# A Ramp Model for Distributed Key Distribution Schemes

Carlo Blundo[1], Paolo D'Arco[1], 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 i Telemàtica
Universitat Politècnica de Catalunya, C. Jordi Girona, 1–3, 08034 Barcelona, Spain

e-mail: matcpl@mat.upc.es

February 13, 2002

**Abstract**

A Key Distribution Center (KDC) of a network is a server enabling private communications within groups of users. The center provides the secret keys for encrypting and decrypting the messages. A Distributed Key Distribution Center (DKDC) is a set of servers that *jointly* realizes a Key Distribution Center. In this paper we introduce a ramp model for unconditionally secure Distributed Key Distribution Centers. In the ramp approach, the required resources (randomness, information storage, messages to be exchanged,...) can be reduced at the cost of a security degradation which depends on the size of the coalition of users who tries to break the scheme. We show lower bounds on the amount of information needed to setup and manage such a DKDC and describe a simple protocol meeting the bounds.

**keyword:** Key Distribution, Protocols, Distributed Systems.

## 1 Introduction

Key Distribution is an intriguing and deeply studied problem in Cryptography. A huge amount of literature can be found on this subject. Moreover, the most remarkable cryptographic idea of the last century, public key algorithms, was motivated by the necessity of solving this issue.

Roughly speaking, the problem can be described as follows: a group of users of a network, to privately communicate, could decide to use symmetric encryption algorithms, for example DES, RC6, or RIJNDAEL. These algorithms are fast and supposed to be secure. But to apply this strategy, they need a common key with which to encrypt and to decrypt the messages they will send to each other. On the other hand, the solution offered by public key cryptography is fascinating and it permits getting over the preliminary question of the common key: each user can generate a pair of keys, a public one and a private one. The first one can be used by any other user across the network to send encrypted messages to the owner of that key. The private key enables only the legitimate receiver to decrypt the encrypted messages. The security of the communication relies on the "computational in-feasibility" of recovering the private key from the public one. No preliminary common key must be hold by the communicating parties.

Currently, asymmetric algorithms are far away from symmetric ones in terms of computational efficiency, and this "efficiency-distance" grows up enormously if the group of users, usually referred to as a *conference*, has a big size: each user, to send a message to all the other members of the conference, needs to encrypt the message many times with different public keys. Thus, the computational effort required to the user can be dramatically heavy.

Moreover, notice that, once all users in a conference have a common key to be used in a symmetric encryption algorithm, a user has to encrypt a message just once to send it to all other users of the conference. Hence, the overall communication complexity is quite better than with the public key approach[1]. These few reasons help to figure out why it is still necessary to find good solutions for the key distribution problem.

In traditional models of networks, a frequently used approach is the *Key Distribution Center*, a server of the network 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 privately 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 [18] began this approach that is implemented most notably in the Kerberos System [19] and formally defined and studied in [7], where it is referred to as the *three party case*.

The scheme implemented by the Key Distribution Center to give each conference a key is referred to as 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.

Different kinds of Key Distribution Schemes have been considered: 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 [2, 16, 5]. BESs are designed to enable secure broadcast transmissions and have been introduced in [13]. The broadcast encryption idea has grown in various directions: traitor tracing [10], anonymous broadcast transmission [14], re-keying protocols for secure multi-cast communications [8, 9, 20].

Several unconditionally secure key distribution schemes have been proposed so far [22]. However, it is not difficult to see that all the previous designs of unconditionally secure key distribution schemes consider a centralized environment. Models and protocols assume the presence of a single server that accomplishes the key distribution task.

The unpleasant situation which often arises with a Key Distribution Center is that the center knows all the conference keys. Therefore, it must be *trusted*. Moreover, the Key Distribution Center could become a performance bottleneck and a point of failure for the system [17]. In effect, all users have to communicate with it every time they wish to obtain a conference key. Besides, a crash of the server stalls the whole system.

As has been pointed out in [17], in a multi-cast communication environment with support for virtual meetings involving thousands of clients, and data streams transmission to a large group of

[1] 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 of the conference, before starting the communication using a symmetric algorithm. However, this solution is still not efficient and it is possible to do better.

recipients, the availability and security issues of a centralized environment become even more relevant and difficult to solve than with unicast communication.

Well known and applied solutions to the availability and reliability issues are *replication* of the Key Distribution Center in several points of the network and *partition* of the network in several domains with dedicated Key Distribution Centers, responsible of the key management for only a fixed local area. However, these solutions are partial and expensive solutions [17].

A robust and efficient solution can be a Distributed Key Distribution Center [17] (DKDC, for short). A Distributed Key Distribution Center is a set of $n$ servers of a network that jointly realizes the same function of a Key Distribution Center. In this setting, each user shares *private channels* with all the servers. When a user needs to participate to a conference, he sends a key-request message to a subset at his choice of the $n$ servers. The contacted servers answer with some private information enabling the user to compute the conference key. With a DKDC the concentration of secrets and the slow down factor which arise in a network with a single Key Distribution Center are eliminated. A single server by itself does not know the secret keys, since they are *shared* between the $n$ servers. Moreover, each user can send a key-request in parallel to different servers. Hence, there is no loss in time to compute a conference key compared to a centralized environment. Besides, the users can obtain the keys they need even if they are unable to contact some of the servers.

This approach to key distribution has been proposed and developed in [17]. In [3, 12], the notion of Distributed Key Distribution Center has been studied under an information theoretic point of view. Besides, the authors have proved that the protocol described in [17], which uses bivariate polynomials, is optimal with respect to the resources required to set up and to manage the distributed center.

In this paper we extend the model presented in [3, 12]. Given the high complexity of the distribution mechanism therein described, we investigate the *ramp* approach, introduced in [1] in the context of secret sharing schemes. Basically, it allows to reduce the required resources (randomness, information storage, messages to be exchanged, ...) at the cost of a security degradation which depends on the size of the coalition of users who tries to break the scheme. More precisely, we want to consider a *ramp structure* for the DKDC, characterized by two thresholds $t_1$ and $t_2$, where coalitions of users of size $t$, with $t < t_1, t_1 \leq t < t_2, t \geq t_2$, are able, colluding with at most $k-1$ servers, respectively, to gain *no* information on a new conference key, *some* information, or the *whole* key. Basically, the ramp approach enables to gain a factor $\frac{1}{t_2 - t_1}$ in terms of memory storage, communication complexity and randomness, compared to the one-threshold case, by "splitting" the whole key in smaller pieces that can be recovered separately. The drawback is that coalitions of users, whose size is in between the two thresholds, from the values they have received from some servers in order to compute some keys, can gain partial information about new ones. In some situations this *trade-off* resources vs security can be suitable.

**Organization of the paper.** In Section 2, we introduce the model. Then, in Section 3, we give some technical lemmas needed in the rest of the paper. In Section 4 we show properties and lower bounds holding on the model. Finally, in Section 5, we describe a protocol which meets the bounds.

