Bounds and Constructions for Unconditionally Secure Distributed Key Distribution Schemes for General Access Structures*

Carlo Blundo[1], Paolo D'Arco[1], Vanessa Daza[2], and Carles Padró[2]

[1] Dipartimento di Informatica ed Applicazioni
Università di Salerno, 84081 Baronissi (SA), Italy
e-mail: {carblu, paodar}@dia.unisa.it
[2] Departament de Matemàtica Aplicada IV
Universitat Politècnica de Catalunya, Barcelona, Spain
e-mail: {vdaza, matcpl}@mat.upc.es

April 22, 2002

Abstract

In this paper we investigate the issues concerning with the use of a single server across a network, the *Key Distribution Center*, to enable private communications within groups of users. After providing several motivations, showing the advantages related to the *distribution* of the task accomplished by this server, we describe a model for such a distribution, and present bounds on the amount of resources required in a real-world implementation: random bits, memory storage, and messages to be exchanged. Moreover, we introduce a linear algebraic approach to design optimal schemes distributing a Key Distribution Center and we point out that some previous constructions belong to the proposed framework.

Keywords: Key Distribution, Protocols, Distributed Systems.

1 Introduction

Private communications over insecure channels can be carried out using encryption algorithms. If a public key infrastructure is available, public key algorithms can be employed. However, in this setting, if a user wishes to send the same message to n different users, he has to compute n encryptions of the message using n different public keys, and he has to send the message to each of them. Moreover, public key encryption and decryption are slow operations and, when the communication involves a group of users, hereafter referred to as a *conference*, this communication strategy is completely inefficient from a computational and communication point of view as well.

An improvement on the "trivial" use of public key algorithms can be the *hybrid* approach: a user chooses at random a key and sends it, in encrypted form (public key), to all the other members

*The work of the first and second authors was partially supported by Italian *Ministero dell'Istruzione, dell'Università e della Ricerca* in the framework of the project "Azioni Integrate Italia-Spagna". The work of the third and the fourth authors was partially supported by Spanish *Ministerio de Ciencia y Tecnología* under project TIC 2000-1044.

of the conference. Then, they can privately communicate using a symmetric algorithm. Indeed, symmetric encryption algorithms are a few orders of magnitude more efficient than public key ones. Triple-DES, RC6, and RIJNDAEL, for example, are fast algorithms, spreadly used, and supposed to be secure. Besides, if a broadcast channel is available, a message for different recipients needs to be sent just once. Hence, better performances can be achieved with symmetric algorithms.

However, the hybrid protocol described before is still not efficient, and it is possible to do better. Actually, the question is how can be set up an *efficient* protocol to provide a common key to each conference.

A common solution is the use of a Key Distribution Center (KDC, for short), a server responsible of the distribution and management of the secret keys. The idea is the following. Each user shares a common key with the center. When he wants to securely communicate with other users, he sends a request for a conference key. The center checks for membership of the user in that conference, and distributes in encrypted form the conference key to each member of the group. Needham and Schroeder [31] began this approach, implemented most notably in the Kerberos System [32], and formally defined and studied in [3], where it is referred to as the *three party model*.

The scheme implemented by the Key Distribution Center to give each conference a key is called a *Key Distribution Scheme* (KDS, for short). The scheme is said to be *unconditionally secure* if its security is independent from the computational resources of the adversaries.

Several kinds of Key Distribution Schemes have been considered so far: Key Pre-Distribution Schemes (KPSs, for short), Key Agreement Schemes (KASs, for short) and Broadcast Encryption Schemes (BESs, for short) among others. The notions of KPS and KAS are very close to each other [6, 29, 10]. BESs are designed to enable secure broadcast transmissions and have been introduced in [23]. The broadcast encryption idea has grown in various directions: traitor tracing [21], anonymous broadcast transmission [26], re-keying protocols for secure multi-cast communications [18, 20, 34].

Our attention in this paper focusses on a model improving upon the weaknesses of a *single KDC*. Indeed, in the network model outlined before, a KDC must be *trusted*; moreover, it could become a communication *bottleneck* since all key request messages are sent to it and, last but not least, it could become a point of failure for the system: if the server crashes, secure communications cannot be supported anymore.

In [30] a new approach to key distribution was introduced to solve the above problems. A Distributed Key Distribution Center (DKDC, for short) is a set of n servers of a network that jointly realizes the same function of a Key Distribution Center. A user who needs to participate to a conference, sends a key-request to a subset at his choice of the n servers. The contacted servers answer with some information enabling the user to compute the conference key. In such a model, a single server by itself does not know the secret keys, since they are *shared* between the n servers, the communication bottleneck is eliminated, since the key-request messages are distributed, on average, along different paths, and there is no single point of failure, since if a server crashes, the other are still able to support conference key computation.

In subsequent papers [17, 7], the notion of DKDC has been studied from an information theoretic point of view. Therein, the authors introduced the concept of a distributed key distribution scheme (DKDS, for short), a scheme realizing a DKDC, showing that the protocol proposed in [30], based on ℓ-wise independent functions, is optimal with respect to the amount of information needed to set up and manage the system.

In [17, 7], a *threshold access structure* was considered on the set of servers, that is, the subsets of servers authorized to help the users in recovering the conference keys were determined in terms of their cardinality. In this paper, we extend the model studied in [17, 7] by considering a general *access structure* on the set of servers, that is, we consider an arbitrary family of qualified subsets of servers. Any user, in order to recover a conference key, has to contact all the server in any set belonging to the access structure.

We present bounds holding on the model using a reduction technique which relates DKDSs to Secret Sharing Schemes [5, 35]. This technique enables us to prove lower bounds on the memory storage, on the communication complexity and on the randomness needed to set up the scheme in an easy and elegant way. Moreover we describe a linear algebraic approach to design DKDS using a linear secret sharing scheme and a family of linear ℓ-wise independent forms. The optimality of the obtained constructions relies on the optimality of the secret sharing scheme used as building block. Finally, we emphasize the suitability of this approach that allows a unified description of seemingly different schemes, pointing out that some previous constructions can be seen as instances of the proposed framework.

Organization of the paper. A short overview of secret sharing schemes is given in Section 2, where basic definitions and results are recalled. A model for distributed key distribution schemes and the notation we use in the paper are given in Section 3. Some lower bounds on the amount of information stored by the servers and sent to reply to the key-request messages, and on the number of random bits required to set up the scheme are given in Section 4. In Sections 5 a linear algebraic method to construct DKDSs from any linear secret sharing scheme is described, and some examples are presented in Section 6. Finally, Section 7 is devoted to conclusions and some open problems.

B

2 Secret Sharing Schemes

A secret sharing scheme is a method by means of which a secret can be shared among a set $\mathcal{P}$ of n participants in such a way that qualified subsets of $\mathcal{P}$ can recover the secret, but any non-qualified subset has absolutely no information. Secret sharing were introduced in 1979 by Blakley [5] and Shamir [35]. The reader can find an excellent introduction in [39]. The collection of subsets of participants qualified to reconstruct the secret is usually referred to as the *access structure* of the secret sharing scheme. Formally, we have:

Definition 2.1 *Let $\mathcal{P}$ be a set of participants, a* monotone *access structure $\mathcal{A}$ on $\mathcal{P}$ is a subset $\mathcal{A} \subseteq 2^{\mathcal{P}} \setminus \{\emptyset\}$, such that*

$$A \in \mathcal{A}, A \subseteq A' \subseteq \mathcal{P} \Rightarrow A' \in \mathcal{A}.$$

Since, as we will see later, the reconstruction property of a secret sharing scheme *naturally induces* the monotonicity property, all access structures we are going to consider are monotone.

For any participant $P \in \mathcal{P}$, let us denote by $K(P)$ the set of all possible shares given to participant P. Suppose a *dealer* D wishes to a share the secret $s \in S$ among the participants in $\mathcal{P}$ (we shall assume that $D \notin \mathcal{P}$). To this aim, he gives to each participant $P \in \mathcal{P}$ a share from $K(P)$, chosen according to some (non necessarily uniform) probability distribution. Given a set of participants $A = \{P_{i_1}, \ldots, P_{i_r}\} \subseteq \mathcal{P}$, where $i_1 < \ldots < i_r$, denote by $K(A) = K(P_{i_1}) \times \cdots \times K(P_{i_r})$.

Any secret sharing scheme for secrets in S and a probability distribution $\{p_S(s)\}_{s \in S}$ naturally induce a probability distribution on $K(A)$, for any $A \subseteq \mathcal{P}$. Denote such probability distribution by $\{p_{K(A)}(a)\}_{a \in K(A)}$. To avoid overburdening the notation, with the same symbol A we will denote both a subset of participants and the random variable taking values in $K(A)$ according to the probability distribution $\{p_{K(A)}(a)\}_{a \in K(A)}$; analogously, with S we will denote both the set of secrets and the random variable taking values in S according to $\{p_S(s)\}_{s \in S}$. For any $s \in S$ and $a \in K(A)$ with $p_{K(A)}(a) > 0$ denote by $p(s|a)$ the probability that the secret is equal to s given that the shares held by participants in A are equal to a. In terms of Shannon's entropy[1], we say that a secret sharing scheme is a *perfect* secret sharing scheme with secrets chosen in S, or simply a secret sharing scheme with secrets chosen in S, for the monotone access structure $\mathcal{A} \subseteq 2^{\mathcal{P}}$ if

1. *Any subset $A \subseteq \mathcal{P}$ of participants enabled to recover the secret can compute the secret:* Formally, for all $A \in \mathcal{A}$, it holds that $H(\mathbf{S}|\mathbf{A}) = 0$.

[1] The reader is referred to the Appendix A for the definition of the entropy function and some basic properties.

2. *Any subset $A \subseteq \mathcal{P}$ of participants not enabled to recover the secret has no information on the secret value:*
Formally, for all $A \notin \mathcal{A}$, it holds that $H(\mathbf{S}|\mathbf{A}) = H(\mathbf{S})$.

Property 1 means that the value of the shares held by $A \in \mathcal{A}$ completely determines the secret $s \in S$. On the other hand, Property 2 means that the probability that the secret is equal to s given that the shares held by $A \notin \mathcal{A}$ are a, is the same as the *a priori* probability of the secret s.

