Analysis and Design of Distributed Key Distribution Centers*

Carlo Blundo and Paolo D'Arco

Dipartimento di Informatica ed Applicazioni
Università di Salerno, 84081 Baronissi (SA), Italy

e-mail: {carblu,paodar}@dia.unisa.it

November 8, 2004

Abstract

A Key Distribution Center of a network is a server who *generates* and *distributes* secret keys to groups of users for secure communication. A Distributed Key Distribution Center is a set of servers that *jointly* realizes a Key Distribution Center. In this paper we describe in terms of information theory a model for distributed key distribution centers, and we present lower bounds holding in the model for the main resources needed to set up and manage a distributed center, i.e., memory storage, randomness, and bandwidth. Then, we show that a previously proposed protocol which uses a bidimensional extension of Shamir's secret sharing scheme meets the bounds and it is, hence, optimal.

Keywords: Key Establishment, Cryptographic Protocols, Distributed Systems.

1 Introduction

Key Establishment. Key Establishment is an intriguing, deeply studied and, partially, still open problem in Cryptography. Loosely speaking, it can be described as follows: a group of users of a public network would like to use encryption and authentication algorithms to securely communicate. In terms of computation symmetric algorithms are more efficient than asymmetric ones. Moreover, if a broadcast channel is available, a user has to encrypt, authenticate, and send a message *just once* in order to reach all members of the group. Unfortunately, the group needs common keys to encrypt, decrypt, and authenticate the messages the users wish to send to each other *before* starting the communication. Hence, the problem is how to design an efficient protocol by means of which the members of the group can establish a common key.

A basic solution to this problem, provided by public key algorithms, consists of the so called *hybrid* approach. This approach assumes that users' public keys are contained in a public available and authenticated bulletin board. Before starting a secure communication, one of the users of the group chooses at random a common key. Then, for each other user

*An extended abstract of this paper appeared at the 7-th Italian Conference on Theoretical Computer Science (ICTCS 2001), Lecture Notes in Computer Science, vol. 2202, pp. 357-369, Springer-Verlag , 2001.

belonging to the group, he encrypts a message containing the chosen key with the user's public key, and sends this encrypted message to the user. The user, by using his private key, recovers the common key from the encrypted message. Later on, the users of the group can encrypt, decrypt and authenticate the messages they wish to send to each other with symmetric algorithms.

Nevertheless, this solution is not efficient, expecially in presence of a large group, and it has some drawbacks: public key encryptions and decryptions are slow operations, and the user who chooses the key must be trusted, i.e., the other users have to believe that he is choosing a good key.

The large amount of literature on the Key Establishment problem[1] can be roughly divided into two classes: the class of *key agreement* schemes and the class of *key distribution* schemes. In the former, users interact in order to agree on a common key. Schemes in the latter are based on a third party who helps in computing a common key by distributing information.

Moreover, schemes are classified according to the assumptions made on the power of the adversary: computationally secure schemes are based on the existence of (presumed to be) hard problems; while, unconditionally secure schemes consider a computationally unbounded adversary.

Research efforts of the last years have given rise to computationally secure schemes based on extensions of the Diffie-Hellman key agreement scheme [25], unconditionally secure schemes based on some a-priori common knowledge among the parties, and to unconditionally or computationally secure schemes based on the use of a trusted third party. In the following, we give a quick look at these approaches. We refer the reader to [37] (chapters 12 and 13) for a general overview of both computationally and unconditionally secure schemes as well as for a detailed discussion of the practical issues associated with real implementations of several schemes.

Extensions of the Diffie-Hellman Scheme. The Diffie-Hellman scheme [25] (DH scheme, for short) enables two parties, who have never met before, to agree on a common key by exchanging messages over a public channel. The original proposal assumes a passive adversary, who just taps the channel. It is based on the difficulty of computing discrete logarithms in finite cyclic groups. In order to deal with active adversaries and to exhibit formal proofs of security, several variants of the basic scheme (e.g., [26, 3, 30, 4, 18, 19, 20]) have been proposed; nowadays, it is almost general opinion that the two-party case is well understood.

The first attempt to extend the DH scheme to groups was given by Ingemarson [28], where the members of the group are arranged in a logical ring.

Another extension was proposed by Steer et al. in [47], while Becker and Wille [2] found lower bounds on the communication complexity of protocols for group key agreement.

The so-called natural extensions of the DH scheme were studied by Steiner et al. in [45, 46], and Ateniese et al. in [1], while Bresson et al. in [13, 14, 15], have defined a formal adversarial model and have proved secure slight variants of the protocols given in [45, 46, 1].

Another interesting scheme was introduced by Bermester and Desmedt [17], further studied and generalized by Just and Vaudenay in [30]; recently Katz and Yung [31] have shown that Bermester and Desmedt's scheme is provable secure against a passive adversary

[1]Several surveys (e.g., [42, 21]) and books (e.g., [12]) have been written along the years on this subject.

when implemented over a cyclic group of prime order which satisfies the decisional Diffie-Hellman assumption [11].

However, at the state of current knowledge, an efficient and provable secure protocol for managing dynamic group (i.e., from time to time, users can join and leave the group) is not available. The natural extensions of the DH scheme are not efficient in terms of round-complexity, while Burmester and Desmedt's scheme, need to be started from scratch when the group structure changes. For the three-party case there is a round-complexity efficient scheme due to Joux [29] but no general extension is available, yet. Finding an efficient and provable secure (under standard assumptions) key agreement scheme for dynamic group is an open challenge.

Unconditionally secure schemes based on a-priori common knowledge. Maurer's research group [27] has studied under which conditions (and how) two entities can agree on a common key when no computational assumptions on the power of the adversary are made. It has been shown that if the two parties do not have any common knowledge, key agreement is impossible against active or passive adversaries if the communication channel they have access to is completely insecure. On the other hand, if some common knowledge is available, a common key can be established.

In a recent three-part paper [34, 35, 36], which summarises part of this foundational work, a clear presentation of the setting and of some of the results obtained along the years is given. The paper considers the case of a completely insecure communication channel. A general model where the common knowledge is represented by a joint probability distribution of three random variables associated with the two parties and the adversary, respectively, is given. Furthermore, two special cases are analysed: in the first one, the parties have access to the outcome of n independent realizations of a random experiment. In the second, a common *partially secret* string is available to the parties and, by means of a process called *privacy amplification*, the same parties are able to extract a totally secret (shorter) key.

Several problems are still open, as well as the multi-party case has not been considered, yet. We refer the interested reader to [34, 35, 36] for details and references to other works on this interesting approach.

Schemes based on a trusted third party. An approach exploited by several papers assumes the presence of a trusted third party, who plays a certain role in order to enable a group of users to establish a common key. The trusted third party sends, during a set-up phase, some information to the users. By using this information, the users can compute the group keys they need to securely communicate. Depending on the scheme, the trusted third party might be or not be active during the computation of a common key by a group of users, as well as there might be or not be interaction among the users of the group, in order to compute the common key. Such schemes are generally referred to as *key distribution schemes*. We refer to Stinson's survey [48] and to [9] for an overview of some interesting (mainly unconditionally secure) schemes following the above approach.

A frequently used strategy applied in traditional models of a network to solve the Key Establishment Problem consists of using a *Key Distribution Center* (for short, KDC), a server of the network who generates and distributes *on-demand* the group keys. The idea is the following: each user shares a secure point-to-point channel with the center. When the user wants to securely communicate with other users, he sends a request message for a session key. The center checks for membership of the user in the group, and distributes in encrypted form the common key to each member of the group. Needham and Schroeder [39] began this approach for the two-party case, implemented most notably in the Kerberos System [40] (see again [37] for more references).

With this approach the KDC is a bottleneck, since all users have to communicate with it every time they wish to obtain a key. A crash of the KDC stalls the entire system. Besides, the KDC is a valuable target to an adversary.

Well-known and applied solutions to the availability and reliability issues are *replication* of the KDC in several points of the network and *partition* of the network in several domains with dedicated KDCs, responsible for key management in only a fixed local area. However, these solutions are partial and expensive solutions [38].

Distributed Key Distribution Center. 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 KDC. In this setting, users have secure point-to-point channels with all servers. A user who needs to securely communicate with other users, sends a key-request message to a subset at his choice of at least k out of the n servers. The contacted servers answer with some information enabling him to compute the group key (see Figure 1).

