar{\psi}$ .

Let two matrices  $A$  and  $B$  be double-labelled by the functions  $\psi_1$  and  $\psi_2$  for the rows and by the functions  $\bar{\psi}_1$  and  $\bar{\psi}_2$  for the columns. Define  $A \boxtimes B$  to be a  $(\sum_i |\psi_1(i)| |\psi_2(i)|) \times (\sum_i |\bar{\psi}_1(i)| |\bar{\psi}_2(i)|)$  matrix consisting of sub-matrices  $A_{i,j} \otimes B_{i,j}$ :

$$A = \begin{pmatrix} A_{1,1} & \dots & A_{1,n} \\ \vdots & \dots & \vdots \\ A_{n,1} & \dots & A_{n,n} \end{pmatrix}, \quad B = \begin{pmatrix} B_{1,1} & \dots & B_{1,n} \\ \vdots & \dots & \vdots \\ B_{n,1} & \dots & B_{n,n} \end{pmatrix}, \quad \text{and}$$

$$A \boxtimes B = \begin{pmatrix} A_{1,1} \otimes B_{1,1} & \dots & A_{1,n} \otimes B_{1,n} \\ \vdots & \dots & \vdots \\ A_{n,1} \otimes B_{n,1} & \dots & A_{n,n} \otimes B_{n,n} \end{pmatrix}. \quad (3)$$

Note that any MSP can be seen as a double-labelled matrix and that the operation we just introduced is a generalization of the operation diamond. Let  $\mathcal{M}_1 = (\mathbb{F}, M^{(1)}, \boldsymbol{\varepsilon}^1, \psi_1)$  and  $\mathcal{M}_2 = (\mathbb{F}, M^{(2)}, \boldsymbol{\varepsilon}^2, \psi_2)$  be MSPs, then it is easy to verify that  $M^{(1)}(M^{(2)})^T$  is doubly-labelled matrix with labelled functions  $\psi_1$  for the rows and  $\psi_2$  for the columns. Now we prove the following properties of the new operation, which establish a relation between MSPs and diamond operation on the one hand and doubly-labelled matrices and the new operation on the other hand.

**Lemma 2.** Let MSPs  $\mathcal{M}_i = (\mathbb{F}, M^{(i)}, \boldsymbol{\varepsilon}^i, \psi_i)$  for  $i = 1, 2, 3, 4$  be such that  $M^{(i)}$  are  $m_i \times d_i$  matrices and let  $d_1 = d_3$ ,  $d_2 = d_4$ . Then the following equality holds

$$(M^{(1)} \diamond M^{(2)})(M^{(3)} \diamond M^{(4)})^T = (M^{(1)}(M^{(3)})^T) \boxtimes (M^{(2)}(M^{(4)})^T).$$

*Proof.* Let  $M^{(i)} = \begin{pmatrix} M_1^{(i)} \\ \dots \\ M_n^{(i)} \end{pmatrix}$ . Then we need to show that the following holds:

$$\begin{aligned} & \begin{pmatrix} M_1^{(1)} \otimes M_1^{(2)} \\ \dots \\ M_n^{(1)} \otimes M_n^{(2)} \end{pmatrix} \begin{pmatrix} M_1^{(3)} \otimes M_1^{(4)} \\ \dots \\ M_n^{(3)} \otimes M_n^{(4)} \end{pmatrix}^T \\ &= \begin{pmatrix} M_1^{(1)}(M_1^{(3)})^T \otimes M_1^{(2)}(M_1^{(4)})^T \dots M_1^{(1)}(M_n^{(3)})^T \otimes M_1^{(2)}(M_n^{(4)})^T \\ \vdots \\ M_n^{(1)}(M_1^{(3)})^T \otimes M_n^{(2)}(M_1^{(4)})^T \dots M_n^{(1)}(M_n^{(3)})^T \otimes M_n^{(2)}(M_n^{(4)})^T \end{pmatrix}. \end{aligned}$$

From the properties of the tensor (Kronecker) product we have

$$(M_i^{(1)} \otimes M_i^{(2)})(M_j^{(3)} \otimes M_j^{(4)})^T = (M_i^{(1)}(M_j^{(3)})^T) \otimes (M_i^{(2)}(M_j^{(4)})^T),$$

which concludes the proof.  $\square$

By applying Lemma 1 it is easy to verify that the following relation is satisfied.

