

UNIVERSITY OF CALGARY

Towards Designing a More Efficient Conference Key Distribution Scheme With Unconditional
Security

by

Fatemeh Arbab

A THESIS

SUBMITTED TO THE FACULTY OF GRADUATE STUDIES
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE
DEGREE OF MASTER OF SCIENCE

DEPARTMENT OF COMPUTER SCIENCE

CALGARY, ALBERTA

September, 2011

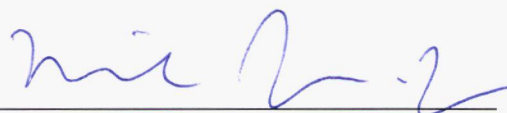
© Fatemeh Arbab 2011

UNIVERSITY OF CALGARY
FACULTY OF GRADUATE STUDIES

The undersigned certify that they have read, and recommend to the Faculty of Graduate Studies for acceptance, a thesis entitled "Towards Designing a More Efficient Conference Key Distribution Scheme With Unconditional Security" submitted by Fatemeh Arbab in partial fulfillment of the requirements for the degree of Master of Science.



Supervisor, Dr Reyhaneh Alsadat Safavi-Naeini
Department of Computer Science



Internal Examiner, Dr Michael John Jacobson Jr.
Department of Computer Science



External Examiner, Dr Christoph Simon
Department of Physics and Astronomy



Date

Abstract

Conference key distribution (CKD) schemes enable a group of eligible users, referred to as a conference, to compute a common secret key. Assumptions on the size of an eligible conference or the communication model are often made when designing such schemes. In an attempt to make CKD schemes more applicable to real life scenarios, one might challenge these assumptions. In particular, having a perfectly secure and relatively efficient CKD scheme that enables conferences of arbitrary sizes to compute a common key is desirable. Also, in real life communication between two users is confined by spatial restrictions such as wired connectivity in wired networks, or coverage domain in wireless networks and so, it is not realistic to assume broadcast channels where a message sent by a user is received by every other users, unchanged. Although communication graphs are known to be better tools to model the communication infrastructure, they have not been assumed in CKD schemes as far as we know.

In this thesis, we study a variety of CKD schemes and propose three schemes that extend the most efficient existing CKD scheme with constant conference size, to allow conferences of varying sizes. To our knowledge, all previous CKD schemes that are known so far assume broadcast model. We also design a new CKD scheme that uses communication graphs to model conference connectivity, and show that the scheme is more efficient compared to the other known schemes.

Acknowledgements

To my supervisor, Dr Reyhaneh Alsadat Safavi-Naeini (Rei); thank you for giving me the opportunity to work in your research group, being extremely patient with me, wanting me to achieve the best I could be and above all, caring me to smile. I will always be proud of getting *your* approval on this work.

To my parents; thank you for always reminding me of my goals and values, supporting me long distance and lovingly loving me even when I didn't deserve it.

To my brother; thank you for being the best emergency contact person, on the forms and the phone.

To my dear friends, their spouses and in some cases even their parents; thank you for making my time in Calgary so memorable, for your words of encouragement, technical help, scientific discussions and critical comments. Thank you for being with me during the fun times as well as the less fortunate times. To Daniel Medeiros de Castro, Fatemeh Keshavarz, Hadi Ahmadi, Maryam Elahi, Maryam Khajepour, Mohammed Ashraful Alam Tuhin, Seyed Hossein Ahmadinejad, Shreya Rawal and Zahra Aghazadeh.

And finally, I would like to acknowledge the impact of the people of my home country, Iran, who gave my life a new dimension during and after the 2009 presidential election.

Table of Contents

iv

Abstract	ii
Acknowledgements	iii
Table of Contents	iv
List of Tables	vi
List of Figures	vii
Nomenclature	viii
1 Introduction	1
2 Background and definitions	5
2.1 Information theory	5
2.2 Communication model	7
2.3 Cryptography	9
2.4 Adversary model	11
2.5 Security model	12
2.5.1 Perfect secrecy	12
2.6 Conference key distribution	14
2.7 Secret Sharing Schemes	17
2.8 Graph theory	19
2.9 Definitions from Combinatorics	20
2.10 New Definitions for this thesis	22
3 Previous Work	24
3.1 Samples of CKP Schemes	24
3.1.1 Trivial (g, b) -CKP Scheme	25
3.1.2 Blom's $(2, b)$ -CKP Scheme	26
3.1.3 Fiat-Naor's $(\leq n, b)$ -CKP Scheme	27
3.1.4 Blundo et al.'s (g, b) -CKP Scheme	29
3.1.5 More on Symmetric Polynomials	29
3.2 Communication in Key Distribution Schemes	32
3.2.1 Non-interactive Schemes	32
3.2.2 Interactive Schemes	35
3.3 Samples of CKA Schemes	44
3.3.1 Blundo et al.'s 1-restricted $(\leq n, b)$ -CKA Scheme	44
3.3.2 Blundo et al.'s 1-Restricted (g, b) -CKA Scheme	46
3.3.3 Blundo et al.'s τ -Restricted (g, b) -CKA Scheme	49
3.4 Analysis of the τ -Restricted (g, b) -CKA scheme of Blundo et al.	50
3.4.1 Soundness of the construction	51
3.4.2 Growth of the Field Sizes, k_i 's	52
3.4.3 Comparison of the τ -Restricted (g, b) -CKA scheme to previous CKA schemes	54
3.5 Conclusions	57
4 A τ -restricted CKA scheme for conferences of varying sizes	59
4.1 Introduction	59

4.2	Scheme 1	61
4.2.1	Efficiency	62
4.3	Scheme 2	63
4.3.1	Performance	66
4.4	Scheme 3	66
4.4.1	Performance	67
4.5	Comparison	68
4.5.1	Key rates	68
4.5.2	Communication rate	72
4.6	Conclusion	72
5	CKD scheme for Tree Structured Conferences	76
5.1	Motivation	76
5.2	The model	77
5.3	Tree Structured Key Agreement	78
5.4	Security of the scheme	83
5.5	Performance	86
5.6	Comparison	87
5.6.1	Technical lemma	87
5.6.2	1-restricted m -tree (g, b) -CKA scheme vs. (g, b) – 1RD scheme	90
5.7	Simulation	91
5.8	Conclusions	95
6	Conclusions and Future Work	96
6.1	Future Work	97
A	99
A.1	Key rates	99
A.1.1	Graphic presentation	99
A.1.2	Comparing key rates of Scheme 1 and 2	100
A.1.3	Comparing key rates of Scheme 2 and 3	100
A.2	Communication rates	101
B	103
B.1	Key rate	103
	Bibliography	105

List of Tables

2.1	Notations look up table	23
3.1	Summary of the studied CKD schemes	58

List of Figures

2.1	The conditional entropy, joint entropy and mutual information of two random variables, \mathbf{X}, \mathbf{Y}	8
3.1	Mapping permutations to the coefficients of a symmetric polynomial	30
3.2	Computation field sizes' fluctuations	53
3.3	Field sizes line up	53
4.1	Scheme 1's key rates for \mathcal{G}_{3i+2}	69
4.2	Scheme 2's key rates for \mathcal{G}_{3i+2}	70
4.3	Scheme 3's key rates for \mathcal{G}_{3i+2}	70
4.4	Scheme 1's communication rates for \mathcal{G}_{3i+2}	73
4.5	Scheme 2's communication rates for \mathcal{G}_{3i+2}	73
4.6	Scheme 3's communication rates for \mathcal{G}_{3i+2}	74
5.1	Gnutella peer to peer file sharing network from August 4th 2002 for about 100 nodes	77
5.2	Leaf-to-root phase of the conference key computation	82
5.3	Root-to-leaf phase of the conference key computation	83
5.4	1-restricted m -tree (g, b) -CKA scheme's key rate vs. $(g, b) - 1RD$ scheme's	93
5.5	Cross over section of the key rates for 1-restricted m -tree (g, b) -CKA scheme and the $(g, b) - 1RD$ scheme	94

Nomenclature

$(g, b) - 1RD$ scheme	1-restricted (g, b) -CKA scheme of Blundo et al.
$(g, b) - \tau RD$ scheme	τ -restricted (g, b) -CKA scheme of Blundo et al.
CKA scheme	Conference Key Agreement scheme
CKD scheme	Conference Key Distribution scheme
CKP scheme	Conference Key Pre-distribution scheme
TA	Trusted Authority

Chapter 1

Introduction

When users communicate over public channels such as the internet, they are sharing their information not only with the intended recipients but potentially to other users that have access to the channel. To ensure that messages are private between the sender and the intended receiver, the messages being sent over the channel can be encrypted. The original message is often called the *plaintext* and the encrypted message is often called the *ciphertext*. In a cryptosystem, the *encryption algorithm* describes how to transform a plaintext into a ciphertext and the *decryption algorithm* is the process of retrieving the plaintext from the ciphertext. In symmetric cryptosystems, the encryption and decryption algorithms share a common *secret key*. Without the knowledge of the secret key, the ciphertext is unreadable. In this thesis we only work with symmetric cryptosystems.

In key distribution schemes, the goal is to ensure that two users who wish to apply a symmetric cryptosystem are equipped with identical secret keys. In situations where more than two users wish to communicate over public channels, it is desirable to devise a method by which all the users get access to the same secret key. This leads to conference key distribution (CKD) schemes. A CKD scheme is a method of enabling a group of users, referred to as a conference, to agree on a common key. Generally speaking, a CKD scheme consists of two phases: *initialization* and *conference key computation*. In the initialization phase, some private information is distributed among all the users which will later be used in the conference key computation phase to generate the common key. The conference key computation phase itself can be completed non-interactively or interactively. In a non-interactive CKD scheme, often referred to as conference key pre-distribution (CKP) scheme, the users can individually complete the conference key computation phase whereas in an interactive CKD scheme, often referred to as a conference key agreement (CKA) scheme, the users in a

conference are required to interact among themselves in order to complete the conference key computation phase.

In CKA schemes, different communication models might be assumed. In this thesis we only study two of such models, *broadcast model* and *communication graph model*. In a broadcast model, it is assumed that once a user sends a message, all the other users receive it. However, in a communication graph model, every user is presented as a node of a graph and every edge connecting two nodes represents a communication channel between the respective users.

Since in this thesis we study a number of different CKD schemes, we need to have a measure to compare the performance of these schemes. We define two such measures, *key rate* and *communication rate*. The key rate of a CKD scheme measures the ratio of the size of the initially distributed private information to the size of a typical common key that a conference computes at the end of the conference key computation phase. In other words, this measure gives an estimate of the amount of private information that has to be distributed among all the users in order to compute a common key. The communication rate is the ratio of the size of all the communicated messages within a conference to the size of a typical key produced at the end of the conference key computation phase. We compare schemes with respect to either their key rate or communication rate. The scheme with lower rate, key or communication, is considered more efficient.

Since a CKD scheme enables users to compute identical secret keys for future communications, it is essential to ensure that the computed conference keys at the end of the conference key computation phase, satisfy certain security requirements. This propagates to the security of the communicated messages using a symmetric cryptosystem.

Any user who is not an intended receiver for a message is a potential, so called, bad guy. The bad guys might collaborate in different ways to learn about the informations that they are not entitled to know. We often assume that there exists an adversary who corrupts the users, bad guys, and by defining the adversary's capabilities, we model the possible behaviours of the bad guys.

There are two approaches to designate a secure cryptographic scheme: *unconditional* and *computational* models. In an unconditionally secure scheme, no limitations are assumed on the computational power of an adversary and the arguments are made in an information theoretic setting. Information theory provides a tool to quantify the information contained in any mathematically presentable phrase. To prove a scheme is unconditionally secure, the argument often involves showing that the publicly available information does not leak any information about a term that has to remain secret. In a computationally secure scheme, the computational power of the adversary is limited. Adversary's inability to solve a hard mathematical problem concludes the security of such schemes. Note that once a scheme is proved to be computationally secure, it might not remain secure in a later time since an effective solution to the hard problem it had assumed might be discovered during a course of time. One such hard problem is factoring large integers into prime factors. For instance, in [14] Peter Shor shows how the factoring problem can be feasibly solved on a quantum computer. However, there has not been a realization of a quantum computer such that Shor's algorithm can be run on for large integers.

In this thesis, we study a number of CKD schemes with a broadcast communication, unconditionally secure and passive adversary model. A passive adversary can only get access to the private information of a fixed number of corrupted users. The adversary is also static which means he can only decide on who to corrupt at the beginning of the protocol and can not modify them at a later time. According to the efficiency measures, we compare the performance of these schemes. We also review a previous result that analytically proves an optimal key rate in an unconditionally secure model.

As the first contribution of this work, we extend the result on [5], in which Blundo et al. present a CKA scheme to compute τ conference keys for conferences of all the same size, g . In our extension, we study the possibility of having a CKA scheme to compute τ conference keys for conferences of varying sizes, g_1, \dots, g_τ .

The second contribution of this work is a new CKA scheme which is based on a communication graph as its communication model rather than a broadcast model and argue that the communication graphs are more accurate models to simulate real life communication scenarios. We specifically consider conferences whose communication graph has an underlying spanning tree such that by properly choosing a root node for it, every non-leaf node has exactly m children. We show that our new scheme always achieves better communication rate than the previous broadcast based schemes, when applied on a communication graph. We also prove that for certain parameter values, our scheme attains better key rate than the previously known most key rate efficient CKA scheme.

The significance of the contributions is in lowering the key rates of the CKD schemes. A lower key rate means a more efficient use of devices' memory which is a critical factor in wireless sensor networks.

Organization:

In Chapter 2, we present the relative definitions to our work. Chapter 3 contains a review on the previous works including some analysis and performance comparisons. In Chapter 4 we present and analyze the proposals to extend [5] to compute τ conference keys for conferences of varying sizes. Chapter 5 contains our new CKA scheme whose design is based on communication graphs. We conclude the thesis in Chapter 6 with a summary of the results and possible future directions to it.

Chapter 2

Background and definitions

This chapter contains the background knowledge and all the definitions we need in this thesis.

2.1 Information theory

Information Theory enables us to measure the amount of information contained in a mathematically representable phrase. Similarly, it addresses more general questions such as the amount of information that two such phrases jointly contain or the conditional information contained in a mathematically representable phrase given another such phrase, etc.

We can think of any phrase that contains some information, as a random variable over a sample set that takes its values with respect to a probability distribution. We can also define numeric values to the random variable defined over any sample set by defining a one to one function from the sample set to the set of real numbers, \mathbb{R} . The word *random* emphasizes on the fact that we are dealing with experiments governed by laws of chance rather than any deterministic law [12]. The information contained in a random variable, \mathbf{X} , can itself be expressed as how much the entropy, randomness or uncertainty about the value of \mathbf{X} is cleared out once its actual value is known. In other words, how much information one obtains by learning the value of \mathbf{X} .

Definition 1. [12] A **random variable** is a real-valued function defined over the sample space of a random experiment.

Definition 2. [12] A random variable with discrete probability distribution is called a **discrete random variable**.

Remark 1. We denote a random variable with uppercase boldface letters, say \mathbf{X} , and by the cor-

responding *italic uppercase letter*, X , we refer to the respective sample set. Let the *italic lowercase letters* present the numeric value that such an random variable can take, x , with respect to the probability distribution $\{\Pr(x)\}_{x \in X}$.

The amount of information contained in a random variable is regarded as its entropy. In other words, the entropy of a random variable represents how uncertain one can be about the actual value of that random variable. The term *entropy* usually refers to the *Shannon entropy*, which quantifies the expected value of the information contained in a message, usually measured in bits. Equivalently, the Shannon entropy is a measure of the average information content one is missing when he does not know the value of a certain random variable.

Definition 3. [16] For a discrete random variable \mathbf{X} , which takes on values from a finite set X , the **entropy** of \mathbf{X} is defined to be the quantity:

$$H(\mathbf{X}) = - \sum_{x \in X} \Pr(x) \log_b \Pr(x). \quad (2.1)$$

The choice of b determines the unit of measurement.

Throughout this thesis, we take $b = 2$ and hence the unit is *bit*.

Definition 4. [16] The **conditional entropy**, $H(\mathbf{X}|\mathbf{Y})$, is the weighted average, with respect to the probabilities $\Pr(y)$, of the entropies $H(\mathbf{X}|y)$ over all possible values y :

$$H(\mathbf{X}|\mathbf{Y}) = - \sum_{x \in X} \sum_{y \in Y} \Pr(y) \Pr(x|y) \log_2 \Pr(x|y) \quad (2.2)$$

In other words, the conditional entropy measures the average amount of information about \mathbf{X} that is revealed by \mathbf{Y} .

Corollary 1. *For two statistically independent random variables, \mathbf{X} and \mathbf{Y} , we have:*

$$\begin{aligned}
 H(\mathbf{X}|\mathbf{Y}) &= -\sum_{x \in X} \sum_{y \in Y} \Pr(y) \Pr(x|y) \log \Pr(x|y) \\
 &= -\sum_{x \in X} \sum_{y \in Y} \Pr(y) \Pr(x) \log \Pr(x) \\
 &= (\sum_{y \in Y} \Pr(y)) (-\sum_{x \in X} \Pr(x) \log \Pr(x)) \\
 &= H(\mathbf{X}).
 \end{aligned} \tag{2.3}$$

Let $\Pr(x,y)$ be the joint probability for random variables \mathbf{X} and \mathbf{Y} . The following corollary follows.

Corollary 2. *The joint entropy of two random variables, \mathbf{X} and \mathbf{Y} , is defined as:*

$$H(\mathbf{X}, \mathbf{Y}) = H(\mathbf{X}) + H(\mathbf{Y}|\mathbf{X}). \tag{2.4}$$

Another important characteristic of two random variables is their mutual information which, roughly speaking, measures their dependency. For instance, two independent random variables over a sample set have zero mutual information whereas if they are dependent, their mutual information would be a positive value.

Definition 5. [12] *If $\Pr(x|y)$ represents the conditional probability for two random variables, \mathbf{X} and \mathbf{Y} , the **mutual information** of these two random variables is defined as:*

$$I(\mathbf{X}; \mathbf{Y}) = H(\mathbf{X}) - H(\mathbf{X}|\mathbf{Y}). \tag{2.5}$$

Figure 2.1 from [1] depicts a Venn diagram, presenting the joint entropy, conditional entropy and mutual information of two random variables, \mathbf{X} , \mathbf{Y} .

2.2 Communication model

With a communication model we specify the characteristics of the means of communication between the users. In Distributed Computing, the communication model is perhaps the most signifi-

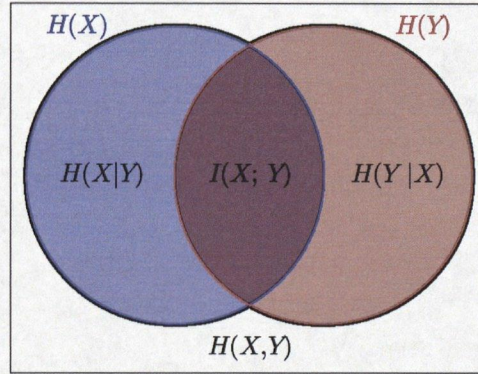


Figure 2.1: The conditional entropy, joint entropy and mutual information of two random variables, X, Y .

cant parameter of the setting. In [10], an inclusive description of possible communication models and their characteristics are given. Regardless of the specifications of the communication model, it contains channels. Below are some of the basic properties of any communication channel. [11]

- A *channel* is a means of conveying information from one user to another.
- A *secured channel* is one from which an adversary does not have the ability to reorder, delete, insert or read.
- An *insecure channel* is one from which parties other than those for which the information is intended can reorder, delete, insert or read.

In this thesis we either work with *point to point* or *broadcast channels*.

In a point to point channel, every message is intended to be received by exactly one user. There are additional assumptions when working in a point to point channel communication model. Firstly, we also assume a global clock such that every action takes place at a tick of this clock. We also assume that our point to point channels are *synchronous*, i.e. once a message is sent at time t , we assume it is received by time $t + 1$. We take *noise free* channels, which are channels through

which a message is received identically as it was sent, i.e. the channel doesn't change the messages that pass through it.

In this thesis, by a broadcast channel we refer to a noise free synchronous channel in which once a message is sent, all the users receive it.

2.3 Cryptography

To introduce cryptography, an understanding of the issues related to information security in general is necessary. Quantifying information enables us to study such issues. Studying any transition with respect to its cryptographic characteristics is of special importance since it ensures the parties that a certain level of information security objectives are obtained. [11]

Definition 6. [11] *Cryptography is the study of mathematical techniques related to aspects of information security such as confidentiality, data integrity, entity authentication and data origin authentication.*

It is important to be advised that in cryptography, we are more concerned with the non-physical means of assuring information security such as mathematical algorithms. Also note that cryptography provides us with a set of possible techniques to do so, and not necessarily an exclusive set of such techniques. Below we introduce each of the cryptographic goals separately [11].

1. **Confidentiality** is a service used to keep the content of information from all but those authorized to have it.
2. **Data integrity** is a service that addresses the unauthorized alternation of data. Data manipulation includes insertion, deletion and substitution.
3. **Authentication** is a service related to identification. This function applies to both entities and information itself. Two parties entering into a communication should identify each other.

Information delivered over a channel should be authenticated as to origin, date of origin, data content, time sent, etc.

4. **Non-repudiation** is a service that prevents users from denying actions they have previously committed.

In this work, our cryptographic goal is to preserve confidentiality. The following sample gives a better intuition of the setting. To model data transitions, it is often assumed that Alice wants to send a message to Bob over an insecure channel such that her messages stays confidential against an eavesdropper, Eve. The initial message that Alice wishes to send to Bob is often referred to as *plaintext*. Alice transforms this message to a *ciphertext* using a predetermined key. The ciphertext is only understandable to Bob who shares some previously determined information, such as the encryption key, with Alice. On the other hand, Eve should not be able to realize the plaintext by only seeing the ciphertext. The primary goal of cryptography is to enable Alice and Bob to securely communicate in such setting. The definition below introduces a cryptosystem which is a formalization of this idea.

Definition 7. [16] A **cryptosystem** is a five-tuple $(\mathcal{P}, \mathcal{C}, \mathcal{K}, \mathcal{E}, \mathcal{D})$, where the following conditions are satisfied:

1. \mathcal{P} is a finite set of possible plaintexts.
2. \mathcal{C} is a finite set of possible ciphertexts.
3. \mathcal{K} , the key space, is a finite set of possible keys.
4. For each $K \in \mathcal{K}$, there is an encryption rule $e_K \in \mathcal{E}$ and a corresponding decryption rule $d_K \in \mathcal{D}$. Each $e_K : \mathcal{P} \rightarrow \mathcal{C}$ and $d_K : \mathcal{C} \rightarrow \mathcal{P}$ are functions such that $d_K(e_K(x)) = x$ for every plaintext element $x \in \mathcal{P}$.

Note that by a *user* we refer to someone or something that sends, receives or manipulates information. For example, Alice and Bob are the users in a cryptosystem. In general, a user can be a person, a computer terminal, etc. Users engage in a series of actions to fulfill a certain task with an expected level of security. These series of actions are often referred to as protocols or schemes.

Definition 8. [11] *A cryptographic protocol, is a distributed algorithm defined by a sequence of steps precisely specifying the actions required of two or more users to achieve a specific security objective.*

2.4 Adversary model

To represent the users who try to learn about the information they are not entitled to know, we consider an adversary who corrupts the users. Once a user is corrupted, his locally private information and all the messages he communicates will be revealed to the adversary. We further classify the adversary type with respect to the extent of the control he obtains over the corrupted users. If an adversary only views the private information of a corrupted user, he is called a *passive adversary*. On the other hand, if an adversary can both access the private information of corrupted users and force them to manipulate the information according to his will, the adversary is called an *active adversary*. In other words, a passive adversary only steals the corrupted users' private information and does not influence their performance whereas an active adversary steals the corrupted users' private information and forces them to behave in his favour.

The behaviour of an adversary can yet be studied from another perspective, his dynamics. Both passive and active adversaries can be *static* or *adaptive*. A static adversary chooses the users he wishes to corrupt once and at the beginning of the protocol and doesn't change them afterward. However, the adaptive adversary can decide who to corrupt at any point through the execution of the protocol, taking advantage of the information he has accumulated by then.

In [7] a formal and well detailed description of different adversary models is given. For this thesis, we only consider passive static adversary model.

2.5 Security model

The security model of any protocol signifies how that protocol is robust against the designated adversary. In general, there are two main frameworks to define security: *computational* and *unconditional security*. The computational security model is based on the hardness of solving certain problems, such as factoring large integers into their prime factors. The argument is that, since the adversary cannot feasibly factor large integers, he cannot break the protocol, provided that the only way to break the protocol is to factor a large integer. Quantum computers are perhaps most threatening for the security of protocols that rely on discrete logarithm or factorization as the underlying hard problem. In [14], Peter Shor gave an algorithm to factor large numbers on a quantum computer in polynomial time. However, there are no realizations of a quantum computer to practically implement Shor's algorithm for large integers on. In the unconditional security model, the adversary is computationally unbounded. The argument is that even if the adversary has unlimited computational power, there is no way for him to learn anything about the information he is not entitled to know in an unconditionally secure protocol. In this thesis, we only work in an unconditionally secure model.

2.5.1 Perfect secrecy

Take $(\mathcal{P}, \mathcal{C}, \mathcal{K}, \mathcal{E}, \mathcal{D})$ a specific cryptosystem in which each key $k \in \mathcal{K}$ is used for only one encryption. Let us assume some fixed probability distribution on the plaintext space, \mathcal{P} . We also assume that the key is chosen by a specific probability distribution over the key space, \mathcal{K} . This allows us to define random variables, \mathbf{X} and \mathbf{K} , to define elements of the plaintext and the key, respectively. Since the key is chosen before Alice knows what plaintext she might be sending, it is a

reasonable assumption to take \mathbf{X} and \mathbf{K} as two independent random variables. The two probability distributions on \mathcal{P} and \mathcal{K} induce a probability distribution on \mathcal{C} , the ciphertext space. Let \mathbf{Y} denote the random variable that defines the elements of \mathcal{C} . To compute this induced probability on \mathcal{K} , for any $k \in \mathcal{K}$ we define $C(k)$ as the set of all possible ciphertexts when k is used as the key:

$$C(k) = \{e_k(x) : x \in \mathcal{P}\}. \quad (2.6)$$

For every $y \in \mathcal{C}$, the induced probability on the ciphertext space is:

$$\Pr(\mathbf{Y} = y) = \sum_{\{k: y \in C(k)\}} \Pr(\mathbf{K} = k) \Pr(\mathbf{X} = d_k(y)). \quad (2.7)$$

We now have obtained the required terms to define perfect secrecy. Intuitively speaking, a cryptosystem has perfect secrecy if the adversary can not learn anything about the plaintext by only observing the ciphertext.

Definition 9. [16] A cryptosystem has **perfect secrecy** if for all $x \in \mathcal{P}$ and $y \in \mathcal{C}$:

$$\Pr(x|y) = \Pr(x). \quad (2.8)$$

That is a posterior probability that the plaintext is x given that the ciphertext y is observed, is identical to the a priori probability that the plaintext is x .

The theorem below, proof of which is provided in [16, Theorem 2.4], gives a characterization, originally by Shannon, of when perfect secrecy can be obtained.

