

# Operations on Access Structures

## (Corrected Version)

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

<sup>1</sup> Department of Mathematics and Informatics,  
Veliko Tarnovo University,  
5000 Veliko Tarnovo, Bulgaria  
vnikov@mail.com

<sup>2</sup> 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

### Changes

S. Fehr [2] pointed out that denoting the complement  $\Gamma_A = \Delta_A^c$  [4] as honest (or good) players structure appears to be a misleading term. Actually its dual access structure  $\Gamma_A^\perp$  should be called the honest (or good) players structure, since for any set  $G$  of good players the complement  $G^c$  is the set of corrupted players from  $\Delta_A$ . This reflects in some changes of the notations in Theorem 7 and Theorem 8 from [4].

**Abstract.** Recently Hirt and Maurer [3] introduced the notion of  $Q^2(Q^3)$  adversary structure. In this paper we generalize this notion by defining new operation on access structures and we describe some applications of the new definition. There is one-to-one correspondence between access structures and monotone Boolean functions, defined on bipartite graphs.

## 1 Introduction

A *secret sharing scheme* (SSS) protects a secret (key) by distributing related information among a group of participants. This is done in such a way that only certain pre-specified groups of these participants (the *access structure*) may reconstruct the secret.

In the “classic” secret sharing schemes, there is assumed to have no faults in the system. Loosely speaking, in a *verifiable secret sharing scheme* (VSS) [1] a dealer shares a secret among a set of participants in such a way that each participant

---

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

can verify if the shares he gets, are consistent with the secret. In other words a malicious adversary is allowed to *corrupt* the dealer and some subset of the participants. All other players are *honest* and the adversary in VSS is *static*. *Proactive* security for secret sharing was first suggested by Ostrovski and Yung in [5]. Proactive security refers to security and availability in the presence of a *mobile* adversary.

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 – *forbidden*. The collection of all qualified groups is denoted by  $\Gamma$ , and the collection of all forbidden groups is denoted by  $\Delta$ . In fact  $\Gamma$  is *monotone increasing*, and  $\Delta$  is *monotone decreasing*. The tuple  $(\Gamma, \Delta)$  is called 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  $(\Gamma, \Delta)$  is *complete* and we denote it by  $\Gamma$ . Otherwise we say that  $(\Gamma, \Delta)$  is *incomplete*. There exists an adversary  $A$  who can *corrupt* a set of servers during any time period. Corrupting a server means learning the secret information in the server, modifying its data, sending out a wrong message, and so on. Since the server can be rebooted, the adversary is a mobile one. The collection of all possible corrupted servers for a fixed time period we call *bad* and we denote it by  $\Delta_A$ . The collection of all possible uncorrupted servers for the same period of time we call *good* and we denote it by  $\Gamma_A^\perp$ . So, we can consider a second complete access structure  $\Gamma_A^\perp$ , which is called an *adversary access structure* [3].

The contribution of this paper is twofold. First, we introduce a new operation for the access structures which extends the notion of  $Q^2(Q^3)$  adversary structure introduced by Hirt and Maurer [3]. This operation characterizes which adversary structure can be tolerated. Second, this operation allows us to study how the participants and the adversary structures are linked.

## 2 New operation on access structures

Recently Hirt and Maurer [3] introduced the notion of  $Q^2(Q^3)$  adversary structure.

**Definition 1** [3] *For a given set of players  $P$  and an adversary structure  $\Delta_A$ , we say that the adversary structure is  $Q^2(P, \Delta_A)$  if no two sets in  $\Delta_A$  cover the full set  $P$  of players, and correspondingly we say that the adversary structure is  $Q^3(P, \Delta_A)$  if no three sets in  $\Delta_A$  cover the full set  $P$  of players.*

Using the notion of  $Q^2(Q^3)$  adversary structure Hirt and Maurer characterize exactly which adversary structures can be tolerated. But Hirt and Maurer do not consider the interaction between the adversary structure and the usual access structure. In [4] we generalize the  $Q^2(Q^3)$  notion, by introducing a new operation on access structure.

**Definition 2** [4] *For the access structure  $(\Gamma, \Delta)$  we define the operation  $*$  as follows:  $n * \Delta = \{A = A_1 \cup A_2; A_1 \in (n - 1) * \Delta, A_2 \in \Delta\}$ , for  $n = 2, 3, \dots$*

Let us consider the tuples  $(\Gamma, \Delta), (\Gamma, 2 * \Delta), \dots, (\Gamma, n * \Delta)$ . They are access structures if and only if  $\Gamma \cap n * \Delta = \emptyset$ , because  $n * \Delta$  is monotone decreasing.