**Lemma 3.** *Let MSPs  $\mathcal{M}_i = (\mathbb{F}, M^{(i)}, \varepsilon^i, \psi_i)$  for  $i = 1, 2, 3, 4$  be such that  $M^{(i)}$  are  $m_i \times d_i$  matrices. Let  $R^{(1)}$  be a  $d_1 \times d_3$  matrix and let  $R^{(2)}$  be a  $d_2 \times d_4$  matrix. Then the following equality holds*

$$(M^{(1)} \diamond M^{(2)})(R^{(1)} \otimes R^{(2)})(M^{(3)} \diamond M^{(4)})^T = (M^{(1)} R^{(1)} (M^{(3)})^T) \boxtimes (M^{(2)} R^{(2)} (M^{(4)})^T).$$

Note that  $M^{(1)} R^{(1)} (M^{(3)})^T$  is also a doubly-labelled matrix.

## 4 Metering Schemes - the Settings

The model of Metering Schemes has been proposed in [12] for threshold access; it has been extended to the general case in [3] and strengthened in [9]. We will follow the settings from [9], where the most general case is considered with a so-called *mixed adversary*.

Consider the following scenario: there are  $n$  clients,  $\tilde{n}$  servers and an audit agency  $\mathbb{A}$  which is interested in counting the client visits to the servers in  $\tau$  different time frames. For any  $i = 1, \dots, n$  and  $j = 1, \dots, \tilde{n}$ , denote the  $i$ -th client (user) by  $\mathcal{U}_i$  and the  $j$ -th server (player) by  $P_j$ . Consider an *access structure*  $\Gamma$  of qualified groups and its complement  $\Delta = \Gamma^c$  of forbidden groups for the set of client's  $\mathcal{U} = \{\mathcal{U}_1, \dots, \mathcal{U}_n\}$ . In a metering scheme realizing the clients access structure  $\Gamma$  any server which has been visited by at least a qualified subset of clients in  $\Gamma$  in a fixed time frame can provide the audit agency with a proof for the visits it has received. A second access structure  $\Gamma_S$  for the set of servers  $\{P_1, \dots, P_{\tilde{n}}\}$  can be considered. A subset of servers is called *corrupt* if they are not in  $\Gamma_S$ , i.e., if they are in  $\Delta_S = \Gamma_S^c$ . Denote the set of possible subsets of

corrupt clients by  $\Delta_A$ ; note that  $\Delta_A \subseteq \Delta$ , where  $\Delta$  is the set of curious clients. In the mixed adversary model a  $(\Delta_A, \Delta_S)$ -adversary is considered. 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 forward to a corrupt server all the private information received by the audit agency during the initialization phase. A corrupt server can forward to another corrupt server the private information received from clients in the previous time frames and in the actual time frame.

Several phases can be defined in the metering scheme.

- 1) There is an *initialization phase* in which the audit agency  $\mathbb{A}$  chooses the access structures, computes the corresponding matrices, makes them public and distributes some information to each client  $\mathcal{U}_i$  through a private channel. For any  $i = 1, \dots, n$  denote by  $V_i^{(t)}$  the shares that the audit agency  $\mathbb{A}$  gives to the client  $\mathcal{U}_i$  for time frames  $t = 1, \dots, \tau$ .
- 2) A *regular operation* consists of a client's 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  $i = 1, \dots, n$ ;  $j = 1, \dots, \tilde{n}$  and  $t = 1, \dots, \tau$ , denote by  $C_{i,j}^{(t)}$  the information that the client  $\mathcal{U}_i$  sends to the server  $P_j$  during the visit in time frame  $t$ .
- 3) During the *proof computation phase* any server  $P_j$  which has been visited by at least a subset of qualified clients in time frame  $t$  is able to compute its proof. For any  $j = 1, \dots, \tilde{n}$  and  $t = 1, \dots, \tau$  denote by  $p_j^{(t)}$  the proof computed by server  $P_j$  at time frame  $t$  when it has been visited by a qualified set of clients.
- 4) During the *proof verification phase* audit agency  $\mathbb{A}$  verifies the proofs received by the servers and decides on the amount of money to be paid to the servers. If the proof received from a server at the end of a time frame is correct, then  $\mathbb{A}$  pays the server for its services.

