Applications of Broadcast Encryption Schemes and Related Technical Mechanisms for Digital Rights Management of Multimedia Broadcasts

DISSERTATION

zur Erlangung des Grades eines
Doktor-Ingenieurs
der Fakultät für Elektrotechnik und Informationstechnik
an der Ruhr-Universität Bochum

vorgelegt von
Ulrich Greveler

Bochum, 2006

Dissertation eingereicht am: 10.05.2006
Tag der mündlichen Prüfung: 03.07.2006
Referent: Prof. Dr. rer. nat. Jörg Schwenk
Korreferent: Prof. Dr.-Ing. Christof Paar

Contents

Abstract

Solutions to the problem of transmitting data to a group of receivers in a way that only the correct sub-set of all possible receivers can decrypt the data are needed by a number of applications: Pay-TV, rights-controlled media, audio streaming, real-time business data, multicast communication are current examples, but there also off-line applications, particularly with regard to storage media for multimedia content (*e. g.,* CD, DVD). The well-established cryptographic solution to the problem is called *Broadcast Encryption* and this term was first coined by Fiat and Naor in 1993.

In the context of Pay-TV that had become a successful commercial model in the 1980ies a technological approach to the problem was developed and data formats were standardized. The known technical solutions use cryptographic mechanisms but do not apply a fully-fledged broadcast encryption scheme. The umbrella term referring to technologies in this area is *Conditional Access Systems* (CAS).

The more general term *Digital Rights Management* (DRM) refers to all technological measures that enforce the rights and restrictions on digital content as defined by the content owners (*e. g.,* copy or content protection, usage rights). Broadcast Encryption is the cryptographic tool regarding DRM for stateless receivers of unidirectional communication.

In this dissertation we focus on the problem how valuable digital multimedia content can be delivered by a global service while enforcing the digital rights as defined by the rights-holders. We consider a general service requiring payment for usage that could be a future global Pay-TV station but also any other provider of digital goods. A new type of broadcast encryption scheme is presented that aims to provide cryptographic mechanisms that are advantageous for the realization of DRM for multimedia content. The proposed schemes offer a tradeoff between parameters like key storage, transmission length etc. and the number of free-riders of a certain transmission.

Moreover, the technical framework of secure multimedia broadcast is analyzed and we propose a new concept of enforcing regional rights on broadcasted digital goods which is an important factor in DRM systems because the value of multimedia content is subject to the geographic region of the potential consumer.

Finally, we study the implementation issues around the proposed technical and cryptographical mechanism and the possible applications to offline media.

This work also contains an annex part that presents simulation results for realistic use cases. By doing this we can demonstrate that the proposed schemes can be used to meet requirements for securing content regarding pay-per-view broadcasts as well as media distribution. The data also indicate the environment parameters that let the proposed schemes be more efficient than established existing schemes.

Kurzdarstellung der Dissertation

Eine Reihe von Anwendungen benötigt Lösungen für das Problem, Daten an eine Empfängermenge in einer Weise zu übertragen, dass nur die spezifizierte Teilmenge der Empfänger diese entschlüsseln können: Pay-TV (Bezahlfernsehen), rechteverwaltete Medien, Audio Streaming, Echtzeitdaten und Multicast-Kommunikation sind aktuelle Beispiele. Darüber hinaus existieren nicht vernetzte Anwendungen, insbesondere im Hinblick auf Speichermedien für digitale Inhalte (CD, DVD). Eine gängige kryptographische Lösung für das Problem stellt *Broadcast Encryption* dar; der Begriff wurde von Fiat und Naor 1993 eingeführt.

Im Bereich Pay-TV, das bereits seit den achtzigern Jahren ein erfolgreiches Geschäftsmodell ist, wurde ein technologischer Ansatz gewählt und es wurden Datenformate standardisiert. Die bekannten technischen Lösungen verwenden kryptographische Mechanismen, enthalten aber kein vollständiges Broadcast-Encryption-System. Der Überbegriff für die in diesem Bereich verwendeten Technologien ist CAS (*Conditional Access System*).

Der allgemeinere Begriff DRM (Digitales Rechtemanagement) bezieht sich auf alle technologischen Maßnahmen, die Rechte und Einschränkungen, so wie die Rechte-Inhaber diese definierren, in Bezug auf digitale Inhalte durchsetzen (z. B. Kopierschutz, detaillierte Nutzungsrechte). Broadcast Encryption ist das kryptographische Hilfsmittel zur Durchsetzung digitaler Rechte bei zustandslosen Empfängern unidirektionaler

Kommunikation.