## 2 The Model

Let $\mathcal{U} = \{U_1, \ldots, U_m\}$ be a set of $m$ users, and let $S_1, \ldots, S_n$ be the $n$ servers of the network. Each user has *private connections* with all the servers (i.e., only the parties at both ends of the connection can read/write messages). 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. We assume that the initialization phase is done by $k$ servers say, without loss of generality, $S_1, \ldots, S_k$. Each of these servers, using a private source of randomness $r_i$, generates some information that it privately distributes to the others. More precisely, for $i = 1, \ldots, k$, server $S_i$ generates and sends to $S_j$, the value $\gamma_{i,j}$, where $j = 1, \ldots, n$. At the end of the initialization phase, each server $S_i$ stores some secret information $a_i = f(\gamma_{1,i}, \ldots, \gamma_{k,i})$, which can be computed from the information he has received. Assume that a group of users $C_h \subseteq \mathcal{U}$, referred to as a *conference*, wants to communicate privately. Each user $U_j$ in $C_h$, requiring a key for the conference $C_h$ (we denote such a key with $\kappa_h$), contacts $k$ servers at least. Then, server $S_i$, contacted by user $U_j$, checks[2] for membership of $U_j$ in $C_h$; if so, the server $S_i$ computes a value $y^h_{i,j} = F(a_i, j, h)$, which is a function of the private information $a_i$, $j$, and the index $h$ of the requested key. Otherwise, the server sets $y^h_{i,j} = \perp$, a special value which does convey no information on the conference key. Finally, the server sends the value $y^h_{i,j}$ to $U_j$. The users in $C_h$ compute the conference key as a function of the information received by the contacted servers, i.e., each user $U_j$ in $C_h$ computes $\kappa_h = G(y^h_{i_1,j}, \ldots, y^h_{i_k,j})$, where $i_1, \ldots, i_k$ are the indices of the servers he has contacted and $G$ is a publicly known function.
A Distributed Key Distribution Center must satisfy the following properties:

- When the initialization phase terminates, each server $S_i$ has to be able to compute his private information $a_i$.
- The private information $a_i$ of the server $S_i$ must be retrieved only if the server has received all the $k$ initializing values from servers $S_1, \ldots, S_k$.
- Each user in a conference $C_h \subseteq \mathcal{U}$ must be able to uniquely compute the conference key, after interacting with at least $k$ servers of his choice.
- A conference key must be secure against attacks performed by coalitions of servers, coalitions of users, and hybrid coalitions (servers and users).

We are interested in formalizing, within an information theoretic framework, the notion of a ramp Distributed Key Distribution Scheme. To this aim, we need to setup our notation.

- Let $\mathcal{C}$ be the family of all possible conferences on $\mathcal{U}$ that need to communicate privately. Suppose that these conferences are indexed by elements of $\mathcal{H} \subseteq \{0, 1, 2, \ldots\}$.
- For any coalition $G = \{U_{j_1}, \ldots, U_{j_g}\} \subseteq \mathcal{U}$ of users, denote by $\mathcal{C}_G$ the set of the conferences containing some user in $G$, and by $\mathcal{H}_G$ the set of the corresponding indices. In other words, $\mathcal{C}_G = \{C_h \in \mathcal{C} : C_h \cap G \neq \emptyset\}$, and $\mathcal{H}_G = \{h : C_h \in \mathcal{C}_G\}$.
- Let $\Gamma_{i,j}$ be the set of values $\gamma_{i,j}$ that, for $i = 1, \ldots, k$, can be sent by server $S_i$ to server $S_j$, for $j = 1, \ldots, n$, and let $\Gamma_j = \Gamma_{1,j} \times \Gamma_{2,j} \times \cdots \times \Gamma_{k,j}$ be the set of values that $S_j$, for $j = 1, \ldots, n$, can receive during the initialization phase. Analogously, for $Y = \{j_1, \ldots, j_s\}$, let $\Gamma_Y = \Gamma_{j_1} \times \cdots \times \Gamma_{j_s}$. Equally, for $X = \{i_1, \ldots, i_r\}$, we consider $\Gamma_{X,j} = \Gamma_{i_1,j} \times \cdots \times \Gamma_{i_r,j}$, and $\Gamma_{X,Y} = \Gamma_{i_1,j_1} \times \cdots \times \Gamma_{i_r,j_1} \times \cdots \times \Gamma_{i_1,j_s} \times \cdots \times \Gamma_{i_r,j_s}$.

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

- Let $K_h$ be the set of possible values of the key $\kappa_h$ corresponding to the conference $C_h \in \mathcal{C}$ and let $A_i$ be the set of possible values $a_i$ that the server $S_i$ can compute during the initialization phase. As before, for each set $X = \{h_1, \ldots, h_r\}$, let $K_X = K_{h_1} \times \cdots \times K_{h_r}$ and, for each set $Y = \{i_1, \ldots, i_s\}$, let $A_Y = A_{i_1} \times \cdots \times A_{i_s}$.

- Let $Y^h_{i,j}$ be the set of possible values $y^h_{i,j}$ that can be sent by the server $S_i$ when it receives a key-request message from user $U_j$ for the conference $C_h$, and, for each set $X = \{i_1, \ldots, i_r\}$, let $Y^h_{X,j} = Y^h_{i_1,j} \times \cdots \times Y^h_{i_r,j}$. Then, for each set $G = \{U_{j_1}, \ldots U_{j_g}\}$ let $Y^h_G = Y^h_{1,j_1} \times \cdots \times Y^h_{1,j_g} \times \ldots \times Y^h_{n,j_1} \times \cdots \times Y^h_{n,j_g}$, be the set of values the $n$ servers can send to the users belonging to $G$ for conference $C_h$. Finally, let $Y^X_G$ be the set of values that can be sent by the $n$ servers to the users of $G$ for computing keys for conferences in $X = \{h_1, \ldots, h_s\} \subseteq \mathcal{H}_G$, i.e., $Y^X_G = Y^{h_1}_G \times \cdots \times Y^{h_s}_G$.

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

Generalizing the definition given in [3, 12] for a Distributed Key Distribution Scheme, we define a *ramp* Distributed Key Distribution Scheme as follows[3]:

**Definition 2.1** *A ramp $(k, t_1, t_2, n, \mathcal{C})$-Distributed Key Distribution Scheme (for short, ramp $(k, t_1, t_2, n, \mathcal{C})$-DKDS) is a protocol which enables each user of $C_h \in \mathcal{C}$ to compute a common key $\kappa_h$ interacting with at least $k$ of the $n$ servers of the network. More precisely, the following properties are satisfied:*

1. *After the initialization phase, each server reconstructs his private information.* Formally, for each $i = 1, \ldots, n$, it holds that
$$H(\mathbf{A}_i | \mathbf{\Gamma}_i) = 0.$$

2. *Each of the $k$ servers is needed to compute $a_i$.* Formally, for each $X \subset \{1, \ldots, k\}$ such that $X \neq \{1, \ldots, k\}$, and $i \in \{1, \ldots, n\}$ it holds that
$$H(\mathbf{A}_i | \mathbf{\Gamma}_{X,i}) = H(\mathbf{A}_i).$$