**Definition 3** [4] For the complete access structure  $\Gamma$  we define the operation  $*$  as follows: First we set  $\Delta = 2^P \setminus \Gamma$  and (as in Definition 2) calculate  $n * \Delta$ . Then we define  $n * \Gamma = 2^P \setminus n * \Delta$ , for  $n = 2, 3, \dots$

Now we can consider the sequence  $\Gamma, 2 * \Gamma, \dots, n * \Gamma$  of access structures if and only if  $n * \Gamma \neq \emptyset$ , i.e. if  $n * \Gamma$  is non-trivial, because  $n * \Gamma$  is monotone increasing.

**Lemma 4** Let  $\Gamma$  be a complete access structure, then  $n * \Gamma \neq \emptyset$  for every  $n$  if and only if there exists a  $P_i \in P$  such that  $\{P_i\} \notin \Delta$  (so  $\{P_i\} \in \Gamma$ ).

Now we are ready to define the adversary power  $p_A$  as follows.

**Definition 5** Let  $\Gamma_A$  be a complete access structure. We define the adversary power  $p_A = \max\{n : n * \Gamma_A \neq \emptyset\}$

Since the logic of Secret Sharing Schemes is the secret to be shared, i.e. no one can reveal the secret on his own, the case when  $\{P_i\} \in \Gamma$  for some  $P_i$  is not of practical interest. For completeness it is easy to see that for such a SSS the adversary power  $p_A = \infty$ .

**Theorem 6** Let  $\Gamma_A$  be a complete adversary access structure so  $\Gamma_A$  is  $Q^2(Q^3)$ , if and only if  $p_A \geq 2$  ( $p_A \geq 3$ ).

Now we present some applications of the definitions given above. One can prove that for VSS the new definition gives rise to the following theorem.

**Theorem 7** [4] There exists an unconditionally secure verifiable secret sharing scheme if the following condition is satisfied:  $(2 * \Gamma_A)^\perp \subseteq \Gamma$ .

Correspondingly for Proactive SSS in [4] we prove the following theorem.

**Theorem 8** [4] There exists an unconditionally secure proactive secret sharing scheme if the following conditions are satisfied:

- a)  $(3 * \Gamma_A)^\perp \subseteq \Gamma$ .
- b) For each group  $N \in \Gamma^-$  the number of rows for this group of matrix  $M$  is equal to the number of columns of matrix  $M$ .

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  $\tau$  different time frames. For any  $i = 1, \dots, n$  and  $j = 1, \dots, k$ , we denote by  $\mathcal{C}_i$  the  $i$ -th

client and by  $S_j$  the  $j$ -th server. Such kinds of schemes are called *metering schemes*.

We consider an access structure  $(\Gamma, \Delta)$  of qualified and forbidden groups for the set of clients  $\{\mathcal{C}_1, \dots, \mathcal{C}_n\}$ . A second access structure (complete)  $\Gamma_S$  can be considered for the set of servers  $\{S_1, \dots, S_k\}$ . We call the servers *corrupt* if they are not in  $\Gamma_S$ . We denote also the set of possible subsets of *corrupt clients* by  $\Delta_C$ , which is in fact monotone decreasing. It is obvious requirement that  $\Gamma \cap \Delta_C = \emptyset$ . To deal with this scenario we enhance Definition 2 in the following way.

**Definition 9** For the access structure  $(\Gamma, \Delta)$  and monotone decreasing set  $\Delta_C$  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  $(n, k, \tau)$  metering scheme realizing the access structures  $(\Gamma, \Delta)$ ,  $\Gamma_S$  and tolerate corrupted set of clients  $\Delta_C$ , we consider the tuple  $(\Gamma, \Delta * \Delta_C)$ . We can prove the following lemma.

**Lemma 10** An  $(n, k, \tau)$  metering scheme realizing the access structures  $(\Gamma, \Delta)$ ,  $\Gamma_S$  and tolerate corrupt set of clients  $\Delta_C$  exists, if and only if  $(\Gamma, \Delta * \Delta_C)$  is an access structure (i.e.  $\Gamma \cap \Delta * \Delta_C = \emptyset$ ).

## References

1. **B. Chor, S. Goldwasser, S. Micali, 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, 383-395.
2. **S. Fehr, V. Nikov, S. Nikova**, private communication.
3. **M. Hirt, U. Maurer**, Player Simulation and General Adversary Structures in Perfect Multiparty Computation, *J. of Cryptology* 13, 2000, 31-60.
4. **V. Nikov, S. Nikova, B. Preneel, J. Vandewalle**, Applying General Access Structure to Proactive Secret Sharing Schemes, *Proc. of the 23rd Symposium on Inf. Theory in the Benelux*, May 29-31, 2002, UCL, Lovain-la-Neuve, Belgium, pp.197-206.
5. **R. Ostrovsky, M. Yung**, How to withstand mobile virus attack, ACM Symposium on principles of distributed computing, 1991, 51-59.