In dieser Dissertation konzentrieren wir uns auf die Fragestellung, wie hochwertige digitale Multimediainhalte bei gleichzeitiger Durchsetzung der digitalen Rechte über einen global verfügbaren Dienst übertragen werden können. Wir betrachten einen allgemeinen Dienst, der Bezahlung bei Nutzung vorsieht und sowohl für einen zukünftigen, globalen Pay-TV-Dienstleister stehen kann als auch für andere Anbieter digitaler Waren. Eine neue Variante des Broadcast Encryption wird vorgestellt, die auf die Bereitstellung kryptographischer Mechanismen für die Realisierung eines Multimedia-DRM abzielt. Die vorgeschlagenen kryptographischen Verfahren stellen einen gegenseitigen Ausgleich zwischen Parametern wie Schlüsselspeicherbedarf, Länge der Übertragung etc. auf der einen Seite und der Anzahl der geduldeten unberechtigten Empfänger einer Übertragung auf der anderen Seite bereit.

Darüberhinaus werden technische Rahmenbedingungen einer sicheren Übertragung multimedialer Inhalte analysiert und ein neues Konzept zur Durchsetzung der regionalen digitalen Rechte an übertragenen Inhalten bereitgestellt. Diese regionalen Rechte spielen eine herausgehobene Rolle für globale DRM-Lösungen, da der Wert bzw. Preis multimedialer Inhalte abhängig vom geographischen Ort der Konsumption ist.

Abschließend werden Implementierungsaspekte zu den vorgeschlagenen technischen und kryptographischen Mechanismen beleuchtet, und es werden Anwendungsmöglichkeiten für Speichermedien, deren Konsum ohne Netzzugang erfolgt, entwickelt.

Diese Arbeit enthält darüber hinaus einen Anhang mit Simulationsergebnissen für wirklichkeitsnahe Anwendungsfälle. Auf diese Weise können wir zeigen, dass die

vorgeschlagenen Verfahren genutzt werden können, um Anforderungen an den Schutz von Inhalten sowohl bei der Absicherung einzelner hochwertiger Ausstrahlungen als auch bei der Verteilung über Medien zu erfüllen. Die Daten geben desweiteren Hinweise, unter welchen Randbedingungen die vorgeschlagenen Verfahren effizienter als die etablierten, vorhandenen Verfahren sind.

Preface

The work presented in this dissertation would never have been possible without the help of many people that supported me throughout the last years.

First, and foremost my acknowledgments go to my advisor Prof. Dr. Jörg Schwenk for providing me with the chance to study broadcast encryption systems in the context of digital rights management and to pursue a PhD project on this topic. His guidance, patience and persistence helped me to remove the obstacles that I encountered throughout the research work conducted for this dissertation. I also enjoyed the joint work on teaching in the brand-new IT security course programs and on supervising students.

My heartfelt appreciation goes out to my fiancée, Daniela Thümer, and my family who never ceased to provide encouragement when it was needed most.

Special credits go to André Adelsbach who was the *senior* researcher among the PhD students and helped me and others a lot to write our first papers for an international conference. He is also the co-author of one paper that presented major results of the research work to the *DRMtics* conference in Sydney, Australia. I always enjoyed the productive discussions with him on a variety of research issues. I also appreciate the collaboration with the other colleagues at NDS group, particularly with Sebastian Gajek, Mark Manulis, Michael Psarros and Lijun Liao. Their stimulating suggestions and constructive criticism inspired and encouraged me. I thankfully acknowledge the input of Ahmad-Reza Sadeghi who read an early draft of my ideas regarding a new approach to broadcast encryption schemes and had valuable suggestions.

I also acknowledge the collaboration with Andreas Krügersen who partially implemented one of the proposed schemes for simulation purposes which helped me to find realistic parameters for the schemes.

Finally, I would like to mention the European Network of Excellence for Cryptology (ECRYPT[1]) that made it possible for me to finance travel and accommodation costs that are connected with presenting ideas at scientific conferences and the Horst-Görtz-Stiftung that greatly funded the research facilities at Bochum University regarding IT security and cryptography letting the NDS research group come to existence.

[1] ECRYPT is a 4-year network of excellence funded within the Information Societies Technology (IST) Programme of the European Commission's Sixth Framework Programme (FP6) under contract number IST-2002-507932.

List of Figures

List of Tables

List of Algorithms

1 Introduction

In the recent past we witnessed the quick emergence and growth of electronic commerce (e-commerce), integrating business processes with information technology and Internet based sales channels using the world-wide web and the Internet technology as a bearer [89].

Pay-per-Download or Pay-per-View is nowadays a known concept for e-commerce applications where web-based virtual shops use the web not only for the product offering but also for *shipping* the goods to the customer. The Pay-per-view business model for selling digital goods (*e. g.,* music tracks, e-books, journal articles) via the Internet is well established and it particularly has initiated the ongoing research and development work regarding robust and secure digital rights management (DRM) solutions [10].

Before e-commerce became a commonly used term there was already the world of analog Pay-TV where the subscription based content distribution of television programs and the Pay-per-View model for single high-value transmissions (*e. g.,* sports events or movie premieres) had been in place for years and had become a successful business model for several broadcasters. Pay-TV in Europe and the US could deliver audiences numbered in millions already in the early 1990s with analog communication networks (satellite, cable-TV, terrestrial) when e-commerce over the Internet was not yet invented.