Theorem 1. [16] Suppose $(\mathcal{P}, \mathcal{C}, \mathcal{K}, \mathcal{E}, \mathcal{D})$ is a cryptosystem where $|\mathcal{K}| = |\mathcal{C}| = |\mathcal{P}|$. Then the cryptosystem provides perfect secrecy if and only if every key is used with equal probability $\frac{1}{|\mathcal{K}|}$, and for every $x \in \mathcal{P}$ and every $y \in \mathcal{C}$, there exists a unique key $k \in \mathcal{K}$ such that $e_k(x) = y$.

A very classic example of a cryptosystem with perfect secrecy is the One-time Pad, which was first introduced by Gilbert Vernam in 1917.

Definition 10. [16] One-time Pad

Let $n \geq 1$ be an integer and take $\mathcal{P} = \mathcal{C} = \mathcal{K} = (\mathbb{Z}_2)^n$. For $k \in (\mathbb{Z}_2)^n$, define $e_k(x)$ to be the vector sum modulo 2 of k and x , or equivalently the exclusive-or of the two associated bit strings. So, if $x = (x_1, \dots, x_n)$ and $k = (k_1, \dots, k_n)$, then:

$$e_k(x) = (x_1 + k_1, \dots, x_n + k_n) \mod 2. \quad (2.9)$$

Decryption is identical to encryption. If $y = (y_1, \dots, y_n)$, then:

$$d_k(y) = (y_1 + k_1, \dots, y_n + k_n) \mod 2. \quad (2.10)$$

2.6 Conference key distribution

Generally speaking, a conference key distribution (CKD) scheme is a method of distributing initial secrets shares among the users such that later when they form specific groups, called *conferences*, they can use their secret shares to compute a secret common value, referred to as the *conference key*. In an unconditionally secure setting, a CKD scheme consists of two phases: initialization and conference key computation.

Initialization:

In the initialization phase, a *trusted authority (TA)* privately distributes some secret information, referred to as user key, among all the users. The user keys will later be used to compute the conference keys. The TA does the initialization once at the beginning of the protocol and goes offline afterwards. Note that the TA has no prior knowledge about which conferences will later be formed.

Conference key computation:

The conference key computation phase can be *non-interactive*, which means that the user keys distributed by the TA and the available public information are sufficient for each conference member

to compute the conference key individually, or *interactive*, where users need to communicate in order to compute the conference key.

We refer to the non-interactive CKD schemes as *conference key pre-distribution (CKP)* schemes and by *conference key agreement* schemes we refer to the interactive CKD schemes.

The interaction among the users are done through public noise free channels, i.e. every user can see all the transmitted messages and the messages are delivered correctly to the recipients. We denote the set of all possible messages sent by the users in conference G by M_G and M represents the set of all communicated messages among all the conferences.

We take $\mathcal{U} = \{u_1, \dots, u_n\}$ as the set of n users with i the public id of user u_i . We interchangeably use i or u_i to refer to the same user in \mathcal{U} , as long as no confusion is caused. Let U_i denote the user key of user i , the set of all possible information that the TA sends privately to user i . For any conference $G = \{u_{i1}, \dots, u_{ig}\}$ we define $U_G = U_{i1} \times \dots \times U_{ig}$ and by K_G we denote the set of all possible values for G_i conference key. We assume that TA distributes $U_{\mathcal{U}}$ according to a probability distribution, $\{\Pr(u_{\mathcal{U}})\}_{u_{\mathcal{U}} \in U_{\mathcal{U}}}$. This induces a probability distribution $\{\Pr(k_G)\}_{k_G \in K_G}$ on K_G . The security model is an unconditional one with a passive static adversary.

We formally define a CKP scheme as follows.

Definition 11. [5] Let $\mathcal{U} = \{u_1, \dots, u_n\}$ be a set of n users with i the public id of user u_i . If g and b are two positive integers such that $g + b \leq n$, we define a (g, b) -**CKP scheme** to be a distribution method that satisfies the following:

1. *Correctness:* Each user i in a conference, G , of size g , can uniquely compute the conference key:

$$H(\mathbf{K}_G | \mathbf{U}_i) = 0, \quad \forall i \in G \subseteq \mathcal{U}, |G| = g.$$

2. *Perfect Secrecy:* No coalition of adversaries, A , of size at most b , disjoint from a conference, G , of size g , can learn anything about the conference key k_G :

$$H(\mathbf{K}_G | \mathbf{U}_A) = H(\mathbf{K}_G), \quad \forall A \subset \mathcal{U}, |A| \leq b, A \cap G = \emptyset.$$

Note that as the term pre-distribution suggests, in a CKP scheme the ultimate conference keys are pre-distributed among the users and hence no interaction is required to compute the conference keys. We formally define a CKA scheme as follows.

Definition 12. [5] Let $\mathcal{U} = \{u_1, \dots, u_n\}$ be a set of n users with i the public id of user u_i . If g and b are two positive integers such that $g + b \leq n$, we define a (g, b) -CKA scheme to be a key distribution method that satisfies the following:

1. *Interactive Property:* No conference member $i \in G$, with $|G| = g$ can compute the conference key without interaction with other conference members:

$$H(\mathbf{K}_G | \mathbf{U}_i) = H(\mathbf{K}_G), \quad \forall i \in G \subseteq \mathcal{U}, |G| = g. \quad (2.11)$$

2. *Correctness:* The transmitted messages among conference members and the private information given to each of the conference members uniquely determines the conference key:

$$H(\mathbf{K}_G | \mathbf{U}_i \mathbf{M}_G) = 0, \quad \forall i \in G \subseteq \mathcal{U}, |G| = g. \quad (2.12)$$

3. *Perfect Secrecy:* No coalition of adversaries, A , of size at most b , disjoint from a conference, G , of size g , can learn any information about the conference key, k_G , given all possible transmitted messages for all possible conferences, G_i , of size g and the collection of all the private information given to members of A :

$$H(\mathbf{K}_G | \mathbf{U}_A \mathbf{M}) = H(\mathbf{K}_G), \quad A \cap G = \emptyset, |A| = b, \mathbf{M} = \bigcup_{G \subseteq \mathcal{U}, |G|=g} \mathbf{M}_G. \quad (2.13)$$

In [5], Blundo et al. introduced the idea a CKA scheme that is perfectly secure for forming τ conference keys, namely τ -restricted CKA scheme. We will extensively study their scheme in Chapter 3.

Definition 13. [5] Let $\mathcal{U} = \{u_1, \dots, u_n\}$ be a set of n users with i the public id of user u_i . If g and b are two positive integers such that $g + b \leq n$, we define a τ -restricted (g, b) -CKA scheme to be a distribution method that satisfies the following:

1. *Interactive Property: No conference member $i \in G$, with $|G| = g$ can compute the conference key without interaction with other conference members:*

$$H(\mathbf{K}_G | \mathbf{U}_i) = H(\mathbf{K}_G), \forall i \in G \subseteq \mathcal{U}, |G| = g. \quad (2.14)$$

2. *Correctness: The transmitted messages among conference members and the private information given to each of the conference members uniquely determines the conference key:*

$$H(\mathbf{K}_G | \mathbf{U}_i \mathbf{M}_G) = 0, \forall i \in G \subseteq \mathcal{U}, |G| = g. \quad (2.15)$$

3. *Perfect Secrecy: For any τ conferences G_1, \dots, G_τ , with $|G_i| = g$ for $1 \leq i \leq \tau$, and for any $m_{G_1}, \dots, m_{G_\tau}$, no coalition of adversaries, A , of size b such that $A \cap G_i = \emptyset$, has any information on the k_{g_i} :*

$$H(\mathbf{K}_{G_i} | \mathbf{U}_A \mathbf{M}_{G_1} \dots \mathbf{M}_{G_\tau}) = H(\mathbf{K}_{G_i}), \quad A \cap G_i = \emptyset, |A| = b. \quad (2.16)$$

2.7 Secret Sharing Schemes

In secret sharing schemes, we study the methods to split a secret into shares and assign each user with a share such that only when an eligible subset of users put their shares together, the original secret can be reconstructed. There is a designated user in secret sharing schemes called the *dealer* who distributes the shares among the users correctly and privately. If a group of users do not satisfy the eligibility requirement, their shares will not result in a valid secret.

More formally, given a set of users, \mathcal{U} , and an access structure, \mathcal{A} , which is the set of authorized subsets of \mathcal{U} ; a secret sharing scheme provides the tool to compute and distribute secret shares among all the users in \mathcal{U} such that by pooling the secret shares of the users in any subset of \mathcal{A} , the original secret is reconstructed. At the same time, such scheme should ensure that the secret shares of any group of unauthorized users, $B \subset \mathcal{U}$ and $B \notin \mathcal{A}$, will not result in reconstructing the original secret. [11]

Threshold schemes are a special class of secret sharing schemes in which the eligibility requirement is on the size of the group of users who intend to reconstruct the key. More specifically, every group of at least t users, $t \in \mathbb{N}^+$, should be able to recompute the secret and groups of size less than t should not be able to reconstruct the secret when only using their secret shares.

In [13], Shamir gives a threshold secret sharing scheme. Here we briefly introduce Shamir's threshold scheme. Assume that $\mathcal{U} = \{u_1, \dots, u_n\}$ is a set of users such that i is the public identity of user u_i . In Shamir's scheme, the secret, s , is a random value of the field, $GF(q)$. The dealer randomly chooses a polynomial of degree $t - 1$ in one variable, f , over $GF(q)$ such that $f(0) = s$. The dealer computes $f(i)$ as the secret share for user u_i , for $1 \leq i \leq n$. Hence any group of at least t users have sufficiently many points of f to interpolate it and hence recompute s . However, every group of users of size less than t cannot interpolate f and hence have no information about $s = f(0)$ by only using their secret shares.

Note that although there are similarities in the general structure of a secret sharing scheme and a CKD scheme, i.e. an initialization and computation phase, there are a few distinctive remarks to be considered:

1. In a secret sharing scheme, users are merely share holders whereas in a CKD schemes, users are responsible to compute the conference key. This distinguishes between the potential application scenarios that each scheme can be used for with respect to the users characteristics. For instance, secret sharing schemes are suitable methods for access control whereas CKD schemes are more suitable to provide secure communications.
2. In a secret sharing scheme, there is one single secret that is shared among all the users and hence the shares of any eligible group of users result in reconstructing the same secret. However, in a CKD scheme, the conference key computed for any two distinct conferences should be statistically independent.

3. Following the previous point, a secret sharing scheme can be considered as a key pre-distribution scheme that facilitates one-time key establishment. [11]

2.8 Graph theory

This section is allocated to introduce some relative definitions from Graph Theory [9].

Definition 14. A **graph**, $G = \langle V, E \rangle$, is an ordered pair of parameters where V represents a set of vertices or nodes and E contains a collection of all subsets of V of size 2. Each $e = \{v_i, v_j\} \in E$ represent a relation between the respective nodes, v_i and v_j .

Definition 15. For a graph $G = \langle V, E \rangle$ and every pair of nodes $v_i, v_j \in V$, if $\{v_i, v_j\} \in E$ then we call v_i and v_j **neighbours**. Also the **degree** of a vertex, $d(v_i)$, is defined as the number of neighbours of $v_i \in V$.

Definition 16. In a graph $G = \langle V, E \rangle$, a **path** between two vertices, $v_i, v_j \in V$ is a sequence of vertices with v_i and v_j at the two ends such that every two consecutive vertices are connected via an edge. The number of edges in a path determines its **length**. A **cycle** in a graph is a path with identical end vertices. If there is a path between every two vertices of a graph, it is called a **connected graph**.

Definition 17. A **tree**, $T = \langle V, E \rangle$, is a connected graph with no cycles. Nodes of degree 1 in a tree are called **leaf nodes**.

It is not hard to show that in a tree $T = \langle V, E \rangle$ with $|V| = n$, there are $|E| = n - 1$ edges. We designate a non leaf node $v_r \in V$ and refer to it as the **root node**.

In a rooted tree $T = \langle V, E \rangle$ with root node v_r , we define a parent/child relation between every pair of neighbour nodes by considering the length of the path that connects each of the nodes to the root node. The node with the shorter path is called the *parent* and the one with longer path to the root is referred to as the *child*.

Definition 18. In a rooted tree $T = \langle V, E \rangle$, if all the non leaf nodes have exactly m children, the tree is called an **m -balanced rooted tree**.

2.9 Definitions from Combinatorics

In this section, we introduce the terms that are later used in Chapter 3 to construct some CKD schemes.

Definition 19. For q prime, a polynomial $P(x_1, \dots, x_t) = \sum_{0 \leq j_1, \dots, j_t \leq k} a_{j_1, \dots, j_t} (x_1)^{j_1} (x_2)^{j_2} \dots (x_t)^{j_t}$ of degree k , where $a_{j_1, \dots, j_t} \in GF(q)$, is said to be a **symmetric polynomial** if $P(x_1, \dots, x_t) = P(x_{\sigma(1)}, \dots, x_{\sigma(t)})$ for any permutation $\sigma : \{1, 2, \dots, t\} \rightarrow \{1, 2, \dots, t\}$. [6]

If a term in a symmetric polynomial can be obtained from the other by applying a permutation function, those two terms are called **equivalent terms**.

Remark 2. Few remarks regarding symmetric polynomials include:

1. In a symmetric polynomial in t variables, the coefficients of any two equivalent terms are identical. So a_{i_1, \dots, i_t} and a_{j_1, \dots, j_t} are equal if i_1, \dots, i_t is a permutation of j_1, \dots, j_t .
2. The equivalent relation between the terms is itself an equivalence relation, i.e. it is reflexive, symmetric and transitive. So the terms in a symmetric polynomial can be organized into equivalence classes with designated terms as class representatives.

The example below presents two polynomials, symmetric and non-symmetric.

Example 1. Let $f(x_1, x_2, x_3) = x_1^2 + x_2^2 + x_3^2 + 4x_1x_2x_3$ and $g(x_1, x_2, x_3) = 4x_1x_2 + 3x_1x_3$. It is not hard to see that f is symmetric, for instance let us switch x_1 and x_2 :

$$f(x_2, x_1, x_3) = x_2^2 + x_1^2 + x_3^2 + 4x_2x_1x_3 = f(x_1, x_2, x_3).$$

On the other hand, once switching x_1 and x_2 in g , we get:

$$g(x_2, x_1, x_3) = 4x_2x_1 + 3x_2x_3 \neq g(x_1, x_2, x_3),$$

hence g is not a symmetric polynomial.

As we will see later in Chapter 3, the scheme of Blundo et al. in [5] is based on objects from *Design Theory*. Design theory is a subdivision of Combinatorics in which numeric characteristics of sets of objects are related to their representation in specific structures.

Definition 20. A **design** is a pair (V, \mathcal{B}) , where V is a set of n elements, called points, and \mathcal{B} is a set of subsets of V of fixed size k , $k \geq 2$, called blocks.

Definition 21. A **parallel class** of (V, \mathcal{B}) consists of $\frac{n}{k}$ blocks from \mathcal{B} which partition the set V .

Definition 22. The design (V, \mathcal{B}) is said to be a **resolvable design** if the set of blocks, \mathcal{B} , can be partitioned into parallel classes. If \mathcal{B} consists all k subsets of V , then (V, \mathcal{B}) is called the *Complete k -Uniform Hypergraph on V* and in this case, there will be exactly $\binom{n-1}{k-1}$ many parallel classes.

The example below presents a resolvable design with its parallel classes.

Example 2. Let $V = \{1, 2, 3, 4, 5, 6\}$, $n = 6$, and $k = 3$, then

$$\begin{aligned} \mathcal{B} = & \{ \{1, 2, 3\}, \{1, 2, 4\}, \{1, 2, 5\}, \{1, 2, 6\}, \{1, 3, 4\}, \{1, 3, 5\}, \{1, 3, 6\}, \{1, 4, 5\}, \{1, 4, 6\}, \{1, 5, 6\}, \\ & \{2, 3, 4\}, \{2, 3, 5\}, \{2, 3, 6\}, \{2, 4, 5\}, \{2, 4, 6\}, \{2, 5, 6\}, \\ & \{3, 4, 5\}, \{3, 4, 6\}, \{3, 5, 6\}, \\ & \{4, 5, 6\} \} \end{aligned}$$

with $\binom{n}{k} = \binom{6}{3} = 20$ blocks is a complete 3-uniform hypergraph on V . Note that

$$C = \{ \{1, 2, 3\}, \{4, 5, 6\} \}$$

is a possible parallel class that contains $\frac{n}{k} = \frac{6}{3} = 2$ disjoint blocks of \mathcal{B} and the union of its components is V . But

$$C' = \{ \{1, 2, 3\}, \{1, 5, 6\} \}$$

does not form a parallel class. The classification below shows how \mathcal{B} can be partitioned into $r = \frac{\binom{6}{3}}{2} = 10 = \binom{5}{2} = \binom{k-1}{n-1}$ parallel classes:

$$\begin{aligned}
C_1 &= \{\{1,2,3\}, \{4,5,6\}\}, & C_2 &= \{\{1,2,4\}, \{3,5,6\}\}, \\
C_3 &= \{\{1,2,5\}, \{3,4,6\}\}, & C_4 &= \{\{1,2,6\}, \{3,4,5\}\}, \\
C_5 &= \{\{1,3,4\}, \{2,5,6\}\}, & C_6 &= \{\{1,3,5\}, \{2,4,6\}\}, \\
C_7 &= \{\{1,3,6\}, \{2,4,5\}\}, & C_8 &= \{\{1,4,5\}, \{2,3,6\}\}, \\
C_9 &= \{\{1,4,6\}, \{2,3,5\}\}, & C_{10} &= \{\{1,5,6\}, \{2,3,4\}\}.
\end{aligned}$$

The following theorem of Baranyai, proof of which is provided in [17, Theorem 36.1], gives a numerical relation between the parameters of a design to assure that it is a uniform complete hypergraph.

Theorem 2. *The complete k -uniform hypergraph on n points is resolvable if $n \equiv 0 \pmod k$.*

2.10 New Definitions for this thesis

As a part of the contributions of this thesis, we will introduce a CKA scheme that takes point to point channels as its communication model. We will represent such channels by using communication graphs in which each node represents a user and each edge represent a possible communication channel. We define an m -tree (g, b) CKA scheme as a key agreement scheme in which the communication channels are modelled by an m -balanced rooted tree.

Definition 23. *Let $T = \langle V, E \rangle$ be an m -balanced rooted tree with $V = \mathcal{U}$ in which every edge in E represents a communication channel between the respective users. An m -tree (g, b) -CKA scheme is a (g, b) -CKA scheme in which the messages can only travel through the edges of an m -balanced rooted tree, such as T .*

To measure the performance of a CKD scheme, we generalized the previous efficiency measure, *key rate*, for computing one conference key to τ conference keys. We also introduce another efficiency ratio, *communication rate*, that has not been used in the schemes we have studied so far.

Table 2.1: Notations look up table

Parameter	Description
n	total number of users
\mathcal{U}	the set of n users
u_i	user with the public id, i
G	a conference of size g
A	adversary set of size b
U_i	set of user key given to u_i
U_G	collection of all secret values given to users of G
M_G	set of all communicated messages between the users of G
M	set of all communicated messages
K_G	set of all possible values for the key of conference G
r	number of parallel classes for a resolvable design
χ	number of blocks in each parallel class for a resolvable design

The key rate of a scheme gives a measure on the rate of user keys distributed among all the users per bit of a typical conference key.

Definition 24. *The key rate of a CKD scheme is defined as the total size (in bits) of the user keys per bit of a conference key:*

$$\frac{\log_2 |U_{\mathcal{U}}|}{\log_2 |K_G|}. \quad (2.17)$$

Definition 25. *The communication rate of a CKD scheme is defined as the total size (in bits) of communicated messages in a conference per bit of that conference's key:*

$$\frac{\log_2 |M_G|}{\log_2 |K_G|}. \quad (2.18)$$

Remark 3. *According to Definition 11, the communication rate of any CKP scheme is 0.*

Table 2.10 summarizes the notations and parameters that will be used in future chapters.

Chapter 3

Previous Work

In this chapter, we review a number of constructions that represent different approaches to realize a perfectly secure CKD scheme and evaluate their performance according to Equation (2.17) and (2.18) [15]. We also recall the lower bound on the size of user keys for a perfectly secure CKP scheme, presented in [3]. In the next two chapters, we compare the key rate of our proposed schemes to the best key rate of the schemes presented in this chapter.

3.1 Samples of CKP Schemes

Let $\mathcal{U} = \{u_1, \dots, u_n\}$ denote the set of n users and $g, b \in \mathbb{N} \cup \{0\}$ such that $g + b \leq n$. We also take the finite field $GF(q)$ such that $q = p^k > n$ for p prime and $k \in \mathbb{N}$. As noted earlier, in a CKP scheme, users do not need to interact in order to compute the conference key and hence, for the communication rate of all the schemes presented in this section we have, $\frac{|M_G|}{|K_G|} = 0$.

Notation:

To make distinctions between the efficiency ratios of the different CKD schemes that we study in this chapter, we use the indices below:

CKD scheme	Index
Trivial (g, b) -CKP	(Trivial)
Blom's $(2, b)$ -CKP	(Blom)
Fiat-Naor's $(\leq n, b)$ -CKP	(FiatNaor)
Blundo et al.'s (g, b) -CKP	(Blundo)
Blundo et al.'s 1-Restricted $(\leq n, b)$ -CKA	(1Blundo)
Blundo et al.'s 1-Restricted (g, b) -CKA	(1RD)
Blundo et al.'s τ -Restricted (gn, b) -CKA	(τ RD)

3.1.1 Trivial (g, b) -CKP Scheme

The trivial scheme is the most straight forward way of realizing a CKP scheme. We assume that all possible conferences of size g are publicly enumerated. That is there exists a public table that contains the index of each conference and its members, so every users knows the index of the conferences he is a member of.

Initialization:

For a conference G_i , the TA randomly chooses a key, $k_{G_i} \in GF(q)$, and privately sends it to the users in G_i .

Conference Key Computation:

Each users in G_i can individually compute the conference key by using the distributed key, k_{G_i} .

Since no user outside of a certain conference, G_i , receives k_{G_i} , no information about k_{G_i} is given to a disjoint adversary set and hence the trivial scheme is unconditionally secure.

To compute the key rate for the trivial scheme, note that for every user $u_i \in \mathcal{U}$, there are $\binom{n-1}{g-1}$ conferences of size g that u_i can be a member of. As each $u_i \in \mathcal{U}$ receives a separate secret key for each of these conferences as his user key and a typical conference key is one of these user keys,

the key rate for the trivial scheme is:

$$\frac{|U_{\mathcal{U}}|}{|K_G|}^{(Trivial)} = n \binom{n-1}{g-1}. \quad (3.1)$$

We used an index (*Trivial*) to refer to the key rate of the trivial (g, b) -CKP scheme.

3.1.2 Blom's $(2, b)$ -CKP Scheme

This scheme was first proposed by Blom, [4], and is designed to produce pairwise keys between every two users, i.e. $g = 2$ and hence $b \leq n - 2$.

Initialization:

The TA randomly chooses a symmetric polynomial in 2 variables, $f(x_1, x_2)$, of degree b in each variable. TA evaluates $f(x_1, x_2)$ on its first variable for the public identity, i , of each user $u_i \in \mathcal{U}$, and sends the resulting 1-variable polynomial, $p_i(x_2) = f(i, x_2)$, to user u_i , privately.

Conference key computation

Two users can obtain a shared key by evaluating their polynomial on the public identity of the other user and use the resulting value as the conference key. That is, two users u_i and u_j individually compute:

$$f(i, j) = p_i(j) = p_j(i) \quad (3.2)$$

for their conference key.

An adversary who corrupts at most b users, say $A = \{u_{b1}, \dots, u_{bb}\}$, will get access to $p_{b1}(x_2) = f(b1, x_2), \dots, p_{bb}(x_2) = f(bb, x_2)$, each a polynomial in 1 variable of degree at most b . Since TA's symmetric polynomial, $f(x_1, x_2)$, is of degree at most b in each variable, at least $b + 1$ distinct shares are required to interpolate $f(x_1, x_2)$. Hence the adversary cannot interpolate $f(x_1, x_2)$ using the user keys of the corrupted users. This concludes the perfect secrecy of Blom's $(2, b)$ -CKP scheme.

As we will later see in Section 3.1.5, each user receives $(n - 1)$ secret values of $GF(q)$ from

the TA to uniquely identifies his 1-variable polynomial of degree at most b . Since the conference key is also an element of $GF(q)$, the key rate for this scheme is:

$$\frac{|U_{\mathcal{U}}|^{(Blom)}}{|K_G|} = n(n-1). \quad (3.3)$$

We used an index $(Blom)$ to refer to the key rate of Blom's $(2, b)$ -CKP scheme.

3.1.3 Fiat-Naor's $(\leq n, b)$ -CKP Scheme

In [8], Fiat and Naor proposed a CKP scheme that works for computing conference keys for conferences of any size. In their scheme, the maximum size of the adversary set is b and the distribution is based on the possible adversary sets. Let \mathcal{A} be the collection of adversary sets of size at most b , i.e. \mathcal{A} can be the collection of all subsets of \mathcal{U} of sizes 0 to b . Their scheme consists of two phases as follows:

Initialization:

Let $b < n$ denote the maximum size of an adversary set. For every $A \in \mathcal{A}$ with $|A| \leq b$, the TA chooses a random value $s_A \in GF(q)$, and distributes it to all users in $\mathcal{U} \setminus A$.

Conference key computation:

The key associated with any conference $G \subseteq \mathcal{U}$ is defined as:

$$k_G = \sum_{A \in \mathcal{A}: A \cap G = \emptyset} s_A.$$

Note that k_G is only known to the users in G , since it is composed of the portions of the user key that all members of G share. Such values, s_A 's, are not distributed among any user in a disjoint adversary set, $A \in \mathcal{A}$, $A \cap G = \emptyset$. Hence the conference key remains perfectly secure.

Since every user in \mathcal{U} receives a random secret value from $GF(q)$ with respect to every possible adversary set of size at most b , and the observation that a conference key itself is a random value

from $GF(q)$, we compute the key rate for Fiat-Naor's scheme as:

$$\frac{|U_{\mathcal{U}}|}{|K_G|}^{(FiatNaor)} = n \sum_{j=0}^b \binom{n}{j}. \quad (3.4)$$

We used an index (*FiatNaor*) to refer to the key rate of Fiat-Naor $(\leq n, b)$ -CKP scheme.

The example below illustrates how this scheme works.