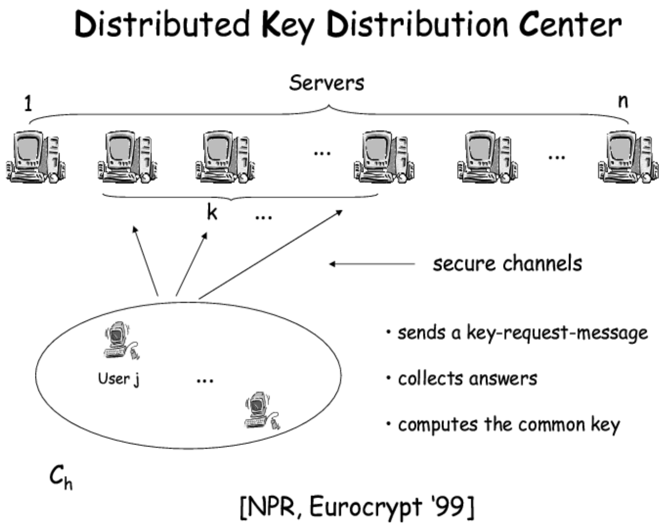

Figure 1: Example of a Distributed Key Distribution Center

With this approach, the concentration of secrets and the slow down factor which arise in a network with a single KDC are removed. 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 of time to compute a key, compared to a centralised setting. Finally, the users can obtain the keys they need even if they are unable to contact some of the servers.

Related Researches. Kurosawa et al. [33] studied the security of the trusted center in a b-secure t-group key distribution scheme. In this scheme the center issues, in a set-up phase, a private information v_i to each user U_i. Later on, user U_i computes from v_i the common key with other $t-1$ users in such a way that any dishonest b users cannot learn anything about keys they should not know. The schemes considered in [33] are usually referred to as key pre-distribution schemes since, after the set-up phase, users compute by themselves the keys associated to the groups they belong to. Kurosawa et al. showed how to distribute some key pre-distribution schemes among m servers, so that even if $\ell < m$ servers and b users collaborate, they learn nothing about keys the b users are not entitled to compute. However, the model introduced in [38], which is analysed in this paper from an information theoretic point of view, is pretty much different from the one considered in [33]. The model proposed in [38] requires *interaction* among users and servers when the users need to compute a common key; while, the schemes considered in [33] do not.

During the last two years some other papers dealing with distributed key distribution centers have been published. Given the resource complexity of the distribution mechanism described in this paper, in [7] the *ramp* approach, introduced in [5] in the context of secret sharing schemes, has been investigated. 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, it has been considered a *ramp structure* for the DKDC, characterized by two thresholds t_1 and t_2, where coalitions of malicious users of size t, with $t < t_1$, or $t_1 \leq t < t_2$, or $t \geq t_2$, 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 of this approach 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.

In [8] the model here studied has been extended by considering a general family of subsets of servers, referred to as the *access structure*, authorized to help the users in recovering the conference keys. Therein, lower bounds holding on the model have been shown in an easy and elegant way by using a reduction technique which relates DKDSs (schemes realizing DKDCs, defined later on) to Secret Sharing Schemes. Moreover, a linear algebraic approach to designing DKDSs has been proposed. Namely, a method for constructing a DKDS from a linear secret sharing scheme and a family of linear ℓ-wise independent functions has been described. This approach has allowed a unified description of seemingly different schemes.

Finally, in [23, 41] *robust* DKDCs have been proposed: each server can *verify* that the information it stores and uses to answer the users' key-request messages is consistent with the information stored by the other servers; at the same time, the users are *guaranteed* that they can compute the same key for a given conference in which they belong to. Moreover, time is divided in *periods*, and at the beginning of each period the servers are involved in an update procedure that "refreshes" the private information they store while the conference keys they provide stay the same. This property is referred to as *proactive security.* The design of the DKDC is based on unconditionally secure proactive verifiable secret sharing scheme.

A new *computationally secure* DKDC has been instead proposed in [24]. The model the authors consider is slightly different from the one introduced in [38], and it seems to be in a certain sense more realistic in terms of assumptions.

Our Contribution. In this paper we study unconditionally secure DKDCs. We propose an information theoretic model for a *Distributed Key Distribution Scheme* (DKDS, for short), a scheme realizing a DKDC. Then, we analyze the relations between the sizes of the different pieces of information needed to setup and maintain a DKDC. We show lower bounds on the amount of information that each server has to store, on the amount of information that each server has to send to answer a key-request message, and on the size of the messages that have to be generated and sent in the setup phase to initialize the DKDC. Moreover, we quantify the randomness needed to setup a DKDC. Finally, we show that the bounds are tight, since they are met by a protocol [38] which uses a bidimensional extension of Shamir's secret sharing scheme, based on polynomial interpolation.

Motivation. Why an information theoretic analysis of DKDCs? There are several reasons: the entropy function enables us to describe in a general, elegant and compact form the properties that a Distributed Key Distribution Center has to satisfy. Then, it simplifies the task of analysing the resources needed to set-up such a distributed center. Moreover, recent trends in Cryptography, related to the investigation of new computational models (e.g, bounded storage model, models with noise channels, ...) as well as the progress obtained by researchers in quantum computing, have promoted a renaissance of interest in designing unconditionally secure schemes, which do not rely on any computational assumption on the power of the adversary.

Often unconditionally secure schemes, which can be used for a finite number of times, can be easily turned into computationally secure schemes by standard cryptographic techniques (e.g., by moving the computation to the exponent, given a finite cyclic group and a generator of the group) or they can be used as intermediate modules in more complex constructions.

In our case, an unconditionally secure distributed key distribution center might be used to give a long-term key to each group of users from which the group can derive several session keys for short-term uses. Last but not least, the analysis of unconditionally secure schemes provides a term of comparison to designers of computationally secure schemes: it is always helpful to know how many resources are needed in order to get perfect security.

2 Information Theory

In this section we briefly recall some basic notions of Information Theory [22].

A *discrete random experiment* is defined by a finite set, called *sample space*, consisting of all elementary events, and a *probability measure* assigning a non-negative real number to every elementary event, such that the sum of all these probabilities is equal to 1. An *event* of a discrete random experiment is a subset of the sample space, and the probability assigned to it is the sum of the probabilities of its elementary events.

A *discrete random variable* $\mathbf{X}$ is a mapping from a sample space to a certain range X, and is characterized by its probability distribution $\{P_{\mathbf{X}}(x)\}_{x\in X}$ that assigns to every $x \in X$ the probability $P_{\mathbf{X}}(x)$ of the event that $\mathbf{X}$ takes on the value x.

The *entropy* of $\mathbf{X}$, denoted by $H(\mathbf{X})$, is a real number that measures the uncertainty about the value of $\mathbf{X}$ when the underlying random experiment is carried out. It is defined by

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

assuming that the terms of the form $0 \log 0$ are excluded from the summation, and where the logarithm is relative to the base 2. The entropy of a random variable 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 $x \in X$. The deviation of the entropy $H(\mathbf{X})$ from its maximal value can be used as a measure of non-uniformity of the distribution $\{P_{\mathbf{X}}(x)\}_{x \in X}$.

Given two random variables $\mathbf{X}$ and $\mathbf{Y}$, taking values on sets X and Y, respectively, according to a probability distribution $\{P_{\mathbf{XY}}(x, y)\}_{x \in X, y \in Y}$ on their Cartesian product, the conditional uncertainty of $\mathbf{X}$, given the random variable $\mathbf{Y}$, called *conditional entropy* and denoted by $H(\mathbf{X}|\mathbf{Y})$, is defined as

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

Notice that the conditional entropy is not the entropy of a probability distribution but the *average* over all entropies $H(\mathbf{X}|\mathbf{Y} = y)$. Simple algebra shows that

$$H(\mathbf{X}|\mathbf{Y}) \geq 0 \tag{1}$$

with equality if and only if X is a function of Y.