Various new technologies have transformed the consumer entertainment landscape allowing digital content – such as movies, electronic games, books, television and music – to be delivered digitally. Because exact copies of digital creative work can be easily and quickly distributed, the digital rights holders and distributors often employ technical systems to protect copyrighted content. This technology is mainly characterized as *digital rights management* (DRM) technology.

The technical solutions for enforcing the television content restriction to the subscribers are referred to by *Conditional Access Systems* (CAS). This term was used before digital goods became available for consumption, distribution and copying by using personal computers. As television broadcasting becomes *more digital* and adopts the same standards for multimedia transmissions as Internet-enabled PCs [63, 35] and as these PCs can get access to the data channels used by television broadcasters (DVB-S, DVB-C) as well as to high-bandwidth Internet connections, the question comes up: when do the technologies converge toward a point that any user connected to the Internet could use any pay-per-view television offering? Reasoning that by following e-commerce strategies the Pay-TV broadcasters or the digital rights holders could reduce their costs of delivering the content significantly, we can analyze the barriers that might prevent the global access to a Pay-TV offering. Digital media have gained intensely in popularity over analog media both because of technical supremacy associated with their production, reproduction, and editing, and also because they are in general of higher perceptual quality than their analog counterparts.

A major barricade to overcome before digitalization reaches a mark where valuable digital content is transmitted everywhere to everybody is rights enforcement: a new distribution channel will probably not realize its potential before the digital rights holders are convinced that the new way to deliver their valuable content to the consumer does not technologically facilitate unauthorized copying or consumption of content [16].

There are a number of international groups working towards specifications for DRM standards and a large number of proprietary DRM applications competing in the marketplace. Until now, there is no internationally agreed specification for a full DRM system but there are some formal standardization processes (*e. g.,* DVB-CPCM [27]) that might prevent the existence of a dominant proprietary system becoming a de facto standard of the future.

One major facet of DRM is content protection of digital media (*e. g.,* CD, DVD and their successors). It might not be obvious for somebody not being familiar with DRM technology, but several of the mechanisms that are designed for securing broadcasted transmissions are also applicable to the protection of digital storage media. As a matter of fact, the similarity is well-founded because broadcasts do not rely on a feedback channel from the receiver back to the sender and this one-way communication precondition is also postulated for broadcast encryption schemes. Anyhow, a one-way digital transmission can be recorded and become a preceding block of data stored on media, thus, the same cryptographic protocols can be executed by broadcast receivers as well as media players. Hence, broadcast encryption is a major building block for content protection schemes among other technical measures, mainly hardware tamper resistance, or organizational measures, *e. g.,* legal enforcement of regulations. Vice versa, existing content protection and conditional access technologies are reconsidered in this dissertation in the context of broadcast encryption schemes in order to obtain a consistent terminology permitting a comparison of the diverse proposals.

The next chapters of this dissertation are organized as follow: Chapter 2 describes the terminology and mathematical results being used in this work, Chapter 3 summarizes related work from the scientific community and describes related technical standards. The following chapter 4 presents new broadcast encryption schemes for relaxed requirements while Chapter 5 depicts the more technical requirements for a global broadcast system with regional rights enforcement. Implementations of the proposed schemes and mechanisms are focused on in Chapter 6. Conclusions are drawn in Chapter 7 and the Appendix A offers some exemplary data regarding achieved parameters of the proposed new schemes.

Publications

The results of the research work conducted for this dissertation is partially published ([3], [45], [46], [47], [44]) with the following conference proceedings.

1. Ulrich Greveler: *How Pay-TV becomes E-Commerce*
   7th International IEEE Conference on E-Commerce Technology, Munich 2005, IEEE Press P2277

2. André Adelsbach and Ulrich Greveler: *A Broadcast Encryption Scheme with Free-Riders but Unconditional Security*
   Safavi-Naini, Reihaneh; Yung, Moti (Eds.): First International Conference on Digital Rights Management, Sydney 2005, Lecture Notes in Computer Science Vol. 3919, Springer-Verlag 2006

3. Ulrich Greveler: *Enforcing Regional DRM for Multimedia Broadcasts with and without Trusted Computing*
   Safavi-Naini, Reihaneh; Yung, Moti (Eds.): First International Conference on

Digital Rights Management, Sydney 2005, Lecture Notes in Computer Science Vol. 3919, Springer-Verlag 2006

4. Ulrich Greveler: *DRM für Multimedia-Broadcasts – wie sieht das Pay-TV der Zukunft aus?*
   in Patrick Horster (Hrsg.), D.A.CH Security '06, syssec, Düsseldorf 2006, pp. 260-267

5. Ulrich Greveler: *Patentierung kryptographischer Verfahren, die an Hochschulen entwickelt wurden*
   GI Fachtagung *Sicherheit 2006*, Magdeburg, Feb. 2006, LNI Proceedings P-77, pp. 329-332

Patents

One particular result of the research work conducted for this dissertation regarding the new approach for broadcast encryption schemes was subject of a patent application. The patent application was filed on August 30th, 2004, and the patent was issued and published on September 22nd, 2005, by the German patent agency (DPMA) as Patent Number `DE 102004042094 B3`, titled *Datenübertragungsverfahren und Datenübertragungsanordung* (inventor: Ulrich Greveler).