**Definition 4.** [9] An  $(n, \tilde{n}, \tau)$  metering scheme realizing the access structures  $\Gamma, \Gamma_S$  and secure against an  $(\Delta_A, \Delta_S)$ -adversary, is a protocol to measure the interaction between clients  $\mathcal{U}_1, \dots, \mathcal{U}_n$  with an access structure  $\Gamma$  and servers  $P_1, \dots, P_{\tilde{n}}$  with an access structure  $\Gamma_S$  during  $\tau$  time frames in such a way that the following properties are satisfied:

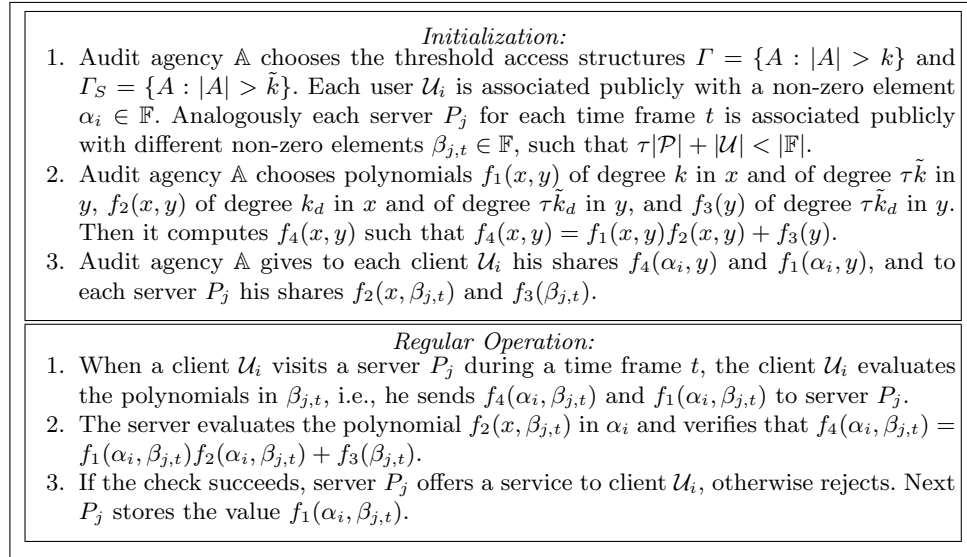
- For any time frame  $t$  any client is able to compute the information needed to visit any server.
- Correctness. For any time frame  $t$  any server  $P_j$  which has been visited by a qualified subset of clients  $A \in \Gamma$  in time frame  $t$  can compute its proof for time frame  $t$ .
- Privacy. Let  $B_2$  be a coalition of corrupt servers, i.e.,  $B_2 \in \Delta_S$  and let  $B_1$  be a coalition of corrupt clients, i.e.,  $B_1 \in \Delta_A$ . 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.,  $B_3 \in \Delta$ , such that we still have  $B_3 \cup B_1 \in \Delta$ . Then the

servers in coalition  $B_2$  have no information about their proofs for a time frame  $t$ , even if they are helped by the corrupt clients in  $B_1$ .