The efficiency of a secret sharing scheme is measured by means of an "information rate", which relates the size of the secret with the size of the shares given to the participants. More precisely, given a secret sharing scheme Σ for the access structure $\mathcal{A}$, on the set of secrets S, we define the information rate $\rho(\Sigma, \mathcal{A}, S)$ as

$$\rho(\Sigma, \mathcal{A}, S) = \frac{\log |S|}{\max_{P \in \mathcal{P}} \log |K(P)|}$$

and the optimal information rate of $\mathcal{A}$ as

$$\rho(\mathcal{A}) = \sup \rho(\Sigma, \mathcal{A}, S)$$

where the supremum is taken over the space of all possible sets of secrets S, $|S| \geq 2$, and all secret sharing schemes for $\mathcal{A}$. Secret sharing schemes with information rate equal to one, which is the maximum possible value of this parameter, are called *ideal*, and an access structure $\mathcal{A}$ on $\mathcal{S}$ is said to be *ideal* if there exists an ideal secret sharing scheme Σ realizing it.

Secret sharing schemes have been extensively studied during the last years, and a huge amount of results can be found in the literature (see [38]). One of the basic issue in the area of secret sharing schemes is that of estimating the *information rate* of the scheme, that is, the ratio between the size of the secret and that of the largest share given to any participant. This problem has received considerable attention in the last few years (e.g., [2, 9, 13, 14, 15, 16, 40, 8, 33]). The practical relevance of this issue is based on the following observations: Firstly, the security of any system tends to degrade as the amount of information that must be kept secret, i.e., the shares of the participants, increases. Secondly, if the shares given to participants are too long, the memory requirements for the participants will be too severe and, at the same time, the shares distribution algorithms will become inefficient. Therefore, it is important to derive significative upper and lower bounds on the information rate of secret sharing schemes.

A special class of secret sharing schemes, on which our constructions of DKDSs will be based on, is the class of *linear secret sharing schemes* (LSSS, for short). We briefly recall some basic facts. Let E be a vector space of finite dimension over a finite field $GF(q)$. For every $P_i \in \mathcal{P} \cup \{D = P_0\}$, let E_i be a vector space over $GF(q)$, and let $\pi_i : E \rightarrow E_i$ be a surjective linear mapping. Let us suppose that these linear mappings satisfy the following properties: for any $A \subset \mathcal{P}$,

$$\bigcap_{P_i \in A} \ker \pi_i \subset \ker \pi_0 \quad \text{or} \quad \bigcap_{P_i \in A} \ker \pi_i + \ker \pi_0 = E.$$

The family of vector spaces and the linear surjective mappings above defined determine the following access structure

$$\mathcal{A} = \left\{ A \subset \mathcal{P} \, : \, \bigcap_{P_i \in A} \ker \pi_i \subset \ker \pi_0 \right\}.$$

A linear secret sharing scheme with secrets chosen in E_0 for the access structure $\mathcal{A}$ can be defined as follows: for a secret $k \in E_0$, the dealer uniformly chooses a vector $v \in E$ such that $\pi_0(v) = k$ and sends to each participant $P_i \in \mathcal{P}$ the vector $a_i = \pi_i(v) \in E_i$ as its share. A formal proof that this is a secret sharing scheme for the access structure $\mathcal{A}$ with secrets chosen in E_0 can be derived by a straightforward application of the following lemma.

Lemma 2.2 *Let E, E_0 and E_1 be vector spaces over a finite field $GF(q)$. Let us consider two linear mappings $\varphi_0 : E \to E_0$ and $\varphi_1 : E \to E_1$, where φ_0 is surjective. Let us suppose that a vector $x \in E$ is chosen uniformly at random and let us consider the random variables $\mathbf{X}_0$ and $\mathbf{X}_1$ corresponding to $x_0 = \varphi_0(x)$ and $x_1 = \varphi_1(x)$, respectively. Then,*

1. $H(\mathbf{X}_0|\mathbf{X}_1) = 0$ *if and only if* $\ker \varphi_1 \subset \ker \varphi_0$,
2. $H(\mathbf{X}_0|\mathbf{X}_1) = H(\mathbf{X}_0)$ *if and only if* $\ker \varphi_1 + \ker \varphi_0 = E$.

Proof. Let $x_1 = \varphi_1(x)$. Then, $x_0 \in \varphi_0(x') + \varphi_0(\ker \varphi_1)$, where $x' \in E$ is any vector such that $\varphi_1(x') = x_1$. Besides, all values in $\varphi_0(x') + \varphi_0(\ker \varphi_1)$ are equiprobable and it is easy to see that x_0 can be uniquely determined from x_1 if and only if $\varphi_0(\ker \varphi_1) = \{0\}$, i.e., if and only if $\ker \varphi_1 \subset \ker \varphi_0$. On the other hand, the value x_1 does not provide any information about the value x_0 if and only if $\varphi_0(\ker \varphi_1) = E_0$. In any other case, the value of x_1 provides partial information about x_0. We can prove that $\varphi_0(\ker \varphi_1) = E_0$ if and only if $\ker \varphi_1 + \ker \varphi_0 = E$. Indeed, let us suppose that $\varphi_0(\ker \varphi_1) = E_0$. Then, for any $x \in E$, there exists $y \in \ker \varphi_1$ such that $\varphi_0(x) = \varphi_1(x)$. Therefore, $x = y + (x - y)$, where $y \in \ker \varphi_1$ and $x - y \in \ker \varphi_0$. Reciprocally, if $\ker \varphi_1 + \ker \varphi_0 = E$, then $E_0 = \varphi_0(E) = \varphi_0(\ker \varphi_1 + \ker \varphi_0) = \varphi_0(\ker \varphi_1)$. Hence, the result holds. ■

Notice that the above result, applied to our linear algebraic framework, says that the sets in A whose linear mappings satisfy the condition $\bigcap_{P_i \in A} \ker \pi_i \subset \ker \pi_0$ are sets allowed to recover the secret. On the contrary, the ones whose mappings satisfy the condition $\bigcap_{P_i \in A} \ker \pi_i + \ker \pi_0 = E$ obtain no information on the secret.

The information rate of this scheme is $\rho = \dim E_0 / (\max_{1 \le i \le n} \dim E_i)$. In a LSSS the secret is computed by a linear mapping. More precisely, for every $A = \{P_{i_1}, \ldots, P_{i_r}\} \in \mathcal{A}$, there exists a linear mapping $\chi_A : E_{i_1} \times \cdots \times E_{i_r} \to E_0$ that enables the participants in A to compute the secret.

Linear secret sharing schemes were first introduced by Brickell [12], who considered only ideal linear schemes with $\dim E_i = 1$ for any $P_i \in \mathcal{P} \cup \{D\}$. General linear secret sharing schemes were introduced by Simmons [36], Jackson and Martin [25] and Karchmer and Wigderson [27] under other names such as geometric secret sharing schemes or monotone span programs.

3 The Model

Let $\mathcal{U} = \{U_1, \ldots, U_m\}$ be a set of m users and let $\mathcal{S} = \{S_1, \ldots, S_n\}$ be a set of n servers. Each user has private connections with *all* the servers. Let us consider an access structure $\mathcal{A} \subset 2^{\mathcal{S}}$ on the set of servers and two families $\mathcal{C}, \mathcal{G} \subset 2^{\mathcal{U}}$ of subsets of the set of users. $\mathcal{C}$ is the family of *conferences*, i.e., the family of group of users which want to securely communicate, and $\mathcal{G}$ is the family of *tolerated coalitions*, i.e., the family of coalitions of users who can try to break the scheme in some way. A distributed key distribution scheme is divided in three phases: an *initialization phase*, which involves only the servers; a *key-request phase*, in which users ask for keys to servers; and a *key-computation phase*, in which users retrieve keys from the messages received from the servers contacted during the key-request phase.

Initialization phase We assume that the initialization phase is performed by a *privileged* subset of servers $P_I = \{S_1, \ldots, S_t\} \in \mathcal{A}$. Each of these servers, using a *private source* of randomness r_i, generates some information that securely distributes to the others. More precisely, for $i = 1, \ldots, t$, S_i sends to S_j the value $\gamma_{i,j}$, where $j = 1, \ldots, n$. At the end of the distribution, for $i = 1, \ldots, n$, each server S_i *computes and stores* some secret information $a_i = f(\gamma_{1,i}, \ldots, \gamma_{t,i})$, where f is a publicly known function.

Key-request phase Let $C_h \in \mathcal{C}$ be a conference. Each user U_j in C_h, contacts the servers belonging to some subset $P \in \mathcal{A}$, requiring a key for the conference C_h. We denote such a key by κ_h. Server $S_i \in P$, contacted by user U_j, checks[2] for membership of U_j in C_h; if $U_j \in C_h$, then S_i computes a value $y_{i,j}^h = F(a_i, j, h)$, where F is a public known function. Otherwise, S_i sets $y_{i,j}^h = \perp$, a special value which does convey no information about κ_h. Finally, S_i sends the value $y_{i,j}^h$ to U_j.

Key-computation phase Once having received the answers from the contacted servers, each user U_j in C_h computes $\kappa_h = G_P(y_{i_1,j}^h, \ldots, y_{i_{|P|},j}^h)$, where $i_1, \ldots, i_{|P|}$ are the indices of the contacted servers, and G_P is a publicly known function.

We are interested in formalizing, within an information theoretic framework the notion of a DKDS, in order to quantify *exactly* the amount of resources that a *real-world* implementation of such a system can require. We use the entropy function because it enables a compact, elegant, and concise description of the model, and permits to take into account all possible probability distributions on the entities of the system. To this aim, we need to setup our notation.