Example 3. Let $\mathcal{U} = \{u_1, u_2, u_3, u_4\}$ and $b = 2$ with $GF(7)$. So \mathcal{A} consists of all the subsets of \mathcal{U} of size at most 2.

$$\mathcal{A} = \{\emptyset, \{1\}, \{2\}, \{3\}, \{4\}, \{1, 2\}, \{1, 3\}, \{1, 4\}, \{2, 3\}, \{2, 4\}, \{3, 4\}\}$$

The TA should send secret values $s_A \in GF(7)$ with respect to any adversary set, $A \in \mathcal{A}$. If the TA chooses the following values:

$$\begin{aligned} s_{\emptyset} &= 6, & s_{\{1\}} &= 3, & s_{\{2\}} &= 3, & s_{\{3\}} &= 2, \\ s_{\{4\}} &= 5, & s_{\{1,2\}} &= 3, & s_{\{1,3\}} &= 3, & s_{\{1,4\}} &= 1, \\ s_{\{2,3\}} &= 6, & s_{\{2,4\}} &= 2, & s_{\{3,4\}} &= 4, \end{aligned}$$

then TA sends s_A to all users in $\mathcal{U} \setminus A$. For instance, s_{\emptyset} is given to u_1, u_2, u_3, u_4 , $s_{\{1\}}$ is given to u_2, u_3, u_4 and $s_{\{1,2\}}$ goes to u_3, u_4 . For $G = \{2, 4\}$, the conference key is:

$$k_G = s_{\emptyset} + s_{\{1\}} + s_{\{3\}} + s_{\{1,3\}} = 0 \pmod{7}.$$

We note that both u_2 and u_4 receive $s_{\emptyset}, s_{\{1\}}, s_{\{3\}}, s_{\{1,3\}}$ as their user keys and hence are able to compute $k_{\{2,4\}}$ individually. Similarly we compute the conference key of all the other conferences:

$$\begin{aligned} k_{\{1,2,3,4\}} &= 6, & k_{\{2,3,4\}} &= 2, & k_{\{1,3,4\}} &= 2, & k_{\{1,2,4\}} &= 1, \\ k_{\{1,2,3\}} &= 4, & k_{\{3,4\}} &= 1, & k_{\{2,4\}} &= 0, & k_{\{2,3\}} &= 1, \\ k_{\{1,4\}} &= 3, & k_{\{1,3\}} &= 2, & k_{\{1,2\}} &= 3. \end{aligned}$$

Note that for any conference, G , and any adversary set, $A \in \mathcal{A}$, $A \cap G = \emptyset$, none of the summands to form k_G is given to any users in A and hence k_G remains completely unknown to A . On the other hand, all users in G receive all the summands in their user keys and are able to compute k_G individually.

3.1.4 Blundo et al.'s (g, b) -CKP Scheme

This scheme is a generalization of Blom's scheme [6]. Let $\mathcal{U} = \{u_1, \dots, u_n\}$ be the set of all users with i the public identity of user u_i .

Initialization:

The TA randomly chooses a symmetric polynomial, $f(x_1, \dots, x_g)$, in g variables of degree at most b in each variable, where g is the size of the conference and $b \leq n - g$ is the size of the adversary set. The TA evaluates $f(x_1, \dots, x_g)$ on its first variable for the public identity of each user and sends the resulting $(g - 1)$ -variable polynomial to the respective user, privately. For instance, user i receives:

$$p_i(x_2, \dots, x_g) = f(i, x_2, \dots, x_g).$$

Conference key computation:

When a group of g users decide to form a conference, each user evaluates his private polynomial on the public identity of the other $g - 1$ conference members. Since the original polynomial, $f(x_1, \dots, x_g)$, is a symmetric, every conference member evaluates the same value. This common value will be taken as the conference key. For instance, the conference key for $G = \{u_1, \dots, u_g\}$ is:

$$k_G = f(1, \dots, g) = p_1(2, \dots, g) = \dots = p_g(1, \dots, g - 1).$$

3.1.5 More on Symmetric Polynomials

To compute the efficiency of this scheme, we need to know the number of coefficients required to uniquely represent a symmetric polynomial, as this determines U_i and consequently $U_{\mathcal{U}}$. To do so, we show a correspondence between the terms in a symmetric polynomial and a combinatorial problem. The combinatorial problem is to enumerate the different possible arrangement of two distinct sets of identical objects. Let one of the sets contain b identical objects, referred to as “power objects” and another set of g identical objects, referred to as “variable objects”. For a given arrangement of these $g + b$ objects, we interpret the number of “power objects” on the left



Figure 3.1: Mapping permutations to the coefficients of a symmetric polynomial

hand side of the i -th “variable object” to be the value of α_i in the term $x_1^{\alpha_1} \dots x_i^{\alpha_i} \dots x_t^{\alpha_t}$. Once deciding all the α_i ’s for a single term we have essentially found a class representative term, see Remark 2. We randomly choose a coefficient for this term, $a_{\alpha_1, \dots, \alpha_t} \in GF(q)$, and put the same coefficient for all the terms equivalent to it. We repeat this process for all possible arrangements of the “variable objects” and “power objects”. Note that each possible arrangement introduces a class representatives, see Remark 2.

Figure 3.1 illustrates how the mapping between each arrangement and terms of a symmetric polynomial works. Let the ovals represent the “power objects” and sticks, the “variable objects”. This arrangement suggests the term $x_1^0 x_2^0 x_3^3 x_4^7 = x_3^3 x_4^7$. Once choosing a coefficient for this term, all other equivalent terms will take the same coefficient. In this case, the equivalent terms are:

$$x_1^3 x_2^7, x_1^3 x_3^7, x_1^3 x_4^7, x_2^3 x_1^7, x_2^3 x_3^7, x_2^3 x_4^7, x_3^3 x_1^7, x_3^3 x_2^7, x_3^3 x_4^7, x_4^3 x_1^7, x_4^3 x_2^7, x_4^3 x_3^7.$$

The total number of possible ways to arrange g identical elements together with another b identical elements is:

$$\frac{(g+b)!}{b! \times g!} = \binom{g+b}{g}.$$

This is the number of random variables that the TA needs to choose in order to form a symmetric polynomial in g variables and of degree b for each variable. Once evaluating such symmetric polynomial for one of it’s variables, we argue that the resulting polynomial is still a symmetric polynomial in $g - 1$ variables and hence TA needs to send $\binom{g+b-1}{g-1}$ coefficients to each user u_i as his user key, U_i .

Here we prove a lemma that states by fixing a variable, say x_1 , a symmetric polynomial in $t \geq 1$ variables, x_1, \dots, x_t , contains all the symmetric terms in the other $(t-1)$ variables, x_2, \dots, x_t .

Lemma 1. *Let f be a symmetric polynomial in $t \geq 1$ variables, x_1, \dots, x_t , and of degree at most k for each variable. By fixing any variable, x_i , $1 \leq i \leq t$, the symmetric polynomial, $f(x_1, \dots, x_t)$, includes all symmetric terms in the other $t - 1$ variables, $x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_t$.*

Proof. Let us represent f in the following format:

$$f(x_1, \dots, x_t) = c_1 h_1(\alpha_{12}, \dots, \alpha_{1t}) + \dots + c_m h_m(\alpha_{m1}, \dots, \alpha_{mt}),$$

with $m = \binom{k+t}{t}$, $c_i \in GF(q)$ and each $h_i(\alpha_{i1}, \dots, \alpha_{it})$ the class representative (see Remark 2) of all symmetric terms in t variables with $0 \leq \alpha_{ij} \leq k$ the power of the j -th variable, $1 \leq j \leq t$ and $1 \leq i \leq m$. Without loss of generality, we show the lemma is true when fixing x_1 and the same argument can be applied for any of the other variables. We can rewrite f as:

$$f(x_1, \dots, x_t) = x_1^0 r_0(x_2, \dots, x_t) + x_1^1 r_1(x_2, \dots, x_t) + \dots + x_1^k r_k(x_2, \dots, x_t),$$

where r_i , is a polynomial in $t - 1$ variables, x_2, \dots, x_t , and is obtained by factoring x_1^i out from all the terms in f in which x_1^i appears, $1 \leq i \leq k$. Note that r_i 's together should include all symmetric polynomials in $t - 1$ variables. Otherwise, there exists a term $x_2^{\beta_2} \times \dots \times x_t^{\beta_t}$ with $0 \leq \beta_2, \dots, \beta_t \leq k$ such that at least one of its equivalent terms is not showing up in any of the r_i 's. Multiplying x_1^i 's back, under the permutation σ_1 , the permutation function that is identical for input 1, i.e. $\sigma_1(1) = 1$, the missing equivalent term to $x_2^{\beta_2} \times \dots \times x_t^{\beta_t}$ remains missed. This contradicts the symmetry of f and hence no such term can appear in any of the r_i 's. This is equivalent to saying that r_i 's together should include all symmetric terms in $t - 1$ variables and of degree at most k , for each variable. Since f is symmetric, the same argument works when fixing any of the variables other than x_1 . \square

The argument above justifies the following representation for f :

$$f(x_1, \dots, x_t) = g_1(x_1)h_1(\beta_{12}, \dots, \beta_{1t}) + \dots + g_t(x_t)h_t(\beta_{t2}, \dots, \beta_{tt}),$$

where $g_i(x_1)$ is a polynomial of degree at most k and h_i is the class representative for all symmetric polynomials in $(t-1)$ variables with $0 \leq \beta_{ij} \leq k$, the power of the j -th term, $1 \leq j \leq t$ and $1 \leq i \leq \ell$. Note that $\ell = \binom{k+t-1}{t-1}$ is the number of equivalence classes.

Hence we conclude that in the (g, b) -CKP scheme of Blundo et al., the TA only needs to send $g_1(i), \dots, g_t(i)$ to each user $u_i \in \mathcal{U}$ obtained from $f(i, x_2, \dots, x_g)$, to uniquely identify $p_i(x_2, \dots, x_g) = f(i, x_2, \dots, x_g)$.

Remark 4. For $g = 2$, Blom's $(2, b)$ -CKP scheme becomes a special case of Blundo et al.'s (g, b) -CKP scheme.

Similar to the scheme of Section 3.1.2, the perfect secrecy of this scheme follows. For the efficiency measures, we compute:

$$\frac{|U_{\mathcal{U}}|}{|K_G|}^{(Blundo)} = n \binom{g+b-1}{g-1}, \quad (3.5)$$

We used an index $(Blundo)$ to refer to the key rate of Blundo et al.'s (g, b) -CKP scheme.

3.2 Communication in Key Distribution Schemes

One possible question at this stage is, does allowing interaction among the users at the conference key computation phase help to reduce the size of user keys? In [3], Beimel and Chor address this question. They prove a bound on the user key size for non-interactive and interactive CKD schemes. We present their results in the following two subsections.

3.2.1 Non-interactive Schemes

In [6], Blundo et al. proved a lower bound on the key rate of a perfectly secure CKP scheme and showed that their (g, b) -CKP scheme meets this bound with equality. Beimel and Chor in [3] prove the same lower bound for a slightly different security model. They show that even if one relaxes

the security requirements for a CKP scheme, as defined in Definition 26, the same size of user key has to be distributed at the initialization phase as is required in an unconditionally secure CKP scheme. In the new security setting, weak security, we assume that the adversary might have some guesses about the key by knowing the user key of the corrupted users. We start with the definition of the weak security property.

Definition 26. [3] **(Weak Security Property)** *Let $A \subset \mathcal{U}$ be an adversary set and let $G \subseteq \mathcal{U}$ be a conference, such that $G \cap A = \emptyset$. Then users in A having their user keys can not rule out any possible value for k_G . In other words, for every possible vector of user keys, $\vec{u} = \langle u_1, \dots, u_n \rangle$, that TA might distribute with positive probability among all the users and every possible key, $k_G \in K_G$, there exists a vector of user keys \vec{u}' that agrees with \vec{u} when restricted to A and results in the same key, k_G , for G :*

$$P(\mathbf{K}_G = k | \mathbf{U}_A) > 0, \quad \forall k \in K. \quad (3.6)$$

CKD schemes studied in [3] and [6], are namely threshold schemes, i.e. any g -subset of \mathcal{U} is a conference and any disjoint subset of size $b \leq n - g$ is an adversary set. We represent these schemes as (g, b) -CKD schemes.

The theorem below gives a lower bound on the size of a user's user key in a weakly secure CKP scheme.

Theorem 3. [3] *In a weakly secure (g, b) -CKP scheme with $g + b \leq n$ users, we have:*

$$|U_i| \geq |K|^{\binom{g+b-1}{g-1}}, \quad (3.7)$$

where K is the set of all possible conference keys and U_i is the set of user keys for user i , with $1 \leq i \leq n$.

The proof is given in [3], here we rewrite their proof.

Proof. Let $\mathcal{U} = \{user_1, \dots, user_n\}$ be the set of all users with $n = g + b$. Without loss of generality, we show the bound for the size of $user_1$'s user key. Let $G_1, \dots, G_\ell \subseteq \mathcal{U}$ be all the distinct conferences with $user_1$ as a member, so $\ell = \binom{g+b-1}{g-1}$. This gives a clue that $user_1$'s user key should include at least ℓ pieces of secret values. In fact we show that for every ℓ -vector of keys, $\vec{k} \in K^\ell$, there exists a *matching* vector of user keys $\vec{u} \in U_{\mathcal{U}}$. By a *matching* vector of user key, \vec{u} , to an ℓ -vector of keys, \vec{k} , we refer to the user keys that result in evaluating $k_j \in \vec{k}$ as the key of conference G_j , for all $1 \leq j \leq \ell$

If we can show that this correspondence between \vec{k} 's and \vec{u} 's exists, then we can conclude that for every distinct ℓ -vector of keys, $\vec{k} \in K^\ell$, there should exist a distinct value for u_1 in $\vec{u} \in U_{\mathcal{U}}$. This is because the scheme is non-interactive and hence $user_1$ computes the elements of \vec{k} by only using his user key, $u_1 \in \vec{u}$. So for every distinct ℓ -vector of keys, \vec{k} , there should exist a distinct vector of user keys, \vec{u} , in which u_1 takes a different value. This proves the inequality in Equation (3.7).

To show the correspondence between \vec{k} 's and \vec{u} 's, assume $\vec{k} = \langle k_1, \dots, k_\ell \rangle \in K^\ell$ is an ℓ -vector of conference keys without any matching vector of user keys in $U_{\mathcal{U}}$. Let i be the maximal index, due to non-uniqueness, such that for some $\vec{u}' \in U_{\mathcal{U}}$ and its matching $\vec{k}' = \langle k'_1, \dots, k'_\ell \rangle$, the first $(i-1)$ elements of \vec{k} and the first $(i-1)$ elements of \vec{k}' are identical and the first inconsistency appears on the i -th entity:

$$\forall 1 \leq j < i, k_j = k'_j \text{ and } k_i \neq k'_i.$$

Note that such index, $i \geq 2$, exists since for $i = 1$ we can come up with \vec{u}' such that the corresponding \vec{k}' and \vec{k} differ in their first component, as long as there are more than one possible value for k_1 .

Let $B = \mathcal{U} \setminus G_i$, so $|B| = b$. Note that B intersects with all the G_j 's, $j \neq i$, and given \vec{u}' , the members of B should be able to compute $k'_1, \dots, k'_{i-1}, k'_{i+1}, \dots, k'_\ell$ as the elements of the matching key vector, \vec{k}' . Since \vec{k} and \vec{k}' agreed on their first $(i-1)$ entities, members of B compute the first

$(i-1)$ entities of \vec{k} as well. Now, any positive probability of computing k_i , given $\mathbf{U}_B = \bigcup_{j \in B} u'_j$, with $u'_j \in \vec{u}'$, contradicts the maximality of i . Hence:

$$\Pr(\mathbf{K}_{G_i} = k_i | \bigcup_{j \in B} u'_j) = 0,$$

which contradicts the weak security property of the scheme. So the assumption of not having a matching user key vector to \vec{k} should be false. Hence, for any given ℓ -vector of conference keys, $\vec{k} \in K^\ell$, there exists a matching vector of user keys, $\vec{u} \in U_{\mathcal{U}}$. This completes the proof. \square

We can similarly conclude that in any CKP scheme with weak security property where each user can be a member of at least ℓ , $\ell \leq \binom{g+b-1}{g-1}$, distinct conferences, we have:

$$|U_i| \geq |K|^\ell. \quad (3.8)$$

Remark 5. *The (g, b) -CKP scheme of Blundo et al. introduced in Section 3.1.4, satisfies this lower bound.*

3.2.2 Interactive Schemes

By showing how to map a CKA scheme to a weakly secure CKP scheme, Beimel and Chor argue that including interaction cannot help to reduce the size of user key. Their proof method consists of showing how a proper choice of user key can compensate for the broadcast messages and hence a CKA scheme can be studied as a CKP scheme.

Theorem 4. [3] *For an unrestricted (g, b) -CKA scheme with $n \geq g + b$ users, we have:*

$$|U_i| \geq |K|^{\binom{g+b-1}{g-1}}, \quad (3.9)$$

where K is the set of conference keys and U_i is the set of user key given to user i .

Proof. The proof is given in [3], here we represent their proof in the following steps:

1. The idea is to show that every unrestricted (g, b) -CKA scheme can be transformed to a weakly secure (g, b) -CKP scheme without changing the domain of user keys. Hence the lower bound of Theorem 3 applies here too.
2. Communicated messages, \vec{M} , during an application of a CKA scheme are determined by the user key that TA distributes, $\vec{u} = \langle u_1, \dots, u_n \rangle$, and the local randomness of conference members, $\vec{r} = \langle r_1, \dots, r_n \rangle$. For a conference G we can write $\vec{M}_G = M(\vec{u}_G, \vec{r}_G)$, where M is the function that determines the communicated messages and \vec{r}_G is the restriction of \vec{r} to the members of conference G ; similarly for \vec{u}_G and \vec{M}_G . The communicated vector \vec{M} includes all \vec{M}_G for every possible conference G .
3. We first fix and publish \vec{M} , a vector of communicated messages for every possible conference, G . Then distribute the initial secret information, \vec{u} , such that it is *compatible* with \vec{M} . By a *compatible* user key vector, \vec{u}_G , to a communicated message vector, \vec{M}_G , we mean a vector of secret information for which there exists a vector of local random values, \vec{r}_G , such that $\vec{M}_G = M(\vec{u}_G, \vec{r}_G)$.
4. In the new scheme the users only need to use their user key and the publicly available communicated messages to reconstruct respective conference keys without interacting among themselves. Note that the TA will not need to distribute any larger amount of data than it did for the non-communicating scheme.
5. We first show how to choose and fix the public communicated messages, then determine a user key vector compatible with it and finally show that the new scheme satisfies the weak security property. The method is by showing how a real run of a CKA scheme can be obtained in a non-interactive fashion. We refer to the parameters of the CKA scheme with an index 1 and the same parameters in the CKP scheme with index 2. Parameters without indices are not specific to any of the schemes.

In the following we formalize each of the steps above.

1. Determining \vec{M} :

Let $\vec{u}_1 = \langle u_1, \dots, u_n \rangle$ be an arbitrary vector of user keys that TA_1 distributes in the CKA scheme and let $\vec{r} = \langle r_1, \dots, r_n \rangle$ be an arbitrary vector of local random values. Let $\vec{M} = M(\vec{u}_1, \vec{r})$ be the respective communicated messages. If \vec{M} is aired, clearly \vec{u}_1 is compatible with \vec{M} .

2. Choosing the matching \vec{u}_2 :

TA_2 randomly chooses $\vec{u}_2 = \langle u_2, \dots, u_n \rangle$ from the set of all user keys for which there exists a vector of local randomness \vec{r}' that makes \vec{u}_2 compatible to \vec{M}_G for every conference G . Note that \vec{u}_2 is also compatible to \vec{M} . On the other hand, since the scheme is an unrestricted one, such \vec{u}_2 exists.

3. Consistency of the new scheme:

Note that in the new CKP scheme, every conference member can reconstruct the key of the respective conferences by only using his user key and the publicly aired vector of communicated messages, without interacting with other users.

So far we have shown that any given CKA scheme can be transformed correctly to a CKP scheme without increasing the size of the user key. We are only left to show that the new scheme satisfies the weak security property.

4. Weak Security Property:

Let G be an arbitrary conference and A the corresponding disjoint adversary set. By the assumption, the original CKA scheme can be used to securely compute an unrestricted number of conference keys. This guarantees that for any $k_0 \in K$ with a given vector of communicated messages, \vec{M}_G , there exists a vector of user keys $\vec{v} \in U_{\mathcal{U}}$ such that $\vec{v}_A = \vec{u}_A$ and \vec{v}_G is compatible with \vec{M}_G such that $K_G = k_0$. According to the statement of weak security property, Equation (3.6), it follows that the transformed scheme is a weakly secure CKP scheme.

5. Concluding the proof:

We have shown that any unrestricted (g, b) -CKA scheme can be viewed as a weakly secure (g, b) -CKP scheme with the same domain of user keys distributed by the TA. Using Theorem 3, we conclude that the size of any user's user key should satisfy:

$$|U_i| \geq |K|^{\binom{g+b-1}{g-1}},$$

which completes the proof. □

According to Theorem 4, if a CKA scheme is designed to compute an unrestricted number of conference keys, the minimum user key size is the same as the minimum user key size for a CKP scheme. Another option to obtain a lower user key size is to relax the security requirement, i.e. CKA scheme that remains perfectly secure for only a limited number of conference keys. This results in the definition of τ -restricted CKA schemes which we have already seen in Definition 13. Note that for $\tau > \binom{g+b-1}{g-1}$, from Equation (3.7) one concludes that interaction cannot help to reduce the size of user keys in a τ -restricted CKA scheme.

The next theorem gives a lower bound on the size of user key in a τ -restricted (g, b) -CKA scheme.

Theorem 5. *In any τ -restricted (g, b) -CKA scheme with $\tau \leq \binom{g+b-1}{g-1}$, the size of a user key is lower bounded by:*

$$|U_i| \geq |K|^\tau. \tag{3.10}$$

Proof. The proof given in [3] is an adaptation of the proof of Theorem 4. We show how a τ -restricted (g, b) -CKA scheme can be transformed to a weakly secure (g, b) -CKP scheme. Here we fix the communication of τ predefined sets and require the TA to distribute compatible user keys to the communication of these conferences using a similar method as of Theorem 4. The rest of the theorem follows in a similar way. □

We observe that the $(2, b)$ -CKP scheme of Blom, see section 3.1.2, requires distributing user key of size $|K|^{b+1}$ which is consistent with the statement of Theorem 3, Equation (3.7). On the other hand, from Equation (3.10) one obtains the lower bound on the user key size in a 1-restricted $(2, b)$ -CKA scheme to be $|K|$. We also know that a 1-restricted $(2, b)$ -CKA scheme can be transformed to a $(2, b)$ -CKP scheme which causes the lower bound on $|U_i|$ to increase from $|K|$ to $|K|^{b+1}$. This suggests that the lower bound obtained from Theorem 5, Equation (3.10), is not the tightest. In the next section, we investigate the lower bound on user key for a 1-restricted $(2, b)$ -CKA scheme.

Lower bound on user key size of a 1-restricted $(2, b)$ -CKA scheme

To proceed with this task, we include some results from Maurer's work, as stated in [3]. Maurer studies the setting with two coin flipping users, each with private piece of information, who execute a protocol by communicating over a broadcast channel. After the execution of the protocol, the two users generate a common key such that an adversary who eavesdrop all the communication does not have any information about the key. Maurer shows that the conditional mutual information of the initial secrets that the two users have is at least equal to the entropy of the generated key. In other words, *the mutual information of any pair of random variables held by the two users cannot be increased after a conversation over a broadcast channel.*

More formally, let U_1, U_2 and U_3 be random variables representing the secret information held by $user_1, user_2$ and $user_3$, respectively. Let $user_1$ and $user_2$ communicate over a broadcast channel for a number of rounds. Let \mathbf{M} denote the random variable corresponding to the communicated messages. After execution of the protocol, $user_1$ and $user_2$ will agree on a key, $k \in K$, known to both of them such that $user_3$ has no information about it:

$$H(K|U_1\mathbf{M}) = H(K|U_2\mathbf{M}) = 0$$

and

$$H(\mathbf{K}|\mathbf{U}_3\mathbf{M}) = H(\mathbf{K}),$$

then

$$I(\mathbf{U}_1; \mathbf{U}_2|\mathbf{U}_3) \geq H(\mathbf{K}). \quad (3.11)$$

In the next lemma, we prove a tighter lower bound for the 1-restricted $(2, b)$ -CKA scheme than Equation (3.10).

Lemma 2. [3] *For a 1-restricted $(2, b)$ -CKA schemes with uniform distribution over the key space, K , the cardinality of the user key is at least $|K|^{b+1}$, that is:*

$$|U_i| \geq |K|^{b+1}. \quad (3.12)$$

This Lemma is proved in [3]. Here we rewrite the proof as follows.

Proof. Without loss of generality, we assume that $n = b + 2$, and prove the bound for the size of $user_1$'s user key. The proof method is by transforming the 1-restricted $(2, b)$ -CKA scheme to a 3-user communication scenario of Maurer. Let $user'_1$ receive the user key of $user_1$ in the original scheme and for every $2 \leq i \leq b + 2$, let $user'_2$ receive $user_i$'s user key and $user'_3$ gets $user_{i+1}, \dots, user_{b+2}$'s user keys¹. Following the steps of the original protocol, $user'_1$ and $user'_2$ qualify to compute a common key while $user'_3$ will not learn anything about their key. So by applying Equation (3.11), we can write:

$$I(\mathbf{U}'_1; \mathbf{U}'_2|\mathbf{U}'_3) = I(\mathbf{U}_1; \mathbf{U}_i|\mathbf{U}_{i+1} \dots \mathbf{U}_{b+2}) \geq H(\mathbf{K}). \quad (3.13)$$

On the other hand, from the definition of conditional mutual information, we have:

¹Although this is how the proof proceeds in [3], it seems to hold true for $U'_3 = \bigcup_{j \in \mathbb{Z} \setminus \{1, i\}} U_j$.

$$\begin{aligned}
& \sum_{i=2}^{b+2} I(\mathbf{U}_1; \mathbf{U}_i | \mathbf{U}_{i+1} \dots \mathbf{U}_{b+2}) \\
&= \sum_{i=2}^{b+2} H(\mathbf{U}_1 | \mathbf{U}_{i+1} \dots \mathbf{U}_{b+2}) - \sum_{i=2}^{b+2} H(\mathbf{U}_1 | \mathbf{U}_i \mathbf{U}_{i+1} \dots \mathbf{U}_{b+2}) \\
&= H(\mathbf{U}_1) - H(\mathbf{U}_1 | \mathbf{U}_2 \dots \mathbf{U}_{b+2}) \\
&\leq H(\mathbf{U}_1).
\end{aligned}$$

Combining these arguments, we get:

$$H(\mathbf{U}_1) \geq \sum_{i=2}^{b+2} I(\mathbf{U}'_1; \mathbf{U}'_2 | \mathbf{U}'_3) = I(\mathbf{U}_1; \mathbf{U}_i | \mathbf{U}_{i+1} \dots \mathbf{U}_{b+2}) \geq (b+1)H(\mathbf{K}).$$

On the other hand, we know that the distribution over the key space is a uniform one, which translates to $H(\mathbf{K}) = \log |K|$. Note that for any random variable \mathbf{X} , $H(\mathbf{X}) \leq \log |X|$. So we can rewrite the above inequality as:

$$\log |U_1| \geq H(\mathbf{U}_1) \geq (b+1)H(\mathbf{K}) = (b+1)\log |K|,$$

which completes the proof. \square

This lemma strengthens the idea of having more precise bound than that of Theorem 5 when τ , g or b take specific values. In the next theorem, a general lower bound on the size of user key is given for any τ -restricted (g, b) -CKD scheme.