The *mutual information* $I(\mathbf{X}; \mathbf{Y})$ between $\mathbf{X}$ and $\mathbf{Y}$ is a measure of the amount of information by which the uncertainty about $\mathbf{X}$ is reduced by learning $\mathbf{Y}$, and vice versa. It is defined by

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

Since

$$I(\mathbf{X}; \mathbf{Y}) = I(\mathbf{Y}; \mathbf{X}) \text{ and } I(\mathbf{X}; \mathbf{Y}) \geq 0, \tag{2}$$

it is easy to see that

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

with equality if and only if $\mathbf{X}$ and $\mathbf{Y}$ are independent. Along the same lines, 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

$$\begin{aligned} I(\mathbf{X}; \mathbf{Y}|\mathbf{Z}) &= H(\mathbf{X}|\mathbf{Z}) - H(\mathbf{X}|\mathbf{Z}\,\mathbf{Y}) \qquad (4) \\ &= H(\mathbf{Y}|\mathbf{Z}) - H(\mathbf{Y}|\mathbf{Z}\,\mathbf{X}) = I(\mathbf{Y}; \mathbf{X}|\mathbf{Z}). \end{aligned}$$

Since the conditional mutual information $I(\mathbf{X}; \mathbf{Y}|\mathbf{Z})$ is always non-negative, it holds that

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

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

$$H(\mathbf{X}_1 \ldots \mathbf{X}_n|\mathbf{Y}) = H(\mathbf{X}_1|\mathbf{Y}) + H(\mathbf{X}_2|\mathbf{X}_1\mathbf{Y}) + \cdots + H(\mathbf{X}_n|\mathbf{X}_1 \ldots \mathbf{X}_{n-1}\mathbf{Y}). \tag{6}$$

Therefore, given n random variables, $\mathbf{X}_1 \ldots \mathbf{X}_n$, it holds that

$$H(\mathbf{X}_1 \ldots \mathbf{X}_n) = \sum_{i=1}^{n} H(\mathbf{X}_i | \mathbf{X}_1 \ldots \mathbf{X}_{i-1}) \leq \sum_{i=1}^{n} H(\mathbf{X}_i). \tag{7}$$

Moreover, the above relations imply that, for any $k \leq n$,

$$H(\mathbf{X}_1 \ldots \mathbf{X}_n) \geq H(\mathbf{X}_1 \ldots \mathbf{X}_k). \tag{8}$$

3 The Model

In this section we formally describe the model hereafter we will deal with.

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 secure connections with *all* the servers. We assume that users and servers are honest-but-curious: they follow the protocol but, by pooling together the available information (secret keys, transcript of the communication, et cetera ...) they might try to gain some "extra" knowledge.

A distributed key distribution scheme is a protocol divided in three phases: a *set up 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.

Set up Phase. We assume that the set-up phase is started by k servers say, without loss of generality, $S_1, \ldots, S_k$. Each of these servers, using a *private source* of random bits r_i, generates some information which is securely distributed to the others. More precisely, for $i = 1, \ldots, k$, server S_i generates/sends to S_j, the value $\gamma_{i,j}$, for all $j = 1, \ldots, n$.

At the end of the set-up phase server S_i, for $i = 1, \ldots, n$, stores some secret information $a_i = f(\gamma_{1,i}, \ldots, \gamma_{k,i})$, where f is a publicly known k-argument function, which can be computed from the values he has received from $S_1, \ldots, S_k$.

Key-request Phase. Let $\mathcal{C}$ be the set of all possible groups of users, referred to as a *conferences*, who want to securely communicate, and let $C_h \in \mathcal{C}$. Each user U_j in C_h, to compute a key for conference C_h (we denote such a key with κ_h), contacts at least k servers. Then, server S_i, contacted by user U_j, checks[2] for membership of U_j in C_h; if the check is successful, then S_i computes a value $y^h_{i,j} = F(a_i, j, h)$, which is a function of the private information a_i, the index j indicating user U_j, and the index h of the requested key. Otherwise, S_i sets $y^h_{i,j} = \perp$, a special value which gives no information on the conference key. Finally, S_i sends the value $y^h_{i,j}$ to U_j.

Key-computation Phase. The users in C_h compute the conference key as a function of k values received from any k 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 $y^h_{i_1,j}, \ldots, y^h_{i_k,j}$ are k values received from servers $S_{i_1}, \ldots, S_{i_k}$, and G is a publicly known k-argument function.

Informally, a Distributed Key Distribution Center must satisfy the following properties:

- **Complete Setup.** When the set up phase correctly terminates, every server S_i must be able to compute his private information a_i.

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

- **Consistence.** Each user in a conference $C_h \subseteq \mathcal{U}$ must be able to uniquely compute *the same* conference key, after interacting with at least k servers of his own choosing.
- **Security.** 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 Distributed Key Distribution Scheme. We use the entropy function mainly because this leads to a compact and simple description of the scheme, and because the entropy approach takes into account all probability distributions on the keys. To this aim, we need to setup our notation.

- Let $\mathcal{C} \subset 2^{\mathcal{U}}$ be the set of conferences on $\mathcal{U}$ who need to securely communicate, and assume they are indexed by elements of $\mathcal{H} = \{1, 2, \ldots\}$.
- Let $\mathcal{G} \subset 2^{\mathcal{U}}$ be the set of tolerated coalitions, i.e., the coalitions of malicious users who try to break the scheme. For any coalition $G \in \mathcal{G}$, denote by $\mathcal{C}_G$ the set of conferences containing some user in G, and with $\mathcal{H}_G$ the set of corresponding indices. In other words, $\mathcal{C}_G = \{C_h \in \mathcal{C} : C_h \cap G \neq \emptyset\}$, and $\mathcal{H}_G = \{h \in \mathcal{H} : C_h \in \mathcal{C}_G\}$.
- For $i = 1, \ldots, k$ and for $j = 1, \ldots, n$, 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 $j = 1, \ldots, n$, let $\Gamma_j = \Gamma_{1,j} \times \cdots \times \Gamma_{k,j}$ be the set of values that S_j, can receive during the set up phase.
- For $i = 1, \ldots, n$, let A_i be the set of values a_i the server S_i can compute during the set up phase from the values received from $S_1, \ldots, S_k$.
- For $i = 1, \ldots, m$, for $j = 1, \ldots, n$, and for each $h \in \mathcal{H}$, let $Y^h_{i,j}$ be the set of values $y^h_{i,j}$ that can be sent by S_i when it receives a key-request message from U_j for conference C_h.
- Finally, for each $h \in \mathcal{H}$, let K_h be the set of possible values for the conference key κ_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 sets T_i, $T_{i,j}$ and $T^h_{i,j}$, where $i \in X, j \in Y$, and $h \in H$, we denote by $T_X = \prod_{i \in X} T_i$, by $T_{X,Y} = \prod_{i \in X, j \in Y} T_{i,j}$, and by $T^H_{X,Y} = \prod_{i \in X, j \in Y, h \in H} T^h_{i,j}$, the corresponding Cartesian products. According to this notation, we will consider the following Cartesian products, defined on the sets of our interest:

Γ_Y	Set of values that can be received by server S_j, for all $j \in Y$
$\Gamma_{X,j}$	Set of values that can be sent by server S_i to S_j, for all $i \in X$
$\Gamma_{X,Y}$	Set of values that can be sent by server S_i to S_j, for all $i \in X$ and all $j \in Y$
K_X	Set of $\lvert X \rvert$-tuple of conference keys
A_X	Set of $\lvert X \rvert$-tuple of servers' private information a_i
$Y^h_{X,j}$	Set of values that can be sent by S_i, for all $i \in X$, to U_j for the conference C_h
Y^h_G	Set of values that can be sent by $S_1, \ldots, S_n$ to U_j, for all $U_j \in G$, for C_h
Y^H_G	Set of values that can be sent by $S_1, \ldots, S_n$ to U_j, for all $U_j \in G$, for C_h $\forall h \in H$

Table 1: Cartesian Products

As done in Section 2, given a set W, we will denote in bold the random variable $\mathbf{W}$ assuming values on W according to the probability distribution $\mathcal{P}_{\mathbf{W}}$.

A Distributed Key Distribution Scheme can be defined as follows:

Definition 3.1 *A $(k, n, \mathcal{C}, \mathcal{G})$-Distributed Key Distribution Scheme (for short, $(k, n, \mathcal{C}, \mathcal{G})$-DKDS) is a protocol which enables each user of $C_h \in \mathcal{C}$ to compute securely w.r.t. $\mathcal{G}$ a common key κ_h by interacting with at least k out of the n servers of the network. More precisely, the following properties are satisfied:*

1 *Each server computes his private information at the end of the set up phase.*
Formally, for each $i = 1, \ldots, n$, and for each $X \subset \{1, \ldots, k\}$ it holds that

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

2 *Each server can answer the key-request messages.*
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

$$H(\mathbf{Y}^h_{i,j}|\mathbf{A}_i) = 0.$$

3 *Each user in $C_h \in \mathcal{C}$ can compute a common key κ_h after contacting at least k servers.*
Formally, for each conference $C_h \in \mathcal{C}$, for each subset of $t \geq k$ indices $X = \{i_1, \ldots, i_t\} \subseteq \{1, \ldots, n\}$, and for each user $U_j \in C_h$, it holds that

$$H(\mathbf{K}_h|\mathbf{Y}^h_{X,j}) = 0.$$

4 *Any coalition $G \in \mathcal{G}$ of users and at most $k-1$ servers, by putting together their private information and the values sent by the n servers to users in G during some previous key-request phases, does not gain any information on any new key.*
Formally, for each conference $C_h \in \mathcal{C}$, for any coalition of users $G \in \mathcal{G}$, and for any $(k-1)$-subset $X =\subset \{1, \ldots, n\}$, it holds that

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

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

■

The adversary neither obstaculates the computation of some key nor substitutes the inputs to any party. In other words, the model we consider is a semi-honest model. Notice that, property 4 formalizes the security requirements we are looking for. Indeed, the worst case scenario consists of coalitions of users in G (the information they can acquire during the run of the protocol is represented by $\mathbf{Y}_G^{\mathcal{H}_G \setminus \{h\}}$) and $k-1$ corrupted servers knowing $\mathbf{\Gamma}_X$ and $\mathbf{\Gamma}_{Z,N}$ (the random variable $\mathbf{\Gamma}_{Z,N}$ takes into account the possibility that among the corrupted servers there are some which send out information in the set up phase to other servers). The property guarantees that such coalitions of users and servers, do not gain any information on a new key.