- Let $\mathcal{C} \subset 2^{\mathcal{U}}$ be the set of conferences on $\mathcal{U}$ indexed by elements of $\mathcal{H} = \{1, 2, \ldots\}$.
- For any subset $G = \{U_{j_1}, \ldots, U_{j_g}\} \subset \mathcal{U}$ of users, denote by $\mathcal{C}_G = \{C_h \in \mathcal{C} : C_h \cap G \neq \emptyset\}$ the set of conferences containing some user in G, and by $\mathcal{H}_G = \{h \in \mathcal{H} : C_h \in \mathcal{C}_G\}$ the set of corresponding indices. Let $\ell = \max_{G \in \mathcal{G}} |\mathcal{C}_G|$ be the maximum number of conferences that are controlled by any coalition in $\mathcal{G}$.
- For $i = 1, \ldots, t$, let $\Gamma_{i,j}$ be the set of values $\gamma_{i,j}$ that can be sent by server S_i to server S_j, for $j = 1, \ldots, n$, and let $\Gamma_j = \Gamma_{1,j} \times \cdots \times \Gamma_{t,j}$ be the set of tuples that S_j, for $j = 1, \ldots, n$, can receive during the initialization phase.
- Let K_h be the set of possible values for the key κ_h, and let A_i be the set of values a_i the server S_i can compute during the initialization phase.
- Finally, let $Y_{i,j}^h$ be the set of values $y_{i,j}^h$ that can be sent by S_i when it receives a key-request message from U_j for the conference C_h.

Given three sets of indices $X = \{i_1, \ldots, i_r\}$, where $i_1 < i_2 \ldots < i_r$, $Y = \{j_1, \ldots, j_s\}$, where $j_1 < j_2 \ldots < j_s$, and $H = \{h_1, \ldots, h_t\}$, where $h_1 < h_2 \ldots < h_t$, and three families of sets $\{T_i\}$, $\{T_{i,j}\}$ and $\{T_{i,j}^h\}$, we denote by $T_X = T_{i_1} \times \cdots \times T_{i_r}$, by $T_{X,Y} = T_{i_1,j_1} \times \cdots \times T_{i_r,j_1} \times \cdots \times T_{i_1,j_r} \times \cdots \times T_{i_r,j_s}$, and by $T_{X,Y}^H = T_{i_1,j_1}^{h_1} \times \cdots \times T_{i_1,j_s}^{h_1} \times \cdots \times T_{i_r,j_1}^{h_1} \times \cdots \times T_{i_r,j_s}^{h_1} \times \cdots \times T_{i_1,j_1}^{h_t} \times \cdots \times T_{i_1,j_s}^{h_t} \times \cdots \times T_{i_r,j_1}^{h_t} \times \cdots \times T_{i_r,j_s}^{h_t}$, the corresponding Cartesian products. According to this notation, we will consider several Cartesian products, defined on the sets of our interest (see Table 1).

Γ_Y	Set of tuples that can be received by server S_j, for $j \in Y$
$\Gamma_{X,j}$	Set of tuples that can be sent by server S_i to S_j, for $i \in X$
$\Gamma_{X,Y}$	Set of tuples that can be sent by server S_i to S_j, for $i \in X$ and $j \in Y$
K_X	Set of tuples of conference keys
A_X	Set of tuples of private information a_i
$Y_{X,j}^h$	Set of tuples that can be sent by S_i, for $i \in X$, to U_j for the conference C_h
Y_G^h	Set of tuples that can be sent by $S_1, \ldots, S_n$ to U_j, with $j \in G$, for C_h
Y_G^H	Set of tuples that, for any $h \in H$, can be sent by $S_1, \ldots, S_n$ to U_j, with $j \in G$, for C_h

Table 1: Cartesian Products

We will denote in boldface the random variables $\mathbf{\Gamma}_{i,j}, \mathbf{\Gamma}_j, \ldots, \mathbf{Y}_G^X$ assuming values on the sets $\Gamma_{i,j}, \Gamma_j, \ldots, Y_G^X$, according to the probability distributions $\mathcal{P}_{\mathbf{\Gamma}_{i,j}}, \mathcal{P}_{\mathbf{\Gamma}_j}, \ldots, \mathcal{P}_{\mathbf{Y}_G^X}$.

Roughly speaking, a DKDC must satisfy the following properties:

[2] We do not consider the underline authentication mechanism involved in a key request phase.

- **Correct Initialization Phase.** When the initialization phase correctly terminates, each server S_i must be able to compute his private information a_i. On the other hand, if server S_i misses/does-not-receive *just one* message from the servers[3] in P_I sending information, then S_i must not gain any information about a_i. We model these two properties by relations 1 and 2 of the formal definition.
- **Consistent Key Computation.** Each user in a conference $C_h \subseteq \mathcal{U}$ must be able to compute *the same* conference key, after interacting with the servers of a subset $P \in \mathcal{A}$ at his choice. Relations 3 and 4 of the formal definition ensure these properties. More precisely, relation 3 establishes that each server uniquely determines an answer to any key-request message; while, property 4 establishes that each user uniquely computes the same conference key, using the messages received by the subset of authorized servers he has contacted for that conference key.
- **Conference Key Security.** A conference key must be secure against attacks performed by coalitions of servers, coalitions of users, and hybrid coalitions (servers and users). This is the most intriguing and difficult property to formalize. Indeed, the worst case scenario to look after consists of a coalition of users $G \in \mathcal{G}$ that honestly run the protocol many times, retrieving several conference keys and, then, with the cooperation of some dishonest servers, try to gain information on a new conference key, which was not requested before. Notice that, according to our notation, the maximum amount of information the coalition can acquire honestly running the protocol is represented by $\mathbf{Y}_G^{\mathcal{H}_G \setminus \{h\}}$; moreover, dishonest servers, belonging to $F \notin \mathcal{A}$, know $\mathbf{\Gamma}_F$ and, maybe, $\mathbf{\Gamma}_{Z,N}$. This random variable takes into account the possibility that some of the dishonest servers send information in the initialization phase (i.e. $Z \subseteq F \cap P_I$). Hence, they know the messages they send out to the other servers in this phase. Relation 5 ensures that such coalitions of adversaries, do not gain information on any new key.

Formally, a Distributed Key Distribution Scheme with access structure $\mathcal{A}$ on $\mathcal{S}$ can be defined as follows:

Definition 3.1 Let $\mathcal{U} = \{U_1, \ldots, U_m\}$ be a set of users and let $\mathcal{S} = \{S_1, \ldots, S_n\}$ be a set of servers. Let us consider an access structure $\mathcal{A} \subset 2^{\mathcal{S}}$ on the set of servers and two families $\mathcal{C}, \mathcal{G} \subset 2^{\mathcal{U}}$ of subsets of the set of users. An $(\mathcal{A}, \mathcal{C}, \mathcal{G})$-Distributed Key Distribution Scheme (for short, $(\mathcal{A}, \mathcal{C}, \mathcal{G})$-DKDS) is a protocol which enables each user of $C_h \in \mathcal{C}$ to compute a common key κ_h interacting with a subset of authorized servers in $\mathcal{A}$ of the network. More precisely, the following properties are satisfied:

1 For each $i = 1, \ldots, n$, $H(\mathbf{A}_i | \mathbf{\Gamma}_i) = 0$.
2 For each $X \subset P_I$, $X \neq P_I$, and $i \in \{1, \ldots, n\}$,, $H(\mathbf{A}_i | \mathbf{\Gamma}_{X,i}) = H(\mathbf{A}_i)$.
3 For each $C_h \in \mathcal{C}$, for each $U_j \in C_h$, and for each $i = 1, \ldots, n$, $H(\mathbf{Y}_{i,j}^h | \mathbf{A}_i) = 0$.
4 For each $C_h \in \mathcal{C}$, for each $P \in \mathcal{A}$, and for each $U_j \in C_h$, $H(\mathbf{K}_h | \mathbf{Y}_{P,j}^h) = 0$.
5 For each $C_h \in \mathcal{C}$, for each $G \in \mathcal{G}$, and for each subset $F \notin \mathcal{A}$

$$H(\mathbf{K}_h | \mathbf{Y}_G^{\mathcal{H}_G \setminus \{h\}} \mathbf{\Gamma}_F \mathbf{\Gamma}_{Z,N}) = H(\mathbf{K}_h),$$

where $Z = F \cap P_I$ and $N = \{1, \ldots, n\}$.

Notice that a DKDC implemented by a DKDS is a *deterministic* system at all. Random bits are needed only at the beginning (i.e. initialization of the system), when each server in P_I uses his own random source to generate messages to deliver to the other servers of the network.

In the following, without loss of generality and to emphasize the *real-world oriented* motivations of our study, we assume that the conference keys are *uniformly chosen* in a set K. Hence, for different $h, h' \in \mathcal{H}$, $H(\mathbf{K}_h) = H(\mathbf{K}_{h'}) = \log |K|$.

[3] Without loss of generality, we choose P_I as one of the smallest subsets in $\mathcal{A}$ because one of our aim is to minimize the randomness (i.e., the number of random bits needed to set up the scheme) and the communication complexity of the initialization phase.

4 Communication Complexity, Memory Storage, and Randomness of a DKDS

A basic relation between $(\mathcal{A}, \mathcal{C}, \mathcal{G})$-DKDS and Secret Sharing Schemes enables us to derive some lower bounds on the *memory storage*, on the *communication complexity*, and on the number of *random bits* needed to set up the scheme.

4.1 Preliminaries

We state some results which will be useful in proving the lower bounds.

The following simple lemma establishes that, given three random variables $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{C}$, if $\mathbf{B}$ is a function of $\mathbf{C}$, then $\mathbf{B}$ gives less information on $\mathbf{A}$ than $\mathbf{C}$.

Lemma 4.1 *Let* $\mathbf{A}$, $\mathbf{B}$, *and* $\mathbf{C}$ *be three random variables such that* $H(\mathbf{B}|\mathbf{C}) = 0$. *Then,* $H(\mathbf{A}|\mathbf{B}) \geq H(\mathbf{A}|\mathbf{C})$.