Theorem 6. [3] *Let τ, g, b be positive integers such that $\tau \leq \binom{g+b-1}{g-1}$. Consider a τ -restricted (g, b) -CKA scheme with $n \geq b + g$ users and a uniform distribution over the domain of keys, K . We have:*

$$|U_i| \geq |K|^e,$$

with $e = \max\{\tau, (\frac{b-1}{g})\tau^{1-\frac{1}{g-1}}\}$, and for $\tau = 1$ the lower bound is:

$$|K|^{1+\lfloor \frac{b}{g-1} \rfloor}.$$

Proof. We rewrite the proof as appeared in [3]. From Theorem 5, we can conclude this theorem for cases where $\tau \geq (\frac{b-1}{g})\tau^{1-\frac{1}{g-1}}$ and hence $e = \tau$. So we only need to investigate the case for:

$$\tau \leq (\frac{b-1}{g})\tau^{1-\frac{1}{g-1}} \longrightarrow \tau^{\frac{1}{g-1}} \leq (\frac{b-1}{g}). \quad (3.14)$$

The proof idea is to transform the τ -restricted (g, b) -CKA scheme to a 1-restricted $(2, c)$ -CKA scheme, where c is a function of τ , g and b . Then we apply the result of Lemma 2 to conclude the proof. Without loss of generality, we assume that there are $n = g + b$ users in the τ -restricted (g, b) -CKA scheme and $n' = 2 + c$ users in the 1-restricted $(2, c)$ -CKA scheme. We will show the bound for the size of $user_1$'s user key. Since users communicate to produce one conference key in the 1-restricted $(2, c)$ -CKA scheme, this only conference key computation should result in all the τ conference keys that are produced in the τ -restricted (g, b) -CKA scheme. So we take the keys in the new scheme from a domain of size $|K|^\tau$. Let $user'_1$ in the new scheme behave just like he does in the original scheme, so it only receives U_1 . The strategy is to give each user enough shares from the old scheme so that every two users in the new scheme are capable of computing τ conference keys. Let us map a -many distinct users from the old scheme to $user'_i$, $2 \leq i \leq c + 2$, in the new scheme. Note that from combinatorics we know $\binom{X}{Y} \geq (\frac{X}{Y})^Y$. If we choose a such that:

$$\binom{a}{g-1} \geq (\frac{a}{g-1})^{g-1} \geq \tau \longrightarrow a \geq \lceil (g-1)\tau^{\frac{1}{g-1}} \rceil, \quad (3.15)$$

then we conclude that every $user'_i$, $i \neq 1$, in the new scheme has enough pieces of user key to compute τ conference keys when interacting with another user, $user'_j$, $1 \leq j \neq i \leq c + 2$.

On the other hand, since every user in the new scheme represents a users from the old scheme, except for $user'_1$, who acts just like $user_1$, we have:

$$a(n' - 1) + 1 = n \longrightarrow a(c + 1) + 1 = n = g + b \longrightarrow c = \lfloor \frac{(g + b - 1)}{a} \rfloor - 1.$$

Now that we have obtained the parameters, we can formalize the steps of the new protocol as follows. Let $U'_1 = U_1$ and for every $2 \leq i \leq c + 2$, $U'_i = (U_{a(i-1)+1}, \dots, U_{ai})$. Also note that

according to the choice of a and c we distribute at most as much user key as was distributed in the original scheme. On the other hand, according to Equation (3.15), for every conference G of size g in the original scheme, there exists $i \neq j$ such that $U_G \subseteq U'_i \cup U'_j$. So the key that $user'_i$ and $user'_j$ compute in the new scheme would be a combination of at least τ keys in the old scheme, containing k_G .

So far, we have shown that the transformation is sound. To complete the proof we also need to show that the distribution over the key space in the new scheme remains uniform. We show this by contradiction.

Let C be a coalition of users disjoint from $user'_i$ and $user'_j$, $1 \leq i \neq j \leq c+2$, such that using U'_C and the communicated messages, enables members of C to learn about the key of $user'_i$ and $user'_j$, k'_{ij} , in the new scheme. Note that according to the construction, k'_{ij} includes the key of τ conferences, say G_1, \dots, G_τ , of the old scheme. We can rewrite this with respect to the original scheme as, $\exists 1 \leq \ell \leq \tau$ and an adversary set A_ℓ disjoint from G_ℓ such that the adversaries can learn about the key of G_ℓ using their user key, communicated messages and the key of $G_1, \dots, G_{\ell-1}$. Since this argument can hold true for a coalition of size $|A_\ell| = n - |G_\ell| = b$, it contradicts the security of the original scheme for computing τ conference keys. So the distribution over the key space in the new scheme remains uniform.

By applying Lemma 2 for the new scheme, we have:

$$|U_1| = |U'_1| \geq |K'|^{c+1} = |K|^{\tau(c+1)} = |K|^{\tau(\lfloor \frac{b+g-1}{a} \rfloor)} > |K|^{(\frac{b-1}{g})\tau(1-\frac{1}{g-1})},$$

and for $\tau = 1$ we have $a = g - 1$, thus:

$$|K|^{\tau(\lfloor \frac{b+g-1}{a} \rfloor)} = |K|^{\lfloor \frac{b+g-1}{g-1} \rfloor} = |K|^{1+\lfloor \frac{b}{g-1} \rfloor},$$

which completes the proof. □

3.3 Samples of CKA Schemes

In this section we present a number of CKA schemes and compare their key rates. In the last section, we noted that if the number of conferences is limited, the interactive schemes can potentially achieve lower key rate than the non-interactive schemes. In this section we review a number of CKA schemes and compute their key rates and communication rates.

3.3.1 Blundo et al.'s 1-restricted $(\leq n, b)$ -CKA Scheme

In [6], Blundo et al. present a CKA scheme that can be used to compute conference keys for conferences of sizes $2, \dots, n$ and an adversary set of size b . In their scheme, as well as the scheme of Section 3.1.3, once the size of the adversary set, b , is fixed the scheme enables computing conference keys for conferences of size g for $2 \leq g \leq n - b$.

Initialization:

Let $\mathcal{U} = \{u_1, \dots, u_n\}$ be a set of n users with i the public identity of user u_i . The TA randomly chooses a symmetric polynomial in two variables, $f(x_1, x_2)$, and of degree b in each variable. For a user $u_i \in \mathcal{U}$, the TA evaluates $f(x_1, x_2)$ by substituting the first variable with the public identity of user u_i and sends the resulting 1-variable polynomial, $p_i(x_2) = f(i, x_2)$, to user u_i . This provides two users, $u_i, u_j \in \mathcal{U}$, with a common key:

$$k_{\{i,j\}} = f(i, j) = p_i(j) = p_j(i).$$

Conference key computation:

When a group of g users, G , decide to form a conference, the user with minimum identity in G , say u_ℓ , randomly chooses a secret, $s \in GF(q)$, as the conference key and encrypts it using the pairwise key he shares with every other users in that conference. Finally, u_ℓ sends:

$$m_{\ell,j} = s \oplus p_\ell(j),$$

for all j such that $u_j \in G \setminus \{u_\ell\}$. Each user $u_j \in G$ can uniquely compute the conference key using his user key, $p_j(x_2)$, and the message he receives from user u_ℓ as follows:

$$s = m_{\ell,j} \oplus p_j(\ell).$$

Example 4. For $G = \{u_1, \dots, u_g\}$, u_1 will randomly choose $s \in GF(q)$ and to each user $u_j \in G$, $j \neq 1$ sends $m_{1j} = s \oplus p_1(j)$. Every user $u_j \in G$ with $j \neq 1$, can compute the conference key by performing the following computation:

$$m_{1j} \oplus p_j(1) = s.$$

Without loss of generality, we assume $n = g + b$. From section 3.1.5 we know that the size of user key for each user is $\binom{g+b-1}{1} \log q = (n-1) \log q$ and since the conference key is a random element of $GF(q)$, the key rate is:

$$\frac{|U_{\mathcal{U}}|}{|K_G|}^{(1Blundo)} = n(n-1). \quad (3.16)$$

On the other hand, there will be $(g-1)$ messages that user u_ℓ , a typical conference member with minimum id, sends to the other conference members, hence:

$$\frac{|M_G|}{|K_G|}^{(1Blundo)} = (g-1). \quad (3.17)$$

We use the index $(1Blundo)$ to refer to the $(\leq n, b)$ -CKA scheme of Blundo et al. The 1 appearing in the index implies that this scheme can only be used to compute one conference key, securely. We discuss this point below.

After one conference key is computed by this scheme, the user keys used to encrypt the conference key, s , are no longer secure. For instance once user u_2 receives m_{21} from u_1 in Example 4, he can compute s and by XOR-ing s with the other communicated messages, m_{1j} for $2 < j \leq g$, he can obtain the shared key between user u_1 and u_j . Hence this scheme is only perfectly secure to compute one conference key.

3.3.2 Blundo et al.'s 1-Restricted (g, b) -CKA Scheme

Let $\mathcal{U} = \{1, \dots, n\}$ be a set of n users, $G \subseteq \mathcal{U}$ be a conference of size g , and $b \leq n - g$ denote the size of the adversary set. Suppose that $2 \leq \ell \leq g$ is an integer such that $g \equiv 1 \pmod{\ell - 1}$ and that $k \in \mathbb{N}^+$.

Initialization:

The TA distributes user keys corresponding to an $(\ell, b + g - \ell)$ -CKP scheme as described in Section 3.1.4, implemented over $(GF(p^k))^\ell$, with p prime and $k \in \mathbb{N}$ such that $p^k > n$. We denote the key of an ℓ -subset of users, $L \subseteq \mathcal{U}$, $|L| = \ell$, with k_L and we think of it as being made up of ℓ independent keys over $GF(p^k)$ which we denote by $k_{L,1}, \dots, k_{L,\ell}$.

Conference key computation:

Each user $h \in G$, $|G| = g$, performs the following steps:

1. Chooses a random value $m^{(h)} = (m_1^h, \dots, m_r^h) \in (GF(p^k))^r$, where $r = \binom{g-2}{n-2}$. (According to Definition 22, here r is the number of parallel classes that can be formed over a complete $(\ell - 1)$ -uniform hypergraph on $G \setminus \{h\}$.)
2. Partitions the complete $(\ell - 1)$ -uniform hypergraph on $G \setminus \{h\}$ into r parallel classes C_1, \dots, C_r , where each consists of $\chi = \frac{g-1}{\ell-1}$ blocks. We denote each block with $B_{i,j}^h$, for $1 \leq i \leq r$ and $1 \leq j \leq \chi$.
3. For each block $B_{i,j}^h$, denote with $B(i, j, h)$ the set $B_{i,j}^h \cup \{h\} = \{x_1, \dots, x_\ell\}$ and let $\alpha_{i,j}^h$ denote the index such that $x_{\alpha_{i,j}^h} = h$. (Note that we are implicitly assuming the users are ordered increasingly with respect to their public identities inside each conference, so $\alpha_{i,j}^h$ is uniquely determined.)
4. Encrypts each m_i^h using the χ keys $k_{B(i,j,h), \alpha_{i,j}^h}$:

$$b_{i,j}^h = k_{B(i,j,h), \alpha_{i,j}^h} + m_i^h \pmod{p^k},$$

for $1 \leq i \leq r$ and $1 \leq j \leq \chi$.

5. Broadcasts the vector:

$$b^{(h)} = (b_{1,1}^h, \dots, b_{1,\chi}^h, \dots, b_{r,1}^h, \dots, b_{r,\chi}^h).$$

The conference key is:

$$k_G = m^{(1)} || \dots || m^{(g)},$$

which can be computed by anyone in G using the broadcast messages $b_G = (b^{(1)}, \dots, b^{(g)})$ and their user keys.

Here we give an example to clarify how this scheme works.

Example 5. Let $\mathcal{U} = \{1, 2, 3, 4, 5, 6, 7\}$, $G = \{1, 2, 3, 4, 5\}$ with $b = 2$ and $GF(q) = \mathbb{Z}_{11}$. Since $5 \equiv 1 \pmod{2}$, we take $\ell = 3$.

Initialization:

The TA distributes user keys according to an $(3, \leq 4)$ -CKP scheme of Section 3.1.4 over $(\mathbb{Z}_{11})^3$, so the coefficients are in forms of 3-tuples. Let $f(x, y, z) = (7, 2, 1)xyz$, be TA's random symmetric polynomial.

Conference key computation:

Without loss of generality, we only follow user u_4 's actions. His user key is $p_4(y, z) = f(4, y, z) = (7, 8, 4)yz$.

1. User u_4 chooses a random vector $m^{(4)} = (m_1^4, m_2^4, m_3^4) \in (\mathbb{Z}_{11})^3$.
2. He divides the design $(G \setminus \{4\}, \mathcal{B})$ into $r = \binom{3}{1} = 3$ parallel classes, where \mathcal{B} consists of all 2-subset of $G \setminus \{4\}$.
3. Computes the corresponding α values as follows:

$$C_1^4 = \{\{1, 2\}, \{3, 5\}\}, \quad C_2^4 = \{\{1, 3\}, \{2, 5\}\}, \quad C_3^4 = \{\{1, 5\}, \{2, 3\}\}.$$

$$\alpha_{1,1}^4 = 3, \alpha_{1,2}^4 = 2, \quad \alpha_{2,1}^4 = 3, \alpha_{2,2}^4 = 2, \quad \alpha_{3,1}^4 = 2, \alpha_{3,2}^4 = 3.$$

4. Encrypts each m_i^4 with the appropriate keys:

$$\begin{aligned} b_{1,1}^4 &= m_1^4 + k_{\{1,2,4\},3}, & b_{1,2}^4 &= m_1^4 + k_{\{3,4,5\},2}, \\ b_{2,1}^4 &= m_2^4 + k_{\{1,3,4\},3}, & b_{2,2}^4 &= m_2^4 + k_{\{2,4,5\},2}, \\ b_{3,1}^4 &= m_3^4 + k_{\{1,4,5\},2}, & b_{3,2}^4 &= m_3^4 + k_{\{2,3,4\},3}. \end{aligned}$$

The arithmetic is done in \mathbb{Z}_{11} .

5. Broadcasts $b^{(4)} = (b_{1,1}^4, b_{1,2}^4, b_{2,1}^4, b_{2,2}^4, b_{3,1}^4, b_{3,2}^4)$.

Note that $k_{\{1,2,4\}} = (3, 5, 8)$ and hence $k_{\{1,2,4\},3} = 8$. The same process has to be completed by other users in G . Note that once user u_1 is doing his part, at some point he needs to encrypt his secret with $k_{\{1,2,4\}}$ and since $\alpha_{1,1}^1 = 1$, he will pick the value 3 as the encryption key. This should clarify why all the encryptions in this scheme are samples of one time pad.

The security of this scheme directly results from the security of the (g, b) -CKP scheme of Section 3.1.4.

Without loss of generality, assume $g + b = n$. From Section 3.1.5 we know that each user has to get $\binom{g+b-1}{\ell-1}$ pieces of random values from $(GF(q))^\ell$. On the other hand, the final conference key is obtained by concatenating the g random values that conference members share, each consists of $r = \binom{g-2}{\ell-2}$ pieces from $GF(q)$. Hence the key rate is:

$$\frac{|U_{\mathcal{U}}|^{(1RD)}}{|K_G|} = n \frac{\ell \binom{n-1}{\ell-1}}{g \binom{g-2}{\ell-2}}. \quad (3.18)$$

To compute the communication rate, note that each user broadcasts a vector that contains $r \times \chi$ entities, each from $GF(q)$. Hence the communication rate is:

$$\frac{|M_G|^{(1RD)}}{|K_G|} = \frac{gr\chi}{gr} = \chi, \quad (3.19)$$

with $\chi = \frac{g-1}{\ell-1}$.

We used the index (1RD) to refer to the 1-restricted (g, b) -CKA scheme of Blundo et al. The 1 in the index refers to the fact that this scheme can be used to compute one conference key, securely. The RD in the index refers to the application of *Resolvable Designs* in their scheme.

3.3.3 Blundo et al.'s τ -Restricted (g, b) -CKA Scheme

In this section we introduce the τ -restricted scheme of Blundo et al., [5], and analyze its efficiency.

The τ -restricted (g, b) -CKA scheme of Blundo et al. is based on their 1-restricted (g, b) -CKA scheme, described in the previous section. The idea is to use τ copies of the 1-restricted (g, b) -CKA scheme while recycling the user keys that have not been used in the computation of the i -th conference key, to compute the $(i + 1)$ -th conference key, where $1 \leq i \leq \tau - 1$. The protocol can be used to compute conference keys for τ conferences of all the same size, g .

Initialization:

The TA distributes $\tau - 1$ sets of user keys according to the initialization phase of a 1-restricted $(g, b + 1)$ -CKA scheme of Blundo et al., see Section 3.3.2. TA also distributes one set of user keys according to the 1-restricted (g, b) -CKA scheme of Blundo et al. Let Δ_i refer to the i -th 1-restricted scheme of Blundo et al. We randomly choose Δ_i from $(GF(p^{k_i}))^\ell$, where p is prime and $p^{k_i} > n$ with $k_i \leq k_1$, $\forall 1 \leq i \leq \tau$.

Conference key computation:

1. When users in conference G_1 , want to compute a common key, they do so by using the conference key computation of Δ_1 .
2. For all the other conferences, G_i with $2 \leq i \leq \tau$, all conference members follow the conference key computation phase of Δ_i .
3. Since G_{i-1} and G_i are distinct conferences, $G_i \setminus G_{i-1} \neq \emptyset$. Let $u_h \in G_i \setminus G_{i-1}$ be the user with the smallest identity. Note that u_h has not used his user key from Δ_{i-1} . User u_h randomly

chooses a value $r^{(h)} \in GF(p^{\ell k_{i-1}})$ and encrypts it by following the conference key computation phase of Δ_{i-1} , when all the ℓ entities of his shared key is used for this encryption. The final conference key for $G_i = \{u_{i1}, \dots, u_{ig}\}$ is:

$$k_{G_i} = m^{(i1)} || \dots m^{(ig)} || r^{(h)}.$$

This scheme requires that all users hold a counter which is incremented each time a conference key is generated.

- Remark 6.** 1. *The conference key piece that user $u_h \in G_i \setminus G_{i-1}$ appends to k_{G_i} is randomly chosen from $GF(p^{\ell k_{i-1}})$ and will be encrypted using all the ℓ entries of u_h 's user key from Δ_{i-1} .*
2. *In some cases it is desired to have all the conference keys of a same size. For G_1 we have $|k_{G_1}| = grk_1 \log p$ and for all $2 \leq i \leq \tau$, the size of the final conference key is $|k_{G_i}| = (grk_i + r\ell k_{i-1}) \log p$. So we need to ensure that for all i , $2 \leq i \leq \tau$:*

$$gk_1 = gk_i + \ell k_{i-1}. \quad (3.20)$$

Since all elements in both sides of the equation above are positive values, we conclude that in this scheme the size of the i -th field, $GF(p^{k_i})$ used to initialize the Δ_i , is smaller than the size of the field $GF(p^{k_1})$, used to initialize Δ_1 . In Section 3.4.2 we study the field size of the conferences in more detail.

3.4 Analysis of the τ -Restricted (g, b) -CKA scheme of Blundo et al.

We analyze the τ -restricted (g, b) -CKA scheme of Blundo et al. from three perspectives: (i) we show that a realization of this scheme does exist regarding the field sizes to preserve the fixed size of the conference keys, (ii) growth of the computation field sizes and (iii) comparing the efficiency

of the τ -restricted (g, b) -CKA scheme with τ independent copies of the 1-restricted $(\leq n, b)$ -CKA scheme (see Section 3.3.1) and τ copies of the 1-restricted (g, b) -CKA scheme (see Section 3.3.2).

3.4.1 Soundness of the construction

In this section we review the analysis presented in [5] to investigate the possibility of having a collection of k_i 's that satisfy Equation (3.20).

In the following we show that for any given integers n, g, b, ℓ and τ such that $g + b \leq n$ and $2 \leq \ell \leq g$ with $g \equiv 1 \pmod{\ell - 1}$, there always exist positive integers k_1, \dots, k_τ satisfying Equation (3.20).

Lemma 3. [5] *Let g and ℓ be two positive integers such that $2 \leq \ell \leq g$. Let $I_1 = 1$ and $I_t = g^{t-1} - \ell I_{t-1}$, for $2 \leq t \leq \tau$. If $gk_1 = gk_t + \ell k_{t-1}$, for $2 \leq t \leq \tau$, then it holds that $k_t = k_1 \cdot \frac{I_t}{g^{t-1}}$ for $1 \leq t \leq \tau$.*

Proof. The proof is by induction on t . For $t = 1$, we have $k_1 = k_1 \frac{I_1}{1}$. Now suppose that the lemma is true for some $t < \tau$, then we prove it is also true for $t + 1$, i.e. $k_{t+1} = k_1 \frac{I_{t+1}}{g^t}$. According to the assumption, we know:

$$gk_1 = gk_{t+1} + \ell k_t \rightarrow gk_{t+1} = gk_1 - \ell k_t, \quad (3.21)$$

and also according to the inductive step, for t we have:

$$k_t = k_1 \frac{I_t}{g^{t-1}}. \quad (3.22)$$

Substituting k_t from Equation (3.22) in Equation (3.21), we get:

$$\begin{aligned} gk_{t+1} &= gk_1 - \ell k_1 \frac{I_t}{g^{t-1}} = k_1 \left(g - \frac{\ell I_t}{g^{t-1}} \right) = k_1 \left(\frac{g^t - \ell I_t}{g^{t-1}} \right) = k_1 \left(\frac{I_{t+1}}{g^{t-1}} \right), \\ k_{t+1} &= k_1 \frac{I_{t+1}}{g^t}. \end{aligned}$$

□

3.4.2 Growth of the Field Sizes, k_t 's

In this section, we study the growth of k_t for $1 \leq t \leq \tau$. We start with comparing k_1, k_2 and k_3 . We have the following equations that relate these three variables:

$$\begin{cases} k_i \leq k_1, \ 1 < i \leq \tau, \\ gk_1 = gk_2 + \ell k_1 \quad \rightarrow \quad g(k_2 - k_3) = \ell(k_3 - k_1) \quad \rightarrow \quad k_2 \leq k_3 \leq k_1 \\ gk_1 = gk_3 + \ell k_2 \end{cases}$$

In general for every $1 \leq i, j \leq \tau - 1$ we have:

$$gk_i + \ell k_{i-1} = gk_j + \ell k_{j-1},$$

$$g(k_{i+1} - k_{j+1}) = \ell(k_j - k_i),$$

which leads to the following results:

1. Let $i = 1$, then $g(k_2 - k_{j+1}) = \ell \underbrace{(k_j - k_1)}_{\leq 0} \rightarrow k_2 \leq k_j, \ \forall 1 \leq j \leq \tau$.
2. Let $i = 2$, then $g(k_3 - k_{j+1}) = \ell \underbrace{(k_j - k_2)}_{\geq 0} \rightarrow k_3 \geq k_j \ \forall 2 \leq j \leq \tau$.
3. Let $i = 3$, then $g(k_4 - k_{j+1}) = \ell \underbrace{(k_j - k_3)}_{\leq 0} \rightarrow k_4 \leq k_j \ \forall 3 \leq j \leq \tau$.
4. Let $i = 4$, then $g(k_5 - k_{j+1}) = \ell \underbrace{(k_j - k_4)}_{\geq 0} \rightarrow k_5 \geq k_j \ \forall 4 \leq j \leq \tau$.
- \vdots
5. $\begin{cases} \text{For } \tau - 1 \text{ odd, } & g(k_\tau - k_{j+1}) = \ell \underbrace{(k_j - k_{\tau-1})}_{\leq 0} \rightarrow k_\tau \leq k_{\tau-1}, \\ \text{For } \tau - 1 \text{ even, } & g(k_\tau - k_{j+1}) = \ell \underbrace{(k_j - k_{\tau-1})}_{\geq 0} \rightarrow k_\tau \geq k_{\tau-1}. \end{cases}$

In Figure 3.2 we show pictorially why the field sizes fluctuate between k_1 and k_2 . Figure 3.3, pictures how the field sizes line up, smallest to the largest.

Finally, to measure the efficiency of this scheme we compute:

$$|U_i| = \ell(k_1 + \dots + k_{\tau-1}) \binom{g+b}{\ell-1} \log p + \ell k_{\tau} \binom{g+b-1}{\ell-1} \log p$$

and

$$|K_G| = grk_1 \log p = gk_1 \binom{g-2}{\ell-2} \log p,$$

$$\frac{|U_{\mathcal{U}}|^{(\tau RD)}}{|K_G|} = n \left(\frac{\ell(k_1 + \dots + k_{\tau-1}) \binom{g+b}{\ell-1}}{gk_1 \binom{g-2}{\ell-2}} + \frac{\ell k_{\tau} \binom{g+b-1}{\ell-1}}{gk_1 \binom{g-2}{\ell-2}} \right). \quad (3.23)$$

With respect to the communication rate, we compute:

$$\begin{aligned} \frac{|M_{G_1}|^{(\tau RD)}}{|K_{G_1}|} &= \frac{gr\chi}{grk_1} = \frac{\chi}{k_1}, \\ \frac{|M_{G_i}|^{(\tau RD)}}{|K_{G_i}|} &= \frac{gr\chi + \ell r\chi}{grk_1} = \frac{g+\ell}{gk_1} \chi, \quad 2 \leq i \leq \tau. \end{aligned} \quad (3.24)$$

3.4.3 Comparison of the τ -Restricted (g, b) -CKA scheme to previous CKA schemes

In this section we show that the τ -restricted (g, b) -CKA is more key rate efficient than using τ copies of a 1-restricted CKA schemes, i.e. the 1-restricted (g, b) -CKA scheme and the $(\leq n, b)$ -CKA scheme. We begin with comparing the efficiency of the τ -restricted (g, b) -CKA scheme with the efficiency of the scheme obtained by τ copies of the 1-restricted (g, b) -CKA scheme.

The following lemma from [5] states that under certain conditions, the τ -restricted (g, b) -CKA scheme is more efficient than using τ copies of the 1-restricted (g, b) -CKA scheme, with respect to their key rates.

Lemma 4. *Let τ be an integer greater than 1 and let g and ℓ be two positive integers such that $2 \leq \ell \leq g$. If $gk_1 = gk_t + \ell k_{t-1}$ for $2 \leq t \leq \tau$, then there exist integers k_1, \dots, k_{τ} such that:*

$$\begin{aligned} \frac{|U_{\mathcal{U}}|^{(\tau RD)}}{|K_G|} &\leq \frac{|U_{\mathcal{U}}|^{(1RD)}}{|K_G|} \\ \frac{(k_1 + \dots + k_{\tau-1}) \binom{g+b}{\ell-1}}{k_1 \binom{g-2}{\ell-2}} + \frac{k_{\tau} \binom{g+b-1}{\ell-1}}{k_1 \binom{g-2}{\ell-2}} &\leq \tau \frac{\binom{g+b-1}{\ell-1}}{\binom{g-2}{\ell-2}} \end{aligned} \quad (3.25)$$

iff $\ell^2 - (b+1)\ell - g \leq 0$.