Remark. Notice that a DKDC implemented by a DKDS is a *deterministic* system at all. Random bits are only needed at the beginning (i.e., initialization of the system), when each server who sends out information during the set up phase uses his own random source to generate the messages to be delivered to the other servers of the network.

In the following, to simplify the analysis but without loss of generality, we assume that for different $h, h' \in \mathcal{H}$, the entropies of the keys are equal, i.e., $H(\mathbf{K}_h) = H(\mathbf{K}_{h'})$.

4 Some Technical Lemmas

To show the main properties of our model, we need some technical lemmas. These lemmas are the content of this section.

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 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, (1) and (5) imply that

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

On the other hand, from (2), we have that

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

Hence, $H(\mathbf{A}|\mathbf{C}) = H(\mathbf{A}|\mathbf{BC})$. Since (5) implies that $H(\mathbf{A}|\mathbf{B}) \geq H(\mathbf{A}|\mathbf{BC})$ then, we have that $H(\mathbf{A}|\mathbf{B}) \geq H(\mathbf{A}|\mathbf{C})$, 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 lines of the above proof, we can show that

$$H(\mathbf{A}|\mathbf{BD}) \geq H(\mathbf{A}|\mathbf{CD}). \tag{9}$$

It is not difficult 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 values that at least k out of the n servers send to the users in G when invoked for those conference keys. This is formally stated by the next lemma.

Lemma 4.2 *Let $G = \{U_{j_1}, \ldots, U_{j_g}\}$ be a group of users and, for each $r = 1, \ldots, \ell$, 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 $r = 1, \ldots, \ell$, we get:

$$\begin{aligned} 0 &\leq H(\mathbf{K}_{S_r}|\mathbf{Y}_G^{S_r}) \text{ (from (1))} \\ &\leq \sum_{j=1}^{r} H(\mathbf{K}_{s_j}|\mathbf{Y}_G^{s_j}) \text{ (from (6) and (5))} \\ &\leq \sum_{j=1}^{r} H(\mathbf{K}_{s_j}|\mathbf{Y}_{X,t}^{s_j}) \text{ (from (5) where } t \in C_{s_j} \cap G \text{ and } X = \{i_1, \ldots, i_k\}) \\ &= 0 \text{ (from Property 3 of Definition 3.1).} \end{aligned}$$

Hence, the lemma holds. ■

For any group $G \in \mathcal{G}$ of malicious users, we have denoted by $\mathcal{C}_G$ the family of conferences of $\mathcal{C}$ in which belong at least one user in G, i.e., $\mathcal{C}_G = \{C_h \in \mathcal{C} \text{ and } C_h \cap G \neq \emptyset\}$. Hereafter, we set $\ell_G = |\mathcal{C}_G|$, i.e., ℓ_G denotes the number of conference keys that G can retrieve interacting with the servers. Moreover, we indicate with ℓ the maximum number of conference keys that *any* coalition $G \in \mathcal{G}$ can retrieve interacting with the servers, i.e., $\ell = \max_{G \in \mathcal{G}} \ell_G$.

5 Properties and Bounds

In this section we show some properties of our model. We prove lower bounds on the amount of information that each server has to send once a key-request message from a user has been received, and on the amount of information that each server has to store to answer the key-request messages. Finally, we prove a lower bound on the randomness needed to setup a Distributed Key Distribution Center.

Conference Keys Independence. We start by showing that, in any distributed key distribution scheme, conference keys are ℓ-wise independent.

Lemma 5.1 *Let $G \in \mathcal{G}$ be a coalition of malicious users and let $\mathcal{H}_G = \{s_1, \ldots, s_\ell\}$. Then, for each $r = 1, \ldots, \ell$, it holds that $H(\mathbf{K}_{s_r} | \mathbf{K}_{\mathcal{H}_G \setminus \{s_r\}}) = H(\mathbf{K}_{s_r})$.*

Proof. For each $r = 1, \ldots, \ell$, from (3), we get that, $H(\mathbf{K}_{s_r} | \mathbf{K}_{\mathcal{H}_G \setminus \{s_r\}}) \leq H(\mathbf{K}_{s_r})$. Since, from Lemma 4.2 it results

$$H(\mathbf{K}_{\mathcal{H}_G \setminus \{s_r\}} | \mathbf{Y}_G^{\mathcal{H}_G \setminus \{s_r\}}) = 0,$$

then, setting $\mathbf{A} = \mathbf{K}_{s_r}$, $\mathbf{B} = \mathbf{K}_{\mathcal{H}_G \setminus \{s_r\}}$, and $\mathbf{C} = \mathbf{Y}_G^{\mathcal{H}_G \setminus \{s_r\}}$, we have that

$$\begin{aligned} H(\mathbf{K}_{s_r} | \mathbf{K}_{\mathcal{H}_G \setminus \{s_r\}}) &\geq H(\mathbf{K}_{s_r} | \mathbf{Y}_G^{\mathcal{H}_G \setminus \{s_r\}}) \text{ (from Lemma 4.1)} \\ &\geq H(\mathbf{K}_{s_r} | \mathbf{Y}_G^{\mathcal{H}_G \setminus \{s_r\}} \mathbf{\Gamma}_X \mathbf{\Gamma}_{Z,N}) \text{ (from (5))} \\ &= H(\mathbf{K}_{s_r}) \text{ (from Property 4 of Definition 3.1)}, \end{aligned}$$

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

Size of Servers' Answers. Using some basic properties of the entropy function, it is possible to obtain a lower bound on the amount of information that each server, contacted by a user, has to send once a key-request message has been received.

Theorem 5.2 *In any $(k, n, \mathcal{C}, \mathcal{G})$-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*

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

Proof. Let $X = \{i_1, \ldots, i_{k-1}\} \subset \{1, \ldots, n\}$. For $i \notin X$, relation (3) implies that $H(\mathbf{Y}_{i,j}^h) \geq H(\mathbf{Y}_{i,j}^h | \mathbf{Y}_{X,j}^h)$. Applying (5), we have that

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

According to Property 3 of Definition 3.1, we get that $H(\mathbf{K}_h | \mathbf{Y}_{X,j}^h \mathbf{Y}_{i,j}^h) = 0$. Moreover, we can prove that $H(\mathbf{K}_h | \mathbf{Y}_{X,j}^h) = H(\mathbf{K}_h)$ and since (1) implies that $H(\mathbf{Y}_{i,j}^h | \mathbf{Y}_{X,j}^h \mathbf{K}_h) \geq 0$, we conclude that

$$H(\mathbf{Y}_{i,j}^h) \geq H(\mathbf{Y}_{i,j}^h | \mathbf{Y}_{X,j}^h) \geq H(\mathbf{K}_h) + H(\mathbf{Y}_{i,j}^h | \mathbf{Y}_{X,j}^h \mathbf{K}_h) \geq H(\mathbf{K}_h) = H(\mathbf{K}),$$

which proves the theorem. To prove the equality $H(\mathbf{K}_h|\mathbf{Y}^h_{X,j}) = H(\mathbf{K}_h)$, notice that, from Property 2 of Definition 3.1, 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 4.1, it results that

$$H(\mathbf{K}_h|\mathbf{Y}^h_{X,j}) \geq H(\mathbf{K}_h|\mathbf{A}_X). \tag{10}$$

Moreover, applying Property 1 of Definition 3.1 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 Lemma 4.1, it results that

$$H(\mathbf{K}_h|\mathbf{A}_X) \geq H(\mathbf{K}_h|\mathbf{\Gamma}_X). \tag{11}$$

But, from (11) and (3), and from Property 4 of Definition 3.1 we get

$$H(\mathbf{K}_h|\mathbf{\Gamma}_X) \geq H(\mathbf{K}_h|\mathbf{Y}^{\mathcal{H}_G\setminus\{s\}}\mathbf{\Gamma}_X\mathbf{\Gamma}_{Z,N}) \geq H(\mathbf{K}_h), \tag{12}$$

where $Z = X \cap \{1, \ldots, k\}$, and $N = \{1, \ldots, n\}$. Since (3) implies that $H(\mathbf{K}_h|\mathbf{Y}^h_{X,j}) \leq H(\mathbf{K}_h)$ then, from (10), (11), and (12), we conclude that $H(\mathbf{K}_h|\mathbf{Y}^h_{X,j}) = H(\mathbf{K}_h)$. ∎

Server Memory Storage. We also show that each server, to answer key-request messages from users, has to store some information whose size is lower bounded by $\ell \cdot H(\mathbf{K})$.