3. *Each server can answer to the key requests.* Formally, for each conference $C_h \in \mathcal{C}$, for each $U_j \in C_h$, and for each $i = 1, \ldots, n$, it holds that
$$H(\mathbf{Y}^h_{i,j} | \mathbf{A}_i) = 0.$$

4. *Each user in $C_h \in \mathcal{C}$ can compute a common key $\kappa_h$ after contacting at least $k$ servers.* Formally, for each conference $C_h \in \mathcal{C}$, for each subset of $r \geq k$ indices $X = \{i_1, \ldots, i_r\} \subseteq \{1, \ldots, n\}$, and for each user $U_j \in C_h$, it holds that
$$H(\mathbf{K}_h | \mathbf{Y}^h_{X,j}) = 0.$$

[3]The reader unfamiliar with the entropy function is referred to Appendix A for the definition and some basic properties

5. *Each conference key is completely secure against coalitions $G$ of users of size $|G| < t_1$ and at most $k-1$ servers; on the other hand, it leaks some information if $t_1 \leq |G| < t_2$.* Formally, for each conference $C_h \in \mathcal{C}$, for each coalition of users $G = \{U_{j_1}, \ldots, U_{j_g}\}$, and for each subset $X = \{i_1, \ldots, i_{k-1}\} \subset \{1, \ldots, n\}$ it holds that

$$H(\mathbf{K}_h | \mathbf{Y}_G^{\mathcal{H}_G \setminus \{h\}} \mathbf{\Gamma}_X \mathbf{\Gamma}_{Z,N}) = \begin{cases} H(\mathbf{K}_h), & \text{if } |G| < t_1 \\ \frac{t_2 - |G|}{t_2 - t_1} \cdot H(\mathbf{K}_h), & \text{if } t_1 \leq |G| < t_2 \end{cases}$$

*where $Z = X \cap \{1, \ldots, k\}$ and $N = \{1, \ldots, n\}$.*

Notice that, Property 5 formalizes the security of the scheme. The worst case scenario to be aware of consists of coalitions of users $G$ (the information they can acquire during the run of the protocol is represented by $\mathbf{Y}_G^{\mathcal{H}_G \setminus \{h\}}$) and $k-1$ corrupt servers knowing $\mathbf{\Gamma}_{i_1} \ldots \mathbf{\Gamma}_{i_{k-1}}$ and $\mathbf{\Gamma}_{Z,N}$ (the random variable $\mathbf{\Gamma}_{Z,N}$ takes into account the possibility that among the corrupt servers there are some, involved in the initialization phase, which send out information to other servers). The condition assures that coalitions of users of size $|G| < t_1$, do not gain any information on a new key, while coalitions of users of size $|G|$, where $t_1 \leq |G| < t_2$, are able to recover some partial information on a new key. We point out that the above condition considers even the possibility that $G \cap C_h \neq \emptyset$. In this case, if a user who belongs to $C_h$ does not interact with a subset of $k$ servers, the information obtained by $G$ from previous executions of the protocol does not help to gain any information on a new key (in the terms of the ramp definition).

In the following, without loss of generality, we assume that for different $h, h' \in \mathcal{H}$, $H(\mathbf{K}_h) = H(\mathbf{K}_{h'})$, that is, we suppose that all conference keys $\kappa_h$ have the same size.

# 3 Some Technical Results

Definition 2.1 implies some important results. We need some technical lemmas in order to prove them.

The following simple lemma shows 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 3.1** *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 (7) and (13) of Appendix A imply

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

Since from the symmetry property of the mutual information (property (12) of Appendix A) we have that

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

then, $H(\mathbf{A}|\mathbf{C}) = H(\mathbf{A}|\mathbf{BC})$. On the other hand, from (13) of Appendix A, it results $H(\mathbf{A}|\mathbf{B}) \geq H(\mathbf{A}|\mathbf{BC})$. Therefore, $H(\mathbf{A}|\mathbf{B}) \geq H(\mathbf{A}|\mathbf{C})$, which proves the lemma. ■

For 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 proof of Lemma 3.1, we can show that

$$H(\mathbf{A}|\mathbf{BD}) \geq H(\mathbf{A}|\mathbf{CD}) = 0. \tag{1}$$

It is easy to see that, for any group $G$ of users the set of conference keys $\{K_h : C_h \in \mathcal{C}_G\}$ is univocally determined by the information that the $n$ servers send to users in $G$ when invoked for those conference keys. This is formally stated by next lemma.

**Lemma 3.2** *Let $G = \{U_{j_1}, \ldots, U_{j_g}\}$ be a group of users, and for each $r = 1, \ldots, \mathcal{H}_G$, let $S_r = \{s_1 \ldots, s_r\} \subseteq \mathcal{H}_G$. Then, it holds that $H(\mathbf{K}_{S_r}|\mathbf{Y}_G^{S_r}) = 0$.*

**Proof.** For $r = 1, \ldots, \mathcal{H}_G$, notice that,

$$\begin{aligned}
0 &\leq H(\mathbf{K}_{S_r}|\mathbf{Y}_G^{S_r}) \text{ (from (7) of Appendix A)} \\
&\leq \sum_{j=1}^{r} H(\mathbf{K}_{s_j}|\mathbf{Y}_G^{s_j}) \text{ (from (8) and (13) of Appendix A)} \\
&\leq \sum_{j=1}^{r} H(\mathbf{K}_{s_j}|\mathbf{Y}_{X,t}^{s_j}) \text{ (from (13) of Appendix A where } t \in C_{s_j} \cap G \text{ and } X = \{i_1, \ldots, i_k\}) \\
&= 0 \text{ (from Property 4 of Definition 2.1).}
\end{aligned}$$

Hence, the lemma holds. ■

The following two lemmas are useful to show lower bounds on the size of the information that each of the initializing server $S_1, \ldots, S_k$ has to send to the other servers during the initialization phase and on the randomness (to be defined later) needed to setup a ramp $(k, t_1, t_2, n, \mathcal{C})$-DKDS.

For every $s = 1, 2, \ldots, t_2 - t_1$, define $g_s = t_1 + s - 1$ (observe that $g_1 = t_1$ and $g_{t_2-t_1} = t_2 - 1$). Let $\ell_1, \ell_2, \ldots, \ell_{t_2-t_1}$ be integers representing the maximum of the cardinalities of $\mathcal{H}_G$ for all $G$ of size, respectively, $g_1, g_2, \ldots, g_{t_2-t_1}$ (i.e., $l_i = \max_{G:|G|=g_i} |\mathcal{H}_G|$). Finally, we consider $\ell = \sum_{s=1}^{t_2-t_1} \ell_s$.

