LNCS 2200

George I. Davida
Yair Frankel (Eds.)

Information Security

4th International Conference, ISC 2001
Malaga, Spain, October 2001
Proceedings

Springer

George I. Davida   Yair Frankel (Eds.)

Information Security

4th International Conference, ISC 2001
Malaga, Spain, October 1-3, 2001
Proceedings

Springer

Series Editors

Gerhard Goos, Karlsruhe University, Germany
Juris Hartmanis, Cornell University, NY, USA
Jan van Leeuwen, Utrecht University, The Netherlands

Volume Editors

George I. Davida
University of Wisconsin-Milwaukee, Department of EECS
Milwaukee, WI 53201, USA
E-mail: davida@cs.uwm.edu

Yair Frankel
Techtegrity, LLC
122 Harrison, Westfield NJ 07090, USA
E-mail: yfrankel@cryptographers.com

Cataloging-in-Publication Data applied for

Die Deutsche Bibliothek - CIP-Einheitsaufnahme

Information security : 4th international conference ; proceedings / ISC
2001, Malaga, Spain, October 1 - 3, 2001. George I. Davida ; Yair Frankel
(ed.). - Berlin ; Heidelberg ; New York ; Barcelona ; Hong Kong ; London ;
Milan ; Paris ; Tokyo : Springer, 2001
(Lecture notes in computer science ; Vol. 2200)
ISBN 3-540-42662-0

CR Subject Classification (1998): E.3, D.4.6, K.6.5, F.2.1, C.2, J.1, C.3

ISSN 0302-9743
ISBN 3-540-42662-0 Springer-Verlag Berlin Heidelberg New York

This work is subject to copyright. All rights are reserved, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, re-use of illustrations, recitation, broadcasting, reproduction on microfilms or in any other way, and storage in data banks. Duplication of this publication or parts thereof is permitted only under the provisions of the German Copyright Law of September 9, 1965, in its current version, and permission for use must always be obtained from Springer-Verlag. Violations are liable for prosecution under the German Copyright Law.

Springer-Verlag Berlin Heidelberg New York
a member of BertelsmannSpringer Science+Business Media GmbH

http://www.springer.de

© Springer-Verlag Berlin Heidelberg 2001
Printed in Germany

Typesetting: Camera-ready by author, data conversion by PTP-Berlin, Stefan Sossna
Printed on acid-free paper SPIN: 10840698 06/3142 5 4 3 2 1 0

Preface

The Information Security Conference 2001 brought together individuals involved in multiple disciplines of information security to foster the exchange of ideas. The conference, an outgrowth of the Information Security Workshop (ISW) series, was held in Málaga, Spain, on October 1–3, 2001. Previous workshops were ISW '97 at Ishikawa, Japan; ISW '99 at Kuala Lumpur, Malaysia; and ISW 2000 at Wollongong, Australia. The General Co-chairs, Javier López and Eiji Okamoto, oversaw the local organization, registration, and performed many other tasks.

Many individuals deserve thanks for their contribution to the success of the conference. José M. Troya was the Conference Chair. The General Co-chairs were assisted with local arrangements by Antonio Maña, Carlos Maraval, Juan J. Ortega, José M. Sierra, and Miguel Soriano.

This was the first year that the conference accepted electronic submissions. Many thanks to Dawn Gibson for assisting in developing and maintaining the electronic submission servers. The conference received 98 submissions of which 37 papers were accepted for presentation. These proceedings contain revised versions of the accepted papers. Revisions were not checked and the authors bear full responsibility for the contents of their papers.

The Program Committee consisted of Elisa Bertino, Università di Milano; G. R. Blakely, Texas A&M University; John Daugman, Cambridge University; Jorge Dávila, Polytechnic Univ. of Madrid; Giovanni DiCrescenzo, Telcordia; Josep Domingo-Ferrer, Univ. Rovira i Virgili; Dieter Gollmann, Microsoft Research; Sigrid Guergens, GMD; Hiroaki Kikuchi, Tokai University; Chi-Sung Laih, Natl. Cheng Kung Univ.; Wenbo Mao, HP Laboratories; Masahiro Mambo, Tohoku University; Catherine Meadows, NRL; Sang-Jae Moon, Kyungpook Natl. University; Yuko Murayama, Iwate Prefectural University; René Peralta, Yale University; Josef Pieprzyk, University of Wollongong; Sihan Qing, Chinese Academy of Sciences; Susie Thomson, Datacard; Routo Terada, Univ. of S. Paulo; Yiannis Tsiounis, InternetCash; Moti Yung, Certco; Yuliang Zheng, Monash University; Jianying Zhou, Oracle Corp. Members of the committee spent numerous hours in reviewing papers and providing advice. The committee was also assisted by our colleagues: Marc Alba, Clemente Galdi, Sang-Wook Kim, Yi Mu, Anna Oganian, Francesc Sebé, Rajan Shankaran, Igor Shparlinski. We apologize for any inadvertent omissions. Our thanks to the program committee and all reviewers.