2 Mathematical Background

This chapter is a collection of basic material on probability theory, cryptographic tools, graph theory and general notations that will be used throughout this dissertation.

2.1 Basic Definitions

First, we fix the basic notations.

Definition 2.1.1. Let $\mathbb{N}$ denote the set of positive integers $\{1, 2, 3, \ldots\}$.

Definition 2.1.2. For $n \in \mathbb{N}$ let $\log n$ denote $\log_2(n)$, *i. e.,* the logarithm to the base 2 of n.

Definition 2.1.3. (negligible) A function $f : \mathbb{N} \to \mathbb{R}$ is called *negligible* if for every positive polynomial p

$$f(n) < \frac{1}{p(n)}$$

for all $n > n_p \in \mathbb{N}$.

Definition 2.1.4. (big O notation) Let f and g be two functions defined on real numbers. Then $f(x)$ is $O(g(x))$ iff $\exists\ x_0 \in \mathbb{R}, \exists\ M \in \mathbb{R}^+$ such that $\forall x > x_0\ :\ |f(x)| \leq M|g(x)|$.

Definition 2.1.5. (bit strings) The set $\{0,1\}^n$ for a $n \in \mathbb{N}$ denotes the set of all n-tuples whose items are from $\{0,1\}$ and the set $\{0,1\}^*$ is defined as the set of all (finite) tuples whose items are from $\{0,1\}$.

Set $\{0,1\}^*$ is also called the set of all *bit strings* as we interpret a bit string as a tuple consisting of the bits being assigned the values 1 for TRUE and 0 for FALSE. We also assume the empty string to be an element of $\{0,1\}^*$.

2.2 Probability Theory

2.2.1 Random Variables and Distributions

We will need some notions and results from probability theory because this theory provides a groundwork for the definitions regarding different levels of security, (pseudo-) randomness, probabilistic algorithms and also one-wayness. Moreover we will use probabilistic constructions with respect to key pre-distribution. To keep the introduction of probability theory short we restrict ourselves to discrete probability theory being sufficient for our purposes.

Definition 2.2.1. (sample space, event, probability) The *sample space* Ω is the non-empty set of possible outcomes of a random process. (In this thesis) Ω is finite or countably infinite. A sub-set of Ω is called *event*. The *probability measure* P assigns a value from the real interval $[0,1]$ to each event called the *probability* of this event such that $P(\Omega) = 1$ and for all disjoint events A and B it holds that $P(A \cup B) = P(A) + P(B)$.

Definition 2.2.2. (random variable, distribution) A (discrete) *random variable* X is a mapping from the sample space Ω into a non-empty and at most countably infinite set $X(\Omega)$ called the *range* or the the set of *possible values* of the random variable.

The *probability distribution* P_X of a random variable X is a mapping from $X(\Omega)$ into the real interval $[0, 1]$ with the property $P_X(x) = P(X = x)$ where "$X = x$" (also written as $\{X = x\}$) denotes the set $\{\omega \in \Omega : X(\omega) = x\}$ and is called an *atomic event.*

Definition 2.2.3. (probability ensemble) A *probability ensemble* $\mathbf{X}$ is a family $\mathbf{X} = \{X_n\}_{n\in\mathbb{N}}$ where each X_n is a random variable.

From these definitions follows immediately

Lemma 2.2.4. *Let P_X be a probability distribution of a discrete random variable X with sample space Ω and event $\Omega_1 \subseteq \Omega$. Then we have for P_X and the probability measure P on events from Ω*

$$P(\emptyset) = 0$$

and

$$P_X(x) \geq 0 \quad \forall x \in X(\Omega)$$

and

$$\sum_{x\in X(\Omega)} P_X(x) = 1$$

and

$$P(\Omega_1) = \sum_{x\in X(\Omega_1)} P_X(x) \quad .$$

We will use random variables in algorithms that need internal random values for further processing and having an output that is a random variable itself. This motivates the following definition of chosen values which will heavily be used for algorithmic descriptions.

Definition 2.2.5. (randomly chosen, randomly and uniformly chosen) Let P_X be a probability distribution of a discrete random variable X with sample space Ω. We say that a value $X(\omega)$ is *randomly chosen* from $X(\Omega)$ by X if $\omega \in \Omega$. Moreover, we say that values are chosen *uniformly* by X if $P_X(x) = P_X(x')$ for all $x, x' \in X(\Omega)$.