**Lemma 3.3** *Let $X_i = \{1, \ldots, k\} \setminus \{i\}$, for $i = 1, ..., k$, and let $A \subset \{1, \ldots, n\} \setminus \{j\}$, for $j = 1, ..., n$, be a set of size at most $k - 1$. Then, in any ramp $(k, t_1, t_2, n, \mathcal{C})$-DKDS it holds that*

$$H(\mathbf{\Gamma}_{i,j}|\mathbf{\Gamma}_A \mathbf{\Gamma}_{X_{i,j}}) \geq \ell \cdot \frac{H(\mathbf{K})}{t_2 - t_1}.$$

**Proof.** Let us consider a sequence of subsets of users

$$G_1 \subset G_2 \subset \cdots \subset G_{t_2-t_1} \subset \mathcal{U}$$

such that $G_s$ has cardinality $g_s$. We take $\mathcal{H}_1 = \mathcal{H}_{G_1}$ and, for any $s = 2, \ldots, t_2 - t_1$, we set $\mathcal{H}_s = \mathcal{H}_{G_s} \setminus \mathcal{H}_{G_{s-1}}$.

Let $B \subset \{1, 2, \ldots, n\} \setminus (A \cup \{j\})$ be a set of size $|B| = k - 1 - |A|$. Setting $\mathbf{\Gamma}^{(1)} = \mathbf{\Gamma}_A \mathbf{\Gamma}_{X_{i,j}} \mathbf{\Gamma}_B$, $\mathbf{\Gamma}^{(2)} = \mathbf{\Gamma}^{(1)} \mathbf{\Gamma}_{i,j}$, and $S = \mathcal{H}_{G_{t_2-t_1}}$, we will prove that

$$H(\mathbf{K}_S|\mathbf{\Gamma}^{(1)}) \geq \ell \cdot \frac{H(\mathbf{K})}{t_2 - t_1} \text{ and } H(\mathbf{K}_S|\mathbf{\Gamma}^{(2)}) = 0. \tag{2}$$

If we suppose that the equalities in (2) are satisfied, then from relation (13) of Appendix A, it holds that

$$H(\mathbf{\Gamma}_{i,j}|\mathbf{\Gamma}_A\mathbf{\Gamma}_{X,j}) \geq H(\mathbf{\Gamma}_{i,j}|\mathbf{\Gamma}^{(1)}).$$

Besides, relation (12) of Appendix A implies that

$$\begin{aligned}
H(\mathbf{\Gamma}_{i,j}|\mathbf{\Gamma}^{(1)}) &= H(\mathbf{K}_S|\mathbf{\Gamma}^{(1)}) - H(\mathbf{K}_S|\mathbf{\Gamma}^{(2)}) + H(\mathbf{\Gamma}_{i,j}|\mathbf{K}_S\mathbf{\Gamma}^{(1)}) \\
&\geq \ell \cdot \frac{H(\mathbf{K})}{t_2 - t_1} + H(\mathbf{\Gamma}_{i,j}|\mathbf{K}_S\mathbf{\Gamma}^{(1)}) \text{ (from (2))} \\
&\geq \ell \cdot \frac{H(\mathbf{K})}{t_2 - t_1} \text{ (from (7) of Appendix A)}
\end{aligned}$$

Hence, the lemma holds.

We are left with proving that equalities (2) are satisfied. Let $K_{(s)} = K_{\mathcal{H}_s}$. First, we notice that, from (8) in Appendix A,

$$H(\mathbf{K}_S|\mathbf{\Gamma}^{(1)}) = H(\mathbf{K}_{(1)}\mathbf{K}_{(2)}\ldots\mathbf{K}_{(t_2-t_1)}|\mathbf{\Gamma}^{(1)}) = \sum_{s=1}^{t_2-t_1} H(\mathbf{K}_{(s)}|\mathbf{K}_{(1)}\ldots\mathbf{K}_{(s-1)}\mathbf{\Gamma}^{(1)}) \tag{3}$$

On the other hand, from (13) in Appendix A and from Lemma 3.2, we have that, for any $h \in \mathcal{H}_s$,

$$H(\mathbf{K}_{\mathcal{H}_{G_s}\setminus\{h\}}|\mathbf{Y}_{G_s}^{\mathcal{H}_{G_s}\setminus\{h\}}) = 0.$$

Then, setting $\mathbf{A} = \mathbf{K}_h$, $\mathbf{B} = \mathbf{K}_{\mathcal{H}_{G_s}\setminus\{h\}}$, $\mathbf{C} = \mathbf{Y}_{G_s}^{\mathcal{H}_{G_s}\setminus\{h\}}$, and $\mathbf{D} = \mathbf{\Gamma}^{(1)}$ and applying inequality (1), we obtain

$$H(\mathbf{K}_h|\mathbf{K}_{\mathcal{H}_{G_s}\setminus\{h\}}\mathbf{\Gamma}^{(1)}) \geq H(\mathbf{K}_h|\mathbf{Y}_{G_s}^{\mathcal{H}_{G_s}\setminus\{h\}}\mathbf{\Gamma}^{(1)}). \tag{4}$$

Therefore,

$$\begin{aligned}
H(\mathbf{K}_{(s)}|\mathbf{K}_{(1)}\ldots\mathbf{K}_{(s-1)}\mathbf{\Gamma}^{(1)}) &\geq \sum_{h\in\mathcal{H}_s} H(\mathbf{K}_h|\mathbf{K}_{\mathcal{H}_{G_s}\setminus\{h\}}\mathbf{\Gamma}^{(1)}) \text{ (from (8) in Appendix A)} \\
&\geq \sum_{h\in\mathcal{H}_s} H(\mathbf{K}_h|\mathbf{Y}_{G_s}^{\mathcal{H}_{G_s}\setminus\{h\}}\mathbf{\Gamma}^{(1)}) \text{ (from (4))} \\
&\geq \sum_{h\in\mathcal{H}_s} \frac{t_2 - |G_s|}{t_2 - t_1} H(\mathbf{K}) \text{ (from Property 5 of Definition 2.1)} \\
&= (\ell_s - \ell_{s-1}) \frac{t_2 - t_1 - s + 1}{t_2 - t_1} H(\mathbf{K}),
\end{aligned}$$

where we put $\ell_0 = 0$. Finally,

$$\begin{aligned}
H(\mathbf{K}_S|\mathbf{\Gamma}^{(1)}) &= \sum_{s=1}^{t_2-t_1} H(\mathbf{K}_{(s)}|\mathbf{K}_{(1)}\ldots\mathbf{K}_{(s-1)}\mathbf{\Gamma}^{(1)}) \text{ (from (3))} \\
&\geq \sum_{s=1}^{t_2-t_1} (\ell_s - \ell_{s-1}) \frac{t_2 - t_1 - s + 1}{t_2 - t_1} H(\mathbf{K})
\end{aligned}$$

$$
\begin{aligned}
&= \ell_1 \cdot \frac{1}{t_2 - t_1} H(\mathbf{K}) + \sum_{s=2}^{t_2-t_1} \ell_s \left( \frac{t_2 - t_1 - s + 1}{t_2 - t_1} - \frac{t_2 - t_1 - s}{t_2 - t_1} \right) H(\mathbf{K}) \\
&= \ell \cdot \frac{H(\mathbf{K})}{t_2 - t_1},
\end{aligned}
$$

where $\ell = \sum_{s=1}^{t_2-t_1} \ell_s$.