We thank all the authors who submitted papers to this conference. Without their submissions this conference could not have been a success.

July 2001

George I. Davida
Yair Frankel

Information Security Conference 2001

October 1–3, 2001, Málaga, Spain

Conference Chair

José M. Troya, University of Málaga (Spain)

General Co-chairs

Javier López, University of Málaga (Spain)
Eiji Okamoto, Toho University (Japan)

Program Co-chair

George I. Davida, University of Wisconsin-Milwaukee (USA)
Yair Frankel, TechTegrity L.L.C. (USA)

Program Committee

Elisa Bertino	Università di Milano (Italy)
G. R. Blakely	Texas A&M University (USA)
John Daugman	Cambridge University (UK)
Jorge Dávila	Polytechnic Univ. of Madrid (Spain)
Giovanni DiCrescenzo	Telcordia (USA)
Josep Domingo-Ferrer	Univ. Rovira i Virgili (Spain)
Dieter Gollmann	Microsoft Research (UK)
Sigrid Guergens	GMD (Germany)
Hiroaki Kikuchi	Tokai University (Japan)
Chi-Sung Laih	Natl. Cheng Kung Univ. (Taiwan)
Wenbo Mao	HP Laboratories (UK)
Masahiro Mambo	Tohoku University (Japan)
Catherine Meadows	NRL (USA)
Sang-Jae Moon	Kyungpook Natl. University (Korea)
Yuko Murayama	Iwate Prefectural University (Japan)
Rene Peralta	Yale University (USA)
Josef Pieprzyk	University of Wollongong (Australia)
Sihan Qing	Chinese Academy of Sciences (China)
Susie Thomson	Datacard (UK)
Routo Terada	Univ. of S. Paulo (Brazil)
Yiannis Tsiounis	InternetCash (USA)
Moti Yung	CertCo (USA)
Yuliang Zheng	Monash University (Australia)
Jianying Zhou	Oracle Corp. (USA)

试读结束：需要全本请在线购买： www.ertongbook.com

Table of Contents

Key Distribution

Protocols

Enhancing Technologies

Privacy

Software Protection

Message Hiding I

PKI Issues and Protocols

Hardware Implementations

Cryptanalysis and Prevention

Implementations

Non-repudiation Techniques

Contracts and Auctions

Message Hiding II

Payments

Security Applications

Network and OS Security

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

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

[1] Dipartimento di Informatica ed Applicazioni
Università di Salerno, 84081 Baronissi (SA), Italy
{carblu, paodar}@dia.unisa.it
[2] Departament de Matemàtica Aplicada IV
Universitat Politècnica de Catalunya, 08034 Barcelona, Spain
{vdaza, matcpl}@mat.upc.es

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

Keywords: Key Distribution, Protocols, Distributed Systems.

1 Introduction

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

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

* The work of the third and the fourth authors was partially supported by Spanish *Ministerio de Ciencia y Tecnología* under project TIC 2000-1044.

G.I. Davida and Y. Frankel (Eds.): ISC 2001, LNCS 2200, pp. 1–17, 2001.
© Springer-Verlag Berlin Heidelberg 2001

试读结束：需要全本请在线购买： www.ertongbook.com

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

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

A common solution is the use of a Key Distribution Center (KDC, for short), a server responsible of the distribution and management of the secret keys. The idea is the following. Each user shares a common key with the center. When he wants to securely communicate with other users, he sends a request for a conference key. The center checks for membership of the user in that conference, and distributes in encrypted form the conference key to each member of the group. Needham and Schroeder [20] began this approach, implemented most notably in the Kerberos System [21], and formally defined and studied in [1], where it is referred to as the *three party model.*

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

Several kinds of Key Distribution Schemes have been considered so far: Key Pre-Distribution Schemes (KPSs, for short), Key Agreement Schemes (KASs, for short) and Broadcast Encryption Schemes (BESs, for short) among others. The notions of KPS and KAS are very close to each other [4,18,6]. 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 [11], anonymous broadcast transmission [16], re-keying protocols for secure multi-cast communications [9,10,22].