Regarding cases where more than one random variable is used we need some tools to determine the probabilities resulting from the combination of these variables.

Definition 2.2.6. (joint distribution, independence) Let $X_1, X_2, \dots, X_N$ be random variables on Ω. The *joint probability distribution* of the random variables is defined as a probability distribution on $X_1(\Omega) \times X_2(\Omega) \times \cdots \times X_N(\Omega)$ by

$$P_{X_1 X_2 \dots X_N}(x_1, x_2, \dots, x_n) := P(\{X_1 = x_1\} \cap \{X_2 = x_2\} \cap \cdots \cap \{X_N = x_N\})$$

and we call the $X_1, X_2, \dots, X_N$ *statistically independent* when

$$P_{X_1 X_2 \dots X_N}(x_1, x_2, \dots, x_N) = P_{X_1}(x_1) P_{X_2}(x_2) \cdots P_{X_N}(x_N)$$

for all N-tuples of atomic events.

Remark 2.2.7. Notice that the definition of a joint distribution considers combinations of atomic events of the random variables but as we restrict ourselves to discrete probabilities these function values can be seen as a probability distribution on the combination of the random variables and be added up for every Cartesian product of sub-sets of the $X_i(\Omega)$ by applying Lemma 2.2.4 to derive the probability measure for the Cartesian product of events.

Definition 2.2.8. (expectation) Let P_X be a probability distribution of a (discrete) random variable X with sample space Ω and let f be a function defined on a superset of $X(\Omega)$ into $\mathbb{R}$. We define the *expectation* or *average value* of $f(X)$ by

$$E(f(X)) := \sum_{x \in X(\Omega)} P_X(x) f(x) \quad .$$

Remark 2.2.9. The definition allows the values of the possible outcome of X to be non-real; in many cases the values are real and the function f is interpreted as the identity function when we write $E(X)$ for the expected (or average) value of a random variable X.

Definition 2.2.10. (conditional probability distribution) Let P_X and P_Y be probability distributions of discrete random variables X and Y with joint sample space Ω and let $x \in X(\Omega)$ and $y \in Y(\Omega)$. The *conditional probability distribution* is defined by

$$P_{Y|X}(y|x) := \frac{P_{XY}(x,y)}{P_X(x)}$$

for all (x, y) with $P_X(x) \neq 0$.

By applying Lemma 2.2.4 we can conclude from the definition the following lemma.

Lemma 2.2.11. *Let P_X and P_Y be probability distributions of discrete random variables X and Y on Ω. Then we have for all x with $P_X(x) \neq 0$.*

$$\sum_{y \in Y(\Omega)} P_{Y|X}(y|x) = 1 \quad .$$

The lemma can be interpreted as a motivation to extent the definition from atomic events to all events which is done in the following more general definition of conditional probability (also applicable to the non-discrete case).

Definition 2.2.12. (conditional probability measure) Let P be probability measure on sample space Ω and let $A, B \subseteq \Omega$ be events. The *conditional probability measure* is defined by

$$P(A|B) := \frac{P(A \cap B)}{P(B)} \quad .$$

Finally, we need for our constructions the notion of conditional expectation of a random variable which we also present by using a real function to be more flexible.

Definition 2.2.13. (conditional expectation) Let P_X be a probability distribution of a (discrete) random variable X with sample space Ω and let $A \subseteq \Omega$ be an event. We define the *conditional expectation* of $f(X)$ *given the occurrence* of A by

$$E(f(X)|A) := \sum_{x \in X(\Omega)} P(X = x|A) f(x) \quad .$$

for a real function f defined on a superset of $X(\Omega)$.

2.2.1.1 Important Probability Distributions

We will encounter certain well-known probability distributions in our constructions: The *normal distribution* (Gaussian distribution) with parameters *mean* and *standard deviation* (mainly as an approximation to other distributions); the *Bernoulli distribution* (for single experiments) and the *binomial distribution* (for describing the number of successes in a series of independent Bernoulli experiments with replacement), the *discrete uniform distribution* (when we choose keys randomly and uniformly) and the *hypergeometric distribution* (when we randomly choose sub-sets of the same size without replacement). For all these distributions we will apply their respective cumulative distribution functions when calculating our schemes' parameters. The normal distribution can be used to approximate to other distributions, notably to the binomial distribution. Since the comprehensive introduction of these stochastic tools would be too voluminous we refer to Bauer's [8] or Krengel's [67] textbook as reference.

2.2.2 Probabilistic Algorithms

We will need probabilistic algorithms that make use of an internal random process. For this thesis we can limit the notion of algorithm to an intuitive machine model that operates on bit strings.