Now, we have to prove that $H(\mathbf{K}_S|\mathbf{\Gamma}^{(2)}) = 0$. Notice that, from Property 1 of Definition 2.1 we get that $H(\mathbf{A}_X|\mathbf{\Gamma}^{(2)}) = 0$, where $X = A \cup B \cup \{j\}$. Applying Property 3 of Definition 2.1 we get that $H(\mathbf{Y}^h_{X,i}|\mathbf{A}_X) = 0$ for any $s \in S$ and for any user $U_i \in C_h \cup G_{t_2-t_1}$. Therefore,

$$
\begin{aligned}
0 \leq H(\mathbf{K}_S|\mathbf{\Gamma}^{(2)}) &\leq \sum_{h\in S} H(\mathbf{K}_h|\mathbf{\Gamma}^{(2)}) \\
&\leq \sum_{h\in S} H(\mathbf{K}_h|\mathbf{A}_X) \text{ (from Lemma 3.1)} \\
&\leq \sum_{h\in S} H(\mathbf{K}_h|\mathbf{Y}^h_{X,i}) \text{ (from Lemma 3.1)} \\
&= 0 \text{ (from Property 4 of Definition 2.1)}
\end{aligned}
$$

Thus, equalities (2) are satisfied and the lemma holds. ■

The next result is a consequence of the above lemma.

**Lemma 3.4** *In any ramp $(k, t_1, t_2, n, \mathcal{C})$-DKDS, for each $j = 1, \ldots, n$, and for each set $Y \subset \{1, \ldots, n\} \setminus \{j\}$ of size at most $k-1$, it holds that*

$$
H(\mathbf{\Gamma}_j|\mathbf{\Gamma}_Y) \geq k \cdot \ell \cdot \frac{H(\mathbf{K})}{t_2 - t_1}.
$$

**Proof**. We have that,

$$
\begin{aligned}
H(\mathbf{\Gamma}_j|\mathbf{\Gamma}_Y) &= \sum_{i=1}^{k} H(\mathbf{\Gamma}_{i,j}|\mathbf{\Gamma}_Y \mathbf{\Gamma}_{1,j} \ldots \mathbf{\Gamma}_{i-1,j}) \text{ (from (8) of Appendix A)} \\
&= \sum_{i=1}^{k} H(\mathbf{\Gamma}_{i,j}|\mathbf{\Gamma}_Y \mathbf{\Gamma}_{X,j}) \text{ (from (13) of Appendix A setting } X = \{1, \ldots, k\} \setminus i) \\
&\geq \sum_{i=1}^{k} \ell \cdot \frac{H(\mathbf{K})}{t_2 - t_1} \text{ (from Lemma 3.3)} \\
&= k \cdot \ell \cdot \frac{H(\mathbf{K})}{t_2 - t_1}.
\end{aligned}
$$

Thus, the lemma holds. ■

# 4 Properties and Bounds

In this section, we show some properties of our model. We present lower bounds on the size of the piece of information that each server distributing information during the initialization phase has to send to the other servers, on the size of the piece each server has to store to answer to the key-request messages, and on the size of the piece of information each server has to send upon receiving a user's key-request message. Finally, we present a lower bound on the randomness needed to setup a Distributed Key Distribution Center.

The following theorem establishes a lower bound both on the size of information $\gamma_{i,j}$ that each of the initializing servers, $S_1, \ldots, S_k$, has to send during the setup phase to $S_1, \ldots, S_n$, and on the size of information $\gamma_i$ that each server must receive in order to be able to compute his private information $a_i$.

**Theorem 4.1** *In any ramp $(k, t_1, t_2, n, \mathcal{C})$-DKDS, for each $i = 1, \ldots, k$, and $j = 1, \ldots, n$, the following inequalities are satisfied:*

$$H(\mathbf{\Gamma}_{i,j}) \geq \ell \cdot \frac{H(\mathbf{K})}{t_2 - t_1}, \ \textit{and} \ H(\mathbf{\Gamma}_j) \geq k \cdot \ell \cdot \frac{H(\mathbf{K})}{t_2 - t_1}.$$

**Proof.** Notice that, from (9) of Appendix A and from Lemma 3.3, we have that

$$H(\mathbf{\Gamma}_{i,j}) \geq H(\mathbf{\Gamma}_{i,j} | \mathbf{\Gamma}_A \mathbf{\Gamma}_{X_i,j}) \geq \ell \cdot \frac{H(\mathbf{K})}{t_2 - t_1}.$$

Along the same line, from (9) of Appendix A and from Lemma 3.4, we have that

$$H(\mathbf{\Gamma}_j) \geq H(\mathbf{\Gamma}_j | \mathbf{\Gamma}_Y) \geq k \cdot \ell \cdot \frac{H(\mathbf{K})}{t_2 - t_1}.$$

Thus, the theorem holds. ∎

Using some basic properties of the entropy function, we can obtain a lower bound on the size of the piece of information that each server, contacted by a user, has to send upon receiving a key-request message. This is formally stated by next theorem.

**Theorem 4.2** *In any ramp $(k, t_1, t_2, n, \mathcal{C})$-DKDS, for any $C_h \in \mathcal{C}$, for any $i = 1, \ldots, n$, and for any $U_j \in C_h$, it holds that*

$$H(\mathbf{Y}^h_{i,j}) \geq H(\mathbf{K}).$$

**Proof.** Let $X = \{i_1, \ldots, i_{k-1}\} \subset \{1, \ldots, n\}$. For $i \notin X$, equation (9) of Appendix A implies that $H(\mathbf{Y}^h_{i,j}) \geq H(\mathbf{Y}^h_{i,j} | \mathbf{Y}^h_{X,j})$. Applying equation (12) of Appendix A, we can write

$$H(\mathbf{Y}^h_{i,j} | \mathbf{Y}^h_{X,j}) \;\; = \;\; H(\mathbf{K}_h | \mathbf{Y}^h_{X,j}) - H(\mathbf{K}_h | \mathbf{Y}^h_{X,j} \mathbf{Y}^h_{i,j}) + H(\mathbf{Y}^h_{i,j} | \mathbf{Y}^h_{X,j} \mathbf{K}_h).$$

According to Property 4 of Definition 2.1, one gets $H(\mathbf{K}_h | \mathbf{Y}^h_{X,j} \mathbf{Y}^h_{i,j}) = 0$. Moreover, we can prove that $H(\mathbf{K}_h | \mathbf{Y}^h_{X,j}) = H(\mathbf{K}_h)$ and since equation (7) of Appendix A implies that $H(\mathbf{Y}^h_{i,j} | \mathbf{Y}^h_{X,j} \mathbf{K}_h) \geq 0$, we can conclude that

$$H(\mathbf{Y}^h_{i,j}) \geq H(\mathbf{K}_h) = H(\mathbf{K}),$$

which proves the theorem. Hence, we have to prove that $H(\mathbf{K}_h | \mathbf{Y}^h_{X,j}) = H(\mathbf{K}_h)$. Property (9) of Appendix A implies that $H(\mathbf{K}_h | \mathbf{Y}^h_{X,j}) \leq H(\mathbf{K}_h)$. On the other hand, notice that from Property 3 of Definition 2.1 we get that $H(\mathbf{Y}^h_{X,j} | \mathbf{A}_X) = 0$; setting $\mathbf{A} = \mathbf{K}_h$, $\mathbf{B} = \mathbf{Y}^h_{X,j}$, and $\mathbf{C} = \mathbf{A}_X$ and applying Lemma 3.1, it results that

$$H(\mathbf{K}_h | \mathbf{Y}^h_{X,j}) \geq H(\mathbf{K}_h | \mathbf{A}_X). \tag{5}$$