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

In [19] a new approach to key distribution was introduced to solve the above problems. A Distributed Key Distribution Center (DKDC, for short) is a set of n servers of a network that jointly realizes the same function of a Key Distribution Center. A user who needs to participate to a conference, sends a key-request to a subset at his choice of the n servers. The contacted servers answer with some information enabling the user to compute the conference key. In such a model, a single server by itself does not know the secret keys, since they are *shared* between the n servers, the communication bottleneck is eliminated, since the key-request messages are distributed, on average, along different paths, and there is no single point of failure, since if a server crashes, the other are still able to support conference key computation.

In a subsequent paper [5], the notion of DKDC has been studied from an information theoretic point of view. Therein, the authors introduced the concept of a distributed key distribution scheme (DKDS, for short), a scheme realizing a DKDC, showing that the protocol proposed in [19], based on ℓ-wise independent functions, is optimal with respect to the amount of information needed to set up and manage the system.

In this paper we extend the model studied in [5], by considering a general family of subsets of servers, referred to as the *access structure*, that are authorized to help the users in recovering the conference keys. In the DKDSs studied in [5] the users must send a key-request to at least a certain number of servers, that is, only *threshold* access structures were considered. Because of the general framework, there will be possible to specify the access structure depending on the features of each one of the servers. We present bounds holding on the model using a reduction technique which relates DKDSs to Secret Sharing Schemes [3, 23]. This technique enables us to prove lower bounds on the memory storage, the communication complexity and the randomness needed to set up the scheme in an easy and elegant way. Moreover, we describe a linear algebraic approach to design DKDSs. Namely, we present a method to construct a DKDS from a linear secret sharing scheme and a family of linear ℓ-wise independent functions. The optimality of the obtained schemes relies on the optimality of the secret sharing schemes that are used in their construction. We show that some known previous constructions belong to the proposed framework. This approach is quite suitable since it allows a unified description of seemingly different schemes.

In the journal version of this paper we will describe two construction techniques to realize DKDS for any access structure. The first construction we will consider is based on the cumulative array method introduced in [25]. The second construction will be based on the technique of Benaloh and Leichter [2] which can be applied to any monotone formula describing the access structure.

Organization of the Paper. Some basic facts about secret sharing are recalled in Section 2. A Model for distributed key distribution schemes and the notation we use in the paper are given in Section 3. Some lower bounds on the amount of information that must be stored by the servers and on the size in bits of the messages each server has to send in order to reply key requests, are given in Section 4. There, it is also shown a lower bound on the randomness needed by the scheme. These bounds are found by using the relation between DKDSs secret sharing schemes and depend on the optimal value for the information rate of secret sharing schemes with access structure in the set of servers. In Section 5, we present a linear algebraic method to construct DKDSs from linear secret sharing schemes.

2 Secret Sharing Schemes

A secret sharing scheme (SSS, for short) is a protocol by means of which a dealer distributes a secret $k \in K$ into shares among a set of participants $\mathcal{S}$ in such a way that only the authorized subsets of $\mathcal{S}$ can reconstruct the secret k, whereas the participants in any non authorized subset of $\mathcal{S}$ cannot determine anything about the value of the secret. Secret sharing were introduced in 1979 by Blakley [3] and Shamir [23]. The readers unfamiliar with secret sharing can find an excellent introduction to this topic in [26].

The authorized subsets form the *access structure* $\mathcal{A} \subset 2^{\mathcal{S}}$ of the SSS. The subsets belonging to $\mathcal{A}$ are called *authorized* subsets, while those not in $\mathcal{A}$ are called *forbidden*. We consider only *monotone* access structures, that is, any set containing an authorized subset must be authorized.