Theorem 5.3 *In any $(k, n, \mathcal{C}, \mathcal{G})$-DKDS, for each $i = 1, \ldots, n$, the private information a_i, stored by server S_i, satisfies*

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

Proof. Fix any $G \in \mathcal{G}$ such that $\ell_G = \ell$. Let $\mathcal{H}_G = \{s_1, \ldots, s_\ell\}$, and 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 (5), we have that

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

It is possible to prove that

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

Therefore, since (3) implies that $H(\mathbf{A}_{i_r}) \geq H(\mathbf{A}_{i_r}|\mathbf{A}_{X\setminus\{i_r\}})$, and (1) implies $H(\mathbf{A}_{i_r}|\mathbf{A}_{X\setminus\{i_r\}}\mathbf{K}_{\mathcal{H}_G}) \geq 0$, then we have that

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

which proves the theorem. To prove the equality $H(\mathbf{K}_{\mathcal{H}_G}|\mathbf{A}_{X\setminus\{i_r\}}) = \ell \cdot H(\mathbf{K})$, notice that, from (3) and (7), it follows that

$$H(\mathbf{K}_{\mathcal{H}_G}|\mathbf{A}_{X\setminus\{i_r\}}) \leq H(\mathbf{K}_{\mathcal{H}_G}) \leq \sum_{r=1}^{\ell} H(\mathbf{K}_{s_r}) = \ell \cdot H(\mathbf{K}).$$

From Property 1 of Definition 3.1 and Lemma 4.1 it holds that

$$
\begin{aligned}
H(\mathbf{K}_{\mathcal{H}_G}|\mathbf{A}_{X\setminus\{i_r\}}) &\geq H(\mathbf{K}_{\mathcal{H}_G}|\mathbf{\Gamma}_{X\setminus\{i_r\}}) \\
&\geq \sum_{r=1}^{\ell} H(\mathbf{K}_{s_r}|\mathbf{\Gamma}_{X\setminus\{i_r\}}\mathbf{K}_{\mathcal{H}_G\setminus\{s_r\}}) \text{ (applying (6))} \\
&\geq \sum_{r=1}^{\ell} H(\mathbf{K}_{s_r}|\mathbf{\Gamma}_{X\setminus\{i_r\}}\mathbf{Y}_G^{\mathcal{H}_G\setminus\{s_r\}}) \text{ (from (5))} \\
&\geq \ell \cdot H(\mathbf{K}) \text{ (from Property 4 of Definition 3.1).}
\end{aligned}
$$

Moreover, from (1), we have $H(\mathbf{K}_{\mathcal{H}_G}|\mathbf{A}_X) \geq 0$; while, applying (6) and (5), we have that

$$
\begin{aligned}
H(\mathbf{K}_{\mathcal{H}_G}|\mathbf{A}_X) &\leq \sum_{r=1}^{\ell} H(\mathbf{K}_{s_r}|\mathbf{A}_X, \mathbf{K}_{s_{r-1}}, \ldots, \mathbf{K}_{s_1}) \\
&\leq \sum_{r=1}^{\ell} H(\mathbf{K}_{s_r}|\mathbf{A}_X) \text{ (from property (5))} \\
&\leq \sum_{r=1}^{\ell} H(\mathbf{K}_{s_r}|\mathbf{Y}_{X,j}^{s_r}) \text{ (from Property 3 of Definition 3.1 and Lemma 4.1)} \\
&\leq 0 \text{ (where } U_j \in G\text{, and applying Property 4 of Definition 3.1)}
\end{aligned}
$$

Therefore, the theorem holds. ■

Randomness of a DKDS. Randomness (i.e., truly random bits) is a useful resource due to its ability to enhance the capabilities of other resources, such as time and space. Therefore, the amount of randomness used in a computation is an important issue in many applications. Considerable effort has been devoted both to reduce the number of random bits used by probabilistic algorithms, and to analyze the amount of randomness required in order to achieve a given performance. Moreover, since truly random bits are hard to obtain, it has also been investigated the possibility of using imperfect sources of randomness in randomized algorithms.

The randomness of a scheme can be measured in different way. Knuth and Yao [32] 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 source described by the probability distribution P. In [32] it has been shown the following result:

Theorem 5.4 $H(\mathbf{P}) \leq T(\mathbf{P}) < H(\mathbf{P}) + 2$.

Thus, the entropy of a random source is very close to the average number of unbiased random bits necessary to simulate the source. Hence, it is a natural measure of the randomness[3].

[3]For this and other interesting relations of the Shannon entropy with other measures of complexity, like Kolmogorov complexity, we advice the reader to consult the very readable account given in [22]. Moreover, we refer the reader to [10] for a detailed analysis of randomness in distribution protocols.

In order to prove a lower bound on the randomness needed to realize a DKDC, let us start by showing two lemmas useful to prove a lower bound on the amount of information each of the servers $S_1, \ldots, S_k$ has to send to the other servers during the set up phase.

Lemma 5.5 *In any $(k, n, \mathcal{C}, \mathcal{G})$-DKDS, for each $i = 1, \ldots, k$, for each $j = 1, \ldots, n$, and for each $Y \subset \{1, \ldots, n\} \setminus \{j\}$ of size at most $k-1$, it holds that*

$$H(\mathbf{\Gamma}_{i,j}|\mathbf{\Gamma}_Y \mathbf{\Gamma}_{X,j}) \geq \ell \cdot H(\mathbf{K}),$$

where $X = \{1, \ldots, k\} \setminus \{i\}$.

Proof. Let $B \subset \mathcal{U} \setminus (Y \cup \{j\})$ be a set of size $k - 1 - |Y|$ and let $G \in \mathcal{G}$ such that $\ell_G = \ell$. Setting $\mathbf{\Gamma}^{(1)} = \mathbf{\Gamma}_Y \mathbf{\Gamma}_{X,j} \mathbf{\Gamma}_B$ and $\mathbf{\Gamma}^{(2)} = \mathbf{\Gamma}^{(1)} \mathbf{\Gamma}_{i,j}$, we can prove that

$$H(\mathbf{K}_{\mathcal{H}_G}|\mathbf{\Gamma}^{(1)}) = \ell \cdot H(\mathbf{K}) \text{ and } H(\mathbf{K}_{\mathcal{H}_G}|\mathbf{\Gamma}^{(2)}) = 0. \tag{13}$$

Assuming true the above equalities, from (5), it holds that,

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

Moreover, (2) implies that

$$\begin{aligned} H(\mathbf{\Gamma}_{i,j}|\mathbf{\Gamma}^{(1)}) &= H(\mathbf{K}_{\mathcal{H}_G}|\mathbf{\Gamma}^{(1)}) - H(\mathbf{K}_{\mathcal{H}_G}|\mathbf{\Gamma}^{(2)}) + H(\mathbf{\Gamma}_{i,j}|\mathbf{K}_{\mathcal{H}_G}\mathbf{\Gamma}^{(1)}) \\ &= \ell \cdot H(\mathbf{K}) + H(\mathbf{\Gamma}_{i,j}|\mathbf{K}_{\mathcal{H}_G}\mathbf{\Gamma}^{(1)}) \text{ (from (13))} \\ &\geq \ell \cdot H(\mathbf{K}) \text{ (from (1))}. \end{aligned}$$

Hence, the lemma holds. We are left with proving that equalities (13) hold. The first one can be shown as follows: from (3) and (7) we get

$$H(\mathbf{K}_{\mathcal{H}_G}|\mathbf{\Gamma}^{(1)}) \leq H(\mathbf{K}_{\mathcal{H}_G}) \leq \sum_{r=1}^{\ell} H(\mathbf{K}_r) = \ell \cdot H(\mathbf{K}).$$