Proof. The proof is by induction on τ . If $\tau = 2$, then by choosing $k_1 = g$ and $k_2 = g - \ell$, Equation (3.25) holds:

$$\frac{\binom{g+b}{\ell-1}}{\binom{g-2}{\ell-2}} + \frac{g-\ell}{g} \frac{\binom{g+b-1}{\ell-1}}{\binom{g-2}{\ell-2}} \leq 2 \frac{\binom{g+b-1}{\ell-1}}{\binom{g-2}{\ell-2}} \rightarrow g \binom{g+b}{\ell-1} \leq (g+\ell) \binom{g+b-1}{\ell-1}$$

\Leftrightarrow

$$\frac{g(g+b)!}{(\ell-1)!(g+b-\ell+1)!} \leq \frac{(g+\ell)(g+b-1)!}{(\ell-1)!(g+b-\ell)!} \rightarrow g(g+b) \leq (g+\ell)(g+b-\ell+1)$$

\Leftrightarrow

$$\ell^2 - (b+1)\ell - g \leq 0.$$

Now suppose that inequality (3.25) is true for some $\tau \geq 2$, we show it holds for $\tau + 1$. To conclude this, we first replace k_t with $k_1 \frac{I_t}{g^{t-1}}$, according to Equation (3.25):

$$\frac{(k_1 + \dots + k_{\tau-1})}{k_1} \frac{\binom{g+b}{\ell-1}}{\binom{g-2}{\ell-2}} + \frac{k_\tau}{k_1} \frac{\binom{g+b-1}{\ell-1}}{\binom{g-2}{\ell-2}} = \left(1 + \frac{I_2}{g} + \dots + \frac{I_{\tau-1}}{g^{\tau-2}}\right) \frac{\binom{g+b}{\ell-1}}{\binom{g-2}{\ell-2}} + \frac{I_\tau}{g^{\tau-1}} \frac{\binom{g+b-1}{\ell-1}}{\binom{g-2}{\ell-2}}.$$

On the other hand, for every pair of positive integers x and y with $x \leq y - 1$, we have that:

$$\binom{y}{x} = \binom{y-1}{x-1} + \binom{y-1}{x},$$

so:

$$\binom{g+b-1}{\ell-1} = \binom{g+b}{\ell-1} - \binom{g+b-1}{\ell-2}.$$

Substituting this into the last equality, results in:

$$\left(1 + \frac{I_2}{g} + \dots + \frac{I_{\tau-1}}{g^{\tau-2}} + \frac{I_\tau}{g^{\tau-1}}\right) \frac{\binom{g+b}{\ell-1}}{\binom{g-2}{\ell-2}} - \frac{I_\tau}{g^{\tau-1}} \frac{\binom{g+b-1}{\ell-2}}{\binom{g-2}{\ell-2}} \leq \tau \frac{\binom{g+b-1}{\ell-1}}{\binom{g-2}{\ell-2}}.$$

Note that the last inequality is obtained by the inductive step and hence is only true when $\ell^2 - (b+1)\ell - g \leq 0$. By rewriting the inequality above and adding the term $\frac{I_{\tau+1}}{g^\tau} \frac{\binom{g+b-1}{\ell-1}}{\binom{g-2}{\ell-2}}$ to both sides we get:

$$\left(1 + \frac{I_2}{g} + \dots + \frac{I_{\tau-1}}{g^{\tau-2}} + \frac{I_\tau}{g^{\tau-1}}\right) \frac{\binom{g+b}{\ell-1}}{\binom{g-2}{\ell-2}} + \frac{I_{\tau+1}}{g^\tau} \frac{\binom{g+b-1}{\ell-1}}{\binom{g-2}{\ell-2}} \leq \tau \frac{\binom{g+b-1}{\ell-1}}{\binom{g-2}{\ell-2}} + \frac{I_\tau}{g^{\tau-1}} \frac{\binom{g+b-1}{\ell-2}}{\binom{g-2}{\ell-2}} + \frac{I_{\tau+1}}{g^\tau} \frac{\binom{g+b-1}{\ell-1}}{\binom{g-2}{\ell-2}}.$$

The left hand side of this inequality is now in the desired form for proving the induction hypothesis.

To give the right hand side in the desired format, using Lemma 3 we replace $I_{\tau+1}$ with $g^\tau - \ell I_\tau$, which results in:

$$\left(1 + \frac{I_2}{g} + \dots + \frac{I_{\tau-1}}{g^{\tau-2}} + \frac{I_\tau}{g^{\tau-1}}\right) \frac{\binom{g+b}{\ell-1}}{\binom{g-2}{\ell-2}} + \frac{I_{\tau+1}}{g^\tau} \frac{\binom{g+b-1}{\ell-1}}{\binom{g-2}{\ell-2}} \leq (\tau+1) \frac{\binom{g+b-1}{\ell-1}}{\binom{g-2}{\ell-2}} + \frac{I_\tau}{g^{\tau-1}} \frac{\binom{g+b-1}{\ell-2}}{\binom{g-2}{\ell-2}} - \frac{\ell I_\tau}{g^\tau} \frac{\binom{g+b-1}{\ell-1}}{\binom{g-2}{\ell-2}}.$$

To conclude the lemma, we need to show that

$$\begin{aligned} & \frac{I_\tau}{g^{\tau-1}} \frac{\binom{g+b-1}{\ell-2}}{\binom{g-2}{\ell-2}} - \frac{\ell I_\tau}{g^\tau} \frac{\binom{g+b-1}{\ell-1}}{\binom{g-2}{\ell-2}} \leq 0 \\ \Leftrightarrow & \quad g \binom{g+b-1}{\ell-2} - \ell \binom{g+b-1}{\ell-1} \leq 0 \\ \Leftrightarrow & \quad \frac{g}{g+b-\ell+1} - \frac{\ell}{\ell-1} \leq 0 \end{aligned}$$

which is indeed true as long as $\ell^2 - (b+1)\ell - g \leq 0$. \square

So for $\ell^2 - (b+1)\ell - g \leq 0$, the τ -restricted (g, b) -CKA scheme of Blundo et al. achieves better key rate than the using τ copies of the 1-restricted (g, b) -CKA scheme of Blundo et al. Now, if we show that the 1-restricted (g, b) -CKA scheme of Blundo et al. achieves better efficiency than the 1-restricted $(\leq n, b)$ -CKA scheme of Blundo et al., we can conclude that for $\ell^2 - (b+1)\ell - g \leq 0$, the τ -restricted (g, b) -CKA scheme is also more key efficient than the 1-restricted $(\leq n, b)$ -CKA scheme.

Let $2 \leq \ell_0 \leq g$, such that $g \equiv 1 \pmod{\ell_0 - 1}$ be the best value for ℓ that minimizes the key rate for the 1-restricted (g, b) -CKA scheme. Without loss of generality, let $n = g + b$. We conclude:

$$\begin{aligned} \frac{|U_{\mathcal{U}}|}{|K_G|} \Big|_{\ell=\ell_0}^{(1RD)} & \leq \frac{|U_{\mathcal{U}}|}{|K_G|} \Big|_{\ell=2}^{(1RD)}, \\ \frac{\ell_0 \binom{n-1}{\ell_0-1}}{g \binom{g-1}{\ell_0-2}} & \leq \frac{2(n-1)}{g}. \end{aligned} \tag{3.26}$$

On the other hand, from Equation (3.16) we know that the key rate for the 1-restricted $(\leq n, b)$ -CKA scheme is:

$$\frac{|U_{\mathcal{U}}|}{|K_G|}^{(1Blundo)} = n - 1.$$

Note that since $g \geq 2$, from Equation (3.16) and Equation (3.26) we conclude that:

$$\begin{aligned} \frac{2(n-1)}{g} &\leq n-1, \\ \left| \frac{U_{\mathcal{U}}}{K_G} \right|_{\ell=\ell_0}^{(1RD)} &\leq \left| \frac{U_{\mathcal{U}}}{K_G} \right|^{(1Blundo)}. \end{aligned}$$

From Lemma 4 we conclude that for $\tau \geq 1$ and $2 \leq \ell \leq g$ such that $g \equiv 1 \pmod{\ell-1}$ and $\ell^2 - (b+1)\ell - g \leq 0$, the τ -restricted (g, b) -CKA scheme has lower key rate than applying τ copies of the 1-restricted $(\leq n, b)$ -CKA scheme as well.

3.5 Conclusions

In this chapter we studied a number of major works in which different methods were examined to realize a CKD scheme. We saw that the (g, b) -CKP scheme of Blundo et al. provides perfect security for forming an unlimited number of conference keys. We showed a lower bound on the size of user keys in an unconditionally secure (g, b) -CKP scheme and showed that Blundo et al.'s schemes achieves this bound with equality. We also studied the novel work of Beimel and Chor, [3], that rules out the possibility of having a CKA scheme that is perfectly secure to compute an unlimited number of conference keys with lower key rate than the (g, b) -CKP scheme of Blundo et al.

We concluded that in order to obtain a lower key rate, we must restrict the number of conferences that can be formed. We reviewed the τ -restricted (g, b) -CKA scheme of Blundo et al. as an example of a CKA scheme that distributes less amount of user key among the users than the efficient (g, b) -CKP scheme of Blundo et al.

Since users do not necessarily form conferences of the same size, it is interesting to study the possibility of generalizing the idea of the τ -restricted (g, b) -CKA scheme of Blundo et al. to a τ -restricted $(g_1, \dots, g_\tau, b_1, \dots, b_\tau)$ -CKA scheme, where g_i is the size of the i -th conference and b_i is the size of the respective adversary set, for all $1 \leq i \leq \tau$. We pursue this idea in Chapter 4.

Table 3.1: Summary of the studied CKD schemes

Scheme	Conference Size	Key Rate	Communication Rate
Trivial (g, b) -CKP	variable	$n \binom{n-1}{g-1}$	0
Blom's $(2, b)$ -CKP	2	$n(n-1)$	0
Fiat-Naor's $(\leq n, b)$ -CKP	variable	$n \sum_{j=0}^b \binom{n}{j}$	0
Blundo et al.'s (g, b) -CKP	g	$n \binom{n-1}{g-1}$	0
Blundo et al.'s 1-Restricted $(\leq n, b)$ -CKA	<i>variable</i>	$n(n-1)$	$(g-1)$
Blundo et al.'s 1-Restricted (g, b) -CKA	g	$n \frac{\ell \binom{n-1}{\ell-1}}{r}$	χ
Blundo et al.'s τ -Restricted (g, b) -CKA	τ conf. of size g	$n \frac{\ell \sum_{i=1}^{\tau-1} k_i \binom{n}{\ell-1} + \ell k_{\tau} \binom{n-1}{\ell-1}}{g r k_1}$	$\frac{g+\ell}{g k_1} \chi$

We also remark that the communication model in all the schemes we have studied in this chapter is a broadcast model. In real life networks, broadcast channels must be implemented using point-to-point or multicast channels. Network constraints defies assuming that every two users can directly transmit messages and users need to rout their messages through other nodes in order to communicate. In Chapter 5, we consider a special type of communication graph and introduce a new 1-restricted (g, b) -CKA scheme assuming this communication model. We show that our new scheme always attains better communication rate and for certain parameter values, better key rate when compared to the 1-restricted (g, b) -CKA scheme of Blundo et al.

Table 3.1 present a summary of the schemes we have studied in this chapter.

Chapter 4

A τ -restricted CKA scheme for conferences of varying sizes

In the previous chapter we studied the τ restricted (g, b) -CKA scheme of Blundo et al. and saw that it attains better key efficiency for computing τ conference keys compared to other CKD schemes that we studied so far. This scheme enables τ conferences, all of size g , to securely compute conference keys. It uses resolvable designs and the communication model is broadcast.

In real life applications, users in a network form conferences of different sizes to fulfill different purposes and so it is desirable to have a CKA scheme that securely and efficiently computes conference keys for conferences of varying sizes, say $2 \leq g_1, \dots, g_\tau \leq n$, with n being the total number of network users.

In this chapter we introduce three protocols to securely compute a τ -restricted CKA scheme for a τ -tuple of conference sizes $\mathcal{G} = (g_1, \dots, g_\tau)$ with the respective adversary set sizes $\mathcal{B} = (b_1, \dots, b_\tau)$ such that for all $1 \leq i \leq \tau$, $g_i + b_i \leq n$. We follow each construction with its performance measures and finally present a practical way to compare their performances.

4.1 Introduction

In this section, we formalize the idea of having τ -restricted CKA schemes for a τ -tuple of conference sizes. To do this, we extend Definition 13 as follows:

Definition 27. Let $\mathcal{U} = \{u_1, \dots, u_n\}$ be a set of n users and $\mathcal{G} = (g_1, \dots, g_\tau)$ be a τ -tuple of conference sizes with respective maximum adversary set sizes $\mathcal{B} = (b_1, \dots, b_\tau)$, such that $2 \leq g_i \leq n$ and $g_i + b_i \leq n$, for all $1 \leq i \leq \tau$. We refer to the i -th conference, $1 \leq i \leq \tau$, of size g_i by G_i . A τ -restricted $(\mathcal{G}, \mathcal{B})$ -CKA scheme is a key agreement scheme that satisfies the following conditions:

1. *Interactive property:* Without knowing the communicated messages within a conference G_i with $|G_i| = g_i \in \mathcal{G}$, no subset of users has any information on k_{G_i} , given all the user keys, $U_{\mathcal{U}}$:

$$H(\mathbf{K}_{G_i} | \mathbf{U}_{\mathcal{U}}) = H(\mathbf{K}_{G_i}). \quad (4.1)$$

2. *Correctness:* For every conference G_i with $|G_i| = g_i \in \mathcal{G}$, all users $u_j \in G_i$, $1 \leq j \leq g_i$, using their secret information and knowing the communicated messages, m_i , can compute the conference key for G_i :

$$H(\mathbf{K}_{G_i} | \mathbf{U}_j \mathbf{M}_i) = 0. \quad (4.2)$$

3. *Perfect secrecy:* No information about the key of any of the conferences, G_i with $|G_i| = g_i \in \mathcal{G}$, can be found by the members of a disjoint adversary set, A_i with $|A_i| = b_i \in \mathcal{B}$, $G_i \cap A_i = \emptyset$, given all the user keys of the members of A_i and all the communicated messages:

$$H(\mathbf{K}_{G_i} | \mathbf{U}_{A_i} \mathbf{M}) = H(\mathbf{K}_{G_i}). \quad (4.3)$$

Remark 7. The τ -restricted (g, b) -CKA scheme of Blundo et al. is a special case of the τ -restricted $(\mathcal{G}, \mathcal{B})$ -CKA scheme when all the g_i 's in \mathcal{G} are initialized to g .

Notation:

We refer to the 1-restricted (g, b) -CKA scheme of Blundo et al. as $(\mathbf{g}, \mathbf{b}) - \mathbf{1RD}$ scheme. This notation is used to emphasize on the use of **Resolvable Designs** for computing **one** conference key. Similarly, by $(\mathbf{g}, \mathbf{b}) - \tau\mathbf{RD}$ scheme we refer to the τ -restricted (g, b) -CKA scheme of Blundo et al.

Remark 8. We assume that for a given tuple of conference sizes, $\mathcal{G} = (g_1, \dots, g_\tau)$, we have $g_i \leq g_{i+1}$ for $i = 1, \dots, \tau - 1$. This ensures that $\forall 1 \leq i \leq \tau - 1$, $\exists u_h \in G_{i+1} \setminus G_i$ such that u_h has unused user keys distributed for computing k_{G_i} that can be brought forward when computing $k_{G_{i+1}}$. We also assume that once a conference key, k_{G_i} , is computed, the conference members remember this

key for future applications. In other words, once a conference key is computed, we do not consider recomputing it. This has been the assumption for all the CKD schemes we have studied so far, too. Making this assumption helps when we argue later how the encryptions are samples of one-time pad.

Remark 9. Given $\mathcal{G} = (g_1, \dots, g_\tau)$, whenever we refer to an application of the $(g_i, b_i) - 1RD$, we assume that the TA chooses the ℓ_i that minimizes the respective key rate for the $(g_i, b_i) - 1RD$ scheme, Equation (3.18), for all $g_i \in \mathcal{G}$.

Remark 10. In this chapter, all the schemes assume a broadcast channel for their communication model.

In the following three sections, we introduce three τ -restricted $(\mathcal{G}, \mathcal{B})$ -CKA schemes.

4.2 Scheme 1

For the first scheme, we use a $(g_i, b_i) - 1RD$ scheme of Blundo et al., for each $g_i \in \mathcal{G}$ and the respective $b_i \in \mathcal{B}$.

Initialization

- Given the ordered tuple of conference sizes, $\mathcal{G} = (g_1, \dots, g_\tau)$, the TA privately distributes τ sets of user keys according to the respective $(g_i, b_i) - 1RD$ scheme to each user in \mathcal{U} over $GF(p^{k_i})$, the respective finite field, for all $1 \leq i \leq \tau$.

Conference Key Computation

- When a group of users, $G_i = \{u_{i1}, \dots, u_{ig_i}\}$, $|G_i| = g_i$, want to compute a conference key, they follow the *conference key computation* phase of the respective $(g_i, b_i) - 1RD$ scheme, for all $1 \leq i \leq \tau$.

Remark 11. Since we want to have conference keys of all the same size, we require that for all $1 \leq i \leq \tau$,

$$\begin{aligned} g_1 r_1 k_1 &= g_i r_i k_i, \\ p^{k_i} &> n, \end{aligned} \tag{4.4}$$

with $r_i = \binom{g_i-2}{\ell_i-2}$.

4.2.1 Efficiency

From Equation 3.18 and Equation 3.19, we can directly evaluate the performance of this scheme.

For the key rate we compute:

$$\frac{|U_{\mathcal{K}}|}{|K_G|} = n \frac{\sum_{i=1}^{\tau} \ell_i \binom{g_i+b_i-1}{\ell_i-1} k_i}{g_1 r_1 k_1} = n \sum_{i=1}^{\tau} \frac{\ell_i \binom{n-1}{\ell_i-1}}{g_i r_i}, \tag{4.5}$$

where k_i is replaced from Equation (4.4). Similarly, the communication rate is:

$$\frac{|M_{G_i}|}{|K_{G_i}|} = \frac{g_i r_i \chi_i k_i}{g_1 r_1 k_1} = \chi_i, \tag{4.6}$$

with $r_i = \binom{g_i-2}{\ell_i-2}$ and $\chi_i = \frac{g_i-1}{\ell_i-1}$, for all $1 \leq i \leq \tau$.

Before we continue with other schemes, we discuss the challenge of having different conference sizes, g_i 's.

Remark 12. Note that in a $(g, b) - \tau$ RD scheme, described in Section 3.3.3, since all the conferences are of a same size, g , once fixing ℓ such that $g \equiv 1 \pmod{\ell-1}$, this will hold for all the τ conferences. However, in a τ -restricted $(\mathcal{G}, \mathcal{B})$ -CKA scheme, after choosing an appropriate ℓ_i for g_i , it might be the case that:

$$\begin{aligned} g_i &\equiv 1 \pmod{\ell_i-1}, \\ g_{i+1} &\not\equiv 1 \pmod{\ell_i-1}. \end{aligned} \tag{4.7}$$

The inconsistency in Equation (4.7) affects the performance of $u_h \in G_{i+1} \setminus G_i$ when he wants to append another piece of secret to the key of G_{i+1} , as his user keys from Δ_i will not match the block sizes in the parallel classes of G_{i+1} .

4.3 Scheme 2

The idea of this scheme is to stay as close as possible to the τRD scheme by handling the potential parameter inconsistencies, mentioned in Remark 12.

Since the key rate in $(g_i, b_i) - 1RD$ is a function of ℓ_i for $2 \leq \ell_i \leq g_i$, there is a possibility the value of ℓ_i that minimize the key rate, happens for $\ell_i > 2$. In this case, we might encounter a similar inconsistency as in Equation (4.7). One way of addressing this problem is to introduce *dummy users*.

The idea is to find the minimum integer, $f_i \in \mathbb{N} \cup \{0\}$, such that for all $g_i \in \mathcal{G}$:

$$g_i + f_i \equiv 1 \pmod{(\ell_{i-1} - 1)}, \quad \forall 2 \leq i \leq \tau. \quad (4.8)$$

Let $u_1^{(d)}, \dots, u_{f_i}^{(d)}$ be dummy users with public identities in $GF(p^{k_i})$. We assume these identities have not been assigned to any other users in \mathcal{U} . Take $G_i^{(d)} = G_i \cup \{u_1^{(d)}, \dots, u_{f_i}^{(d)}\}$, so $|G_i^{(d)}| = g_i + f_i$ and according to the choice of f_i , the $(\ell_{i-1} - 1)$ -uniform hypergraph over $G_i^{(d)} \setminus \{u_h\}$, is a resolvable design with $\chi' = \frac{g_i + f_i - 1}{\ell_{i-1} - 1}$ many $(\ell_{i-1} - 1)$ -subsets in each parallel class, for $u_h \in G_i \setminus G_{i-1}$ with minimum identity. Hence u_h 's user key from Δ_{i-1} is compatible with the size of the blocks in $G_i^{(d)} \setminus \{u_h\}$'s parallel classes and so u_h can append another piece of secret to k_{G_i} by following similar steps as if it is a $(g_i + f_i, b_i) - \tau RD$ scheme.

Remark 13. Given a $\mathcal{G} = (g_1, \dots, g_\tau)$, let $g_{\max} = \max \{g_1, \dots, g_\tau\}$. Since f_i 's satisfy:

$$0 \leq f_i \leq g_i,$$

we conclude that:

$$\max \{f_i, 1 \leq i \leq \tau\} \leq g_{\max}. \quad (4.9)$$

Remark 14. For all $1 \leq i \leq \tau$, we assume that $GF(p^{k_i})$ has at least $n + g_{\max}$ elements.

For this construction, given a τ -tuple of conference sizes, $\mathcal{G} = (g_1, \dots, g_\tau)$, and the respective adversary set sizes, $\mathcal{B} = (b_1, \dots, b_\tau)$, the TA can compute the ℓ_i 's and form another τ -tuple $\mathcal{L} =$

$(l_{01}, \dots, l_{0\tau})$. Finally, let $\mathcal{F} = (0, f_2, \dots, f_\tau)$ be a τ -tuple containing the respective number of dummy users, according to Equation (4.8). We describe the two phases of the scheme below.

Initialization:

- The TA distributes the user keys according to $\tau - 1$ copies of a $(g_i, b_i + 1) - 1RD$ scheme over $(GF(p^{k_i}))^{\ell_i}$ for $1 \leq i \leq \tau - 1$ and one copy of the user keys according to the $(g_\tau, b_\tau) - 1RD$ scheme over $(GF(p^{k_\tau}))^{\ell_\tau}$, for some positive integer values $k_i \leq k_1, \forall 1 \leq i \leq \tau$. See Section 3.3.2.

Conference key computation:

- When users in G_1 want to compute the conference key, they follow the conference key computation phase of the $(g_1, b_1) - 1RD$ scheme.
- For all $2 \leq i \leq \tau$, users in G_i follow the conference key computation phase of the respective $(g_i, b_i) - 1RD$ scheme. Since we assumed that for all $2 \leq i \leq \tau$, $g_{i-1} \leq g_i$, we are assured that there exists a $u_h \in G_i \setminus G_{i-1}$. u_h randomly chooses another value $r^{(h)} \in GF(p^{\ell_{i-1} p^{k_i-1}})$. Note that u_h shares a common key with any $(\ell_{i-1} - 1)$ -subset of $G_i^{(d)}$. Using these keys, u_h encrypts $r^{(h)}$ with all the ℓ_{i-1} components of his shared key with the blocks in $G_i^{(d)}$.

We remark that the key size for the first conference, G_1 , is:

$$|k_{G_1}| = g_1 r_1 k_1 \log_2 p,$$

and for all $2 \leq i \leq \tau$, the key size for conference G_i is:

$$|k_{G_i}| = (g_i r_i k_i + \ell_{i-1} r'_i k_{i-1}) \log_2 p,$$

with $r_i = \binom{g_i-2}{\ell_i-2}$ and $r'_i = \binom{g_i+f_i-2}{\ell_{i-1}-2}$. So we need to make sure that the equation below always holds for an arbitrary choice of \mathcal{G} :

$$g_1 k_1 r_1 = g_i k_i r_i + \ell_{i-1} r'_i k_{i-1}. \quad (4.10)$$

In fact we show that for every choice of $\mathcal{G} = (g_1, \dots, g_\tau)$ and \mathcal{B} , with $2 < g_1 \leq g_2 \leq \dots \leq g_\tau$, and consequently the \mathcal{L} and \mathcal{F} , there are always values for k_2, \dots, k_τ such that a realization of the proposed scheme is possible.

Lemma 5. *Let k_1 be a positive integer such that $p^{k_1} > n + g_{\max}$. Given $\mathcal{G} = (g_1, \dots, g_\tau)$, such that $g_1 > 2$, and all integers $2 \leq \ell_t \leq g_t$ such that $g_t \equiv 1 \pmod{(\ell_t - 1)}$, $1 \leq t \leq \tau$, let $I_1 = g_1 r_1 - \ell_1 r_2'$ and $I_t = g_1 r_1 - \frac{\ell_t r_{t+1}'}{g_t r_t} I_{t-1}$. If $g_1 k_1 r_1 = g_t k_t r_t + \ell_{t-1} k_{t-1} r_t'$ then it holds that $k_t = \frac{k_1}{g_t r_t} I_{t-1}$, $\forall 2 \leq t \leq \tau$.*

Proof. We prove this lemma by induction on t . Let $t = 2$, then:

$$g_1 k_1 r_1 = g_2 k_2 r_2 + \ell_1 k_1 r_2',$$

$$k_2 = \frac{g_1 k_1 r_1}{g_2 r_2} - \frac{\ell_1 k_1 r_2'}{g_2 r_2} = \frac{k_1}{g_2 r_2} (g_1 r_1 - \ell_1 k_1 r_2') = \frac{k_1}{g_2 r_2} I_1.$$

Assume the lemma holds for some $t < \tau$, we show it also holds for $t + 1$:

$$g_1 k_1 r_1 = g_{t+1} k_{t+1} r_{t+1} + \ell_t k_t r_{t+1}' \longrightarrow k_{t+1} = \frac{1}{g_{t+1} r_{t+1}} (g_1 k_1 r_1 - \ell_t k_t r_{t+1}').$$

By the induction hypothesis we can substitute for $k_t = \frac{k_1}{g_t r_t} I_{t-1}$, which leads to:

$$k_{t+1} = \frac{1}{g_{t+1} r_{t+1}} (g_1 k_1 r_1 - \ell_t r_{t+1}' (\frac{k_1}{g_t r_t} I_{t-1})) = \frac{k_1}{g_{t+1} r_{t+1}} (g_1 r_1 - \frac{\ell_t r_{t+1}'}{g_t r_t} I_{t-1}) = \frac{k_1}{g_{t+1} r_{t+1}} I_t,$$

which concludes the proof. \square

Remark 15. For $g_1 = 2$, ℓ_2 must be 2 and hence $k_2 = 0$. To avoid this, we assume $g_1 > 2$. Note that for $g = 2$, a similar issue also occurs in the original paper of Blundo et al., [5].

We give an example that shows how the scheme works.

Example 6. Given \mathcal{U} , \mathcal{G} and \mathcal{B} with $g_i = 7$ and $\ell_{i-1} = 6$. Let $G_i = \{u_1, \dots, u_7\}$. To satisfy Equation (4.8), we take $f_i = 4$:

$$7 + 4 \equiv 1 \pmod{6 - 1}.$$

By adding four dummy users, $u_1^{(d)}, u_2^{(d)}, u_3^{(d)}, u_4^{(d)}$, to G_i we get $G_i^{(d)} = G_i \cup \{u_1^{(d)}, u_2^{(d)}, u_3^{(d)}, u_4^{(d)}\}$.