Using information theory, the two properties a secret sharing scheme must satisfy can be stated as follows: assuming that A denotes both a subset of participants and the set of shares these participants receive from the dealer to share a secret $k \in K$, and indicating the corresponding random variables in bold, it holds

- *Any authorized subset can compute the secret:* formally, for each $A \in \mathcal{A}$, $H(\mathbf{K}|\mathbf{A}) = 0$
- *Any forbidden subset has no information on the secret:* formally, for each $F \notin \mathcal{A}$, $H(\mathbf{K}|\mathbf{F}) = H(\mathbf{K})$

The efficiency of a secret sharing scheme is usually quantified by some measurements; basically the *information rate*, and the *randomness* needed to set up the scheme. The information rate $\rho(\Sigma, \mathcal{A}, K)$ of a secret sharing scheme Σ with access structure $\mathcal{A}$ and set of secrets K is defined as the ratio between the *size of the secret* (measured in bits) and the *maximum size of the shares* given to the participants, while the randomness is the *number of random bits* used to setup the scheme. Secret sharing schemes with information rate $\rho = 1$, which is the maximum possible value of this parameter, are called *ideal*. An access structure $\mathcal{A}$ on $\mathcal{S}$ is said to be *ideal* if there exists an ideal secret sharing scheme with access structure $\mathcal{A}$.

Given an access structure $\mathcal{A}$ on $\mathcal{S}$, we will indicate with $\rho^*(\mathcal{A})$ the optimal information rate for a SSS with access structure $\mathcal{A}$. More precisely, $\rho^*(\mathcal{A}) = \sup \rho(\Sigma, \mathcal{A}, K)$, where the supremum is taken over all possible sets of secrets K with $|\, K \,| \geq 2$ and all secret sharing schemes Σ with access structure $\mathcal{A}$.

We recall next some basic facts about linear secret sharing schemes. Let E be a vector space with finite dimension over the finite field $GF(q)$. For every $S_i \in \mathcal{S} \cup \{D\}$, where $D = S_0 \notin \mathcal{S}$ is a special participant called *dealer*, let us consider a vector space E_i over $GF(q)$ and a surjective linear mapping $\pi_i : E \to E_i$. Let us suppose that these linear mappings verify that, for any $A \subset \mathcal{S}$,

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

This family of vector spaces and linear surjective mappings determines the access structure

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

A secret sharing scheme with set of secrets $K = E_0$ and access structure $\mathcal{A}$ is defined as follows: for a secret value $k \in E_0$, a vector $v \in E$ such that $\pi_0(v) = k$ is taken at random and every participant $S_i \in \mathcal{S}$ receives as its share the vector $a_i = \pi_i(v) \in E_i$. It is not difficult to prove that this is a secret sharing scheme with access structure $\mathcal{A}$. The information rate of this scheme is $\rho = \dim E_0/(\max_{1 \leq i \leq n} \dim E_i)$. Secret sharing schemes constructed in this way are called *linear secret sharing schemes* (LSSSs for short). In a LSSS, the secret is computed by a linear mapping from the shares of the participants in an authorized subset. That is, for every $A = \{S_{i_1}, \ldots, S_{i_r}\} \in \mathcal{A}$, there exists a linear mapping $\chi_A : E_{i_1} \times \cdots \times E_{i_r} \rightarrow E_0$ that enables the participants in A to compute the secret.

Linear secret sharing schemes were first introduced by Brickell [7], who considered only ideal linear schemes with $\dim E_i = 1$ for any $S_i \in \mathcal{S} \cup \{D\}$. General linear secret sharing schemes were introduced by Simmons [24], Jackson and Martin [15] and Karchmer and Wigderson [17] under other names such as geometric secret sharing schemes or monotone span programs. In an ideal linear secret sharing scheme with $\dim E_0 = 1$, we can consider that the surjective linear mappings π_i are non-zero vectors in the dual space E^*. In that case, a subset $A \subset \mathcal{S}$ is authorized if and only if the vector $\pi_0 \in E^*$ can be expressed as a linear combination of the vectors $\{\pi_i \,|\, S_i \in A\}$. The access structures that can be defined in this way are called *vector space access structures*. Threshold access structures are a particular case of vector space access structures. Effectively, if $\mathcal{A}$ is the (t, n)-threshold access structure, we can take $q > n$ a prime power and $x_i \in GF(q)$, for any $p_i \in \mathcal{S}$, non-zero pairwise different elements and consider $E = GF(q)^t$, $\pi_0 = (1, 0, \ldots, 0) \in E^*$ and $\pi_i = (1, x_i, x_i^2, \ldots, x_i^{t-1}) \in E^*$ for any $i = 1, \ldots, n$. The ideal linear scheme we obtain in this way is in fact equivalent to the Shamir's threshold scheme [23].