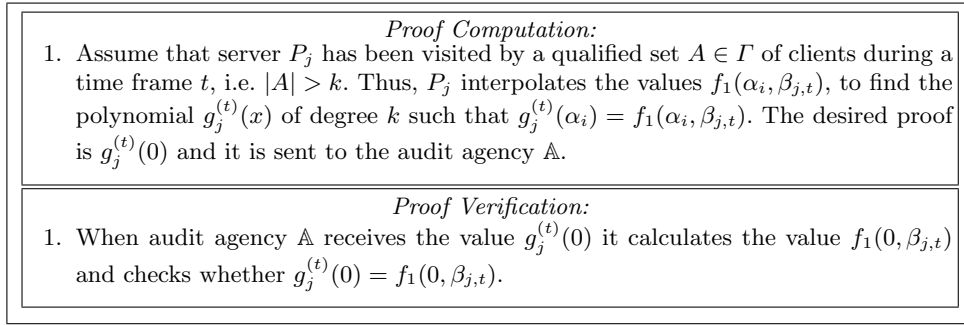
- Stronger Privacy. Let  $B_2$  be a coalition of corrupt servers, i.e.,  $B_2 \in \Delta_S$  and let  $B_1$  be a coalition of corrupt clients, i.e.,  $B_1 \in \Delta_A$ . Assume that in some time frame  $t$  a fixed server in the coalition (e.g.  $P_j \in B_2$ ) has been visited by a subset of forbidden clients  $B_3$ , i.e.,  $B_3 \in \Delta$ , and  $B_3 \cup B_1 \in \Delta$ . Assume that in the same time frame  $t$  any other server in the coalition  $B_2$  has been visited by a subset of qualified clients  $B_4$ . Then the servers in the coalition  $B_2 \setminus \{P_j\}$  are able to compute their proofs for a time frame  $t$ , but they are unable to “help” the server  $P_j$  with the computation of its proofs, even if they are helped by the corrupt clients in  $B_1$ .

## 5 Robust Metering Schemes - the Threshold Case

As noted before the basic model assumes that the clients do not present incorrect evidence to a server, thus preventing the server from constructing its proof. In order to prevent such a behavior robust Metering Schemes were proposed in [12, 13]. For the sake of completeness we start with the robust metering scheme proposed by Naor and Pinkas [12].



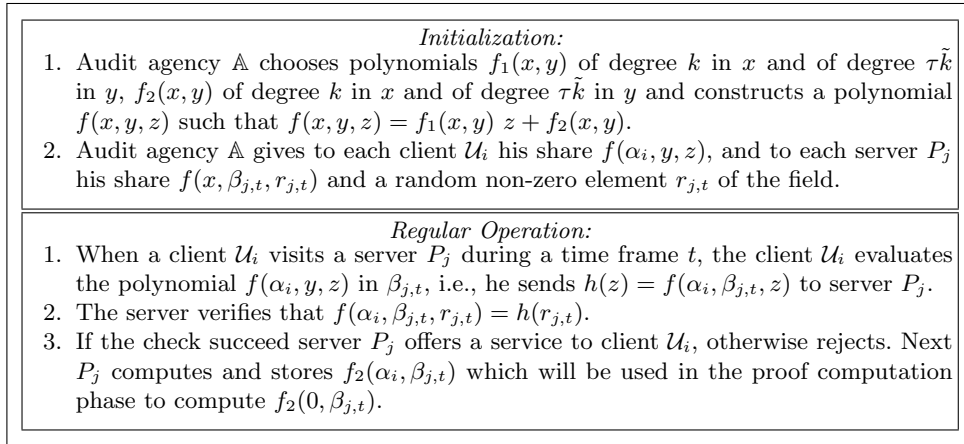
**Fig. 1.** Robust Metering Scheme - threshold case [12]