Without loss of generality, assume $u_h = u_7$. The collection of all subsets of size $\ell_{i-1} - 1 = 5$ in $G_i^{(d)} \setminus \{u_7\}$ is:

$$\begin{aligned} & \{\{1, 2, 3, 4, 5\}, \{1, 2, 3, 4, 6\}, \{1, 2, 3, 4, f_1\}, \\ & \{1, 2, 3, 4, f_2\}, \{1, 2, 3, 4, f_3\}, \{1, 2, 3, 4, f_4\} \\ & \vdots \\ & \dots, \dots, \{6, f_1, f_2, f_3, f_4\} \} \end{aligned}$$

We expect to reorganize these blocks into $r'_i = \binom{g_i + f_i - 2}{\ell_{i-2}} = \binom{9}{4} = 126$ parallel classes such that there are $\chi'_i = \frac{7+4-1}{6-1} = 2$ subsets in each class. Hence by applying the dummy users method, a complete set of parallel classes can be formed in which the size of the elements properly match the parameters of Δ_{i-1} .

4.3.1 Performance

The key rate of this scheme is computed as:

$$\frac{|U_{\mathcal{U}}|}{|K_G|} = n \left(\sum_{i=1}^{\tau-1} \frac{\ell_i k_i \binom{g_i + b_i}{\ell_{i-1}}}{g_1 r_1 k_1} + \frac{\ell_\tau k_\tau \binom{g_\tau + b_\tau - 1}{\ell_{\tau-1}}}{g_1 r_1 k_1} \right), \quad (4.11)$$

and the communication rate is:

$$\frac{|M_{G_1}|}{|K_{G_1}|} = \frac{g_1 r_1 \chi_1 k_1}{g_1 r_1 k_1} = \chi_1, \quad (4.12)$$

$$\frac{|M_{G_i}|}{|K_{G_i}|} = \frac{g_i r_i \chi_i k_i + \ell_{i-1} r'_i \chi'_i k_{i-1}}{g_1 r_1 k_1}, \quad \forall 2 \leq i \leq \tau, \quad (4.13)$$

with $r_i = \binom{g_i - 2}{\ell_{i-2}}$, $\chi_i = \frac{g_i - 1}{\ell_{i-1}}$, $r'_i = \binom{g_i + f_i - 2}{\ell_{i-1} - 2}$ and $\chi'_i = \frac{g_i + f_i - 1}{\ell_{i-1} - 1}$.

4.4 Scheme 3

This last scheme can be viewed as a special case of the previous scheme, when all ℓ_i 's are set to 2. Note that in this case, Equation (4.7) is satisfied and hence with $\mathcal{F} = (0, \dots, 0)$, one can follow

the steps of the previous scheme. We remark that for this scheme, $r_i = r'_i = 1$ and $\chi_i = \chi'_i = g_i - 1$, for all $1 \leq i \leq \tau$.

Initialization:

- The TA distributes the user keys according to the user keys of a $(g_i, b_i) - 1RD$ scheme over $GF(p^{k_i})^2$ with $l_i = 2$, for all $1 \leq i \leq \tau - 1$. The TA also distributes one copy of the user keys distributed according to the $(g_\tau, b_\tau - 1) - 1RD$ scheme over $(GF(p^{k_\tau}))^2$, for some positive integer values $k_i \leq k_1$, $1 \leq i \leq \tau$. See Section 3.3.2.

Conference key computation:

- When users in G_1 want to compute the conference key, they follow the conference key computation phase of the respective $(g_1, b_1) - 1RD$ scheme.
- For all $2 \leq i \leq \tau$, the users in G_i follow the conference key computation phase of Scheme 2, described in Section 4.3 with $\ell_i = 2$.

It is important to make sure that all the computed conference keys are of the same size, i.e. the following equation holds true:

$$g_1 k_1 = g_t k_t + 2k_{t-1}, \quad \forall 2 \leq t \leq \tau. \quad (4.14)$$

The argument that states for every given τ -tuple of conference sizes, \mathcal{G} , a realization of our scheme exists is identical to Lemma 5 when ℓ_i 's are all set to 2.

4.4.1 Performance

Substituting $\ell_i = 2$ and $f_i = 0$, we compute $r_i = \binom{g_i-2}{\ell_i-2} = 1$, $\chi_i = \frac{g_i-1}{\ell_i-1} = g_i - 1$, $r'_i = \binom{g_i+f_i-2}{\ell_{i-1}-2} = 1$ and $\chi'_i = \frac{g_i+f_i-1}{\ell_{i-1}-1} = g_i - 1$. From Equations (4.11), (4.12) and (4.13), we compute the key rate and communication rate of Scheme 3 accordingly. For the key rate we compute:

$$\frac{|U_{\mathcal{K}}|}{|K_G|} = n \left(\sum_{i=1}^{\tau-1} \frac{2k_i(g_i + b_i)}{g_1 k_1} + \frac{2k_\tau(g_\tau + b_\tau - 1)}{g_1 k_1} \right). \quad (4.15)$$

The communication rate is:

$$\frac{|M_{G_1}|}{|K_{G_1}|} = \frac{g_1 r_1 \chi_1 k_1}{g_1 r_1 k_1} = \chi_1 = g_1 - 1, \quad (4.16)$$

$$\frac{|M_{G_i}|}{|K_{G_i}|} = \frac{g_i(g_i - 1)k_i}{g_1 k_1} + \frac{2(g_i - 1)k_{i-1}}{g_1 k_1}, \quad \forall 2 \leq i \leq \tau. \quad (4.17)$$

Remark 16. We remark that the security of Scheme 1, 2 and 3 is directly obtained from the perfect secrecy of the $(g_i, b_i) - 1RD$.

4.5 Comparison

To compare the performance of these three schemes, we used Mathematica 7 to simulate all the parameters that appear in the respective efficiency expressions.

Note that once n , τ , \mathcal{G} , p and k_1 are given, the rest of the parameters can be computed uniquely. To show the result of comparison, we chose to present the key rates in a 2D graph. If the i -th point on the horizontal axis represents having the first i conference sizes from $\mathcal{G} = (g_1, \dots, g_\tau)$, the vertical axis presents the key rate as if $\tau = i$. That is having only the first i conference sizes, $\mathcal{G}_i = (g_1, \dots, g_i)$, from \mathcal{G} to compute conference keys for.

In the following sections, we compare the key rates and communication rates.

4.5.1 Key rates

To compare the key rates of the three schemes, we need to compare the expressions below, from Equation (4.5), (4.11) and (4.15):

$$\begin{aligned} \frac{|U_{\mathcal{G}}|}{|K_{\mathcal{G}}|}^{(Sch1)} &= n \frac{\sum_{i=1}^{\tau} \ell_i \binom{n-1}{\ell_i-1} k_i}{g_1 r_1 k_1} = n \sum_{i=1}^{\tau} \frac{\ell_i \binom{n-1}{\ell_i-1}}{g_i r_i}, \\ \frac{|U_{\mathcal{G}}|}{|K_{\mathcal{G}}|}^{(Sch2)} &= n \left(\sum_{i=1}^{\tau-1} \frac{\ell_i \binom{n}{\ell_i-1} k_i}{g_1 r_1 k_1} + \frac{\ell_\tau \binom{n-1}{\ell_\tau-1} k_\tau}{g_1 r_1 k_1} \right), \\ \frac{|U_{\mathcal{G}}|}{|K_{\mathcal{G}}|}^{(Sch3)} &= n \left(\sum_{i=1}^{\tau-1} \frac{2n k_i}{g_1 k_1} + \frac{2(n-1)k_\tau}{g_1 k_1} \right), \end{aligned} \quad (4.18)$$

with $r_i = \binom{g_i-2}{\ell_i-2}$.

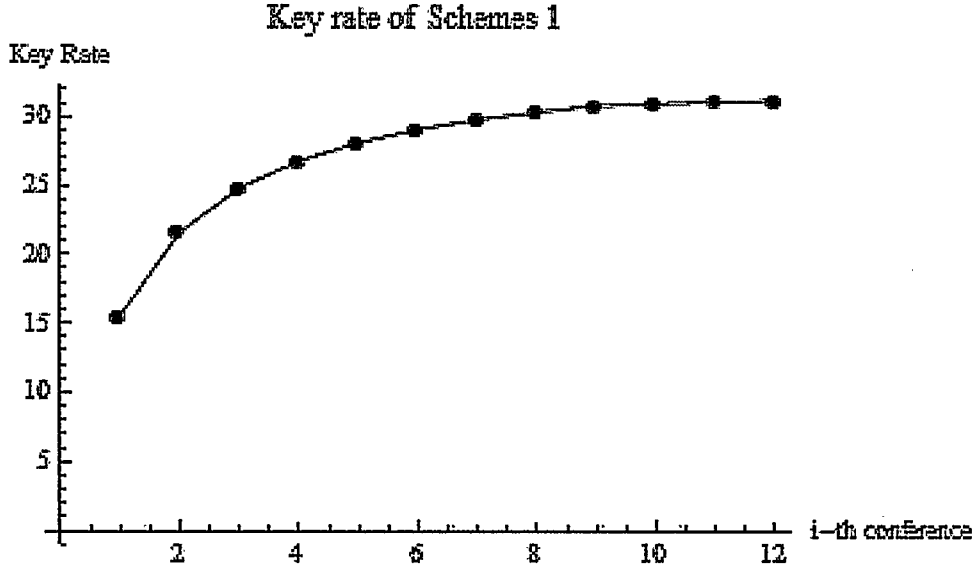


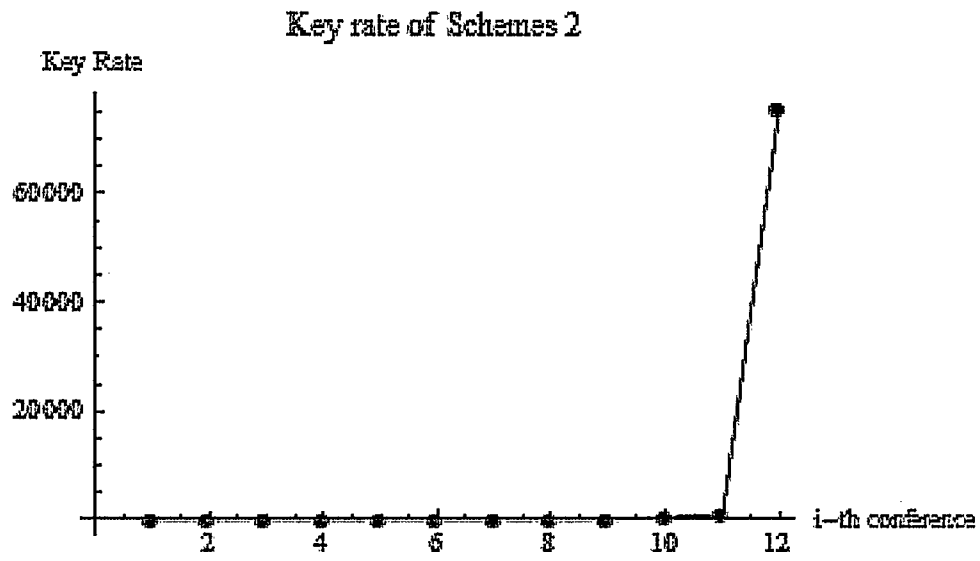
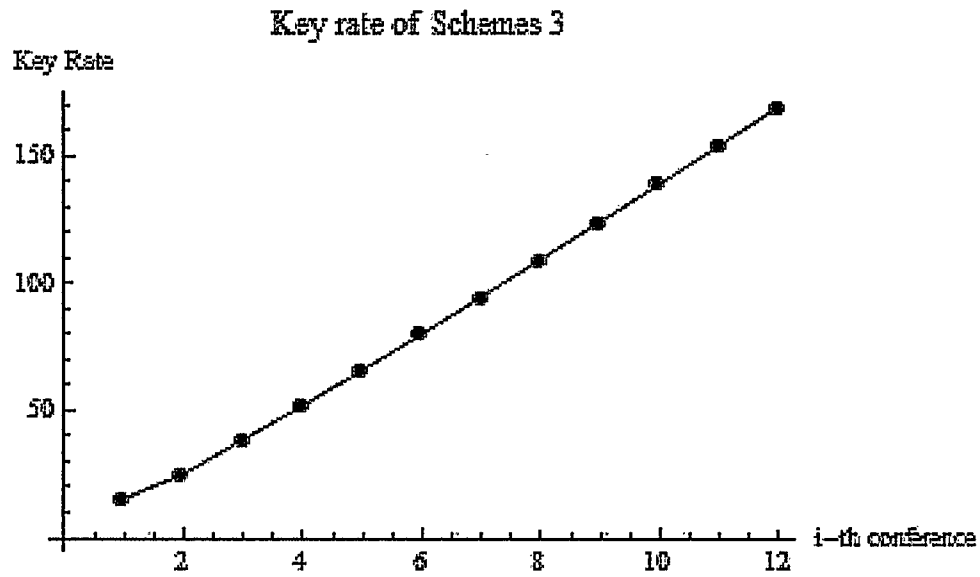
Figure 4.1: Scheme 1's key rates for \mathcal{G}_{3i+2}

Note that without loss of generality, we take $g_i + b_i = n$, $\forall 1 \leq i \leq \tau$. These expressions depend on multiple parameters. To simulate all these parameters, we implemented the key rate of each scheme using Mathematica 7 and conducted the experiment with a range of parameter values.

The Mathematica code sample is provided in Appendix A.1.1. In the next section we present one of the results of running this code for certain parameter values.

Simulation result

Applying the code, sample of which is provided in Appendix A.1.1, we obtained Figures 4.1, 4.2 and 4.3 for the key rate of Scheme 1, 2 and 3, respectively, with \mathcal{G} generated by function $g[i] = 3i + 2$, i.e. $\mathcal{G}_{3i+2} = (5, 8, 11, 14, 17, 20, 23, 26, 29, 32, 35, 38)$. This choice of $g[i]$ results in non-trivial l_i 's and f_i 's, i.e. not all l_i 's are equal to 2 and not all f_i 's are equal to 0. This is of significant importance for our simulation because in case all $l_i = 2$, then Scheme 2 and Scheme 3 are essentially identical. Similarly for the case where all $f_i = 0$.

Figure 4.2: Scheme 2's key rates for \mathcal{G}_{3i+2} Figure 4.3: Scheme 3's key rates for \mathcal{G}_{3i+2}

Analysis

Note that since in Scheme 2:

1. We had recycled the unused user keys from Δ_i in Δ_{i+1} and hence used smaller field sizes, (with Δ_i referring to the section of the scheme for relative parameters to g_i)
2. We used the best values for ℓ_i 's so that the respective key rate of $(g_i, b_i) - 1RD$ is minimized,

we expected Scheme 2 to have the lowest key rate comparing to Scheme 1 and Scheme 3. However, despite our expectation, Scheme 2 does not have better key rate than the other two schemes. To investigate the result, let us look back at Equation (4.18). By comparing the key rate expressions of Scheme 1 and Scheme 2, one would see that the factors that differentiate the growth pace in the key rates of the two schemes are essentially $\binom{n-1}{\ell_i-1} k_i^{(1)}$ and $\binom{n}{\ell_i-1} k_i^{(2)}$. The indices (1) and (2) refer to the respective field sizes of Scheme 1 and Scheme 2. We run another code to compare these two values, a sample of which is provided in Appendix A.1.2.

It turns out that $\binom{n}{\ell_i-1}$ grows faster than $\binom{n-1}{\ell_i-1}$ compared to how faster $k_i^{(2)}$ shrinks compared to $k_i^{(1)}$. Hence the whole expression for $\frac{|U_{\mathcal{Q}}|}{|K_G|}^{(Sch2)}$ grows faster than $\frac{|U_{\mathcal{Q}}|}{|K_G|}^{(Sch1)}$.

On the other hand, when comparing the key rates for Scheme 2 and Scheme 3, we note that the differentiating factor is $\binom{n}{\ell_i-1} \frac{\ell_i k_i^{(2)}}{r_1}$ and $2n k_i^{(3)}$. Using the sample code presented in Appendix A.1.3, we conclude that $\binom{n}{\ell_i-1} \frac{\ell_i}{r_1}$ grows much faster than $2n$, compared to how fast $k_i^{(2)}$ shrinks compared to $k_i^{(3)}$. Hence the whole expression for $\frac{|U_{\mathcal{Q}}|}{|K_G|}^{(Sch2)}$ grows faster than $\frac{|U_{\mathcal{Q}}|}{|K_G|}^{(Sch3)}$.

The simulation result for comparison of Scheme 1 and Scheme 3 shows that choosing the best value for ℓ_i to minimize the respective $(g_i, b_i) - 1RD$ key rate, as is in Scheme 1, is more influential than implementing the $(g_i, b_i) - 1RD$ schemes over smaller fields, as is in Scheme 3.

In conclusion, the idea of recycling the unused user keys that are distributed by the TA for computing the i -th conference key for the computation of the $(i+1)$ -th conference key, enables us to choose the user keys for the $(i+1)$ -th protocol from a smaller field compared to the user keys

chosen for computing the first conference key. However, for schemes that rely on dummy users, it turns out that the reduction caused in the field sizes is smaller than the incrementation caused in the size of user keys.

4.5.2 Communication rate

To compare the communication rates of the three schemes, we need to compare the expressions below, from Equation (4.6), (4.13) and (4.17):

$$\begin{aligned} \frac{|M_{G_i}|}{|K_{G_i}|}^{(Sch1)} &= \chi_i, \\ \frac{|M_{G_i}|}{|K_{G_i}|}^{(Sch2)} &= \frac{g_i r_i \chi_i k_i + \ell_{i-1} r'_i \chi'_i k_{i-1}}{g_1 r_1 k_1}, \\ \frac{|M_{G_i}|}{|K_{G_i}|}^{(Sch3)} &= \frac{g_i (g_i - 1) k_i}{g_1 k_1} + \frac{2(g_i - 1) k_{i-1}}{g_1 k_1}, \end{aligned} \quad (4.19)$$

with $r_i = \binom{g_i - 2}{\ell_i - 2}$, $\chi_i = \frac{g_i - 1}{\ell_i - 1}$, $r'_i = \binom{g_i + f_i - 2}{\ell_{i-1} - 2}$ and $\chi'_i = \frac{g_i + f_i - 1}{\ell_{i-1} - 1}$, for all $2 \leq i \leq \tau$.

Since these expressions depend on many parameters, we simulate them using Mathematica 7.

A sample of the code is provided in Appendix A.2.

Simulation result

Figure 4.4, 4.5 and 4.6 picture the communication rate for Scheme 1, 2 and 3, respectively.

According to the simulation result, Scheme 1 has the lowest communication rate comparing to Scheme 2 and 3.

We remark that for most of the g_i 's, the best value for ℓ_i to minimize the corresponding key rate was 2. However, at times where $\ell_i > 2$, we observed unpredictable increase or decrease in the communication rate values.

4.6 Conclusion

To realize a CKD scheme that works for computing τ conference keys for conferences sizes $\mathcal{G} = (g_1, \dots, g_\tau)$, we adopted the idea of the τ -restricted (g, b) -CKA scheme of Blundo et al. Our goal

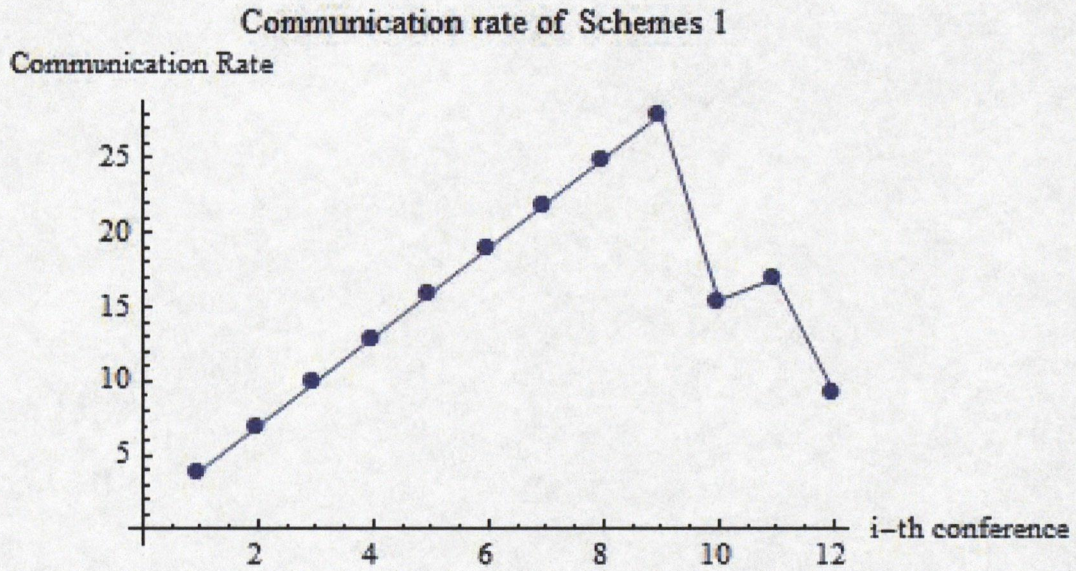


Figure 4.4: Scheme 1's communication rates for \mathcal{G}_{3i+2}

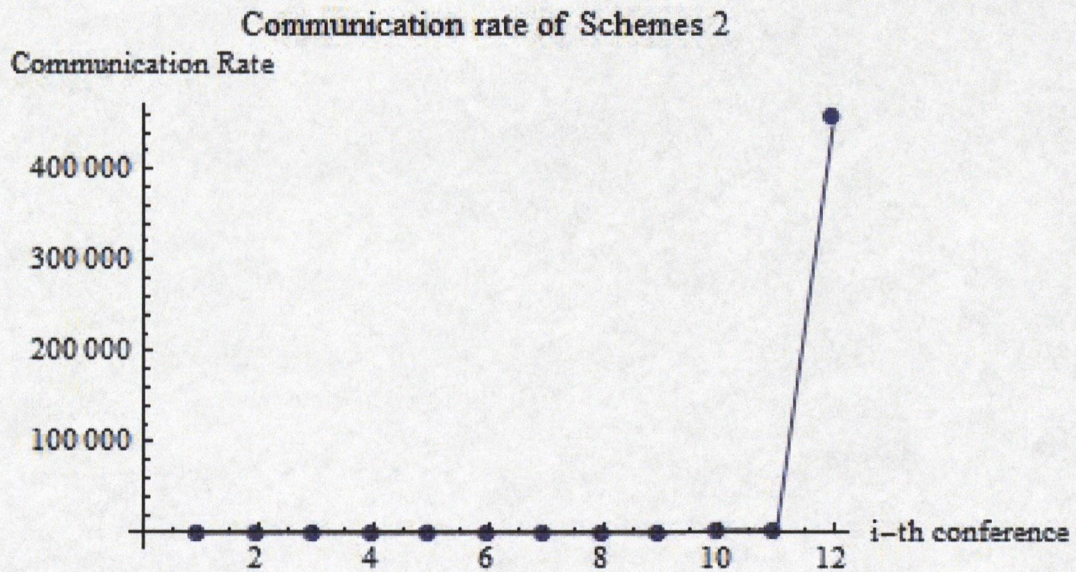


Figure 4.5: Scheme 2's communication rates for \mathcal{G}_{3i+2}

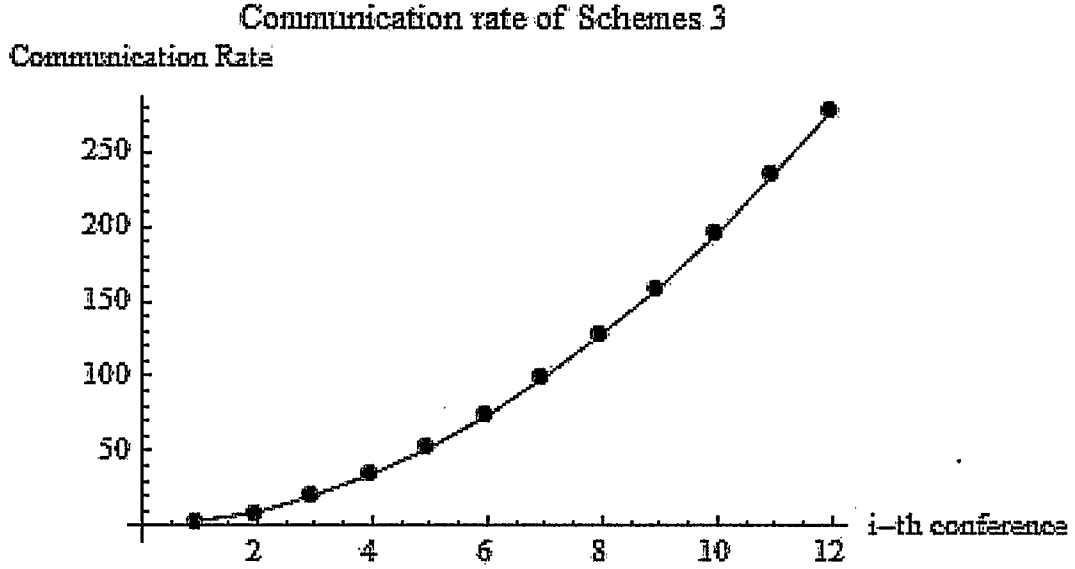


Figure 4.6: Scheme 3's communication rates for \mathcal{G}_{3i+2}

was to incorporate previously distributed secrets to compute the i -th conference key for computing the $(i+1)$ -th conference key. Since the scheme in [5] works for computing τ conference keys for conferences of the same size, g , the main challenge is the inconsistency in the parameter values, as stated in Equation (4.7).

We proposed three ways of realizing such CKD schemes, (i) use a 1-restricted (g_i, b_i) -CKA scheme of Blundo et al. for all $1 \leq i \leq \tau$ (ii) use dummy users to obtain the consistency in the parameter values, and (iii) initialize all the ℓ_i 's to 2.

We provided an implementation of the key rates in these three schemes using Mathematica. Our code can be used to assess the key rate of the three schemes for given parameter values and conference sizes. We saw that despite our expectation, the key rate of Scheme 2 is not always lower than the key rate of Scheme 1 or 3. This contradicts our expectation since in Scheme 2, all field sizes are smaller than k_1 whereas in Scheme 1 the field size remains the same for any conference size, g_i and in Scheme 2, although the field sizes are smaller than k_1 , they are larger than the counter field size in Scheme 2. By doing further comparison, we concluded that the reason

behind the inconsistency of the comparison results to our expectation is the fact that in order to lower the field size in Scheme 2, we had to introduce the fake users and distribute the user keys as if there are such fake users. This increase in the user key size caused by the fake users is more significant than the reduction in the field sizes.

In conclusion, for a given $\mathcal{G} = (g_1, \dots, g_\tau)$, multiple applications of the $(g_i, b_i) - 1RD$ scheme of Blundo et al. results in a better key rate and communication rate than applying Scheme 2 or Scheme 3. On the other hand, we remark that another promising approach to design a $(\mathcal{G}, \mathcal{B})$ -CKA scheme is to initiate a τ_i -restricted (g_i, b_i) -CKA scheme for every $g_i \in \mathcal{G}$, with τ_i the number of times g_i has appeared in \mathcal{G} .