Proof. Notice that, (3) and (8) of Appendix A imply

$$0 \leq H(\mathbf{B}|\mathbf{AC}) \leq H(\mathbf{B}|\mathbf{C}) = 0.$$

Since from (7) of Appendix A,

$$\begin{aligned} I(\mathbf{A}, \mathbf{B}|\mathbf{C}) &= H(\mathbf{A}|\mathbf{C}) - H(\mathbf{A}|\mathbf{BC}) \\ &= H(\mathbf{B}|\mathbf{C}) - H(\mathbf{B}|\mathbf{AC}) = 0. \end{aligned}$$

then, $H(\mathbf{A}|\mathbf{C}) = H(\mathbf{A}|\mathbf{BC})$. But (8) of Appendix A, implies $H(\mathbf{A}|\mathbf{B}) \geq H(\mathbf{A}|\mathbf{BC})$. Therefore, $H(\mathbf{A}|\mathbf{B}) \geq H(\mathbf{A}|\mathbf{C})$, which proves the lemma. ■

Given any four random variables $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$, and $\mathbf{D}$, if $H(\mathbf{B}|\mathbf{C}) = 0$, then, along the line of the above proof, we can show that

$$H(\mathbf{A}|\mathbf{BD}) \geq H(\mathbf{A}|\mathbf{CD}). \tag{1}$$

The next lemma, instead, establishes that the amount of information a subset of servers gains about the conference keys depends on the *membership* of the subset along the access structure $\mathcal{A}$, and is *all-or-nothing* in fashion.

Lemma 4.2 *Let* P *and* F *be two subsets of* $\mathcal{S}$ *such that* $P \in \mathcal{A}$ *and* $F \notin \mathcal{A}$. *Moreover, let* $\mathcal{H}_r = \{h_1 \ldots, h_r\} \subseteq \mathcal{H}$ *be a subset of indices of conferences. Then, it holds that*

$$H(\mathbf{K}_{\mathcal{H}_r}|\mathbf{A}_P) = 0, \; \textit{and} \; H(\mathbf{K}_{\mathcal{H}_r}|\mathbf{A}_F) = H(\mathbf{K}_{\mathcal{H}_r}).$$

Proof. Let $G = \{U_{j_1}, \ldots, U_{j_g}\}$ be a set of users, such that $\mathcal{H}_r \subseteq \mathcal{H}_G$. Notice that,

$$\begin{aligned} 0 &\leq H(\mathbf{K}_{\mathcal{H}_r}|\mathbf{A}_P) \text{ (from (3) of Appendix A)} \\ &\leq H(\mathbf{K}_{\mathcal{H}_r}|\mathbf{Y}_{P,G}^{\mathcal{H}_r}) \text{ (from Lemma 4.1)} \\ &\leq \sum_{j=1}^{r} H(\mathbf{K}_{h_j}|\mathbf{Y}_{P,G}^{h_j}) \text{ (from (4) and (8) of Appendix A)} \\ &\leq \sum_{j=1}^{r} H(\mathbf{K}_{h_j}|\mathbf{Y}_{P,t}^{h_j}) \text{ (from (8) of Appendix A where } t \in C_{h_j} \cap G \text{)} \\ &= 0 \text{ (from Property 4 of Definition 3.1).} \end{aligned}$$

The second equality can be shown in a similar way. Indeed, from the definition of a DKDS easily follows that $H(\mathbf{A}_F|\mathbf{\Gamma}_F) = 0$ and $H(\mathbf{K}_{\mathcal{H}_r\setminus\{h_j\}})|Y_G^{\mathcal{H}_r\setminus\{h_j\}}) = 0$. Applying equation 1, we get

$$H(\mathbf{K}_{\mathcal{H}_r\setminus\{h_j\}}|\mathbf{A}_F\mathbf{K}_{\mathcal{H}_r\setminus\{h_j\}}) \geq H(\mathbf{K}_{h_j}|\mathbf{\Gamma}_F\mathbf{\Gamma}_{Z,N}\mathbf{Y}_G^{\mathcal{H}_r\setminus\{h_j\}}). \tag{2}$$

Hence, a simple algebra shows that

$$\begin{aligned}
H(\mathbf{K}_{\mathcal{H}_r}) &\geq H(\mathbf{K}_{\mathcal{H}_r}|\mathbf{A}_F) \text{ (using (5) of Appendix A)}\\
&= \sum_{j=1}^{r} H(\mathbf{K}_{h_j}|\mathbf{A}_F\mathbf{K}_{\mathcal{H}_r\setminus\{h_j\}}) \text{ (from (4) of Appendix A)}\\
&\geq \sum_{j=1}^{r} H(\mathbf{K}_{h_j}|\mathbf{\Gamma}_F\mathbf{\Gamma}_{Z,N}\mathbf{Y}_G^{\mathcal{H}_r\setminus\{h_j\}}) \text{ (from equation (2))}\\
&= \sum_{j=1}^{r} H(\mathbf{K}_{h_j}) \geq H(\mathbf{K}_{\mathcal{H}_r}) \text{ (applying property 5 of Definition 3.1 and 6 of Appendix A).}
\end{aligned}$$

Thus, the lemma holds. ∎

Finally, the conference keys a coalition of users can retrieve are statistically independent.

Lemma 4.3 *Let $G = \{U_{j_1}, \ldots, U_{j_g}\} \subseteq \mathcal{G}$ be a coalition of users, and let $\mathcal{H}_G = \{h_1, \ldots, h_{\ell_G}\}$. Then, for each $r = 1, \ldots, \ell_G$, it holds that*

$$H(\mathbf{K}_{h_r}|\mathbf{K}_{\mathcal{H}_G\setminus\{h_r\}}) = H(\mathbf{K}_{h_r}).$$

Proof. From property (5) of Appendix A, one has $H(\mathbf{K}_{h_r}|\mathbf{K}_{\mathcal{H}_G\setminus\{h_r\}}) \leq H(\mathbf{K}_{h_r})$, for each $r = 1, \ldots, \ell$. Moreover, noticing that from property (4) and (5) of Appendix A

$$H(\mathbf{K}_{\mathcal{H}_G\setminus\{h_r\}}|\mathbf{Y}_G^{\mathcal{H}_G\setminus\{h_r\}}) \leq \sum_{h\in\mathcal{H}_G\setminus\{h_r\}} H(\mathbf{K}_h|\mathbf{Y}_G^h) = 0,$$

and setting $\mathbf{A} = \mathbf{K}_{h_r}$, $\mathbf{B} = \mathbf{K}_{\mathcal{H}_G\setminus\{h_r\}}$, and $\mathbf{C} = \mathbf{Y}_G^{\mathcal{H}_G\setminus\{h_r\}}$, we can write

$$\begin{aligned}
H(\mathbf{K}_{h_r}|\mathbf{K}_{\mathcal{H}_G\setminus\{h_r\}}) &\geq H(\mathbf{K}_{h_r}|\mathbf{Y}_G^{\mathcal{H}_G\setminus\{h_r\}}) \text{ (from Lemma 4.1)}\\
&\geq H(\mathbf{K}_{h_r}|\mathbf{Y}_G^{\mathcal{H}_G\setminus\{h_r\}}\mathbf{\Gamma}_X\mathbf{\Gamma}_{Z,N}) \text{ (from (8) of Appendix A)}\\
&= H(\mathbf{K}_{h_r}) \text{ (from Property 4 of Definition 3.1),}
\end{aligned}$$

where $N = \{1, \ldots, n\}$, $X = \{i_1, \ldots, i_{k-1}\} \subset N$, and $Z = X \cap \{1, \ldots, k\}$. Hence, the ℓ_G conference keys that the users in G can retrieve are independent. ∎

4.2 Lower Bounds

Lower bounds on the amount of information each server has to store and send to a key-request message, and on the number of random bits needed to set up the scheme can be established exploring the relation existing between a DKDS and SSSs. Since from Appendix A follows that $H(\mathbf{X}) \leq \log|X|$, for each random variable $\mathbf{X}$ assuming values on the set X, we enunciate the lower bounds in terms of the size of the sets of our interest.

First, notice that, the 4-th and the 5-th conditions of the definition of a DKDS "contain" a SSS. More precisely, in any DKDS

- *for each $C_h \in \mathcal{C}$, for each $P \in \mathcal{A}$, and for each $U_j \in C_h$,* $H(\mathbf{K}_h|\mathbf{Y}_{P,j}^h) = 0$

- *for each $C_h \in \mathcal{C}$, for each coalition $G \in \mathcal{G}$, and for each $F \notin \mathcal{A}$ it holds that*

$$H(\mathbf{K}_h | \mathbf{Y}_G^{\mathcal{H}_G \setminus \{h\}} \mathbf{\Gamma}_F \mathbf{\Gamma}_{Z,N}) = H(\mathbf{K}_h)$$

The first relation *is* exactly the *reconstruction property* of a SSS, say Σ_1, for an access structure $\mathcal{A}'$ isomorphic to $\mathcal{A}$, and set of secrets K_h. The isomorphism between the access structures $\mathcal{A}$ and $\mathcal{A}'$ is given, for any fixed pair of values $h \in \mathcal{H}$, and $j \in \{1, \ldots, m\}$, by $\phi : S_i \to Y_{i,j}^h$. In other words, the secret κ_h is shared by means of $y_{1,j}^h, \ldots, y_{n,j}^h$.

The second relation *contains* the *security condition* of Σ_1. Indeed, since the values $y_{i,j}^h$ are function of the private information a_i, computed and stored by each server at the end of the initialization phase, it is easy to check that

$$\begin{aligned} H(\mathbf{K}_h) &= H(\mathbf{K}_h | \mathbf{Y}_G^{\mathcal{H}_G \setminus \{h\}} \mathbf{\Gamma}_F \mathbf{\Gamma}_{Z,N}) \text{ (from property 5 of the definition)} \\ &\le H(\mathbf{K}_h | \mathbf{A}_F) \text{ (since } H(\mathbf{K}_h|\mathbf{\Gamma}_F) \le H(\mathbf{K}_h|\mathbf{A}_F) \text{ and applying (8) of Appendix A)} \\ &\le H(\mathbf{K}_h | \mathbf{Y}_{F,j}^h) \le H(\mathbf{K}_h). \end{aligned}$$

Therefore, recalling that $\rho(\mathcal{A}) = \sup \rho(\Sigma, \mathcal{A}, S)$, and assuming that for any $h \in \mathcal{H}$ it holds that $K_h = K$, the size in bits each answer a server sends to reply to a key request message, must satisfy the inequality given by the following theorem.