Moreover, applying Property 1 of Definition 2.1 we get that $H(\mathbf{A}_X|\mathbf{\Gamma}_X) = 0$; setting $\mathbf{A} = \mathbf{K}_h$, $\mathbf{B} = \mathbf{A}_X$, and $\mathbf{C} = \mathbf{\Gamma}_X$ and applying and Lemma 3.1, it results that

$$H(\mathbf{K}_h|\mathbf{A}_X) \geq H(\mathbf{K}_h|\mathbf{\Gamma}_X). \tag{6}$$

But, from (6) and (9) of Appendix A, and from Property 5 of Definition 2.1 we get

$$H(\mathbf{K}_h|\mathbf{\Gamma}_X) \;\; \geq \;\; H(\mathbf{K}_h|\mathbf{Y}^{\mathcal{H}_G\setminus\{h\}}\mathbf{\Gamma}_X\mathbf{\Gamma}_{Z,N}) \geq \frac{t_2 - |G|}{t_2 - t_1} \cdot H(\mathbf{K}),$$

where $X = \{i_1, \ldots, i_{k-1}\}$, $Z = X \cap \{1, \ldots, k\}$, and $N = \{1, \ldots, n\}$. If we consider a coalition $G$ with $|G| = t_1$, we have

$$H(\mathbf{K}_h|\mathbf{Y}^h_{X,j}) \geq H(\mathbf{K}_h|\mathbf{\Gamma}_X) \geq H(\mathbf{K})$$

and the theorem holds. ∎

We can also show that each server, to answer the user's key-request messages, has to store some information whose size is lower bounded by $\ell \cdot \frac{H(\mathbf{K})}{t_2 - t_1}$.

**Theorem 4.3** *In any ramp $(k, t_1, t_2, n, \mathcal{C})$-DKDS, for each $i = 1, \ldots, n$, the private information $a_i$, stored by the server $S_i$, satisfies*

$$H(\mathbf{A}_i) \geq \ell \cdot \frac{H(\mathbf{K})}{t_2 - t_1}.$$

**Proof.** Let $G = \{U_{j_1}, \ldots, U_{j_g}\}$ be a group of users, and let $\mathcal{H}_G = \{s_1, \ldots, s_\ell\}$. Moreover, let $X = \{i_1, \ldots, i_k\} \subset \{1, \ldots, n\}$. For each $r = 1, \ldots, k$, consider the mutual information between $\mathbf{A}_{i_r}$ and $\mathbf{K}_{\mathcal{H}_G}$ given $\mathbf{A}_{X\setminus i_r}$. Applying equation (12) of Appendix A, we can write

$$H(\mathbf{A}_{i_r}|\mathbf{A}_{X\setminus i_r}) \;\; = \;\; H(\mathbf{K}_{\mathcal{H}_G}|\mathbf{A}_{X\setminus i_r}) - H(\mathbf{K}_{\mathcal{H}_G}|\mathbf{A}_X) + H(\mathbf{A}_{i_r}|\mathbf{A}_{X\setminus i_r}\mathbf{K}_{\mathcal{H}_G}).$$

We can prove that

$$H(\mathbf{K}_{\mathcal{H}_G}|\mathbf{A}_{X\setminus i_r}) \geq \ell \cdot \frac{H(\mathbf{K})}{t_2 - t_1} \quad \text{and} \quad H(\mathbf{K}_{\mathcal{H}_G}|\mathbf{A}_X) = 0.$$

Equation (9) of Appendix A implies that $H(\mathbf{A}_{i_r}) \geq H(\mathbf{A}_{i_r}|\mathbf{A}_{X\setminus i_r})$ and since property (7) of Appendix A implies $H(\mathbf{A}_{i_r}|\mathbf{A}_{X\setminus i_r}\mathbf{K}_{s_1}\ldots\mathbf{K}_{s_\ell}) \geq 0$, we have that

$$H(\mathbf{A}_i) \geq \ell \cdot H(\mathbf{K}),$$

which proves the theorem. Hence, we have to prove that $H(\mathbf{K}_{\mathcal{H}_G}|\mathbf{A}_{X\setminus i_r}) \geq \ell \cdot \frac{H(\mathbf{K})}{t_2 - t_1}$. Indeed, from Property 1 of Definition 2.1, and Lemma 3.1 it holds that

$$\begin{aligned}
H(\mathbf{K}_{\mathcal{H}_G}|\mathbf{A}_{X\setminus i_r}) \;\; &\geq \;\; H(\mathbf{K}_{\mathcal{H}_G}|\mathbf{\Gamma}_{X\setminus i_r}) \\
&= \;\; \sum_{r=1}^{\ell} H(\mathbf{K}_{s_r}|\mathbf{\Gamma}_{X\setminus i_r}\mathbf{K}_{\mathcal{H}_G\setminus s_r}) \text{ (applying (8) of Appendix A)} \\
&\geq \;\; \sum_{r=1}^{\ell} H(\mathbf{K}_{s_r}|\mathbf{\Gamma}_{X\setminus i_r}\mathbf{Y}_G^{\mathcal{H}_G\setminus s_r}) \text{ (from property (13) of Appendix A)} \\
&\geq \;\; \ell \cdot \frac{H(\mathbf{K})}{t_2 - t_1} \text{ (from Property 5 of Definition 2.1).}
\end{aligned}$$

Moreover, from property (7) of Appendix A, $H(\mathbf{K}_{\mathcal{H}_G}|\mathbf{A}_X) \geq 0$, while, applying the chain rule, we have that

$$
\begin{aligned}
H(\mathbf{K}_{\mathcal{H}_G}|\mathbf{A}_X) &\leq \sum_{r=1}^{\ell} H(\mathbf{K}_{s_r}|\mathbf{A}_X \mathbf{K}_{\mathcal{H}_G \setminus i_r}) \\
&\leq \sum_{r=1}^{\ell} H(\mathbf{K}_{s_r}|\mathbf{A}_X) \text{ (from property (13) of Appendix A)} \\
&\leq \sum_{r=1}^{\ell} H(\mathbf{K}_{s_r}|\mathbf{Y}_{X,j}^{s_r}) \text{ (from Property 3 of Definition 2.1 and Lemma 3.1)} \\
&\leq 0 \text{ (choosing } j \in G\text{, and applying Property 4 of Definition 2.1)}
\end{aligned}
$$

Therefore, the theorem holds. ■

**Communication Complexity** The Communication Complexity ($\mathcal{CC}$, for short) of a ramp DKDS is measured by the amount of information sent by the servers $S_1, \ldots, S_k$ during the initialization phase. It is not difficult to see that

$$
\mathcal{CC} = \sum_{i=1}^{k} \sum_{j=1}^{n} H(\mathbf{\Gamma}_{i,j}) \geq k \cdot \ell \cdot n \cdot \frac{H(\mathbf{K})}{t_2 - t_1}.
$$