**Fig. 2.** Robust Metering Scheme - threshold case [12] (cont.)

Note that  $g_j^{(t)}(\alpha_i) = f_1(\alpha_i, \beta_{j,t})$ , thus  $g_j^{(t)}(x) = f_1(x, \beta_{j,t})$ . Hence the proof is  $g_j^{(t)}(0) = f_1(0, \beta_{j,t})$ , which proves the correctness of the scheme. But this scheme is subject to the following attack [13]. Assume that for a server  $P_j$  and for some time frame  $t$  there exists two clients  $\mathcal{U}_{i_1}$  and  $\mathcal{U}_{i_2}$  such that  $f_1(\alpha_{i_1}, \beta_{j,t}) = 0$  and  $f_1(\alpha_{i_2}, \beta_{j,t}) \neq 0$ . Then they can compute  $f_3(\beta_{j,t}) = f_4(\alpha_{i_1}, \beta_{j,t})$  and hence  $f_2(\alpha_{i_2}, \beta_{j,t}) = \frac{f_4(\alpha_{i_2}, \beta_{j,t}) - f_3(\beta_{j,t})}{f_1(\alpha_{i_2}, \beta_{j,t})}$ . Next client  $\mathcal{U}_{i_2}$  computes a random  $\bar{f}_1$  and  $\bar{f}_4$  such that  $\bar{f}_4 = \bar{f}_1 f_2(\alpha_{i_2}, \beta_{j,t}) - f_3(\beta_{j,t})$  and  $\bar{f}_1 \neq f_1(\alpha_{i_2}, \beta_{j,t})$ . Finally, client  $\mathcal{U}_{i_2}$  can fool server  $P_j$  at time frame  $t$  to get a service without providing correct information. The authors of [13] propose the following modification (see Fig. 3).

The server security relies on the well known fact that for any two field elements  $a$  and  $c$  there are  $|\mathbb{F}|$  tuples  $(b, d)$  such that  $d = a b + c$ .



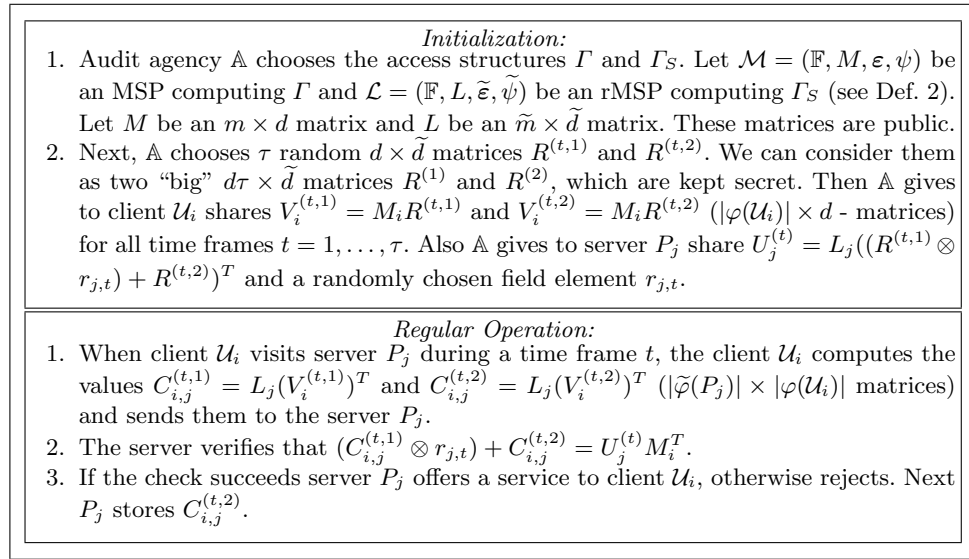
**Fig. 3.** Robust Metering Scheme - the threshold case [13]

We propose another way to fix the metering scheme from Fig. 1. The key for the attack proposed by Ogata and Kurosawa is that the function  $f_3(y)$  does not depend on  $x$ , recall that  $f_4(x, y) = f_1(x, y)f_2(x, y) + f_3(y)$ . Thus the same value  $f_3(\beta_{j,t})$  is used for verification of the shares of all players, which allows the attackers to compute it and then to mount an attack. Hence a way to avoid the described attack is to replace  $f_3(y)$  with a polynomial  $f_3(x, y)$  of degree  $k_d$  in  $x$ . The rest of the protocol described in Fig. 1 and 2 stays the same.