Theorem 4.4 *In any $(\mathcal{A}, \mathcal{C}, \mathcal{G})$-DKDS, for all $j = 1, \cdots, m$ and for each $h \in \mathcal{H}$, it holds that*

$$\max_{i=1,\ldots,n} \log |Y_{i,j}^h| \ge \frac{\log |K|}{\rho(\mathcal{A})}.$$

Analogously, we can show a lower bound on the amount of information each server has to store. To this aim, notice that each server basically holds a *share of the sequence of keys* the users can ask for. According to the definition of a DKDS, the number of conference keys the scheme provides is $|\mathcal{C}|$ but, as stated by Lemma 4.3, only ℓ of them must be independent, where ℓ is the maximum number of conference keys that a coalition G can retrieve. In order to derive the lower bound, we can assume that the scheme enables to compute only ℓ conference keys, where ℓ is the maximum number of conference keys a coalition of users G can retrieve. In this case, the secret the servers share can be seen as an element belonging to the set $T = K_{\mathcal{H}_G}$, for some G such that $\ell_{\mathcal{G}} = \ell$. Applying Lemma 4.3 we can say that

$$H(\mathbf{T}) = H(\mathbf{K}_{\mathcal{H}_\ell}) = \sum_j H(\mathbf{K}_{i_j}) = \ell H(\mathbf{K}).$$

Since Lemma 4.2 establishes that $H(\mathbf{T}|\mathbf{A}_P) = 0$ if $P \in \mathcal{A}$, while $H(\mathbf{T}|\mathbf{A}_F) = H(\mathbf{T})$, when $F \notin \mathcal{A}$, we recover another SSS, say Σ_2, with access structure $\mathcal{A}''$ isomorphic to $\mathcal{A}$, and set of secrets T. Consequently, the next theorem holds:

Theorem 4.5 *In any $(\mathcal{A}, \mathcal{C}, \mathcal{G})$-DKDS, it holds that*

$$\max \log |A_i| \ge \frac{\log |T|}{\rho(\mathcal{A})} \ge \ell \frac{\log |K|}{\rho(\mathcal{A})}.$$

The communication complexity of a $(\mathcal{A}, \mathcal{C}, \mathcal{G})$-DKDS can be lower bounded as follows: notice that relations 1 and 2 of a DKDS are again the properties characterizing a secret sharing scheme, say Σ_3. More precisely, for any subset $F \subset P_I$, it holds that

$$H(\mathbf{A}_i|\mathbf{\Gamma}_i) = 0, \text{ while } H(\mathbf{A}_i|\mathbf{\Gamma}_{F,i}) = H(\mathbf{A}_i).$$

In this case $\mathcal{S} = \{S_1, \ldots, S_t\}$ is the *only* subset in the access structure $\overline{\mathcal{A}}$ of the SSS Σ_3 (i.e., a (t,t) threshold structure), and the shared secret is exactly a_i. Hence, the following holds:

Theorem 4.6 *In any* $(\mathcal{A},\mathcal{C},\mathcal{G})$*-DKDS, for* $j = 1,\ldots,t$*, it holds*

$$\log|\Gamma_{j,i}| \geq \log|A_i|.$$

Moreover, since each server performing the initialization phase uses a private source of random bits, we have:

Theorem 4.7 *In any* $(\mathcal{A},\mathcal{C},\mathcal{G})$*-DKDS, it holds*

$$\log|\Gamma_j| = \log|\Gamma_{1,j}| \times \cdots \times \log|\Gamma_{t,j}| = \sum_{i=1}^{t} \log|\Gamma_{i,j}|.$$

To set up a cryptographic protocol and in this case a Distributed Key Distribution Scheme, we need random bits. This resource is usually referred to as the *randomness* of the scheme[4].

The randomness of a scheme can be measured in different way. Knuth and Yao [28] proposed the following approach: Let `Alg` be an algorithm that generates the probability distribution $P = \{p_1,\ldots,p_n\}$, using only independent and unbiased random bits. Denote by $T(\texttt{Alg})$ the average number of random bits used by `Alg` and let $T(P) = min_{\texttt{Alg}} T(\texttt{Alg})$. The value $T(P)$ is a measure of the average number of random bits needed to simulate the random source described by the probability distribution P.

The randomness $\mathcal{R}$ of a Distributed Key Distribution Scheme can be lower bounded as stated by the following theorem.

Theorem 4.8 *In any* $(\mathcal{A},\mathcal{C},\mathcal{G})$*-DKDS the randomness satisfies*

$$\mathcal{R} \geq t \times \ell \times R_{opt}$$

where $t = |P_I|$*,* ℓ *is the maximum number of conference keys that a coalition of adversaries can retrieve, and* R_{opt} *is the minimum amount of randomness required to* generate and share *a secret according to a secret sharing scheme* Σ *with access structure* $\mathcal{A}$*.*

Indeed, applying a similar argument to the one we have applied before in order to derive the lower bound on the size of the private information stored by each server, we can say that the scheme enables to compute at least ℓ conference keys. Since Lemma 4.3 implies that these ℓ conference keys are independent, the share held by each server can be seen as a sequence of ℓ independent sub-shares, one for each conference key. Therefore, the randomness needed to set up the scheme is at least the randomness needed to share independently ℓ keys among the servers, according to the given access structure. The bound follows observing that each of the t servers setting up the system performs an independent sharing of ℓ values from which the keys are derived.

Hence, all the results and bounds on SSS concerning randomness and information rates related to the study of specific access structures can be used to retrieve corresponding results and bounds holding for $(\mathcal{A},\mathcal{C},\mathcal{G})$-DKDSs.

5 Protocols: Designing DKDSs from LSSSs

In this section, we present a method to construct a $(\mathcal{A},\mathcal{C},\mathcal{G})$-DKDS given a general access structure $\mathcal{A}$ on the set of servers. We start by recalling some preliminary concepts.

Let $E, E_0, E_1, \ldots, E_n$ be vector spaces of finite dimension over a finite field $GF(q)$ and, for $i = 0,\ldots,n$, let $\pi_i : E \rightarrow E_i$ be surjective linear mappings defining a LSSS Σ on $\mathcal{S}$ with access structure $\mathcal{A}$. For any authorized subset $A = \{S_{i_1},\ldots,S_{i_r}\} \in \mathcal{A}$, let χ_A be the linear mapping

[4] A detailed analysis of the randomness in distribution protocols can be found in [11].

$\chi_A : E_{i_1} \times \cdots \times E_{i_r} \to E_0$ enabling the reconstruction of the secret from the shares. Moreover, from every π_i, let $\pi_i^\ell : E^\ell \to E_i^\ell$ be a mapping defined as $\pi_i^\ell(u_1, \ldots, u_\ell) = (\pi_i(u_1), \ldots, \pi_i(u_\ell))$. It is not difficult to see that the mappings π_i^ℓ define a LSSS Σ^ℓ with secrets chosen in E_0^ℓ on the same access structure and with the same information rate of Σ. In this case, the secret is reconstructed from the shares by using the linear mappings $\chi_A^\ell : E_{i_1}^\ell \times \cdots \times E_{i_r}^\ell \to E_0^\ell$ defined from χ_A.

Then, let us consider, for $h = 1, \ldots, |\mathcal{C}|$, a sequence of linear forms $\varphi_h : GF(q)^\ell \to GF(q)$, such that any ℓ different forms $\varphi_{h_1}, \ldots, \varphi_{h_\ell}$ are linearly independent. Notice that a form φ_h can be seen as a vector in the dual space $(GF(q)^\ell)^*$ and it is determined by its coordinates $(\lambda_{h,1}, \ldots, \lambda_{h,\ell})$, where $\varphi_h(v_1, \ldots, v_\ell) = \sum_{j=1}^{\ell} \lambda_{h,j} v_j$. Therefore, if $q \geq |\mathcal{C}|$, such a family of linear forms can be constructed by considering $|\mathcal{C}|$ different values $z_1, \ldots, z_{|\mathcal{C}|}$ in the finite field $GF(q)$ and by taking, for any $h = 1, \ldots, |\mathcal{C}|$, the vector $(\lambda_{h,1}, \ldots, \lambda_{h,\ell}) = (1, z_h, z_h^2, \ldots, z_h^{\ell-1})$. These linear forms can be used to define a *linear key generator* $\mathcal{K}$. More precisely, conference keys are determined as follows: for every conference $C_h \in \mathcal{C}$, a vector $v \in GF(q)^\ell$ is chosen uniformly at random and $\kappa_h = \varphi_h(v)$. The former assumption of independence of any sequence of ℓ forms implies that any set of $\ell - 1$ conference keys does not provide any information on the value of any other conference keys.

Finally, for any vector space U, the linear key generator $\mathcal{K}$, which provides conference keys belonging to the finite field $GF(q)$, can be extended to a linear key generator $\mathcal{K}^U$, whose keys κ_h are *vectors* in U. To this aim, the linear mappings $\varphi_h^U : U^\ell \to U$ can be defined by $\varphi_h^U(u_1, \ldots, u_\ell) = \sum_{j=1}^{\ell} \lambda_{h,j} u_j$, where $(\lambda_{h,1}, \ldots, \lambda_{h,\ell})$ are the coordinates of the linear form φ_h. It is not difficult to see that, as before, any ℓ different conference keys are independent.