Definition 2.2.14. (algorithm) An *algorithm* is a function that can be evaluated by a Turing machine. It *reads* an item from $\{0,1\}^*$ and *outputs* an item from $\{0,1\}^*$.

Regarding the definition of a Turing machine and the Church-Turing thesis about calculations performed by an algorithm we refer to a standard textbook on the theory of computer science [90].

Definition 2.2.15. (probabilistic algorithm) A *probabilistic algorithm* is an algorithm that can be evaluated by a Turing machine that is supplemented by a read-only random tape storing the outcomes of an infinite sequence of unbiased coin tosses, *i. e.,* the machine can randomly and uniformly choose a value from $\{0,1\}$. The output is described by a random variable ranging over $\{0,1\}^*$.

Definition 2.2.16. (polynomial-time algorithm, PPT algorithm) A *polynomial-time algorithm / probabilistic polynomial-time algorithm* is a algorithm / probabilistic algorithm that has a running time of $O(n^k)$ where n is the input size and k is a constant. The running time of the (probabilistic) algorithm is measured as the number of steps of the (probabilistic) Turing machine.

2.3 Cryptographic Tools and Security Notions

2.3.1 Indistinguishability, PRNGs

We introduce the notion of (computational) indistinguishability which is fundamental in complexity theory and originates from Goldwasser and Micali [40].

Definition 2.3.1. (indistinguishability) We call probability ensembles $\mathbf{X} = \{X_n\}_{n\in\mathbb{N}}$ and $\mathbf{Y} = \{Y_n\}_{n\in\mathbb{N}}$ (where X_n and Y_n range over binary strings) *computationally indistinguishable* if for every polynomial-time algorithm A the function

$$d_A(n) := |P(A(X_n) = 1) - P(A(Y_n) = 1)|$$

is negligible.

In the next definition indistinguishability is applied to define pseudo-randomness in way that a pseudo-random generator passes all polynomial-time bounded statistical tests so it cannot be distinguished from random sequences by any PPT algorithm.

Definition 2.3.2. (pseudo-random generator) Let $\ell : \mathbb{N} \to \mathbb{N}$ be a function with $\ell(n) > n$ for all $n \in \mathbb{N}$. An algorithm that for each $i \in \mathbb{N}$ reads an i-bit string (the *seed*) as input and outputs an $\ell(i)$-bit string that is computationally indistinguishable from a randomly and uniformly chosen $\ell(i)$-bit string is called a *pseudo-random generator* (*PRNG*).

Definition 2.3.3. (pseudo-random function) A function which can be evaluated by a polynomial-time algorithm that returns for any n-bit strings s (*seed*) and x (*argument*) an n-bit value $f_s(x)$, is called a *pseudo-random function* if no PPT algorithm can distinguish the values $f_s(x)$ for a randomly and uniformly chosen s from randomly and uniformly chosen strings of the same length.

2.3.2 One-wayness

Definition 2.3.4. A function $f : \{0,1\}^* \to \{0,1\}^*$ is called *one-way* if two conditions are met

(i.) **Easy computation**: There exists a polynomial-time algorithm A outputting on input x the value $f(x)$ for all $x \in \{0,1\}^*$.

(ii.) **Hardness of invertation**: For every PPT algorithm A'

$$P\left(A'(f(x)) \in f^{-1}(f(x))\right)$$

is a negligible function in the length of x.

2.3.3 Security Notions

In modern cryptography several notions of security are used. The two major perceptions are information-theoretic (perfect, unconditional) security versus computational security. There are a lot of settings where information-theoretic security cannot be achieved at all or where only computationally secure schemes are known. Moreover, there are further assumptions that are included in some notions of security, *e. g.,* the hardness of inverting certain functions or we put restrictions regarding the capabilities of an attacker (*e. g.,* ciphertext-only attacks). In this thesis we will show for some of the proposed schemes that information-theoretic security is achieved while for others we can at least show computational security. When further assumptions need to be made they will be named explicitly. In the literature there are several proposed notions for information-theoretic as well as computational security [39, 41, 77].

We provide a definition for the two major security notions being used in this work and give an informal description for another variant. The focus is on encryption schemes as we will use the definitions mainly for the proposed broadcast encryption schemes.

The idea of a unconditionally secure encryption scheme is (loosely speaking): The system is secure if having the ciphertext does not help in learning any information about the plaintext. This is formalized in the next definition, following Goldwasser and Bellare [41].

Definition 2.3.5. (unconditional security for a cryptosystem) Let $n \in \mathbb{N}$ and let the PPT algorithms $\textsc{Setup}$, $\textsc{Encrypt}$, $\textsc{Decrypt}$ constitute a cryptosystem for the message space $\{0,1\}^n$ together with a message probability distribution $D : \{0,1\}^n \rightarrow [0,1]$. The algorithm $\textsc{Setup}$ provides a key K uniformly chosen from $\{0,1\}^k$ for some security parameter k to sender and receiver, $\textsc{Encrypt}$ takes the key K and a message $M \in \{0,1\}^n$ and outputs C with

$$\textsc{Decrypt}_K(C) = M \quad .$$

The sender has access to $\textsc{Encrypt}_K$ and sends the encrypted message to the receiver (and also to the adversary). The receiver has access to $\textsc{Decrypt}_K$ and can thus decrypt the message. The adversary can compute D.