The work in this chapter does not conclude the research line regarding the realization of a τ -restricted $(\mathcal{G}, \mathcal{B})$ -CKA scheme. There are many interesting open questions such as lower bound on the size of user keys in such setting and also constructions in which the conference sizes in \mathcal{G} do not necessarily obey the condition that $g_{i-1} \leq g_i$, for $2 \leq i \leq \tau$, which are left for future work.

Chapter 5

CKD scheme for Tree Structured Conferences

In this chapter we argue that broadcast channels are not suitable tools to model the communication infrastructure with. Instead, communication graphs provide a better model of the communication settings. In a communication graph, every user is represented by a node and two nodes are connected with an edge if there exists a communication channel connecting the respective users. This forces the associated communication graph to a group of users to be a connected one or otherwise there is no other means of communication for that group of users to communicate with one another.

Since from graph theory we know that every connected graph has a spanning tree, we introduce a new CKA scheme in which the communications are based on the spanning tree within a conference's communication graph. Our scheme is designed to compute one conference key for a conference of size g and adversary set of size b such that $g + b \leq n$. We compare the performance of our scheme to a modified version of $(g, b) - 1RD$ scheme in which the messages are communicated through the edges of a communication graph instead of broadcasting them. We prove that our scheme always attains a lower communication rate and for certain parameter values, achieves a lower key rate.

5.1 Motivation

In large networks, two users may not be directly connected and therefore need to find a path through other users to communicate. Communication graphs, where each node represents a network user and an edge represents a point to point communication channel, have been used to model communication channels and communication paths between users.

For instance, Figure 5.1 clearly shows the underlying spanning tree in the communication graph

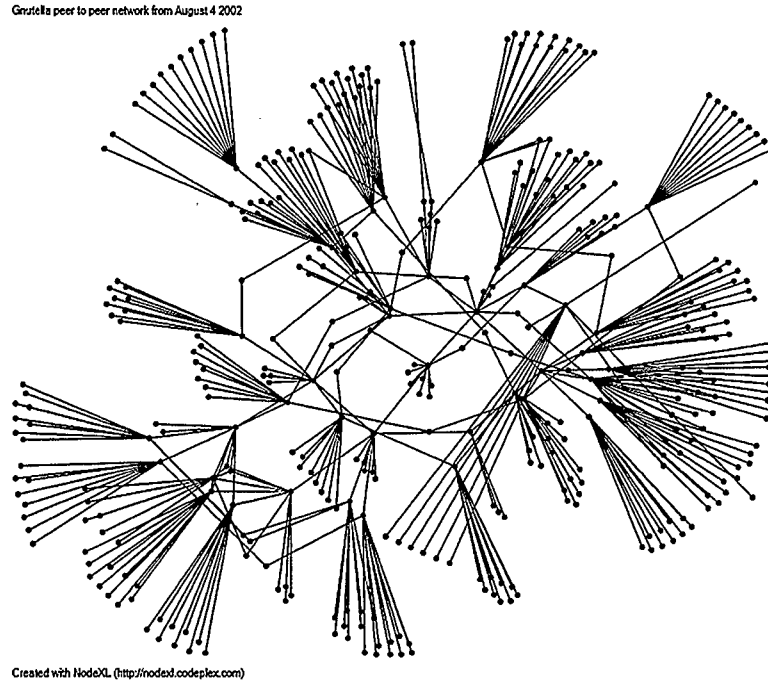


Figure 5.1: Gnutella peer to peer file sharing network from August 4th 2002 for about 100 nodes

of the Gnutella peer to peer file sharing network. This figure is obtained from the data set in [2] for approximately the first 100 nodes out of a total of 10876 nodes. Each node represents a computer and the edges show the communications between the computers.

5.2 The model

In this chapter we consider a setting where users are connected to each other by point to point channels, and use the collection of such channels to represent the communication graph of the users. We assume that all transmitted messages are visible to the adversary, that is the communication channels guarantee authenticated communication but is not private. For two users to communicate over such communication graph, it is necessary that the corresponding nodes be connected through a path in the graph. For a group of users that form a conference, it is necessary to have the subgraph

whose nodes designate that group of users be a connected one. From graph theory, this guarantees a spanning tree within the communication subgraph of group of users, say T . We assume that the communication subgraph associated with a conference of size g has an m -balanced rooted spanning tree: a rooted tree in which every node, except the leaf nodes, has exactly m children, see Definition 18. We also assume that this m -balanced rooted tree is symmetric with respect to the root node and is complete in the sense that the length of the path from any leaf node to the root is d . This assumption is made in order to compute the efficiency measures easier, as in Section 5.5.

In the next section we present a new 1-restricted (g, b) -CKA scheme that is designed specifically for these tree structured networks.

5.3 Tree Structured Key Agreement

In this section we introduce our new CKA scheme for conferences with an m -balanced rooted tree as the communication graph. The idea is to have the TA provide a set of g shared keys between every pair of parent-child nodes in the initialization phase. Later at the conference key computation phase, each user chooses a random value and shares it with other conference members. The final conference key is obtained by concatenating all these random values. We design a method to assure that all these random values get delivered to all the conference members. We do this in two phases of message communication: leaf-to-root and root-to-leaf.

At the leaf-to-root phase, starting from the leaf nodes, each user encrypts and sends his random value and all the other random values he has received from his children, to his parent node. Once the root receives all the random values of the conference members, this phase of message transmission completes.

At the root-to-leaf phase, starting from the root node, each user sends each one of his children an encrypted version of his random value and all the other random values he has received from his parent or other children except the recipient child. This second phase of message transmission

completes once all the leaf nodes receive the $g - 1$ random values of the other conference members.

Note that we assume an order according to which the messages are transmitted. For instance, in the leaf-to-root phase, a non-leaf node does not start sending messages to his parent node unless he has received all the messages he expects from his children. Similarly, in the root-to-leaf phase, a non-root node does not start issuing his messages to any of his children unless he has received all the expected messages from his parent node. Note that the topology of the network is publicly known to all the users.

Here we begin the formal description of our new 1-restricted m -tree (g, b) -CKA scheme.

Let $\mathcal{U} = \{u_1, \dots, u_n\}$ be a set of n users with i denote the public identity of user u_i . Let $g, b \in \mathbb{N}^+$ with $g + b \leq n$ and p be a prime such that $p > m$. Let $T = \langle V, E \rangle$ denote the m -balanced rooted tree underlying the communication graph of the whole network.

Initialization:

1. For every user $u_i \in \mathcal{U}$, except leaf and root nodes, the TA randomly chooses a polynomial, $f_i(x)$, in one variable of degree at most $\min\{b, m - 1\}$ over $(GF(p))^g$. TA privately sends $f_i(x)$ to the respective user, u_i , evaluates $f_i(x)$ on the unique public id of user u_i 's children, and sends each child the resulting g -tuple value. That is $k_{i,j} = (k_{i,j}^1, \dots, k_{i,j}^g) = f_i(j) \in (GF(p))^g$ denotes the g -tuple shared key between a parent node, u_i , and the child, u_j . For $u_i \in \mathcal{U}$ who is not a leaf node nor the root node, the *user key* consists of a random polynomial f_i and one random g -tuple over $(GF(p))^g$.
2. The TA sends the root node only one random one-variable polynomial of degree at most $\min\{b, m - 1\}$ over $(GF(p))^g$.
3. The TA sends each leaf node a g -tuple from $(GF(p))^g$ obtained from the random polynomial given to that leaf node's parent.

Conference key computation:

Let $G = \{u_1, \dots, u_g\} \subseteq \mathcal{U}$ be a conference. Without loss of generality, take u_1 as the root for T_G , the subtree obtained from T with its nodes restricted to G . We define C_i as the set containing i and the public identity of user u_i 's children. Also let $T_{u_i} = (V_{C_i}, E_{C_i})$ denote the subgraph obtained from T_G when its node set, V_{C_i} , is restricted to C_i .

Every user $u_i \in G$ chooses a random value $r_i \in GF(p)$. There are two phases of message transmission: leaf-to-root and root-to-leaf.

1. In the leaf-to-root phase, starting from the leaf nodes in the conference, every user u_i encrypts and sends his secret, r_i , including all the secret values he has received thus far from his children, to his parent node, u_j , using the g -tuple shared key, k_{ji} :

$$m_{i,j}^{(t)} = r_c \oplus k_{j,i}^{(t)},$$

where $c \in C_i$ and $1 \leq t \leq g - 1$.

This phase completes once u_1 , the root node, receives $g - 1$ messages from his children.

2. In the root-to-leaf phase, starting from the root node, u_1 , every user u_i encrypts and sends to each of his children, u_j , his secret value, r_i , and all the secret values that he has received from his parent and children, except u_j , using the unused entries in their g -tuple shared key, $k_{i,j}$:

$$m_{i,j}^{(t)} = r_s \oplus k_{i,j}^{(t)},$$

where $s \in G \setminus V_{C_i}$ and $2 \leq t \leq g - 1$. This phase completes once all the leaf nodes receive $g - 1$ messages from their parents.

Note that t basically counts the number of messages that have already been transmitted between the respective two users and is used to determine which component of the g -tuple shared key should be used for encryption.

It is not hard to see that once these two steps are completed, every node has the secret of all the other conference members. The ultimate conference key is formed by concatenating all these g secrets, according to conference members' identities, smallest to the largest:

$$k_G = r_1 || \dots || r_g.$$

Remark 17. *Note that every two users communicate exactly g messages. For $T = \langle V, E \rangle$, the tree that denotes a conference communication graph, we have $|V| = g$. By removing any edge, $e \in E$, the tree T disconnects into two sub trees $T_1 = \langle V_1, E_1 \rangle$ and $T_2 = \langle V_2, E_2 \rangle$ such that $V_1 \cup V_2 = V$ and $E_1 \cup E_2 \cup \{e\} = E$. So the missing edge, e , is the only means of communication between the users in T_1 and T_2 . Since every node produces exactly one random number, we conclude that exactly $|V_1|$ messages had to pass e from one direction and $|V_2|$ messages from the opposite direction. This adds up to exactly $|V_1| + |V_2| = |V| = g$ messages to pass through any edge.*

The following example is to better illustrate the protocol.

Example 7. *Let $\mathcal{U} = \{u_1, \dots, u_{13}\}$ and $G = \{u_1, \dots, u_7\}$ with $b = 6$ and $m = 3$. Take $p = 11$, so $p > m$. Figure 5.2 and 5.3 shows the communication graph of G , T_G , with u_1 as the root node. According to the protocol, to each non-leaf user, $u_i \in \mathcal{U}$, the TA sends a randomly chosen polynomial, $f_i(x)$, of degree $\min\{b, m-1\} = 2$, with coefficients from $(GF(11))^7$. The TA also sends the shared key obtained from $f_i(x)$ to each of the u_i 's children. Let the random polynomial that the TA gives to u_3 be:*

$$f_3(x) = (2, 4, 3, 8, 0, 1, 8)x^2 + (1, 2, 3, 4, 5, 6, 7)x + (0, 3, 4, 2, 0, 3, 1).$$

Here we have used 7-tuples over $GF(11)$ to show elements of $GF(11)^7$. The TA sends u_3 's children, u_5, u_6, u_7 , the respective shared key, $f_3(5), f_3(6), f_3(7)$:

$$f_3(5) = (0, 3, 6, 2, 3, 3, 5) = k_{3,5},$$

$$f_3(6) = (1, 5, 9, 6, 8, 9, 1) = k_{3,6},$$

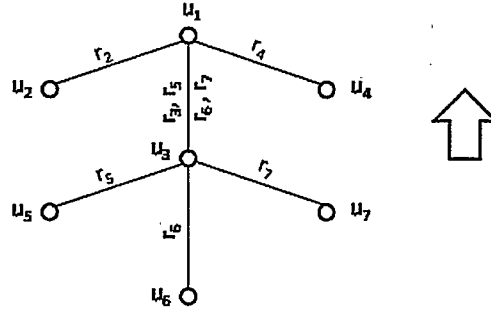


Figure 5.2: Leaf-to-root phase of the conference key computation

$$f_3(7) = (6, 4, 7, 4, 2, 6, 2) = k_{3,7}.$$

Note that u_3 also receives a shared key, $f_1(3) \in (GF(11))^7$, from the TA to communicate with his parent node, u_1 .

In the first phase of conference key computation, starting from the leaf nodes, each user, u_i , randomly chooses a value $r_i \in GF(11)$, encrypts and sends it together with the encrypted version of all the random values he has received from his children, to his parent node.

In the second phase of conference key computation, starting from the root node, u_1 , every user, u_i , sends his children, u_j , the encrypted version of his random value and all the random values he possess and did not receive from u_j .

Now consider the messages that are sent between u_3 and u_5 . In the first phase, u_5 sends u_3 :

$$m_{5,3}^{(1)} = r_5 \oplus k_{5,3}^{(1)} \rightarrow m_{5,3}^{(1)} = r_5 \oplus 0.$$

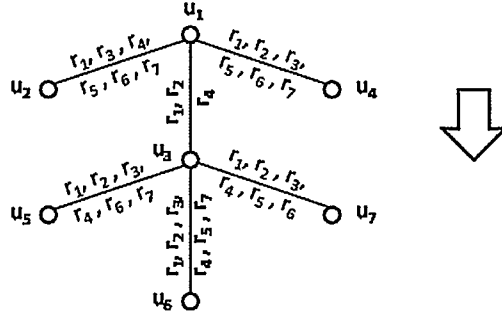


Figure 5.3: Root-to-leaf phase of the conference key computation

In the second round, u_3 sends the following messages to u_5 :

$$\begin{aligned} m_{3,5}^{(2)} &= r_1 \oplus 3, & m_{3,5}^{(3)} &= r_2 \oplus 6 \\ m_{3,5}^{(4)} &= r_3 \oplus 2, & m_{3,5}^{(5)} &= r_4 \oplus 3 \\ m_{3,5}^{(6)} &= r_6 \oplus 3, & m_{3,5}^{(7)} &= r_7 \oplus 5 \end{aligned}$$

The final conference key is $k_G = r_1 || \dots || r_7$.

5.4 Security of the scheme

In this section we show that the scheme in Section 5.3 satisfies the requirements of a 1-restricted m -tree (g, b) -CKA scheme. That is, the requirements of Definition 12 and 23 are satisfied.

1. *Interactive property*: In order to show that the scheme of Section 5.3 satisfies the interactive property, Equation (2.11), we need to show that:

$$H(\mathbf{K}_G | \mathbf{U}_i) = H(\mathbf{K}_G), \quad \forall i \in G \subseteq \mathcal{U}, \quad |G| = g.$$

This is indeed true because the conference key, $k_G = r_{i1} || \dots || r_{ig}$ with $G = \{u_{i1}, \dots, u_{ig}\}$ and r_{ij} is the random value chosen by user $u_{ij} \in G$, and so conference members cannot compute the conference key without interacting with other members.

2. *Correctness*: To show that the new scheme in Section 5.3, satisfies the correctness condition, Equation (2.12), we need to show that:

$$H(\mathbf{K}_G | \mathbf{U}_i \mathbf{M}_G) = 0, \quad \forall i \in G \subseteq \mathcal{U}, |G| = g.$$

According to the description of the scheme, any conference member is capable of uniquely computing the conference key using his user key and the communicated messages. This confirms the *correctness* of our scheme.

3. *Perfect secrecy*: To prove that the scheme is perfectly secure, Equation (2.13), we need to show that:

$$H(\mathbf{K}_G | \mathbf{U}_A \mathbf{M}) = H(\mathbf{K}_G),$$

for $A \cap G = \emptyset$, $|A| = b$ and $\mathbf{M} = \bigcup_{G \subseteq \mathcal{U}, |G|=g} \mathbf{M}_G$. Our scheme is designed for computing one conference key and so $\mathbf{M} = \mathbf{M}_G$, where G is an arbitrary conference with $|G| = g$.

To prove the perfect secrecy of the scheme we argue that the adversary's knowledge consists of two components: (a) the messages that are communicated during the execution of the protocol, and (b) the user key of the corrupted users. We then show that these components are statistically independent from the conference key.

To show (a), note that according to Remark 17, every message is encrypted using a distinct key, hence one-time-pads. Also, from Theorem 1 we conclude that the encrypted messages are perfectly secure and do not leak any information. So from Corollary 1, we can write:

$$\begin{aligned} H(\mathbf{M}_G | \mathbf{U}_A) &= H(\mathbf{M}_G), \\ H(\mathbf{M}_G | \mathbf{K}_G, \mathbf{U}_A) &= H(\mathbf{M}_G). \end{aligned} \tag{5.1}$$

To show (b), note that k_G is built by concatenating fresh random values that the conference members share during the conference key computation phase, whereas the TA gives u_A to members of A at the initialization phase. This means that the two random variables, \mathbf{K}_G and

\mathbf{U}_A , are statistically independent. From Corollary 1 we can write:

$$H(\mathbf{K}_G|\mathbf{U}_A) = H(\mathbf{K}_G). \quad (5.2)$$

To complete the proof, we remark that from Corollary 2, for two random variables \mathbf{X} and \mathbf{Y} , we have:

$$H(\mathbf{X}, \mathbf{Y}) = H(\mathbf{X}) + H(\mathbf{Y}|\mathbf{X}).$$

This can be extended to three random variables, \mathbf{X} , \mathbf{Y} and \mathbf{Z} :

$$\begin{aligned} H(\mathbf{X}, \mathbf{Y}, \mathbf{Z}) &= H(\mathbf{X}, \mathbf{Y}) + H(\mathbf{Z}|\mathbf{X}, \mathbf{Y}) \\ &= H(\mathbf{X}) + H(\mathbf{Y}|\mathbf{X}) + H(\mathbf{Z}|\mathbf{X}, \mathbf{Y}). \end{aligned} \quad (5.3)$$

Using Equation (5.3), we can write:

$$H(\mathbf{K}_G, \mathbf{U}_A, \mathbf{M}) = H(\mathbf{K}_G) + H(\mathbf{U}_A|\mathbf{K}_G) + H(\mathbf{M}|\mathbf{K}_G, \mathbf{U}_A), \quad (5.4)$$

or:

$$H(\mathbf{K}_G, \mathbf{U}_A, \mathbf{M}) = H(\mathbf{U}_A) + H(\mathbf{M}|\mathbf{U}_A) + H(\mathbf{K}_G|\mathbf{U}_A, \mathbf{M}). \quad (5.5)$$

By equating the right hand sides of Equation (5.4) and (5.5), we have:

$$H(\mathbf{K}_G|\mathbf{U}_A, \mathbf{M}) = H(\mathbf{K}_G) + H(\mathbf{U}_A|\mathbf{K}_G) + H(\mathbf{M}|\mathbf{K}_G, \mathbf{U}_A) - H(\mathbf{U}_A) - H(\mathbf{M}|\mathbf{U}_A). \quad (5.6)$$

From Equation (5.1) and (5.2), we have:

$$\begin{aligned} H(\mathbf{K}_G|\mathbf{U}_A, \mathbf{M}) &= H(\mathbf{K}_G) + H(\mathbf{U}_A) + H(\mathbf{M}) - H(\mathbf{U}_A) - H(\mathbf{M}) \\ &= H(\mathbf{K}_G). \end{aligned} \quad (5.7)$$

This completes the argument regarding the perfect secrecy of our new scheme. On the other hand, we just showed that our scheme satisfies the conditions of a 1-restricted (g, b) -CKA scheme. Since the communication model is based on an m -balanced tree, we conclude that our new scheme has all the conditions to be an m -tree (g, b) -CKA scheme.

5.5 Performance

In this section we compute the efficiency of our scheme and compare the result with the $(g, b) - 1RD$ scheme of Blundo et al, when applied on a communication graph. We show that for certain parameter values, our scheme has better key rate and it always achieves a better communication rate.

As mentioned in Section 5.2, we assume that the communication tree is symmetric with respect to the root node and is complete in the sense that the length of the path from any leaf node to the root is d . So, there are m^d leaf nodes and a total of $n = \sum_{i=0}^d m^i = \frac{m^{d+1}-1}{m-1}$ users. To specify a random polynomial, $f(x)$, of degree $\min\{b, m-1\}$, exactly $(\min\{b, m-1\} + 1)$ coefficients are needed. Hence TA sends to every user u_i , except the root and the leaf nodes, $(\min\{b, m-1\} + 2)$ random values from $(GF(p^k))^g$. This includes a polynomial to communicate with their children and one separate key to communicate with their parent. The TA also sends $(\min\{b, m-1\} + 1)$ random values from $(GF(p^k))^g$ to the root node and one such random value to each leaf node. So the TA distributes a total of:

$$\begin{aligned} |U_{\mathcal{U}}| &= (m^d + (\min\{b, m-1\} + 1) + (\min\{b, m-1\} + 2)(n - (m^d - 1)))gk \log_2 p \\ &= (m^d + (\min\{b, m-1\} + 1) + (\min\{b, m-1\} + 2)\frac{m^d - m}{m-1})gk \log_2 p, \end{aligned} \quad (5.8)$$

bits of user keys among all the users. On the other hand, a typical conference key in this scheme consists of g random values from $GF(p^k)$, and:

$$|K_G| = g k \log_2 p. \quad (5.9)$$

Hence for the key rate of our scheme we have:

$$\frac{|U_{\mathcal{U}}|^{(T)}}{|K_G|} = m^d + (\min\{b, m-1\} + 1) + (\min\{b, m-1\} + 2)\frac{m^d - m}{m-1}. \quad (5.10)$$

To compute the total number of transmitted messages, we count the total number of messages that pass through each edge multiplied by the number of edges. We know that for a tree $T = (V, G)$

with $|V| = g$, we have $|E| = g - 1$. From Remark 17, we obtain:

$$|M_G| = g \times (g - 1) k \log_2 p.$$

Hence the communication rate of our scheme is:

$$\frac{|M_G|^{(T)}}{|K_G|} = g - 1. \quad (5.11)$$

By an index (T) we refer to the efficiency ratios of our scheme.

5.6 Comparison

We first compute the minimum value for the communication rate of the $(g, b) - 1RD$ scheme, when adapted to the tree based communication model rather than broadcast model. Next we prove a lemma that gives bounds on the key rate of the $(g, b) - 1RD$ scheme.

5.6.1 Technical lemma

To compute the communication rate of the $(g, b) - 1RD$ scheme, we note that this protocol uses broadcast channels for communication. This means that a sent message is directly received by all other users. In settings where communication is confined to point-to-point channels, such as a tree, nodes may need to forward messages to other nodes in order to allow certain nodes to communicate. Since in the broadcast model, every message gets delivered to all the other users, we think of this as every user emitting $g - 1$ messages, one for every user in G . Similarly, each user expects to receive $g - 1$ messages from the other users. Focusing on the leaf nodes only, this adds up to $2(g - 1)$ messages to travel through the edge that connects any leaf node to his parent node.

Having an m -balanced tree as the communication graph, we argue that the minimum value for communication rate of the $(g, b) - 1RD$ scheme is obtained for the case where G consists of a

node, u_i , together with some of u_i 's children, u_{i1}, \dots, u_{ig-1} . That is, we assure that every leaf node's message gets delivered to its destination via a path of length at most 2, which is the minimum length compared to conferences with $d > 2$. Now, since there are $g - 1$ leaf nodes and $g - 1$ edges such that $2(g - 1)$ message travel through each, we obtain:

$$|M_G| = 2(g - 1)^2 \chi r k \log_2 p. \quad (5.12)$$

From Section 3.3.2, we know:

$$|K_G| = g r k \log_2 p, \quad (5.13)$$

so the communication rate is:

$$\frac{|M_G|^{(1RD)}}{|K_G|} = \frac{2(g - 1)^2 \chi}{g}, \quad (5.14)$$

where $\chi = \frac{g-1}{\ell-1}$. We remark that the key rate of the $(g, b) - 1RD$ scheme is independent of its communication model and hence, according to Equation 3.18:

$$\frac{|U_{\mathcal{U}}|^{(1RD)}}{|K_G|} = n \frac{\ell \binom{n-1}{\ell-1}}{g \binom{g-2}{\ell-2}}.$$

In the following lemma, we give a lower and upper bound on the key efficiency of the $(g, b) - 1RD$ scheme.

Lemma 6. *In the 1-restricted (g, b) -CKA scheme of Blundo et al.:*

$$\frac{1}{2} \frac{|U_{\mathcal{U}}|^{(1RD)}}{|K_G|} \Big|_{\ell=2} < \frac{|U_{\mathcal{U}}|^{(1RD)}}{|K_G|} \leq \frac{|U_{\mathcal{U}}|^{(1RD)}}{|K_G|} \Big|_{\ell=2}.$$

Proof. From Equation (3.18) we have:

$$\frac{|U_i|^{(1RD)}}{|K_G|} = \frac{\ell \binom{n-1}{\ell-1}}{g \binom{g-2}{\ell-2}}.$$

Note that:

$$\frac{|U_i|^{(1RD)}}{|K_G|} = \frac{\ell \binom{n-1}{\ell-1}}{g \binom{g-2}{\ell-2}} = \frac{\ell (n-1)! (\ell-2)! (g-\ell)!}{g (g-2)! (\ell-1)! (n-\ell)!} = \frac{(n-1)!}{g (g-2)!} \frac{\ell}{\ell-1} \frac{(g-\ell)!}{(n-\ell)!}$$

We know that $\ell \leq g \leq n \rightarrow n - \ell > g - \ell$ and we can write $\frac{(g-\ell)!}{(n-\ell)!}$ as:

$$\frac{(g-\ell)!}{(n-\ell)(n-\ell-1)\dots(n-(n-g+\ell)+1)(n-(n-g+\ell))!} = \frac{1}{(n-\ell)\dots(g-\ell+1)}$$

Let $f(\ell) = \frac{1}{(n-\ell)\dots(g-\ell+1)}$. We argue that $f(\ell) \geq f(2)$ because for $\ell = 2$, the terms in the denominator are larger in value and hence the whole fraction takes its smallest value. So:

$$\frac{|U_i|}{|K_G|}^{(1RD)} = \frac{(n-1)!}{g(g-2)!} \times \frac{\ell}{\ell-1} \times f(\ell) \geq \frac{(n-1)!}{g(g-2)!} \times \frac{\ell}{\ell-1} \times f(2).$$

Let us now simplify the terms in the right hand side of the inequality:

$$\frac{(n-1)!}{g(g-2)!} \times \frac{\ell}{\ell-1} \times f(2) = \frac{\ell}{\ell-1} \times \frac{(n-1)(n-2)!}{g(g-2)!} \times \frac{1}{(n-2)\dots(g-1)} = \frac{\ell}{\ell-1} \times \frac{n-1}{g}.$$

So we can write:

$$\frac{\ell}{\ell-1} \times \frac{n-1}{g} \leq \frac{|U_i|}{|K_G|}^{(1RD)}. \quad (5.15)$$

On the other hand we know that for a fixed $g \geq 2$:

$$\min \left\{ \frac{|U_i|}{|K_G|}^{(1RD)}, 2 \leq \ell \leq g \right\} \leq \left. \frac{|U_i|}{|K_G|}^{(1RD)} \right|_{\ell=2} = \frac{2(n-1)}{g}. \quad (5.16)$$

From Equation (5.15) and Equation (5.16) we conclude that:

$$\frac{\ell}{\ell-1} \times \frac{n-1}{g} \leq \frac{|U_i|}{|K_G|}^{(1RD)} \leq \frac{2(n-1)}{g}. \quad (5.17)$$

Also note that $\frac{\ell}{\ell-1}$ is a uniformly decreasing series for $\ell \geq 2$. On the other hand, we remark that although $2 \leq \ell \leq g$, the value of ℓ that minimizes $\frac{|U_i|}{|K_G|}^{(1RD)}$ must happen for some $2 \leq \ell \leq \frac{g}{2}$. So $\frac{g}{g-2} \leq \frac{\ell}{\ell-1} \leq 2$ and:

$$\frac{n-1}{g-2} \leq \frac{\ell}{\ell-1} \times \frac{n-1}{g}. \quad (5.18)$$

For $g > 2$, $\frac{n-1}{g-1} < \frac{n-1}{g-2}$, and from Equation (5.17) we conclude:

$$\frac{n-1}{g} < \frac{|U_i|}{|K_G|}^{(1RD)} \leq \frac{2(n-1)}{g}, \quad (5.19)$$

$$\frac{1}{2} \left. \frac{|U_i|}{|K_G|} \right|_{\ell=2}^{(1RD)} < \frac{|U_i|}{|K_G|}^{(1RD)} \leq \left. \frac{|U_i|}{|K_G|} \right|_{\ell=2}^{(1RD)}.$$

Since in this scheme all users receive the same amount of secret information from TA, $|U_{\mathcal{U}}| = n \times |U_i|$. Multiplying all terms in Equation (5.19) by n preserves the direction of the inequalities and completes the proof. \square

Remark 18. Note that Equation (5.18) gives a tighter lower bound of $\frac{n(n-1)}{g-2}$ for $\frac{|U_i|}{|K_G|}^{(1RD)}$ than $\frac{n(n-1)}{g}$.