**Randomness** When we want to set up a cryptographic protocol as, in this case, a ramp Distributed Key Distribution Scheme, often we need a number of random bits. This resource is usually referred to as the *randomness* of the scheme. The randomness of a scheme can be measured in several ways. Knuth and Yao [15] 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 $T(P) = \min_{\texttt{Alg}} T(\texttt{Alg})$. $T(P)$ is a measure of the average number of random bits needed to simulate the random source described by the probability distribution $P$. In [15] it has been proved the following result

**Theorem 4.4** $H(\mathbf{P}) \leq T(\mathbf{P}) < H(\mathbf{P}) + 2$.

Hence, the entropy is a natural measure of the randomness of a ramp DKDS and the randomness $\mathcal{R}$ of a Distributed Key Distribution Scheme can be lower bounded by $H(\mathbf{\Gamma}_1 \ldots \mathbf{\Gamma}_n)$. The next theorem holds

**Theorem 4.5** *In any ramp $(k, t_1, t_2, n, \mathcal{C})$-DKDS $\mathcal{R}$ satisfies*

$$
\mathcal{R} \geq k^2 \cdot \ell \cdot \frac{H(\mathbf{K})}{t_2 - t_1}.
$$

**Proof.** Notice that, from Theorem 4.4 and property (11) of Appendix A, we have

$$
\begin{aligned}
\mathcal{R} \geq H(\mathbf{\Gamma}_1 \ldots \mathbf{\Gamma}_n) &\geq H(\mathbf{\Gamma}_{j_1} \ldots \mathbf{\Gamma}_{j_k}) \text{ (for each } \{j_1, \ldots, j_k\} \subset \{1, \ldots, n\}) \\
&= \sum_{r=1}^{k} H(\mathbf{\Gamma}_{j_r}|\mathbf{\Gamma}_{j_1} \ldots \mathbf{\Gamma}_{j_r - 1}) \text{ (applying (8))}
\end{aligned}
$$

$$
\begin{aligned}
&\geq \sum_{r=1}^{k} H(\mathbf{\Gamma}_{j_r}|\mathbf{\Gamma}_Y) \text{ (if } Y = \{j_1, \ldots, j_k\} \setminus \{j_r\} \text{ applying (13))} \\
&\geq \sum_{r=1}^{k} k \cdot \ell \cdot \frac{H(\mathbf{K})}{t_2 - t_1} \text{ (from Lemma 3.4)} \\
&= k^2 \cdot \ell \cdot \frac{H(\mathbf{K})}{t_2 - t_1}.
\end{aligned}
$$

Thus, the theorem holds. ∎

## 5 A Protocol meeting the bounds

In this section we propose a protocol which meets the bounds of Theorems 4.1, 4.2, 4.3, and 4.5.

Let $s = t_2 - t_1$ and let $\ell_1, \ell_2, \ldots, \ell_s$ be integers representing the maximum of the cardinalities of $\mathcal{H}_G$ for all coalitions of users $G$ of size, respectively, $g_1, \ldots, g_s$, where $g_1 = t_1$, $g_i = t_1 + i - 1$, for $i = 2, \ldots, s-1$, and $g_s = t_2 - 1$. We assume that a conference key is a tuple of $s$ elements belonging to the finite field $Z_q$, for a certain prime $q$. The protocol is as follows:

Initialization Phase

- For $i = 1, \ldots, k$, server $S_i$, constructs $s$ random bivariate polynomials $P^i_{\ell_1}(x, y), \ldots, P^i_{\ell_s}(x, y)$ of degree $k-1$ in $x$, and, respectively $\ell_1 - 1, \ldots, \ell_s - 1$ in $y$ by choosing $k \cdot \ell_1, \ldots, k \cdot \ell_s$ random elements in $Z_q$.
- Then, for $i = 1, \ldots, k$, $S_i$ computes, for each $\ell \in \{\ell_1, \ldots, \ell_s\}$ and for each $j = 1, \ldots, n$, the polynomial $Q^i_{j,\ell}(y) = P^i_\ell(j, y)$, and sends $Q^i_{j,\ell}(y)$ to the server $S_j$. For $j = 1, \ldots, n$, server $S_j$ computes, for each $\ell \in \{\ell_1, \ldots, \ell_s\}$, the polynomial $Q_{j,\ell}(y) = \sum_{i=1}^{k} Q^i_{j,\ell}(y)$. These $s$ polynomials form the secret information $a_j$ stored by server $S_j$.

Key Request Phase

- A user in conference $C_h$, who wants to compute the conference key, sends a request to at least $k$ servers.
- Each server $S_j$, invoked by the user, checks that the user belongs to $C_h$, and sends to the user the values $Q_{j,\ell_1}(h), \ldots, Q_{j,\ell_s}(h)$.

Key Computation Phase

- Using the $k$ sets of values received by the servers and by polynomial interpolations, each user in $C_h$ recovers the secret key $\kappa_h = [P_{\ell_1}(0, h), \ldots, P_{\ell_s}(0, h)]$, where, for each $\ell \in \{l_1, \ldots, l_s\}$, we have $P_\ell(x, y) = \sum_{i=1}^{k} P^i_\ell(x, y)$.

**Correctness.** It is immediate to see that the protocol satisfies Definition 2.1. Indeed, as required from Properties 1 and 2 of Definition 2.1, for $i = 1, \ldots, n$, server $S_i$ can compute his private information if and only if he receives the values $\gamma_{1,i}, \ldots, \gamma_{k,i}$ sent by the $k$ servers performing the initialization phase. Each server, holding $a_i$, can answer to the conference key requests, as required by Property 3 of Definition 2.1. Moreover, each user interacting with $k$ servers, can recover a key for a conference in which he does belong to, satisfying Property 4 of Definition 2.1.

**Security.** The security property of the protocol can be shown as follows: The worst case scenario we have to consider consists of a coalition of any $k-1$ servers performing the inizialization phase, and a set of users $G \subseteq \mathcal{U}$ which can compute exactly $\ell_i$ different conference keys. Without loss of generality, assume that the servers involved in the attack are $S_1, \ldots, S_{k-1}$. They know the polynomials $Q_{1,\ell_1}(y), \ldots, Q_{1,\ell_s}(y), \ldots, Q_{k-1,\ell_1}(y), \ldots, Q_{k-1,\ell_s}(y)$ and, for $i = 1, \ldots, k-1$ and $j = 1, \ldots, n$, the polynomials $Q^i_{j,\ell_1}(y), \ldots, Q^i_{j,\ell_s}(y)$, i.e., the "partial" polynomials they send to the other servers during the initialization phase. If the users in $G$ have required $\ell_i$ keys for conferences in which they belong to, they are able to interpolate at most the first $\ell_i - \ell_1$ polynomials of degree $\ell_1, \ldots, \ell_i$ in $y$, but they have no information about the other of degree $\ell_{i+1}, \ldots, \ell_s$. Thus, about the $\ell_{i+1}$-th key, they can compute the first $\ell_i - \ell_1$ values but nothing about the other ones. Indeed, they can receive $k-1$ sets of points from the dishonest servers but they have no information about the $k$-th set of points which could send them another server of the system. As we have seen, the protocol establishes that each server, during the initialization phase, can compute its polynomial only if it receives $k$ different partial polynomials (we recall that $Q_{j,\ell_i}(y) = \sum_{i=1}^{k} Q^i_{j,\ell_i}(y)$). Notice that, this means $k-1$ initializing servers do not have information about the polynomials of another server since, for each choice of the $k$-th initializing server, these polynomials can be different. Hence, $S_1, \ldots, S_{k-1}$ cannot guess the values that a $k$-th server of the system can send to a user after a key-request message better than if they choose the value at random. Therefore, Property 5 of Definition 2.1 is satisfied.