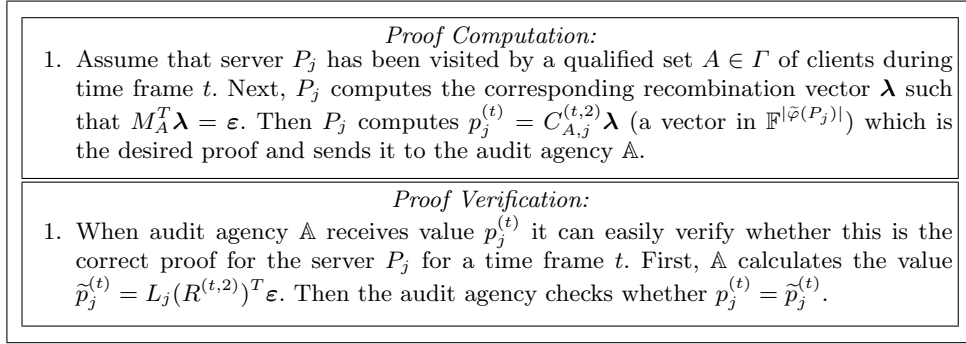
Note that in this model the clients are not protected against corrupt servers which deny to offer their services. One way to protect the clients is to allow them to complain to the audit agency against the server's behavior: if after Step 3 in the Regular Operation Phase, the server  $P_j$  denies service to client  $\mathcal{U}_i$  the latter will broadcast  $(h(z), \mathcal{U}_i, P_j, t)$  as an accusation against the server. Note that if the client is honest then the server is corrupt or the information is broadcasted by a corrupt client. The audit agency rejects payment to server  $P_j$  if in a given time frame  $t$  there is a qualified set of clients which complains (with correct information  $h(z)$ ) against  $P_j$ .

## 6 Robust Metering Schemes - the General Case

In this section we propose two solutions for constructing robust unconditionally secure metering scheme in the general access structure case. We will extend the MSP based approach from [9] to achieve robustness. First, we will generalize the scheme proposed in [13].



**Fig. 4.** Robust Metering Scheme - the general case I



**Fig. 5.** Robust Metering Scheme - the general case I (cont.)

Assume that server  $P_j$  has been visited by a qualified set of clients  $A \in \Gamma$  during time frame  $t$ . Then

$$\begin{aligned} p_j^{(t)} &= C_{A,j}^{(t,2)} \lambda = L_j(V_A^{(t,2)})^T \lambda = L_j(M_A R^{(t,2)})^T \lambda \\ &= L_j(R^{(t,2)})^T M_A^T \lambda = L_j(R^{(t,2)})^T \varepsilon = \tilde{p}_j^{(t)}. \end{aligned}$$

**Theorem 1.** *The metering scheme described in Fig. 4 is robust and perfectly secure against a  $(\Delta_A, \Delta_S)$ -adversary.*