At this point we have all the tools to set up, from any LSSS Σ and any linear key generator $\mathcal{K}$, a $(\mathcal{A}, \mathcal{C}, \mathcal{G})$-DKDS. More precisely, given Σ and $\mathcal{K}$, we can construct Σ^ℓ, the linear key generator $\mathcal{K}^E$ and, for any $i = 0, 1, \ldots, n$, the linear key generator $\mathcal{K}^i = \mathcal{K}^{E_i}$, defined by the linear mappings $\varphi_h^i = \varphi_h^{E_i}$. These choices imply that, for any $i = 0, 1, \ldots, n$ and $C_h \in \mathcal{C}$, we have that $\varphi_h^i \circ \pi_i^\ell = \pi_i \circ \varphi_h^E$. Indeed, for any $u = (u_1, \ldots, u_\ell) \in E^\ell$,

$$
\begin{aligned}
(\varphi_h^i \circ \pi_i^\ell)(u) &= \varphi_h^i(\pi_i(u_1), \ldots, \pi_i(u_\ell)) = \sum_{j=1}^{\ell} \lambda_{h,j} \pi_i(u_j) \\
&= \pi_i \left(\sum_{j=1}^{\ell} \lambda_{h,j} u_j \right) = (\pi_i \circ \varphi_h^E)(u).
\end{aligned}
$$

The above relation is the key point in order to understand the construction and, more precisely, the key computation phase performed by the users. The full protocol can be described as follows:

Initialization Phase

Let $P_I = \{S_1, \ldots, S_t\}$ be the authorized subset of servers $P_I \in \mathcal{A}$ performing the initialization phase.

- For every $i = 1, \ldots, t$, the server S_i chooses at random a vector $r_i \in E^\ell$ and, for every $j = 1, \ldots, n$, sends to server S_j the vector $\pi_j^\ell(r_i) \in E_j^\ell$.
- For $j = 1, \ldots, n$, each server S_j computes his private information summing up the shares it has received from the servers in P_I. That is, server S_j computes $a_j = \pi_j^\ell(r_1) + \cdots + \pi_j^\ell(r_t) = \pi_j^\ell(u) \in E_j^\ell$, where $u = r_1 + \cdots + r_t \in E^\ell$.

Therefore, after the initialization phase, each server S_i has a vector $a_i = (a_{i1}, \ldots, a_{i\ell}) \in E_i^\ell$. This vector is a share of a secret vector $\pi_0^\ell(u) = v = (v_1, \ldots, v_\ell) \in E_0^\ell$ shared according to the LSSS Σ^ℓ. The key corresponding to the conference $C_h \in \mathcal{C}$ is $\kappa_h = (\varphi_h^0 \circ \pi_0^\ell)(u) \in E_0$.

The Key Request Phase is carried out as follows:

Key Request Phase

- A user in a conference C_h who wants to obtain the conference key κ_h, sends a key-request message for the conference key to an authorized subset of servers $A = \{S_{i_1}, \ldots, S_{i_r}\} \in \mathcal{A}$.
- Each server S_i invoked by the user checks that the user belongs to C_h and sends to the user the vector $\varphi_h^i(a_i) = (\varphi_h^i \circ \pi_i^\ell)(u) = (\pi_i \circ \varphi_h^E)(u) \in E_i$, which is a share of the conference key $\kappa_h = (\varphi_h^0 \circ \pi_0^\ell)(u) = (\pi_0 \circ \varphi_h^E)(u) \in E_0$ shared according to the LSSS Σ.

Finally, recovering the conference key requires a simple computation.

Key Computation Phase

- Using the values received from the servers in $A \in \mathcal{A}$ the user in C_h recovers the secret key by computing $\kappa_h = \chi_A(\varphi_h^{i_1}(a_{i_1}), \ldots, \varphi_h^{i_r}(a_{i_r}))$.

It is possible to check, by applying the properties of the linear secret sharing scheme Σ and the linear key generator $\mathcal{K}$, that the proposed scheme verifies conditions 1–4 of Definition 3.1. Moreover, condition 5 is proved in Subsection 5.1.

Finally, we compare the parameters of our scheme with the bounds given in Section 4. Let

$$\rho = \frac{\dim E_0}{\max_{1 \le i \le n} \dim E_i}$$

be the information rate of the LSSS Σ. Let q (a power of a prime) be the cardinality of the finite field $GF(q)$.

The amount of information that a server $S_i \in \mathcal{S}$ has to send to a user $U_j \in C_h$ in the key request phase is $\log |Y_{i,j}^h| = \log |E_i| = \log q \dim E_i$. Observe that

$$\max_{i=1,\ldots,n} \log |Y_{i,j}^h| = \log q \dim E_0 \frac{\max_{i=1,\ldots,n} \dim E_i}{\dim E_0} = \frac{\log |K|}{\rho(\Sigma, \mathcal{A}, \mathcal{S})}.$$

Therefore, the bounds given by Theorems 4.4, 4.5, 4.6 and 4.7 are attained if Σ has optimal information rate, that is, if $\rho(\Sigma, \mathcal{A}, \mathcal{S}) = \rho(\mathcal{A})$.

Remark. The *linearity* property of the secret sharing scheme is *not* necessary to design a DKDS. Actually, from *any* secret sharing scheme, realizing a given access structure, we can set up a DKDS on the same access structure. The reader can easily convince himself noticing that in our protocol each server sums up the shares obtained during the distribution phase, storing in this way a reduced amount of information. This is one of the steps in which the linearity property of the scheme is applied. If the secret sharing scheme is not linear, each server has to store all the shares received from the servers performing the distribution. On the other hand, when a user asks for a conference key, he receives *several* shares that must be processed in order to recover the conference key.

5.1 Security of the Scheme

In this subsection we prove that the above construction is secure, that is, we prove that condition 5 in Definition 3.1 is verified by the constructed scheme.

Let us consider a coalition $F \cup G$, where $G \in \mathcal{G}$ is a set of corrupted users and $F \subset \mathcal{S}$, $F \notin \mathcal{A}$ is a set of corrupted servers. According to Definition 3.1, the maximum amount of information the users in G can acquire honestly running the protocol is $Y_G^{\mathcal{H}_G\setminus\{h\}}$. Furthermore, the servers in $F \notin \mathcal{A}$ know Γ_F and, maybe, $\Gamma_{Z,N}$, where $N = \{1,\ldots,n\}$ and Z is the set of those servers in F that belongs to the initialization subset as well. We have to prove that in this scenario, $H(\mathbf{K}_h|\mathbf{Y}_G^{\mathcal{H}_G\setminus\{h\}}\mathbf{\Gamma}_F\mathbf{\Gamma}_{Z,N}) = H(\mathbf{K}_h)$. In order to do it, we will use Lemma 2.2; therefore, we need to determine the linear maps φ_0 and φ_1 corresponding, respectively, to the random variables $\mathbf{K}_h$ and $\mathbf{Y}_G^{\mathcal{H}_G\setminus\{h\}}\mathbf{\Gamma}_F\mathbf{\Gamma}_{Z,N}$. Recalling that $P_I = \{S_1,\ldots,S_t\}$, let us suppose that the set $F \notin \mathcal{A}$ of dishonest servers is $F = \{S_1,\ldots,S_{t-1},S_{i_t},\ldots,S_{i_m}\}$; then, the set Z is given by $\{S_1,\ldots,S_{t-1}\}$. For every $i = 1,\ldots,t$, let $r_i \in E^\ell$ be the vector chosen by server S_i at random, and let $r = (r_1,\ldots,r_t) \in E^{\ell t}$ be the information the servers in P_I generates during the initialization phase. The servers in $Z = \{S_1,\ldots,S_{t-1}\}$ know $r_1,\ldots,r_{t-1}$. These vectors can be written as $r_i = \sigma_i(r)$ for $i = 1,\ldots,t-1$, where $\sigma_i(r)$ denotes the i-th projection of the vector $r = (r_1,\ldots,r_t)$. Moreover dishonest servers $S_{i_t},\ldots,S_{i_m}$ not belonging to P_I also know[5] the information received from the honest server S_t in the initialization phase, i.e. server S_{i_j} receives $\pi_{i_j}^\ell(r_t) = \pi_{i_j}^\ell(\sigma_t(r))$ for $j = t,\ldots,m$. On the other hand, the information that users in G can acquire is determined by $\phi_j(r_1,\ldots,r_t) = \varphi_j^0 \circ \pi_0^\ell(r_1+\ldots+r_t)$ for those $j \in \mathcal{H}_G\setminus\{h\}$. Therefore, the kernel of the linear map φ_1 associated to the random variable $\mathbf{Y}_G^{\mathcal{H}_G\setminus\{h\}}\mathbf{\Gamma}_F\mathbf{\Gamma}_{Z,N}$ is

$$\ker\varphi_1 = \left(\bigcap_{j=1}^{t-1}\ker\sigma_j\right) \cap \left(\bigcap_{j=t}^{m}\ker\pi_{i_j}^\ell\right) \cap \left(\bigcap_{j\in\mathcal{H}_G\setminus\{h\}}\ker\phi_j\right).$$

Since every key κ_h is defined by $\kappa_h = \phi_h(r_1,\ldots,r_t) = \varphi_h^0 \circ \pi_0^\ell(r_1+\cdots+r_t)$, the kernel of the linear map φ_0 related to the random variable $\mathbf{K}_h$ is

$$\ker\varphi_0 = \ker\phi_h.$$

Hence, we have to show that

$$\ker\varphi_0 + \ker\varphi_1 = E^{\ell t}.$$

Trivially $\ker\varphi_0 + \ker\varphi_1 \subset E^{\ell t}$. The opposite inclusion $E^{\ell t} \subset \ker\varphi_0 + \ker\varphi_1$ can be shown as follows: let $y = (y_1,\ldots,y_\ell)$ be any vector in E^ℓ. The independence of the mappings $\{\phi_j\}_{j\in\mathcal{H}_G}$, implies that

$$\ker\varphi_h^0 + \left(\bigcap_{j\in\mathcal{H}_G\setminus\{h\}}\ker\varphi_j^0\right) = E_0^\ell.$$

Therefore, $\pi_0^\ell(y) = (\pi_0(y_1),\ldots,\pi_0(y_\ell)) = (a_1,\ldots,a_\ell)+(b_1,\ldots,b_\ell)$ where $\phi_j(a_1,\ldots,a_\ell) = 0$ for any $j \in \mathcal{H}_G\setminus\{h\}$ and $\phi_h(b_1,\ldots,b_\ell) = 0$. Since $F \notin \mathcal{A}$, from the properties of the LSSS, it holds that, for any $j = 1,\ldots,\ell$, there exists a $z_j \in E$ such that $\pi_0(z_j) = a_j$ and $\pi_i(z_j) = 0$ for any $S_i \in F$.