**Tightness.** The protocol shows that the bounds of Theorems 4.1, 4.2, 4.3, and 4.5 are tight. Indeed, a key is an $s$-tuple of random values (uniformly chosen) of $Z_q$, and, hence, $H(\mathbf{K}) = s \cdot \log q$. Each server, performing the distribution during the initialization phase, sends exactly $\ell = \sum \ell_i$ values of $Z_q$ to any other server. Hence, any server during the initialization phase receives $k \cdot \ell$ values of $Z_q$. Moreover, each server adds the $k$ sets of values received from each server and stores exactly $\ell$ new values of $Z_q$. The communication complexity of the distribution phase is exactly $k \cdot n \cdot \ell \cdot \log q$ bits. At a user's key-request message, a server answers with an $s$-tuple of values of $Z_q$. Finally, the randomness needed to setup the scheme is $k^2 \cdot \ell \cdot \log q$ bits, since each of the $k$ servers performing the distribution during the initialization phase randomly chooses $k \cdot \ell$ values in $Z_q$.

**Remark.** Notice that when the two thresholds $t_1$ and $t_2$ of a ramp $(k, t_1, t_2, n, \mathcal{C})$-DKDS are defined by $t_2 = t_1 + 1$, then we recover the notion of $(k, n, \mathcal{C})$-DKDS given in [3, 12]. Moreover, the above protocol is exactly the protocol given in [17] and showed to be optimal in [3, 12]. Basically, the ramp approach enables to gain a factor $\frac{1}{t_2 - t_1}$ in terms of memory storage, communication complexity and randomness, compared to the one-threshold case, by "splitting" the whole key in smaller pieces that can be recovered separately. In other words, the *real key* belongs in $Z_{q^s}$ but is represented by an $s$-tuple belonging to $(Z_q)^s$ and can be recovered by applying a certain mapping $\phi : (Z_q)^s \rightarrow Z_{q^s}$. The drawback is that coalitions of users, whose size is in between the two thresholds, from the values they have received in order to compute some keys, can gain partial information about new ones.

## References

[1] G.R. Blakley and C. Meadows, *Security of Ramp Schemes*, Advances in Cryptology: Crypto '84, pp. 547-559, Lecture Notes in Computer Science, vol. 196, pp. 242-268, 1984.

[2] R. Blom, *An Optimal Class of Symmetric Key Generation Systems*, Advances in Cryptology - Eurocrypt 84, Lecture notes in Computer Science, vol. 209, pp. 335–338, 1984.

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

[4] C. Blundo, and P. D'Arco, *An Information Theoretic Model for Distributed Key Distribution*, Proceedings of the 2000 IEEE International Symposium on Information Theory, pp. 270, 2000.

[5] 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.

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

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

[8] 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.

[9] R. Canetti, T. Malkin and K. Nissim, *Efficient Communication-Storage Tradeoffs for Multicast Encryption*, Advances in Cryptology - Eurocrypt 99, Lecture Notes in Computer Science, vol. 1592, pp. 459–474, 1999.

[10] B. Chor, A. Fiat, and M. Naor, *Traicing Traitors*, Advances in Cryptology - Eurocrypt , Lecture Notes in Computer Science, vol. pp. 257–270, 1994.

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

[12] 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.

[13] A. Fiat and M. Naor, *Broadcast Encryption*, Advances in Cryptology - Crypto 92, Lecture Notes in Computer Science, vol. 773, pp. 480–491, 1993.

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

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

[16] T. Matsumoto and H. Imai, *On the Key Predistribution System: A Pratical Solution to the Key Distribution Problem*, Advances in Cryptology - Eurocrypt 87, Lecture Notes in Computer Science, vol. 239, pp. 185–193, 1987.

[17] M. Naor, B. Pinkas, and O. Reingold, *Distributed Pseudo-random Functions and KDCs*, Advances in Cryptology - Eurocrypt 99, Lecture notes in Computer Science, vol. 1592, pp. 327–346, 1999.

[18] 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.

[19] B. C. Neuman and T. Tso, *Kerberos: an authentication service for computer networks*, IEEE Transaction on Communications. vol. 32, pp. 33–38, 1994.

[20] 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 Computer Science, vol. 1666, pp. 624–638, 1999.

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

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

## A Information Theory Elements

This appendix briefly recalls some elements of information theory (see [11] 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 $0 \leq H(\mathbf{X}) \leq \log|X|$, 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

$$H(\mathbf{X}|\mathbf{Y}) \geq 0. \tag{7}$$

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

$$H(\mathbf{X}_1 \ldots \mathbf{X}_n|\mathbf{Y}) = H(\mathbf{X}_1|\mathbf{Y}) + H(\mathbf{X}_2|\mathbf{X}_1\mathbf{Y}) + \cdots + H(\mathbf{X}_n|\mathbf{X}_1 \ldots \mathbf{X}_{n-1}\mathbf{Y}). \tag{8}$$

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

$$H(\mathbf{X}) \geq H(\mathbf{X}|\mathbf{Y}), \tag{9}$$

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

$$H(\mathbf{X}_1 \ldots \mathbf{X}_n) = \sum_{i=1}^{n} H(\mathbf{X}_i|\mathbf{X}_1 \ldots \mathbf{X}_{i-1}) \leq \sum_{i=1}^{n} H(\mathbf{X}_i). \tag{10}$$

Moreover, the above relation implies that, for each $k \leq n$,

$$H(\mathbf{X}_1 \ldots \mathbf{X}_n) \geq H(\mathbf{X}_1 \ldots \mathbf{X}_k). \tag{11}$$

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

$$I(\mathbf{X};\mathbf{Y}|\mathbf{Z}) = H(\mathbf{X}|\mathbf{Z}) - H(\mathbf{X}|\mathbf{Z}\,\mathbf{Y}) = H(\mathbf{Y}|\mathbf{Z}) - H(\mathbf{Y}|\mathbf{Z}\,\mathbf{X}) = I(\mathbf{Y};\mathbf{X}|\mathbf{Z}). \tag{12}$$

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

$$H(\mathbf{X}|\mathbf{Z}) \geq H(\mathbf{X}|\mathbf{Z}\,\mathbf{Y}). \tag{13}$$