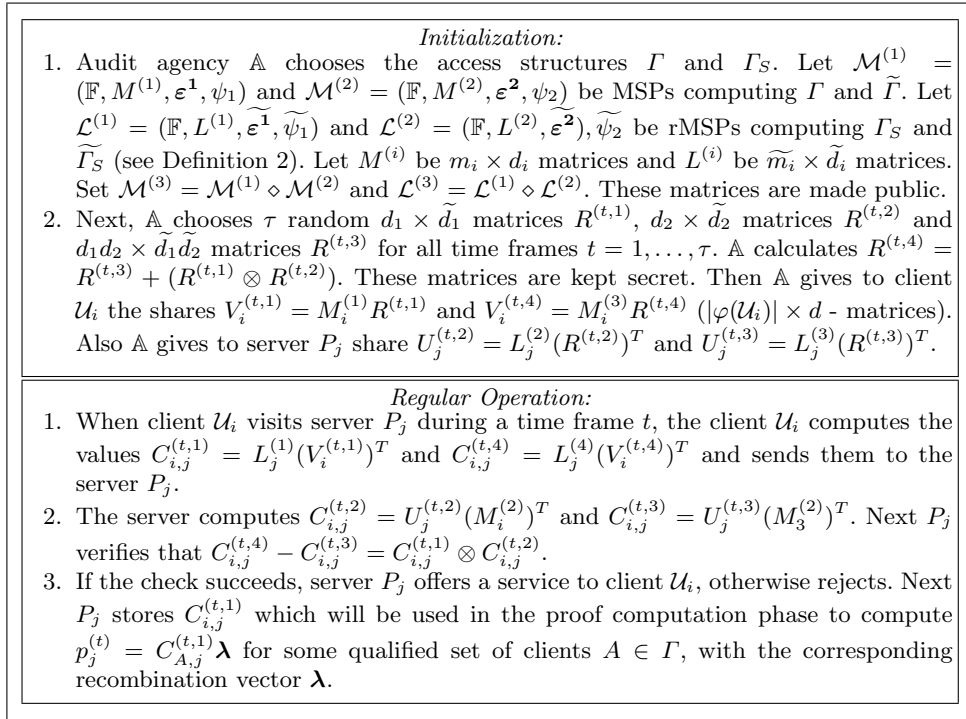
*Proof.* The correctness and (strong) privacy follows directly from the results in [9]. What we need to prove in addition is that the scheme is *robust*, i.e., the server security.

Using the properties of Kronecker (tensor) product it is easy to check that the following relations hold.

$$\begin{aligned} U_j^{(t)} M_i^T &= L_j((R^{(t,1)} \otimes r_{j,t}) + R^{(t,2)})^T M_i^T = L_j(M_i((R^{(t,1)} \otimes r_{j,t}) + R^{(t,2)}))^T \\ &= L_j(M_i(R^{(t,1)} \otimes r_{j,t}) + M_i R^{(t,2)})^T \\ &= L_j((M_i \otimes 1)(R^{(t,1)} \otimes r_{j,t}) + M_i R^{(t,2)})^T \\ &= L_j((M_i R^{(t,1)}) \otimes (1 r_{j,t}) + M_i R^{(t,2)})^T = L_j((V_i^{(t,1)} \otimes r_{j,t}) + V_i^{(t,2)})^T \\ &= L_j(((V_i^{(t,1)})^T \otimes r_{j,t}) + (V_i^{(t,2)})^T) \\ &= ((L_j(V_i^{(t,1)})^T) \otimes r_{j,t}) + L_j(V_i^{(t,2)})^T = (C_{i,j}^{(t,1)} \otimes r_{j,t}) + C_{i,j}^{(t,2)}. \end{aligned}$$

Thus we prove the correctness of the server verification. The robustness of the scheme follows from the arguments of Ogata and Kurosawa [13].  $\square$

Our second proposition is a generalization of the method we proposed in the previous section.



**Fig. 6.** Robust Metering Scheme - the general case II

**Theorem 2.** *The metering scheme described in Fig. 6 is robust and perfectly secure against a  $(\Delta_A, \Delta_S)$ -adversary.*

*Proof.* Again we need to prove only that the scheme is *robust*. First note that

$$C_{i,j}^{(t,z)} = L_j^{(z)} (V_i^{(t,z)})^T = L_j^{(z)} (M_i^{(z)} R^{(t,z)})^T = L_j^{(z)} (R^{(t,z)})^T (M_i^{(z)})^T = U_j^{(t,z)} (M_i^{(z)})^T.$$