On the other hand, for each $r = 1, \ldots, \ell$, from (5), and from Lemma 4.2, we have that $H(\mathbf{K}_r|\mathbf{Y}_G^{\mathcal{H}_G}) = 0$. Then, setting $\mathbf{A} = \mathbf{K}_r$, $\mathbf{B} = \mathbf{K}_{\mathcal{H}_G \setminus \{r\}}$, $\mathbf{C} = \mathbf{Y}_G^{\mathcal{H}_G \setminus r}$, and $\mathbf{D} = \mathbf{\Gamma}^{(1)}$ and applying inequality (9), we get

$$H(\mathbf{K}_r|\mathbf{K}_{\mathcal{H}_G \setminus \{r\}}\mathbf{\Gamma}^{(1)}) \geq H(\mathbf{K}_r|\mathbf{Y}_G^{\mathcal{H}_G \setminus \{r\}}\mathbf{\Gamma}^{(1)}). \tag{14}$$

Therefore, we have that

$$\begin{aligned} H(\mathbf{K}_{\mathcal{H}_G}|\mathbf{\Gamma}^{(1)}) &\geq \sum_{r=1}^{\ell} H(\mathbf{K}_r|\mathbf{K}_{\mathcal{H}_G \setminus \{r\}}\mathbf{\Gamma}^{(1)}) \text{ (from (5) and (6))} \\ &\geq \sum_{r=1}^{\ell} H(\mathbf{K}_r|\mathbf{Y}^{\mathcal{H}_G \setminus \{r\}}\mathbf{\Gamma}^{(1)}) \text{ (from (14))} \\ &\geq \ell \cdot H(\mathbf{K}) \text{ (from Property 4 of Definition 3.1)}. \end{aligned}$$

Now, we have to prove that $H(\mathbf{K}_{\mathcal{H}_G}|\mathbf{\Gamma}^{(2)}) = 0$. As a preliminary step notice that, from Property 1 of Definition 3.1 we get that $H(\mathbf{A}_X|\mathbf{\Gamma}^{(2)}) = 0$, where $X = Y \cup B \cup \{j\}$. Hence, for each $r = 1, \ldots, \ell$, applying Lemma 4.1, it results that

$$H(\mathbf{K}_r|\mathbf{\Gamma}^{(2)}) \geq H(\mathbf{K}_r|\mathbf{A}_X). \tag{15}$$

Moreover, for each $r = 1, \ldots, \ell$, choosing $U_j \in G \cap C_r$ and applying Property 3 of Definition 3.1 we get that $H(\mathbf{Y}^r_{X,j}|\mathbf{A}_X) = 0$. Hence, for each $r = 1, \ldots, \ell$, setting $\mathbf{A} = \mathbf{K}_r$, $\mathbf{B} = \mathbf{Y}^r_{X,j}$, and $\mathbf{C} = \mathbf{A}_X$, and applying Lemma 4.1, it results that

$$H(\mathbf{K}_r|\mathbf{Y}^r_{Y,j}) \geq H(\mathbf{K}_r|\mathbf{A}_X). \tag{16}$$

Then, from (1) we get $H(\mathbf{K}_{\mathcal{H}_G}|\mathbf{\Gamma}^{(2)}) \geq 0$. On the other hand, it results that

$$\begin{aligned}
H(\mathbf{K}_{\mathcal{H}_G}|\mathbf{\Gamma}^{(2)}) &= \sum_{r=1}^{\ell} H(\mathbf{K}_r|\mathbf{K}_{r-1}, \ldots, \mathbf{K}_1, \mathbf{\Gamma}^{(2)}) \text{ (from (6))} \\
&\leq \sum_{r=1}^{\ell} H(\mathbf{K}_r|\mathbf{\Gamma}^{(2)}) \text{ (from (5))} \\
&\leq \sum_{r=1}^{\ell} H(\mathbf{K}_r|\mathbf{A}_X) \text{ (from (15))} \\
&\leq \sum_{r=1}^{\ell} H(\mathbf{K}_r|\mathbf{Y}^r_{X,j}) \text{ (from (16))} \\
&\leq 0 \text{ (from Property 3 of Definition 3.1).}
\end{aligned}$$

Thus, equalities (13) are satisfied and the lemma holds. ∎

The following result is a consequence of the above lemma.

Lemma 5.6 *In any $(k, n, \mathcal{C}, \mathcal{G})$-DKDS, for each $j = 1, \ldots, n$, and for any set $Y \subset \{1, \ldots, n\} \setminus \{j\}$ of size at most $k-1$, it holds that*

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

Proof. We have that,

$$\begin{aligned}
H(\mathbf{\Gamma}_j|\mathbf{\Gamma}_Y) &= \sum_{i=1}^{k} H(\mathbf{\Gamma}_{i,j}|\mathbf{\Gamma}_Y\mathbf{\Gamma}_{1,j}\ldots\mathbf{\Gamma}_{i-1,j}) \text{ (from (6))} \\
&\geq \sum_{i=1}^{k} H(\mathbf{\Gamma}_{i,j}|\mathbf{\Gamma}_Y\mathbf{\Gamma}_{X,j}) \text{ (from (5), setting } X = \{1, \ldots, k\} \setminus \{i\}) \\
&\geq \sum_{i=1}^{k} \ell \cdot H(\mathbf{K}) \text{ (from Lemma 5.5)} \\
&= k \cdot \ell \cdot H(\mathbf{K}).
\end{aligned}$$

Thus, the lemma holds. ■

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

Theorem 5.7 *In any $(k, n, \mathcal{C}, \mathcal{G})$-DKDS, for each $i = 1, \ldots, k$, and for each $j = 1, \ldots, n$, the following inequalities are satisfied:*

$$H(\mathbf{\Gamma}_{i,j}) \geq \ell \cdot H(\mathbf{K}) \text{ and } H(\mathbf{\Gamma}_j) \geq k \cdot \ell \cdot H(\mathbf{K}).$$

Proof. Notice that, for any set $Y \subset \{1, \ldots, n\} \setminus \{j\}$ of size less than or equal to $k-1$ and $X = \{1, \ldots, k\} \setminus \{i\}$, from (3) and from Lemma 5.5, we have that

$$H(\mathbf{\Gamma}_{i,j}) \geq H(\mathbf{\Gamma}_{i,j} | \mathbf{\Gamma}_Y \mathbf{\Gamma}_{X,j}) \geq \ell \cdot H(\mathbf{K}).$$

Moreover, since each of the k servers independently chooses the values $\gamma_{i,j}$, then from the above inequality and from (3) and (7), it results that

$$H(\mathbf{\Gamma}_j) = H(\mathbf{\Gamma}_{1,j} \ldots \mathbf{\Gamma}_{k,j}) = \sum_{i=1}^{k} H(\mathbf{\Gamma}_{i,j}) \geq k \cdot \ell \cdot H(\mathbf{K}).$$

Thus, the theorem holds. ■

It is easy to see that the randomness $\mathcal{R}$ of a Distributed Key Distribution Scheme can be lower bounded by $H(\mathbf{\Gamma}_1 \ldots \mathbf{\Gamma}_n)$. The following theorem shows a lower bound on $\mathcal{R}$.

Theorem 5.8 *In any $(k, n, \mathcal{C}, \mathcal{G})$-DKDS the randomness $\mathcal{R}$ satisfies*

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

Proof. Notice that, for each $\{j_1, \ldots, j_k\} \subset \{1, \ldots, n\}$, from Theorem 5.4, and (8), we get that

$$\begin{aligned}
\mathcal{R} \geq H(\mathbf{\Gamma}_1 \ldots \mathbf{\Gamma}_n) &\geq H(\mathbf{\Gamma}_{j_1} \ldots \mathbf{\Gamma}_{j_k}) \\
&= \sum_{r=1}^{k} H(\mathbf{\Gamma}_{j_r} | \mathbf{\Gamma}_{j_1} \ldots \mathbf{\Gamma}_{j_{r-1}}) \text{ (applying (6))} \\
&\geq \sum_{r=1}^{k} H(\mathbf{\Gamma}_{j_r} | \mathbf{\Gamma}_Y) \text{(from (5), setting } Y = \{j_1, \ldots, j_k\} \setminus \{j_r\}) \\
&\geq \sum_{r=1}^{k} k \cdot \ell \cdot H(\mathbf{K}) \text{(from Lemma 5.6)} \\
&= k^2 \cdot \ell \cdot H(\mathbf{K}).
\end{aligned}$$

Hence, the theorem holds. ■

Communication Complexity in a DKDC The Communication Complexity ($\mathcal{CC}$, for short) of a DKDS is measured by the amount of information sent by the servers $S_1, \ldots, S_k$ during the set up phase. It is not difficult to see, by using Theorem 5.7, that

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

Table 2 summarizes the main bounds obtained by the above analysis, assuming the keys are chosen uniformly at random in a set K, (i.e., $H(\mathbf{K}) = \log |K|$). In filling up the table we use property (1) which states that, for any random variable X, $\log |X| \geq H(\mathbf{X})$.