5.6.2 1-restricted m -tree (g, b) -CKA scheme vs. $(g, b) - 1RD$ scheme

In this section we compare the key rate and communication rate of our scheme, with the $(g, b) - 1RD$ scheme.

Key rate

For $m \leq b$ we compute:

$$\frac{|U_{\mathcal{U}}|}{|K_G|}^{(T)} = (m^d + m) + (m+1) \frac{m^d - m}{m-1}.$$

On the other hand from Equation (3.19), for the $(g, b) - 1RD$ scheme we computed:

$$\frac{|U_{\mathcal{U}}|}{|K_G|}^{(1RD)} = n \times \frac{\ell \binom{n-1}{\ell-1}}{gr},$$

where $g-1 \equiv 0 \pmod{\ell-1}$ and $r = \binom{g-2}{\ell-2}$. From Lemma 6 we know that:

$$\frac{1}{2} \frac{|U_{\mathcal{U}}|}{|K_G|} \Big|_{\ell=2}^{(1RD)} < \frac{|U_{\mathcal{U}}|}{|K_G|}^{(1RD)} \leq \frac{|U_{\mathcal{U}}|}{|K_G|} \Big|_{\ell=2}^{(1RD)}$$

and from Equation 3.18 we compute:

$$\frac{1}{2} \frac{|U_{\mathcal{U}}|}{|K_G|} \Big|_{\ell=2}^{(1RD)} = n \times \frac{n-1}{g},$$

with $n = \frac{m^{d+1}-1}{m-1}$. To find the cases where our scheme achieves better key rate than the $(g, b) - 1RD$

scheme, we considered the cases for which:

$$\begin{aligned}
\frac{|U_{\mathcal{U}}|^{(T)}}{|K_G|} &\leq \frac{1}{2} \frac{|U_{\mathcal{U}}|^{(1RD)}}{|K_G|} \Big|_{\ell=2} \\
(m^d + m) + (m+1) \frac{m^d - m}{m-1} &\leq n \times \frac{n-1}{g} \\
(m^{d-1} + 1)(m-1) + (m+1)(m^{d-1} - 1) &\leq n \times \frac{m^d - 1}{g} \\
g &\leq \frac{n}{2}.
\end{aligned} \tag{5.20}$$

So for all conferences of size at most $\frac{n}{2}$, our scheme achieves better key rate than the $(g, b) - 1RD$ scheme.

However, for $b < m$, the key rate of our scheme becomes

$$\frac{|U_{\mathcal{U}}|^{(T)}}{|K_G|} = m^d + (b+1) + \frac{(b+2)(m^d - m)}{m-1}.$$

Our attempts to analytically compare the key rates for this case did not result in a short or more intuitive expression.

Communication rate

From Equation (5.11) and (5.14) we have:

$$\begin{aligned}
\frac{|M_G|^{(T)}}{|K_G|} &\leq \frac{|M_G|^{(1RD)}}{|K_G|} \\
g-1 &\leq \frac{2(g-1)^2 \chi}{g}, \\
g &\leq 2(g-1)\chi.
\end{aligned} \tag{5.21}$$

Since $\chi \geq 1$, we conclude that for all $g \geq 2$, the inequality above holds.

5.7 Simulation

Since our attempts to conclude the comparison for the case where $b < m$ did not result in a short form or more intuitive expression and also as mentioned in Remark 18, there are tighter lower bounds than which we based our conclusion on, we implemented the key rate of our 1-restricted m -balanced (g, b) -CKA scheme and the key rate of $(g, b) - 1RD$ in Mathematica. This code can be

used to simulate the key rates for different parameter values. We remark that once m, d and g are given, the rest of the parameters can uniquely be computed. The code is provided in Appendix B.

We ran our code for several cases. Here we present the case where $(m, d, n) = (3, 5, 364)$ with $2 \leq g \leq n$. The horizontal axis represent the possible conference sizes, $2 \leq g \leq n$, and the vertical axis represents the key rate. The blue curve shows the key rate for the $(g, b) - 1RD$ scheme and the red curve shows the key rate of our scheme.

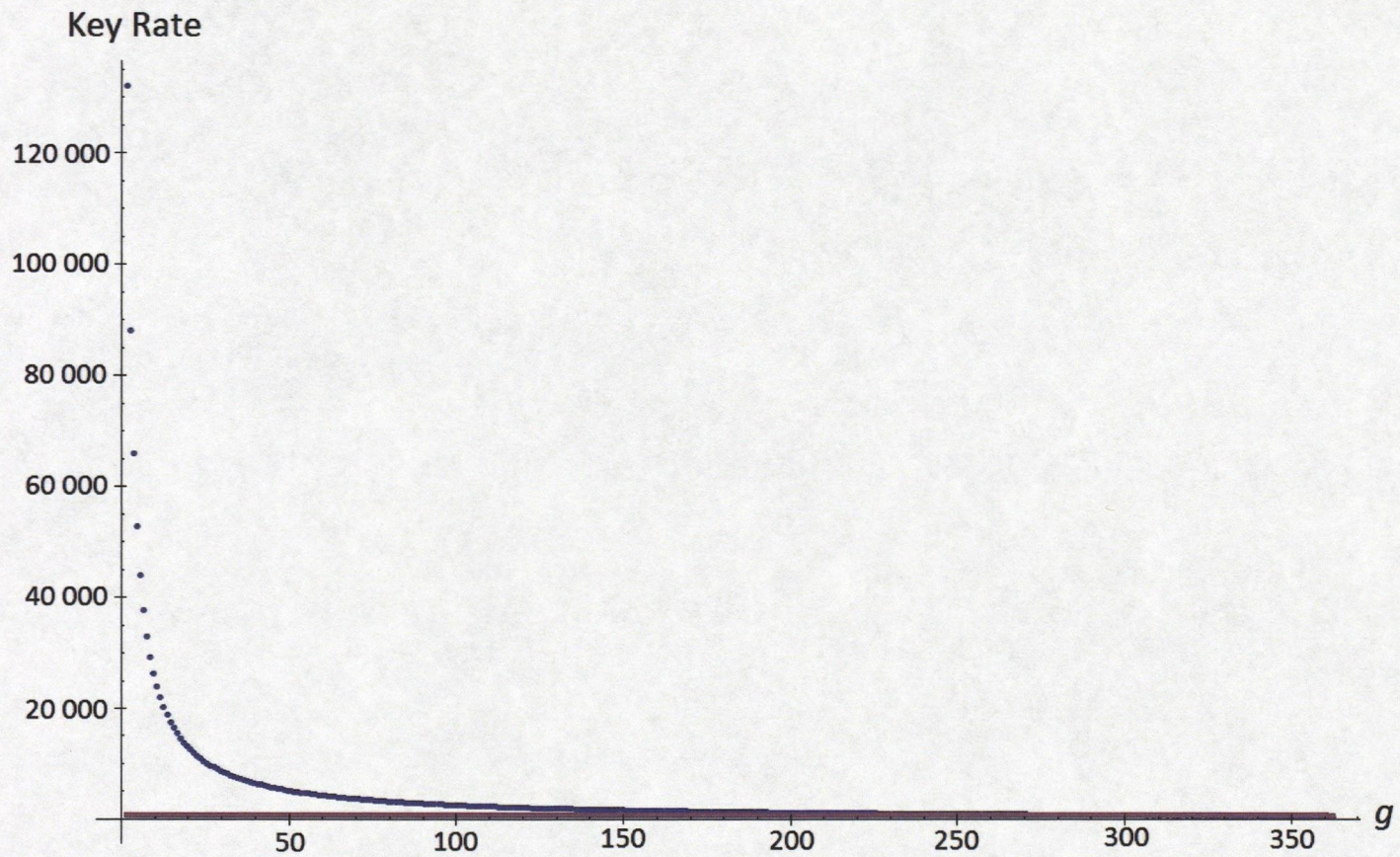


Figure 5.4: 1-restricted m -tree (g, b) -CKA scheme's key rate vs. (g, b) - $1RD$ scheme's

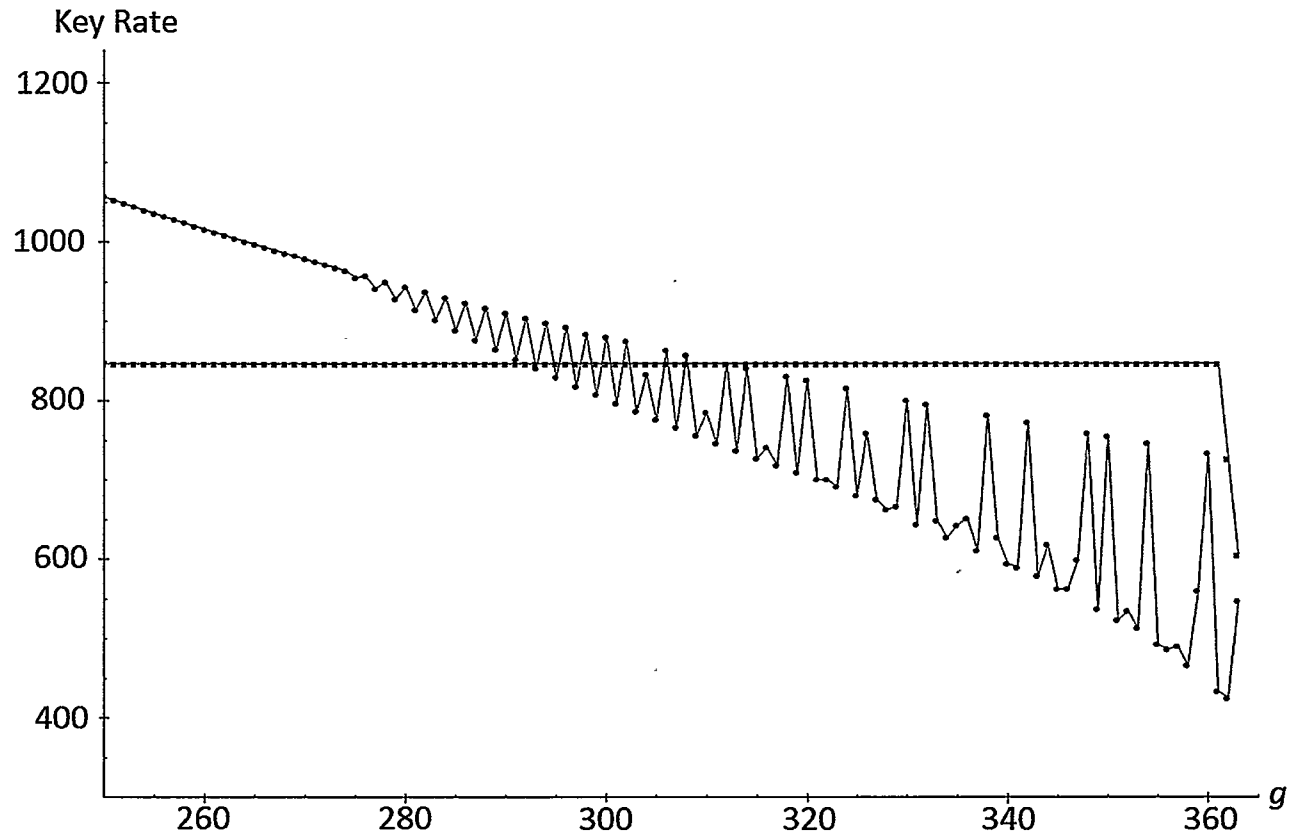


Figure 5.5: Cross over section of the key rates for 1-restricted m -tree (g, b) -CKA scheme and the $(g, b) - 1RD$ scheme

Remark 19. *We remark that this simulation is of importance in at least two perspectives:*

1. *The simulation and the analytic result in Equation (5.20) are compatible. That is, for $g \leq \frac{n}{2} = 182$, our scheme achieves lower key rate. Note that for $g \leq \frac{n}{2}$, $b = n - g \leq \frac{n}{2} = 182$, and hence $m = 3 < b$, which is the assumption we made for our analytical comparison of the two efficiency measures.*
2. *The fluctuating section in Figure 5.5 shows how the two key rate expressions are not comparable, in general.*

5.8 Conclusions

In this chapter, we studied the problem of CKD for networks of users where not all nodes can directly communicate with each other. We presented an interactive key agreement scheme for conferences that have an m -balanced tree as their communication graph. We showed that our new 1-restricted (g, b) -CKA scheme always attains better communication complexity than the 1-restricted (g, b) -CKA scheme of Blundo et al. and for certain parameter values, it achieves better key efficiency.

As a possible extension of this work, it would be interesting to study ways of having tree structured CKA schemes for computing multiple conference keys. Moreover, note that our scheme is based on spanning trees within a conference communication graph. We remark that a communication graph often has more edges than its spanning tree does. This means more communication channels and hence a potential for having other CKA schemes based on the communication graph model with better performances.

Chapter 6

Conclusions and Future Work

The main focus of this thesis was on the study of CKD schemes and design of new schemes to better accommodate the real life situations.

The contributions of this thesis are: (i) extending the work of Blundo et al. to compute τ conference keys for conferences of varying sizes, $\mathcal{G} = (g_1, \dots, g_\tau)$, and (ii) designing a new CKA scheme for conferences that have tree structured communication graphs.

We proposed three schemes to extend the τ -restricted (g, b) -CKA scheme of Blundo et al. To evaluate and compare the performance of the proposed schemes, we developed a code using Mathematica 7 to calculate the key rates and communication rates of these schemes. By initializing the parameters, the code is compatible to evaluate and compare the efficiency measures of any instance of such schemes. We concluded that our first proposed scheme in which a 1-restricted (g_i, b_i) -CKA scheme of Blundo et al. is used for each $g_i \in \mathcal{G}$, has the best key rate and communication rate, compared to the other schemes.

We questioned the suitability of broadcast channels to model the real life communication settings. By the hint of some data obtained from Gnutella peer to peer file sharing networks, we choose to take communication graphs as a more accurate model of communication infrastructure. We introduced a new 1-restricted (g, b) -CKA scheme that takes an m -balanced tree as its communication graph and enables the users to compute one conference key, securely. We compared the performance of this new scheme with a modified version of the 1-restricted (g, b) -CKA scheme of Blundo et al. This is the most efficient CKA scheme that has been discussed in the thesis and the modification was to adjust the $(g, b) - 1RD$ scheme to the communication graph model. We showed that for certain parameter values, our scheme achieves a lower key rate than the modified

$(g, b) - 1RD$ scheme while our new scheme always achieves a lower communication rate.

6.1 Future Work

There are a number of interesting problems and directions that came out from this study for future work. Some of these are outlined below.

1. In Chapter 4 we restricted the conference sizes to be an increasing list of numbers, i.e. in $\mathcal{G} = (g_1, \dots, g_\tau)$, $g_i \leq g_{i+1}$, $\forall 1 \leq i \leq \tau - 1$. It is interesting to study other cases to realize a $(\mathcal{G}, \mathcal{B})$ -CKA scheme without such restrictive conditions. In Chapter 4 we introduced a scheme that can be used for any \mathcal{G} . However, this may not be the most efficient scheme and future research to construct more efficient schemes is needed.
2. Another interesting problem is to find a general lower bounds, similar to [3], on the key rate of a $(\mathcal{G}, \mathcal{B})$ -CKA scheme. Similarly, finding a general bound on the key rate of an optimal CKD scheme with communication graph model, is a problem that have not been studied yet.
3. Our 1-restricted m -tree (g, b) -CKA scheme in Chapter 5 is perfectly secure to compute *one* conference key. It is interesting to study the extension of our construction to compute τ conference keys. This itself can be seen as having τ conferences of the same or varying sizes.
4. In our scheme of Chapter 5, we made the most basic assumption on the communication graph, i.e. being connected. This implies a spanning tree within the communication graph. However, according to the data from [2], real life communication graphs are rather populated. This opens a number of interesting options to design a CKD scheme based on communication graphs. For instance one might assume that between every two nodes, there exists at least $t \geq 1$ paths. Another assumption can be working with special class of graphs

such as bipartite or cluster graphs. Some other characteristics of graphs might also become useful for designing new schemes.

5. Another realistic assumption over the communication graphs is that some of the edges can be secured edges. That is, only a number of designated users in the network have access to communication over these edges. This defines a different area for research where our setting would become a special case of it, where the number of such secure channels is 0.
6. Finally, we remark that in certain situations, the eligibility of a conference is based on other characteristics of the users rather than the size of the conference. For instance, in an institute with designated roles and titles, a conference may consists of two users of *manager* level or a manager can be replaced with two users of *admin* level or an admin can be replaced by two *staffs*. Designing CKD schemes for not uniformly privileged users, and considering eligibility conditions other than the conference size, would be interesting extensions to this work.

Appendix A

Here we present a sample of the Mathematica code we produced to compare the performance of the proposed schemes in Chapter 4.

A.1 Key rates

A.1.1 Graphic presentation

k0: 7

```

l[1] = 1 /. Last[Minimize[{1*Binomial[n - 1, 1 - 1]/
  (g[1]*Binomial[g[1] - 2, 1 - 2])},
  2 <= 1 <= g[1], 1 \[Element] Integers}, 1]];
For[i = 2, i <= tau, i++,
  l[i] = 1 /. Last[Minimize[{1*Binomial[n - 1, 1 - 1]/
    (g[i]*Binomial[g[i] - 2, 1 - 2])},
    2 <= 1 <= g[i], 1 \[Element] Integers}, 1]];
  f[i] = f /. Last[Minimize[{f, Mod[g[i] + f - 1, l[i - 1] - 1] == 0,
    0 <= f <= l[i - 1]}, f]];
r1[i_] := Binomial[g[i] - 2, l[i] - 2];
r2[i_] := Binomial[g[i] + f[i] - 2, l[i - 1] - 2];

(*Scheme 1*)
(*s[i]'s represents the field sizes*)
s[1] := k[1];
s[j_] := If[Floor[g[1] r1[1] k[1]/(g[j] r1[j])] >= k0,
  Floor[g[1] r1[1] k[1]/(g[j] r1[j])], k0];

```

```

Eff1[1] := l[1] Binomial[n - 1, l[1] - 1]/g[1]*r1[1];
Eff1[j_] := Eff1[j - 1] + l[j] Binomial[n - 1, l[j] - 1]
s[j]/(g[j] r1[j] k[1]);

(*Graphs*)
ListLinePlot[{Table[Eff1[j], {j, 1, tau}]], PlotLabel ->
"Key rate of Schemes 1",
AxesLabel -> {"i-th conference", "Key Rate"}, PlotRange -> All,
AxesOrigin -> {0, 0}, PlotMarkers -> {"1"}]

```

A.1.2 Comparing key rates of Scheme 1 and 2

This piece of code is meant to compare $\binom{n-1}{\ell_{0i}-1}k_i^{(1)}$ and $\binom{n}{\ell_{0i}-1}k_i^{(2)}$ which appear in the key rate of Scheme 1 and Scheme 2, respectively:

```

Print["Scheme 1 <? Scheme 2: "];
Print["k^{1}[i] Binomial[n-1, l[i]-1]< k^{2}[i] Binomial[n, l[i]-1]"];
For[i = 1, i <= tau, i++,
Print["tau= ", i, ", ", N[s[i]*Binomial[n - 1, l[i] - 1]] , " < ",
N[k[i]*Binomial[n, l[i] - 1]]]];

```

A.1.3 Comparing key rates of Scheme 2 and 3

This piece of code is meant to compare $\binom{n}{\ell_{0i}-1}\frac{\ell_{0i}k_i^{(2)}}{r_1}$ with $2nk_i^{(3)}$ which appear in the key rate of Scheme 2 and Scheme 3, respectively:

```

Print["Scheme 3 <? Scheme 2: "];
Print["2*k^{3}*n < l[i]*k^{2}[i]*Binomial[n, l[i]-1]/r[1]"];

```

```

For[i = 1, i <= tau, i++,
  Print["tau= ", i, ", ", N[2*m[i]*n] , " < ",
    N[l[i]*k[i]*Binomial[n, l[i] - 1]/r1[1]]];

```

A.2 Communication rates

```

chi1[i_] := (g[i] - 1)/(l[i] - 1);
chi2[i_] := (g[i] + f[i] - 1)/(l[i - 1] - 1);

(*Scheme 1*)
ComRate1[j_] := N[chi1[j]];

(*Scheme 2*)
(*J represents I in the actual scheme *)
(*k[i]'s represent the field sizes*)

J[1] = g[1] r1[1] - l[1] r2[2];
J[i_] := g[1] r1[1] - l[i] r2[i + 1] J[i - 1]/(g[i] r1[i]);
k[i_] := If[Floor[k[1]*J[i - 1]/(g[1] r1[1])] >= k0,
  Floor[k[1] J[i - 1]/(g[1] r1[1])], k0];

ComRate2[1] := N[chi1[1]];
ComRate2[j_] :=
  N[(g[j] r1[j] chi1[j] k[j] + l[j - 1] r2[j] chi2[j] k[j - 1] -
    1)/(g[1] r1[1] k[1])];

(*Graphs*)

```

```
ListLinePlot[{Table[ComRate1[j], {j, 1, tau}]}],  
  PlotLabel -> "Communication rate of Schemes 1",  
  AxesLabel -> {"i-th conference", "Communication Rate"},  
  PlotRange -> All, AxesOrigin -> {0, 0}, PlotMarkers -> {"1"} ]
```

Appendix B

We present the code we ran on Mathematica 7 to compare the key rate of the 1-restricted m -balanced (g, b) -CKA scheme with the $(g, b) - 1RD$ scheme when modified for a communication tree structure.

B.1 Key rate

```

m = 3 ; d = 5;
n = (m^(d + 1) - 1)/(m - 1);
Print["(m, d, n) = (", m, ", ", d, ", ", n, ")"];

dataBlundo =
  Table[{g,
    N[MinValue[{n (1*Binomial[n - 1, 1 - 1])/(g*
      Binomial[g - 2, 1 - 2])},
    Mod[g - 1, 1 - 1] == 0 && 2 <= 1 <= g &&
    Element[1, Integers]]}, {1}, WorkingPrecision -> 10]]], {g, 2,
  n - 1}];

dataTree =
  Table[{g,
    m^d + Min[m, n - g] +
    1 + ((Min[m, n - g] + 2) (m^d - m))/(m - 1)}, {g, 2, n - 1}];

```



```

ListPlot[{dataBlundo, dataTree}, AxesOrigin -> {0, 0},
  PlotRange -> All, AxesLabel -> {"g", "Key Rate"},
  Joined -> {False, False},
  PlotLabel ->
    "Key rate of the 1-restricted (g, b)-CKA scheme of Blundo et al. vs \
the 1-restricted m-tree (g,b)-CKA Scheme"]
ListPlot[{dataBlundo, dataTree}, AxesOrigin -> {250, 300},
  PlotRange -> {{250, 365}, {300, 1200}},
  AxesLabel -> {"g", "Key Rate"}, Axes -> {True, True},
  Joined -> {True, True}, PlotMarkers -> {Automatic, Tiny},
  PlotLabel ->
    "Key rate of the 1-restricted (g, b)-CKA scheme of Blundo et al. vs \
the 1-restricted m-tree (g,b)-CKA Scheme"]

```

Bibliography

- [1] Conditional Entropy, Joint Entropy and Mutual Information Figure. http://en.wikipedia.org/wiki/Mutual_information.
- [2] The SNAP Project. <http://snap.stanford.edu/data/index.html>.
- [3] Amos Beimel and Benny Chor. Communication in key distribution schemes. *IEEE Transactions on Information Theory*, 42:pp., 1996.
- [4] R Blom. An optimal class of symmetric key generation systems. In *Proc. of the EURO-CRYPT 84 workshop on Advances in cryptology: theory and application of cryptographic techniques*, pages 335–338, New York, NY, USA, 1985. Springer-Verlag New York, Inc.
- [5] Carlo Blundo, Paolo D’Arco, and Antonio Giorgio Gaggia. A tau-restricted key agreement scheme. *The Computer Journal*, 42(1):51–61, 1999.
- [6] Carlo Blundo, Alfredo De Santis, Amir Herzberg, Shay Kuten, Ugo Vaccaro, and Moti Yung. Perfectly-secure key distribution for dynamic conferences. In *Proceedings of the 12th Annual International Cryptology Conference on Advances in Cryptology, CRYPTO ’92*, pages 471–486, London, UK, 1993. Springer-Verlag.
- [7] Ronald Cramer. Multiparty computation, an introduction. In *CPT, Lecture Notes, DAIMI*, 2002.
- [8] Amos Fiat and Moni Naor. Broadcast encryption. In *Proceedings of the 13th Annual International Cryptology Conference on Advances in Cryptology, CRYPTO ’93*, pages 480–491, London, UK, 1994. Springer-Verlag.
- [9] Ronald L. Graham, Martin Grtschel, and Lszl Lovsz. *Handbook of Combinatorics*. Elsevier, 1995.

- [10] Nancy A. Lynch. *Distributed Algorithms*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1996.
- [11] Alfred J. Menezes, Scott A. Vanstone, and Paul C. Van Oorschot. *Handbook of Applied Cryptography*. CRC Press, Inc., Boca Raton, FL, USA, 1st edition, 1996.
- [12] Fazlollah M. Reza. *An Introduction to Information Theory*. Dover Publications, September 1994.
- [13] Adi Shamir. How to share a secret. *Commun. ACM*, 22:612–613, November 1979.
- [14] Peter W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM J. on Computing*, pages 1484–1509, 1997.
- [15] Doug R. Stinson. On some methods for unconditionally secure key distribution and broadcast encryption. *Des. Codes Cryptography*, 12:215–243, November 1997.
- [16] Douglas R. Stinson. *Cryptography: Theory and Practice, Second Edition*. Chapman & Hall/CRC, February 2002.
- [17] J. H. van Lint and R. M. Wilson. *A Course in Combinatorics*. Cambridge University Press, 1992.