Thus the verification check  $C_{i,j}^{(t,4)} - C_{i,j}^{(t,3)} = C_{i,j}^{(t,1)} \otimes C_{i,j}^{(t,2)}$  actually relies on the equations

$$\begin{aligned} L^{(3)} (R^{(t,4)} - R^{(t,3)})^T (M^{(3)})^T &= (L^{(1)} \diamond L^{(2)}) (R^{(t,4)} - R^{(t,3)})^T (M^{(1)} \diamond M^{(2)})^T \\ &= (L^{(1)} \diamond L^{(2)}) (R^{(t,1)} \otimes R^{(t,2)})^T (M^{(1)} \diamond M^{(2)})^T \\ &= (L^{(1)} (R^{(t,1)})^T (M^{(1)})^T) \boxtimes (L^{(2)} (R^{(t,2)})^T (M^{(2)})^T), \end{aligned}$$

which are satisfied due to Lemma 3. This proves the correctness of the server verification.  $\square$

## 7 Conclusions

Note that metering schemes can be considered as *two-level* SSSs. In an SSS the players which get the shares from the dealer reconstruct the secret themselves. In two-level structures the players in the first level get the shares from the dealer and later provide some information to the players in the second level who compute certain value related to the secret.

In this paper we have revisited the robust threshold metering schemes from [12, 13]; we have proposed two solutions to build *robust* general metering schemes based on the linear algebra approach. As a tool to achieve this goal we have introduced *doubly-labelled* matrices and an operation on such matrices. We have established a relation between MSPs and the diamond operation on the one hand and doubly-labelled matrices and the new operation on the other hand. Since MSPs and doubly-labelled matrices are used for multiplicative linear SSSs [4, 5] their relations are of independent interest. Finally we have demonstrated that one can protect clients against denied of service attacks of corrupt servers.

## References

1. C. Blundo, A. De Bonis, B. Masucci. Metering Schemes with Pricing, *DISC'00*, LNCS 1914, 2000, pp. 194-208.
2. C. Blundo, A. De Bonis, B. Masucci, D. Stinson. Dynamic Multi-Threshold Metering Schemes, *SAC'00*, LNCS 2012, 2001, pp. 130-144.
3. C. Blundo, S. Martin, B. Masucci, C. Padro. A Linear Algebraic Approach to Metering Schemes, *Cryptology ePrint Archive: Report 2001/087*.
4. R. Cramer, I. Damgard, U. Maurer. General Secure Multi-Party Computation from any Linear Secret Sharing Scheme, *EUROCRYPT'00*, LNCS 1807, 2000, 316-334.
5. R. Cramer, S. Fehr, Y. Ishai, E. Kushilevitz. Efficient Multi-Party Computation over Rings, *EUROCRYPT'03*, LNCS 2656, 2003, pp. 596-613.
6. M. K. Franklin, D. Malkhi. Auditable Metering with Lightweight Security, *Financial Cryptography'97*, LNCS 1318, 1997, pp. 151-160.
7. B. Masucci, D. Stinson. Metering Schemes for General Access Structures, *ESORICS'00*, LNCS 1895, 2000, pp. 72-87.
8. B. Masucci, D. Stinson. Efficient Metering Schemes with Pricing, *IEEE Transactions on Information Theory*, 47 (7), 2001, pp. 2835-2844.
9. V. Nikov, S. Nikova, B. Preneel, J. Vandewalle. Applying General Access Structure to Metering Schemes, *WCC'03*, 2003, *Cryptology ePrint Archive: Report 2002/102*.
10. 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*.
11. V. Nikov, S. Nikova. On a relation between Verifiable Secret Sharing Schemes and a class of Error-Correcting Codes, *Cryptology ePrint Archive: Report 2003/210*.
12. M. Naor, B. Pinkas. Secure and Efficient Metering, *EUROCRYPT'98*, LNCS 1403, 1998, pp. 576-590.
13. W. Ogata, K. Kurosawa. Provably Secure Metering Scheme, *ASIACRYPT'00*, LNCS 1976, 2000, pp. 388-398.
14. P. Pudlak, J.Sgall. Algebraic models of computation and interpolation for algebraic proof systems, *Proc. Feasible Arithmetic and Proof Complexity*, LNCS, 1998, pp. 279-295.