Hence, setting $w = y - z \in E^\ell$, where $z = (z_1\ldots,z_\ell)$, it is easy to check that $\phi_h(\pi_0^\ell(y-z)) = \phi_h(b_1,\ldots,b_\ell) = 0$. Let $x = (x^1,\ldots,x^t)$ be a vector in $E^{\ell t}$. We can prove that $x \in \ker\varphi_0 + \ker\varphi_1$. To this aim, let us define $y = x^t + \sum_{i=1}^{t-1}x^i \in E^\ell$. From the aforementioned results, there exists a vector z such that $\phi_h(\pi_0^\ell(y-z)) = 0$, $\pi_j^\ell(z) = 0$ for every $S_j \in F$, and $\phi_j(z) = 0$ for every $j \in \mathcal{H}_G\setminus\{h\}$. Further, by defining vectors $u = (x^1,\ldots,x^{t-1}, y-z-\sum_{i=1}^{t-1}x^i)$ and $v = (0,0,\ldots,0,z) \in E^{\ell t}$, it follows that $x = u+v \in E^{\ell t}$. At this point, it is not difficult to show that $u \in \ker\varphi_0$ and $v \in \ker\varphi_1$, which closes the proof. Indeed, $\varphi_0(v) = \varphi_h^0 \circ \pi_0^\ell(x^1,\ldots,x^{t-1}, y-z-\sum_{i=1}^{t-1}x^i) = \varphi_h^0\circ\pi_0^\ell(y-x) = 0$ and, on the other hand, $\sigma_j(v) = 0$ for every $j = 1,\ldots,t-1$. Moreover, $\pi_{i_j}^\ell(\sigma_t(v)) = \pi_{i_j}^\ell(z) = 0$ for every $j = t,\ldots,m$ and, finally, $\phi_j(v) = \varphi_j^0\circ\pi_0^\ell(z) = 0$ for every $j \in \mathcal{H}_G\setminus\{h\}$. Therefore, the result holds.

[5] Notice that we do not take into account the information that servers $S_{i_t},\ldots,S_{i_m}$ receive from servers in Z because it can be deduced from $r_1,\ldots,r_{t-1}$ and these values are known by the members of the coalition.

6 Some Examples

We present some examples to explain how can be really applied the construction given in the previous section to set up a DKDS given an arbitrary access structure on the set of servers. Basically, we need simply to re-phrase in the "language" of the LSSSs some well-known constructions of secret sharing schemes for general access structures, such as the monotone circuit technique of Benaloh and Leichter [4] and the Brickell vector space construction for ideal access structures. Then, the design of a DKDS easily follows.

The Benaloh and Leichter monotone circuit technique for secret sharing schemes works as follows: let $\mathcal{A}$ be an access structure on the set of servers $\mathcal{S} = \{S_1, \ldots, S_n\}$, and let $\mathcal{A}_0$ be the basis of $\mathcal{A}$. Moreover, let $X_1 + X_2 + \cdots + X_r$ be a disjunctive normal form boolean formula representing $\mathcal{A}_0$. Each subset in $\mathcal{A}_0$ corresponds to a clause X_i of the formula. For instance, a $(2, 3)$ threshold access structure on the set $\{S_1, S_2, S_3\}$, can be represented by $S_1S_2 + S_2S_3 + S_1S_3$. Moreover, let d_i be the number of minimal subsets in which server S_i belongs to. The value d_i quantifies the number of shares S_i is going to receive. For $i = 1, \ldots, n$, let E_i be a vector space of dimension d_i over a finite field $GF(q)$, let $E_0 = GF(q)$ and let E be a vector space of dimension $\sum_{i=1}^{r}(|X_i| - 1)$. Given a secret $k \in E_0$, a vector $v \in E$ denoted by $v = (v_1^1, \ldots, v_1^{|X_1|-1}, v_2^1, \ldots, v_2^{|X_2|-1}, \ldots, v_r^1, \ldots, v_r^{|X_r|-1})$ is selected uniformly at random. The linear mappings π_i's are defined in such a way that the set of servers corresponding to the clause X_i will hold the sharing $v_i^1, \ldots, v_i^{|X_i|-1}, k - (v_i^1 + \ldots + v_i^{|X_i|-1})$ which allow to recover the secret k.

As a second example, let us suppose that the access structure $\mathcal{A}$ is ideal. Therefore, we can use the Brickell vector space construction [12] that works as follows: let $\mathcal{A}$ be access structure, and let U be a d dimensional vector space over a finite field $GF(q)$. Suppose that there exists a function $\phi : \mathcal{S} \longrightarrow U$ such that the vector $(1, 0, \ldots, 0)$ can be expressed as a linear combination of the vectors in the set $\{\phi(S_i) : S_i \in B\}$ if and only if B is an authorized subset, i.e. $(1, 0, \ldots, 0) \in \langle \phi(S_i) : S_i \in B\rangle \Leftrightarrow B \in \mathcal{A}$. Then, in order to share a secret $k \in GF(q)$, the dealer chooses a random vector $v \in U$ whose first component is k and computes $\{v \cdot \phi(S_i)\}_{i=1}^{n}$. In other words, in this case the linear mappings $\pi_i : U \rightarrow GF(q)$ are defined by $\pi_i(v) = v \cdot \phi(S_i)$.

Using a well-studied access structure on a set of 4 servers, we show that the bounds are attained every time we can construct an optimal linear secret sharing scheme realizing the given access structure. To this aim, let us consider the access structure on a set $\mathcal{S} = \{S_1, S_2, S_3, S_4\}$ of 4 servers whose minimal authorized subsets are $\mathcal{A}_0 = \{\{S_1, S_2\}, \{S_2, S_3\}, \{S_3, S_4\}\}$. This access structure is well-known in the literature concerning secret sharing schemes [14]. It has been proved in [14] that the information rate of any SSS for this access structure is at most $2/3$. Besides, there exists a linear secret sharing scheme Σ with information rate $\rho = 2/3$. Therefore, we can use this construction in order to design a $(\mathcal{A}, \mathcal{C}, \mathcal{G})$-DKDS attaining the bounds in Section 4. Let us see how this construction works: Let E_0, E_1, E_4 be vector spaces over a finite field $GF(q)$ of dimension 2, and let E_2 and E_3 be 3 dimensional vector spaces and E a vector space of dimension 6. Assume that $k = (k_1, k_2) \in E_0$ is the secret. A pre-image v of the secret is given by the vector $v = (\alpha_1, \beta_1, \gamma_1, \alpha_2, \beta_2, \gamma_2)$, where $\alpha_i + \beta_i = k_i$ for $i = 1, 2$. The linear mappings are defined in such a way that the servers receive, respectively:

S_1	$(\alpha_1, \alpha_2 + \beta_2 - \gamma_2)$
S_2	$(\beta_1, \alpha_2, \gamma_2)$
S_3	$(\alpha_1, \gamma_1, \beta_2)$
S_4	$(\alpha_1 + \beta_1 - \gamma_1, \alpha_2)$

Finally, it is interesting to point out that the two constructions presented in [30], based on bivariate polynomials and on monotone span programs, can be seen as instances of the algebraic framework we have described before. In particular, the embedding of the second construction can be done due to the equivalence between monotone span programs and linear secret sharing schemes [1].

7 Conclusion and Open Problems

In this paper we have shown bounds and constructions for unconditionally secure DKDSs with a general access structure on the set of servers. Such schemes enable to setup distributed KDCs which solve many problems related to the presence across a network of a single on-line KDC. Two main contributions can be found in this paper: the reduction technique applied to find the lower bounds, and the linear algebraic framework which unifies many previous proposals.

Some interesting question arise from this study: first of all, we have considered a framework in which each user has *private* connections with all the servers. From a real-life prospective, it would be useful to study a model in which users have only *some* connections with geographically close servers.

Another research direction is to study *computationally secure* distributed key distribution schemes along the line of [30], where some constructions based on pseudo-random functions and the discrete log problem have been proposed.

Finally, for the unconditional and computational frameworks, methods to enhance the constructions with properties like verifiability of the servers' behaviours, proactive security, and anonymity of the conference keys recovered by the users with respect to the servers, are all desirable features to work on.

References

[1] A. Beimel, *Secure Schemes for Secret Sharing and Key Distribution*, PhD Thesis - Department of Computer Science, Technion, 1996.

[2] A. Beimel, A. Gal, and M. Paterson. Lower Bounds for Monotone Span Programs. Proc. 35th IEEE Symp. on Foundations of Computer Science, pp. 674–681, 1995.

[3] M. Bellare and P. Rogaway. Provably Secure Session Key Distribution: The Three Party Case. Proc. 27th Annual Symposium on the Theory of Computing, ACM, 1995.

[4] J. Benaloh and J. Leichter, Generalized Secret Sharing and Monotone Functions. Lecture Notes in Comput. Sci., 403, 27–35, 1990.

[5] G.R. Blakley. Safeguarding Cryptographic Keys. Proceedings of AFIPS 1979 National Computer Conference, Vol. 48, pp. 313–317, 1979.

[6] R. Blom. An Optimal Class of Symmetric Key Generation Systems. Advances in Cryptology - Eurocrypt'84, Lecture Notes in Comput. Sci., vol. 209, pp. 335–338, 1984.

[7] C. Blundo, and P. D'Arco. Unconditionally Secure Distributed Key Distribution Schemes. submitted for publication.

[8] C. Blundo, A. De Santis, R. De Simone, U. Vaccaro. Tight Bounds on the Information Rate of Secret Sharing Schemes. Design, Codes, and Cryptography, vol. 11, no. 1, pp. 1–25, 1997.

[9] C. Blundo, A. De Santis, A. Giorgio Gaggia, and U. Vaccaro. New Bounds on the Information Rate of Secret Sharing Schemes. IEEE Trans. Inform. Theory, Vol. 41, pp. 549–554, 1995.

[10] C. Blundo, A. De Santis, A. Herzberg, S. Kutten, U. Vaccaro and M. Yung. Perfectly-Secure Key Distribution for Dynamic Conferences. Information and Computation, vol. 146, no. 1, pp. 1–23, 1998.