Using the *monotone circuit construction* due to Ito, Saito and Nishizeki [14], Simmons, Jackson and Martin [25] proved that any access structure $\mathcal{A}$ can be realized by a linear secret sharing scheme. The main drawback of the LSSSs that are constructed by using the general method proposed in [25] is that their information rates are in general very small.

Nevertheless, using decomposition techniques, linear secret sharing schemes with much better information rate can be found for some access structures. Those techniques consist of decomposing the given access structure $\mathcal{A}$ into several substructures and combining secret sharing schemes on these substructures in order to obtain a secret sharing scheme for $\mathcal{A}$. For instance, one of the most powerful decomposition techniques to construct secret sharing schemes with good information rate is the *λ-decomposition construction* due to Stinson [27]. A linear secret sharing scheme is obtained when combining linear schemes in a λ-decomposition construction.

3 The Model

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

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

Key-Request Phase. Let $C_h \in \mathcal{C}$ be a conference, that is, a group of users who need to securely communicate. Each user U_j in C_h, contacts the servers belonging to some subset $P \in \mathcal{A}$, requiring a key for the conference C_h. We denote such a key by κ_h. Server $S_i \in P$, contacted by user U_j, checks[1] for membership of U_j in C_h; if the check is satisfied, he computes a value $y^h_{i,j} = F(a_i, j, h)$. Otherwise, he sets $y^h_{i,j} = \perp$, a special value which does convey no information about κ_h. Finally, S_i sends the value $y^h_{i,j}$ to U_j.

Key-Computation Phase. Once having received the answers from the contacted servers, each user U_j in C_h computes $\kappa_h = G_P(y^h_{i_1,j}, \ldots, y^h_{i_{|P|},j})$, with $i_1, \ldots, i_{|P|}$ those indices of the contacted servers, and G_P is a publicly known function.

We are interested in formalizing, within an information theoretic framework[2], the notion of a DKDS, in order to quantify *exactly* the amount of resources that a *real-world* implementation of such a system can require. To this aim, we need to setup our notation.

- Let $\mathcal{C} \subset 2^{\mathcal{U}}$ be the set of conferences on $\mathcal{U}$ indexed by elements of $\mathcal{H} = \{1, 2, \ldots\}$.
- For any coalition $G = \{U_{j_1}, \ldots, U_{j_g}\} \in \mathcal{G}$ of users, denote by $\mathcal{C}_G = \{C_h \in \mathcal{C} : C_h \cap G \neq \emptyset\}$ the set of conferences containing some user in G, and by $\mathcal{H}_G = \{h \in \mathcal{H} : C_h \in \mathcal{C}_G\}$ the set of corresponding indices. Let us

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

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

consider $\ell = \ell_{\mathcal{G}} = \max_{G \in \mathcal{G}} |\mathcal{C}_G|$, the maximum number of conferences that are controlled by a coalition in $\mathcal{G}$.

- For $i = 1, \ldots, t$, let $\Gamma_{i,j}$ be the set of values $\gamma_{i,j}$ that can be sent by server S_i to server S_j, for $j = 1, \ldots, n$, and let $\Gamma_j = \Gamma_{1,j} \times \cdots \times \Gamma_{t,j}$ be the set of values that S_j, for $j = 1, \ldots, n$, can receive during the initialization phase.
- Let K_h be the set of possible values for κ_h, and let A_i be the set of values a_i the server S_i can compute during the initialization phase.
- Finally, let $Y^h_{i,j}$ 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 the conference C_h.

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

Table 1. Cartesian Products

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

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

Roughly speaking, a DKDC must satisfy the following properties:

- **Correct Initialization Phase.** When the initialization phase correctly terminates, each server S_i must be able to compute his private information a_i. On the other hand, if server S_i misses/does-not-receive *just one* message from the servers in P_I[3] sending information, then S_i must not gain any information about a_i. We model these two properties by relations 1 and 2 of the formal definition.
- **Consistent Key Computation.** Each user in a conference $C_h \subseteq \mathcal{U}$ must be able to compute *the same* conference key, after interacting with the servers

[3] Without loss of generality, we choose P_I as one of the smallest subsets in $\mathcal{A}$ because one of our aim is to minimize the randomness and the communication complexity of the initialization phase.

试读结束：需要全本请在线购买： www.ertongbook.com