We say that the cryptosystem is *unconditionally secure with respect to D* if for every $M \in \{0,1\}^n$ and every possible ciphertext C we have

$$P(M|C) = D(M) \quad . \tag{2.1}$$

We say that the cryptosystem is *unconditionally secure* if it is unconditionally secure for every message probability distribution D.

Remark 2.3.6. This definition takes into account that an adversary shall not gain any information about the message text from the ciphertext but he might have some information about the frequencies or likelihoods of certain messages so we define the security of a cryptosystem by making sure that no additional information is gained from the knowledge of an intercepted ciphertext.

Example 2.3.7. The One-Time Pad is an unconditionally secure cryptosystem with respect to Definition 2.3.5. We can set $k := n$ and use the $\textsc{xor}$-operation for encryption and decryption and let key generation be an algorithm which chooses randomly and uniformly from $\{0,1\}^k$. The event that a certain M occurs is then independent from a certain C and the equality in equation 2.1 holds for all $M \in \{0,1\}^n$ and all $C \in \{0,1\}^n$.

Next, we introduce computational security, *i. e.,* the security holds with respect to adversaries of limited computing capability. We consider the ciphertext-only-attack case.

Definition 2.3.8. (computational security) Let $n \in \mathbb{N}$ and let the PPT algorithms $\textsc{Setup}$, $\textsc{Encrypt}$, $\textsc{Decrypt}$ constitute a cryptosystem for the message space $\{0,1\}^n$ together with a message probability distribution $D : \{0,1\}^n \to [0,1]$. The algorithm $\textsc{Setup}$ reads the security parameter k and provides a key K chosen from $\{0,1\}^k$ to sender and receiver, $\textsc{Encrypt}$ takes the key K and a message $M \in \{0,1\}^n$ and outputs C with

$$\textsc{Decrypt}_K(C) = M$$

and the algorithm R selects a message randomly from $\{0,1\}^n$ with respect to distribution D.

The sender has access to $\textsc{Encrypt}_K$ and sends the encrypted message to the receiver (and also to the adversary). The receiver has access to $\textsc{Decrypt}_K$ and can thus decrypt the message. The adversary can compute D and is implemented by a PPT algorithm A that reads ciphertexts.

We say that the cryptosystem is *computationally secure against ciphertext-only-attacks with respect to* D if for every adversary A the function

$$f(k) := \Big| P\left(M = A\left(\textsc{Encrypt}_K(M)\right) \mid K \leftarrow \textsc{Setup}(k), M \leftarrow R\right) - D(M) \Big|$$

is negligible in k.

We say that the cryptosystem is *computationally secure against ciphertext-only-attacks* if it is computationally secure against ciphertext-only-attacks for every message probability distribution D.

2.3.4 Perfect Hash Functions

We will use *perfect hash functions* that are a convenient tool that unlike cryptographic hash function do not contain a compression function taking any input and having a fixed-length output but are rather length preserving as they are hash functions which guarantee to map different keys to different numbers. Collision-freeness is an absolute requirement unlike the very different notion of infeasibility of finding collisions required for cryptographic hash functions.

Definition 2.3.9. *{Fiat and Naor [36]}* Let $\mathcal{N}$ be a non-empty finite set, l and m be positive integers and $\{f_i\}_{i=1}^{l}$ be a family of functions with $f_i : \mathcal{N} \to \{1, 2, \ldots, m\}$

with the property that for every sub-set $\mathcal{S} \subseteq \mathcal{N}$ of size k there exists one i with $u \neq u' \Rightarrow f_i(u) \neq f_i(u')$ for all $u, u' \in \mathcal{S}$ (in other words: f_i is injective on $\mathcal{S}$) then the family $\{f_i\}_{i=1}^{l}$ contains a perfect hash function f_i for every size k sub-set of $\mathcal{N}$.

Remark 2.3.10. The definition is very minimal in the sense that apart from the injectiveness no further requirement is set, in particular there is no one-wayness or randomness required. For our purposes we are interested how to construct a random family of functions containing a perfect hash function or how to calculate the probability that a certain random function being part of a family is a perfect hash function. We collect the relevant results in the next lemma.