Parameters	**Information needed (in bits)**
Server Answer	$\log \|K\|$
Server Memory Storage	$\ell \cdot \log \|K\|$
Randomness	$k^2 \cdot \ell \cdot \log \|K\|$
Communication Complexity	$k \cdot n \cdot \ell \cdot \log \|K\|$

Table 2: Bounds on DKDSs for keys chosen uniformly at random.

6 On the Size of the Coalition of Curious Users

In this section we point out that the model we have studied takes also care of some settings usually considered in designing key distribution schemes.

The model described in Section 3 is a generalization of the model proposed in [6]. In that paper we considered a network with m users, represented by the set $\mathcal{U} = \{U_1, \ldots, U_m\}$, a set $\mathcal{S} = \{S_1, \ldots, S_n\}$ of n servers, and a set of possible conferences $\mathcal{C}$. Moreover, we required the scheme to be secure against coalitions of up to $k-1$ servers, coalition of users of *any* size, and hybrid coalitions. In that simplified scenario a trusted authority, the *dealer*, realizes the set up phase, distributing the private information a_i to each of the n servers of the network. Then, the dealer disappears. It is not difficult to see that such a model is described by relations 3 and 4 of the current model, assuming that in relation 4 the set G is equal to $\mathcal{U}$ and substituting $\mathbf{\Gamma}_X \mathbf{\Gamma}_{Z,N}$ with $\mathbf{A}_X$.

In the previous sections, the analysis has been done making *no assumptions* on the structure of the coalitions of adversaries that will try to break the scheme. Such an approach enable us to model multiple scenarios, usually considered in the analysis of unconditionally secure key distribution schemes [48]. For example, it is too much to assume that all users of a wide area network can collude to break a scheme. It is more realistic to consider an *upper bound* g on the size of a coalition of curious users. Such schemes are often referred to as g-resilient scheme. Moreover, to further reduce resource requirements of a $(k, n, \mathcal{C}, \mathcal{G})$-DKDC, we can consider an environment in which the set of conferences $\mathcal{C}$ is composed only by the subsets of users of size up to t. Such schemes are characterized by two parameters, t and g, and are referred to as g-resilient t-conference schemes.

g-resilient schemes In such a scenario we fix the size of the possible coalitions of adversaries G to be at most equal to g, i.e., $\mathcal{G} = \{G \subset \mathcal{U} | |G| \leq g\}$. Assuming that $\mathcal{C}$ contains all the possible conferences of $\mathcal{U}$, and, hence, $|\mathcal{C}| = 2^m - m - 1$, it is not difficult to see, by applying a counting argument, that the maximum number of conference keys that can be recovered by any coalition $G \in \mathcal{G}$ is

$$\begin{aligned} \ell &= \sum_{j=2}^{g} \binom{g}{j} + \sum_{j=1}^{g} \binom{g}{j} \cdot (2^{m-g} - 1) = \sum_{j=1}^{g} \binom{g}{j} - g + \sum_{j=1}^{g} \binom{g}{j} \cdot 2^{m-g} - \sum_{j=1}^{g} \binom{g}{j} \\ &= \sum_{j=1}^{g} \binom{g}{j} \cdot 2^{m-g} - g. \end{aligned}$$

Hence, the bounds of Theorems 5.2, 5.3, and 5.8, change according to this value.

g-resilient t-conference schemes Suppose that it is known an upper bound t on the maximum size of the conferences in $\mathcal{C}$, and an upper bound g on the maximum size of the coalitions of malicious users in $\mathcal{G}$, i.e., $\mathcal{C} = \{C \subset \mathcal{U} | |C| \leq t\}$ and $\mathcal{G} = \{G \subset \mathcal{U} | |G| \leq g\}$. Then, $|\mathcal{C}| = \sum_{j=2}^{t} \binom{m}{j}$, and

$$\ell = \sum_{j=2}^{t} \binom{g}{j} + \sum_{s=2}^{t} \sum_{j=1}^{s-1} \binom{g}{j} \cdot \binom{m-g}{s-j}.$$

It is easy to see that the bounds of Theorems 5.2, 5.3, and 5.8 are determined by the above value of ℓ.

7 An Optimal Protocol

A construction of a $(k, n, \mathcal{C}, \mathcal{G})$-DKDS has been proposed in [38]. In this section we describe such a protocol and we argue that it satisfies Definition 3.1 and meets the lower bounds we have derived.

The scheme given in [38] is based on a family of ℓ-wise independent functions. A function is ℓ-wise independent if the knowledge of the value of the function in $\ell - 1$ different points of the domain does not convey any information on the value of the function in another point. It enables ℓ conferences in $\mathcal{C} \subseteq 2^{\mathcal{U}}$, *not known* a priori, to securely compute a conference key. The family of ℓ-wise independent functions chosen in [38] to construct the $(k, n, \mathcal{C}, \mathcal{G})$-DKDS is the family of all bivariate polynomials $P(x, y)$ over a given finite field F_q, in which the degree of x is $k - 1$ and the degree of y is $\ell - 1$. The protocol can be described as follows

Set Up Phase

- Let $\ell = \max_{G \in \mathcal{G}} \ell_G$ be the maximum number of conference keys that a coalition G of malicious users can compute.
- Each of the servers $S_1, \ldots, S_k$ constructs a random bivariate polynomial $P^i(x, y)$ of degree $k-1$ in x and $\ell - 1$ in y, by choosing uniformly at random $k \cdot \ell$ elements in F_q.
- Then, for $i = 1, \ldots, k$, server S_i evaluates, for $j = 1, \ldots, n$, the bivariate polynomial $P^i(x, y)$ in the identity j of server S_j, and sends the univariate polynomial $Q_j^i(y) = P^i(j, y)$ to S_j.
- For $j = 1, \ldots, n$, each server S_j computes his private information a_j, summing up the k polynomials of degree $\ell - 1$, obtained from the k servers $S_1, \ldots, S_k$. More precisely,

$$a_j = Q_j(y) = \sum_{i=1}^{k} Q_j^i(y).$$

A user who needs a conference key, sends a key-request message to the servers as follows

Key Request Phase

- A user in conference C_h, who wants to compute the conference key, sends to at least k servers, say $S_{i_1}, \ldots, S_{i_k}$, a request for the conference key.
- Each server S_{i_j}, invoked by the user, checks that the user belongs to C_h, and sends to the user the value $Q_{i_j}(h)$, i.e., the value of the polynomial $Q_{i_j}(y)$ evaluated in $y = h$.

Finally, using the k values received from $S_{i_1}, \ldots, S_{i_k}$, and applying the Lagrange formula, each user in C_h recovers the secret key $P(0, h) = \sum_{i=1}^{k} P^i(0, h)$. More precisely,

Key Computation Phase

- Each user computes, for $j = 1, \ldots, k$, the coefficients

$$b_j = \prod_{1 \leq s \leq k, s \neq j} \frac{i_s}{i_s - i_j}.$$

Then, he recovers $P(0, h)$ computing the $\sum_{j=1}^{k} b_j y_{i_j}$ where y_{i_j}, for $j = 1, \ldots, k$, is the value received from server S_{i_j}, i.e., $y_{i,j} = Q_{i_j}(h)$.

It is not difficult to see that the protocol satisfies Definition 3.1 and meets the bounds established by Theorems 5.7, 5.2, 5.3, and 5.8.

8 Conclusions and Open Problems

Since this paper was written and submitted in February 2000, further researches on this topic have been done, and some related papers have been published. In [7] the ramp approach has been investigated. It allows to reduce the 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. The model analysed in this paper has been extended in [8], by considering a general family of subsets of servers, referred to as the *access structure*, authorized to help the users in recovering the conference keys. Therein, lower bounds on the resources have been shown in an easy and elegant way, by using a reduction technique which relates DKDSs to Secret Sharing Schemes. Moreover, a linear algebraic approach to designing DKDSs has been proposed. In [23, 41] *robust* DKDCs have been proposed: each server can *verify* that the information it stores and uses to answer the users' key-request messages is consistent with the information stored by the other servers; at the same time, the users are *guaranteed* that they can compute the same key for a given conference in which they belong to. Moreover, time is divided in *periods*, and at the beginning of each period the servers are involved in an update procedure that "refreshes" the private information they store while the conference keys they provide stay the same.

Some questions still arise from the analysis we have presented. We have assumed that each user has private connections with *all* the n servers of the network. It can be interesting to study the same problem assuming that, for each user, there are d possible different secure connections with the servers, where $k \leq d \leq n$, or assuming more general network topologies. As well as, it would be of interest to investigate different models for a distributed key distribution center, and to design schemes, expecially in the computationally secure setting, provable secure under standard cryptographic assumptions. In [24] a step towards this direction has been done.