[11] C. Blundo, A. De Santis, and U. Vaccaro. Randomness in Distribution Protocols. Information and Computation, vol. 131, no. 2, pp. 111–139, 1996.

[12] E.F. Brickell. Some ideal secret sharing schemes. J. Combin. Math. and Combin. Comput., 9, 105–113, 1989.

[13] E.F. Brickell and D.R: Stinson. Some Improved Bounds on the Information Rate of Perfect Secret Sharing Schemes. J. Cryptology, Vol. 5, pp. 153-166, 1992.

[14] R.M. Capocelli, A. De Santis, L. Gargano and U. Vaccaro. On the Size of the Shares in Secret Sharing Schemes. Advances in cryptology - CRYPTO'91, Lecture Notes in Comput. Sci., 576, 101–113, 1992.

[15] L. Csimarz. The Size of a Share Must be Large. J. Cryptology, Vol. 10, pp. 223–231, 1997.

[16] M. van Dijk. On the Information Rate of Perfect Secret Sharing Schemes. Design, Codes, and Cryptography, Vol. 6, pp. 143–169, 1995.

[17] P. D'Arco, *On the Distribution of a Key Distribution Center* (extended abstract), Proceedings of ICTCS2001, Lecture Notes in Computer Science, vol. 2202, pp. 357-369, 2001.

[18] R. Canetti, J. Garey, G.Itkins, D. Micciaccio, M. Naor and B. Pinkas. Issues in Multicast Security: A Taxonomy and Efficient Constructions. Proceedings of INFOCOM '99, vol. 2, pp. 708–716, 1999.

[19] W. Jackson, K.M. Martin, and C.M. O'Keefe. Mutually Trusted Authority-Free Secret Sharing Schemes Journal of Cryptology, N.10, pp. 261-289, 1997.

[20] R. Canetti, T. Malkin and K. Nissim. Efficient Communication-Storage Tradeoffs for Multicast Encryption. Advances in Cryptology - Eurocrypt'99, Lecture Notes in Comput. Sci., vol. 1592, pp. 459–474, 1999.

[21] B. Chor, A. Fiat, and M. Naor. Tracing Traitors. Advances in Cryptology - Eurocrypt'94, Lecture Notes in Comput. Sci., vol. 950 pp. 257–270, 1994.

[22] T.M. Cover and J.A. Thomas. Elements of Information Theory. John Wiley & Sons, 1991.

[23] A. Fiat and M. Naor. Broadcast Encryption. Advances in Cryptology - Crypto 92, Lecture Notes in Comput. Sci., vol. 773, pp. 480–491, 1993.

[24] M. Ito, A. Saito and T, Nishizeki. Secret sharing scheme realizing any access structure. Proc. IEEE Globecom'87, 99–102, 1987.

[25] W. Jackson and K. Martin. Geometric Secret Sharing Schemes and Their Duals. Des. Codes Cryptogr., 4, 83–95, 1994.

[26] M. Just, E. Kranakis, D. Krizanc, P. Van Oorschot. Key Distribution via True Broadcasting. Proceedings of the 2nd ACM Conference on Computer and Communications Security, pp. 81–88, 1994.

[27] M. Karchmer, A. Wigderson. On span programs. Proc. of Structure in Complexity'93, 102–111, 1993.

[28] D.E. Knuth and A. Yao, *The Complexity of Nonuniform Random Number Generation*, Algorithms and Complexity, Academic Press, pp. 357-428, 1976.

[29] T. Matsumoto and H. Imai. On the Key Predistribution System: A Practical Solution to the Key Distribution Problem. Advances in Cryptology - Eurocrypt'87, Lecture Notes in Comput. Science, vol. 239, pp. 185–193, 1987.

[30] M. Naor, B. Pinkas, and O. Reingold. Distributed Pseudo-random Functions and KDCs. Advances in Cryptology - Eurocrypt'99, Lecture Notes in Comput. Sci., vol. 1592, pp. 327–346, 1999.

[31] R. M. Needham and M. D. Schroeder. Using Encryption for Authentication in Large Networks of Computers. Communications of ACM, vol. 21, pp. 993–999, 1978.

[32] B. C. Neuman and T. Tso. Kerberos: An Authentication Service for Computer Networks. IEEE Transactions on Communications, vol. 32, pp. 33–38, 1994.

[33] C. Padró and G. Sáez. Secret sharing schemes with bipartite access structure. IEEE Transactions on Information Theory vol. 46, pp. 2596–2605, 2000.

[34] R. Poovendran, J.S.Baras. An Information Theoretic Approach for Design and Analysis of Rooted-Tree Based Multicast Key Management Schemes. Advances in Cryptology - Crypto'99, Lecture Notes in Comput. Sci., vol. 1666, pp. 624–638, 1999.

[35] A. Shamir. How to Share a Secret. Communications of ACM, vol. 22, n. 11, pp. 612–613, 1979.

[36] G.J. Simmons. How to (really) share a secret. Advances in Cryptology, CRYPTO 88, Lecture Notes in Comput. Sci., 403, 390–448, 1990.

[37] G.J. Simmons, W. Jackson and K. Martin. The geometry of secret sharing schemes. Bull. of the ICA, 1, 71–88, 1991.

[38] D.R. Stinson. Bibliography on Secret Sharing Schemes. `http://www.cacr.math.uwaterloo.ca/~dstinson/ssbib.html`.

[39] D.R. Stinson. An explication of secret sharing schemes. Des. Codes Cryptogr., 2, 357–390, 1992.

[40] D.R. Stinson. Decomposition Constructions for Secret Sharing Schemes. IEEE Trans. Inform. Theory, Vol. 40, pp. 118–125, 1994.

[41] D. R. Stinson. On Some Methods for Unconditional Secure Key Distribution and Broadcast Encryption. Designs, Codes and Cryptography, vol. 12, pp. 215–243, 1997.

A Information Theory Elements

This appendix briefly recalls some elements of information theory (see [22] for details). Let $\mathbf{X}$ be a random variable taking values on a set X according to a probability distribution $\{P_{\mathbf{X}}(x)\}_{x\in X}$. The *entropy* of $\mathbf{X}$, denoted by $H(\mathbf{X})$, is defined as

$$H(\mathbf{X}) = -\sum_{x\in X} P_{\mathbf{X}}(x)\log P_{\mathbf{X}}(x),$$

where the logarithm is relative to the base 2. The entropy satisfies $0 \le H(\mathbf{X}) \le \log|X|$, where $H(\mathbf{X}) = 0$ if and only if there exists $x_0 \in X$ such that $Pr(\mathbf{X} = x_0) = 1$; whereas, $H(\mathbf{X}) = \log|X|$ if and only if $Pr(\mathbf{X} = x) = 1/|X|$, for all $x \in X$. Given two random variables $\mathbf{X}$ and $\mathbf{Y}$ taking values on sets X and Y, respectively, according to the joint probability distribution $\{P_{\mathbf{XY}}(x,y)\}_{x\in X, y\in Y}$ on their Cartesian product, the *conditional entropy* $H(\mathbf{X}|\mathbf{Y})$ is defined as

$$H(\mathbf{X}|\mathbf{Y}) = -\sum_{y\in Y}\sum_{x\in X} P_{\mathbf{Y}}(y)P_{\mathbf{X}|\mathbf{Y}}(x|y)\log P_{\mathbf{X}|\mathbf{Y}}(x|y).$$

It is easy to see that

$$H(\mathbf{X}|\mathbf{Y}) \geq 0. \tag{3}$$

with equality if and only if X is a function of Y. Given $n+1$ random variables, $\mathbf{X}_1 \ldots \mathbf{X}_n \mathbf{Y}$, the entropy of $\mathbf{X}_1 \ldots \mathbf{X}_n$ given $\mathbf{Y}$ can be written as

$$H(\mathbf{X}_1 \ldots \mathbf{X}_n|\mathbf{Y}) = H(\mathbf{X}_1|\mathbf{Y}) + H(\mathbf{X}_2|\mathbf{X}_1\mathbf{Y}) + \cdots + H(\mathbf{X}_n|\mathbf{X}_1 \ldots \mathbf{X}_{n-1}\mathbf{Y}). \tag{4}$$

The *mutual information* between $\mathbf{X}$ and $\mathbf{Y}$ is given by

$$I(\mathbf{X};\mathbf{Y}) = H(\mathbf{X}) - H(\mathbf{X}|\mathbf{Y}).$$

Since, $I(\mathbf{X};\mathbf{Y}) = I(\mathbf{Y};\mathbf{X})$ and $I(\mathbf{X};\mathbf{Y}) \geq 0$, it is easy to see that

$$H(\mathbf{X}) \geq H(\mathbf{X}|\mathbf{Y}), \tag{5}$$

with equality if and only if $\mathbf{X}$ and $\mathbf{Y}$ are independent. Therefore, given n random variables, $\mathbf{X}_1 \ldots \mathbf{X}_n$, it holds that

$$H(\mathbf{X}_1 \ldots \mathbf{X}_n) = \sum_{i=1}^{n} H(\mathbf{X}_i|\mathbf{X}_1 \ldots \mathbf{X}_{i-1}) \leq \sum_{i=1}^{n} H(\mathbf{X}_i). \tag{6}$$

Given three random variables, $\mathbf{X}$, $\mathbf{Y}$, and $\mathbf{Z}$, the *conditional mutual information* between $\mathbf{X}$ and $\mathbf{Y}$ given $\mathbf{Z}$ can be written as

$$I(\mathbf{X};\mathbf{Y}|\mathbf{Z}) = H(\mathbf{X}|\mathbf{Z}) - H(\mathbf{X}|\mathbf{Z}\ \mathbf{Y}) = H(\mathbf{Y}|\mathbf{Z}) - H(\mathbf{Y}|\mathbf{Z}\ \mathbf{X}I(\mathbf{Y};\mathbf{X}|\mathbf{Z}). \tag{7}$$

Since the conditional mutual information $I(\mathbf{X};\mathbf{Y}|\mathbf{Z})$ is always non-negative we get

$$H(\mathbf{X}|\mathbf{Z}) \geq H(\mathbf{X}|\mathbf{Z}\ \mathbf{Y}). \tag{8}$$