Acknowledgements

We would like to thank the anonymous referees for their careful reading and useful comments and suggestions, which improved the readability of the paper.

References

[1] G. Ateniese, M. Steiner, and G. Tsudik, *New Multiparty Authentication Services and Key Agreement Protocols*, IEEE Journal of Selected Areas in Communications Vol. 18, No. 4, pp. 1–13, 2000.

[2] C. Becker and U. Wille, *Communication Complexity of Group Key Distribution*, Proceedings of the 5th ACM Conference on Computer and Communication Security, ACM, pp 1–6, 1998.

[3] M. Bellare and P. Rogaway, *Entity Authentication and Key Distribution*, Advances in Cryptology - Crypto '93, Lecture Notes in Computer Science, Vol. 950, pp. 92–111, 1995.

[4] S. Blake-Wilson, D. Johnson, and A. J. Menezes, *Key agreement protocols and their security analysis* Proceedings of the Sixth IMA International Conference on Cryptography and Coding, Lecture Notes in Computer Science, Vol. 1355, pp. 30–45, 1997. Full version available at http://www.cacr.math.uwaterloo.ca/∼ajmeneze/research.html

[5] 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.

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

[7] C. Blundo, P. D'Arco and C. Padrò, *A Ramp Model for Distributed Key Distribution Schemes*, Discrete Applied Mathematics, N. 128, pp. 47–64, 2003.

[8] C. Blundo, P. D'Arco, V. Daza, and C. Padrò, *Bounds and Constructions for Unconditionally Secure Distributed Key Distribution Schemes for General Access Structures*, Theoretical Computer Science, Vol. 320, pp. 269-291, 2004.

[9] 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.

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

[11] D. Boneh, The Decision Diffie-Hellman Problem, *Proceedings of the Third Algorithmic Number Theory Symposium*, Lecture Notes in Computer Science, Vol. 1423, pp. 48–63, 1998.

[12] C. Boyd and A. Mathuria, *Protocols for Authentication and Key Establishment*, Springer-Verlag, 2003.

[13] E. Bresson, O. Chevassut, and D. Pointcheval, *The Group Diffie-Hellman Problems*, Proceedings of SAC '02, Lecture Notes in Computer Science, vol. 2595, pp. 325–338, 2002.

[14] E. Bresson, O. Chevassut, and D. Pointcheval, *Group Diffie-Hellman Key Exchange Secure Against Dictionary Attacks*, Advances in Cryptology - Asiacrypt '02, Lecture Notes in Computer Science, vol. 2501, pp. 497–514, 2002.

[15] E. Bresson, O. Chevassut, and D. Pointcheval, *Dynamic Group Diffie-Hellman Key Exchange under Standard Assumptions*, In Advances in Cryptology - Eurocrypt '02 Lecture Notes in Computer Science vol. 2332, pp. 321–336, 2002.

[16] E. Bresson, O. Chevassut, and D. Pointcheval, *Provably Authenticated Group Diffie-Hellman Key Exchange: The Dynamic Case*, In Advances in Cryptology - Asiacrypt '01 Lecture Notes in Computer Science vol. 2248, pp. 290–309, 2001.

[17] M. Burmester and Y. Desmedt, *A Secure and Efficient Conference Key Distribution System*, Advances in Cryptology - Eurocrypt '94 Lecture Notes in Computer Science vol. 950, pp. 275–286, 1995.

[18] R. Canetti and H. Krawczyk, *Analysis of Key-Exchange Protocols and Their Use for Building Secure Channels*, Advances in Cryptology - Eurocrypt '01, Lecture Notes in Computer Science, vol. 2045, pp. 453–474, 2001.

[19] R. Canetti and H. Krawczyk, *Universally Composable Notions of Key Exchange and Secure Channels*, Advances in Cryptology - Eurocrypt '02, Lecture Notes in Computer Science, vol. 2332, pp. 337–351, 2002.

[20] R. Canetti and H. Krawczyk, *Security Analysis of IKE's Signature based Key Exchange Protocol*, Advances in Cryptology - Crypto '02, Lecture Notes in Computer Science, vol. 2442, pp. 143–161, 2002.

[21] J. Clark and J. Jacob, *A survey of authentication protocol literature: Version 1.0.* http://citeseer.nj.nec.com/clark97survey.html

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

[23] P. D'Arco and D. Stinson, *On Unconditionally Secure Robust Distributed Key Distribution Centers*, Advances in Cryptology, Proceedings of Asiacrypt 2002, Lecture Notes in Computer Science, Vol. 2501, pp. 346-363, 2002.

[24] V. Daza, J. Herranz, C. Padró, and G. Saéz, *A Distributed and Computationally Secure Key Distribution Scheme*, Information Security Conference (ISC 2002), Lecture Notes in Computer Science, Vol. 2433, pp. 342–356, 2002.

[25] W. Diffie and M. Hellman, *New Directions in Cryptography*, IEEE Transactions on Information Theory, N. 22, pp. 644–654, 1976.

[26] W. Diffie, P. C. Van Oorschot, and M. J. Wiener, *Authentication and Authenticated Key Exchanges*, Design, Codes, and Cryptography, vol. 2, pp. 107–125, 1992.

[27] ETH Crypto Group - Home page at http://www.crypto.ethz.ch/

[28] I. Ingemarsson, D. Tang, and C. Wong, *A Conference Key Distribution System*, IEEE Transactions on Information Theory, Vol. 28, No. 5, pp. 714–720, 1982.

[29] A. Joux, *A one round protocol for tripartite Diffie-Hellman*, Algorithmic Number Theory, 4th International Symposium, Lecture Notes in Computer Science, Vol. 1838, pp. 385–393, 2000.

[30] M. Just and S. Vaudenay, *Authenticated Multi-party Key Agreement*, Advances in Cryptology - Asiacrypt 96, Lecture Notes in Computer Science, vol. 1163, pp. 36–49, 1996.

[31] J. Katz and M. Yung, *Scalable Protocols for Authenticated Group Key Exchange*, available at http://eprint.iacr.org/2003/171, report 2003/171, 2003.

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

[33] K. Kurosawa, R. Okada, and Keiichi Sakano, *Security of the Center in Key Distribution Schemes*, Advances in Cryptology - Asiacrypt '94, Lecture Notes in Computer Science, vol. 917, pp. 333–341, 1995.

[34] U. Maurer and S. Wolf, *Secret-Key Agreement Over Unauthenticated Public Channels - Part I: Definition and Completeness Result*, IEEE Transactions of Information Theory, Vol. 49, No. 4, pp. 822–831, April 2003.

[35] U. Maurer and S. Wolf, *Secret-Key Agreement Over Unauthenticated Public Channels - Part II: The Simulatability Condition*, IEEE Transactions of Information Theory, Vol. 49, No. 4, pp. 832–838, April 2003.

[36] U. Maurer and S. Wolf, *Secret-Key Agreement Over Unauthenticated Public Channels - Part III: Privacy Amplification*, IEEE Transactions of Information Theory, Vol. 49, No. 4, pp. 839–851, April 2003.

[37] A.J. Menezes, P.C. Oorschot, and S.A. Vanstone, **Handbook of Applied Cryptography**, CRC Press, 1996.

[38] 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.

[39] 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.

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

[41] V.Nikov, S. Nikova, B. Preneel, and, J. Vandewalle, *On Distributed Key Distribution Centers and Unconditionally Secure Proactive Verifiable Secret Sharing Schemes based on General Access Structure*, INDOCRYPT 2002, Lecture Notes in Computer Science 2551, Springer-Verlag, pp. 422-437, 2002.

[42] R. Rueppel and P.C. van Oorschot, *Modern key agreement techniques*, Computer Communications, vol. 17, no.7, pp.458-465, 1994.

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

[44] V. Shoup, *On Formal Models for Secure Key Exchange*, available at http://eprint.iacr.org/1999/012, report 2003/012, 1999.

[45] M. Steiner, G. Tsudik and M. Waidner, *Diffie-Hellman Key Distribution Extended to Groups*, Proceedings of the 3-rd ACM Conference on Computer and Communications Security, pp. 31–37, 1996.

[46] M. Steiner, G. Tsudik and M. Waidner, *Key Agreement in Dynamic Peer Groups*, IEEE Transactions on Parallel and Distributed Systems, Vol. 11, No. 8, pp. 769–780, 2000.

[47] D. Steer, L. Strawczynski, W. Diffie and M. Wiener, *A secure Audio Teleconference System*, in Advances in Cryptology - Crypto '88, Vol. 403, pp. 520–528, 1988.

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