Lemma 2.3.11. { Fiat and Naor [36] } *Let l, m be positive integers and $\mathcal{N}$ be a non-empty finite set and $\mathcal{K} \subseteq \mathcal{N}$ be a non-empty sub-set and let $n := |\mathcal{N}|$ and $k := |\mathcal{K}|$. Let $\{f_i\}_{i=1}^{l}$ be a family of functions with $f_i : \mathcal{N} \to \{1, 2, \ldots, m\}$, which is constructed by choosing randomly and uniformly a value $f_i(x)$ from $\{1, 2, \ldots, m\}$ for each (i, x) with $1 \leq i \leq l$ and $x \in \mathcal{N}$. Let $f_i|\mathcal{K}$ denote the function that is constructed from f_i by restricting the arguments to the set $\mathcal{K}$.*

(i.) *If $m \geq 2k^2$ then the probability that a certain member of the family $\{f_i|\mathcal{K}\}_{i=1}^{l}$ is a perfect hash function is greater than $\frac{3}{4}$.*

(ii.) *If $m \geq 2k^2$ and $l = \lceil k \log n \rceil$ then the probability that the family $\{f_i|\mathcal{K}\}_{i=1}^{l}$ does not contain a perfect hash function is at most $\dfrac{1}{n^k}$.*

(iii.) If $\mathcal{N}$ and $\mathcal{K}$ are given then there exists a family $\{f_i|\mathcal{K}\}_{i=1}^{l}$ of functions satisfying the parameters $m := 2k^2$ and $l := \lceil k \log n \rceil$ and containing a perfect hash function.

Proof. Fix an i with $1 \leq i \leq l$. We can calculate the number of sub-sets with 2 elements of $\mathcal{K}$ by $\binom{k}{2}$ and see that $f_i|\mathcal{K}$ is injective with probability at least $1 - \frac{\binom{k}{2}}{m}$ for the m possible values of the function. As injectiveness is sufficient for our claim and as

$$\frac{\binom{k}{2}}{m} = \frac{\frac{k!}{2!(k-2)!}}{m} = \frac{\frac{k(k-1)}{2}}{m} \leq \frac{\frac{k(k-1)}{2}}{2k^2} < \frac{1}{4}$$

using the precondition $m \geq 2k^2$ we have proven part (i.) of the lemma.

For proving part (ii.) we use the claim from part (i.) and see that the probability that no $f_i|\mathcal{K}$ is injective is less than $(\frac{1}{4})^l$ as the values of the f_i were chosen independently. By using the precondition $l = \lceil k \log n \rceil$ we have

$$\frac{1}{4^l} \geq \frac{1}{n^{2k}}$$

and we can argue that the probability that one or more f_i are injective is at least

$$1 - \frac{\binom{n}{k}}{n^{2k}} \geq 1 - \frac{1}{n^k}$$

proving part (ii.) of the lemma.

Regarding part (iii.) it is sufficient to show that for given sets $\mathcal{N}$ and $\mathcal{K}$ we can construct an f_1 that is injective when restricted to $\mathcal{K}$ and therefore is the perfect hash function in the family. As the parameter m is set $m := 2k^2$ this is straightforward: the set of possible values for arguments in $\mathcal{K}$ is greater than the set of arguments and both are non-empty finite sets so we can chose a value for $k_1 \in \mathcal{K}$, store it, mark the elements k_1 and $f_1(k_1)$ in the respective sets and then continue with k_2 and $f_1(k_2)$ by choosing unmarked values etc. until all elements of $\mathcal{K}$ are marked; for remaining arguments in $\mathcal{N} \backslash \mathcal{K}$ (if any) an arbitrary value can be set as well as for all other functions $f_2, f_3, \ldots, f_l$ of the family. Clearly, this construction provides a family $\{f_i|\mathcal{K}\}_{i=1}^{l}$ containing a perfect hash function $f_1|\mathcal{K}$. □

Remark 2.3.12. The lemma helps us to select reasonable values for l and m in practice. We do not need to construct families of functions in an explicit way and we do not need to allocate memory for the function values as we can use PRNGs instead to calculate a certain value needed at a certain time and estimate the probability that the constructed family contains a perfect hash function by postulating that the PRNG is comparable to a random function.

2.4 Graphs, Binary Trees, Steiner Trees

We will use basic graph theory as a convenient tool to describe the schemes and protocols in the following sections. Trees are needed rather often and we define and treat them as particular graphs so that we can use the extensive notation provided by graph theory which is detailed in the following.

Definition 2.4.1. Let S be a set and $n \in \mathbb{N}$.

Iff $|S| = n$ we call S an n-set. The empty set is called a 0-set.

With S^n / with $[S]^n$ we denote the set of n-tuples / n-sub-sets whose elements are from S.

Definition 2.4.2. (graph) A *graph* G is a pair (V, E) of sets with $E \subseteq [V]^2$ and $V \cap E = \emptyset$. Elements of V are called *vertices* or *nodes* of G and are denoted by $V(G)$, elements of E are called *edges* of G and are denoted by $E(G)$ and G is also called a graph *on* the set of vertices V.

Definition 2.4.3. (incident, adjacent, degree, path) Let $G = (V, E)$ be a graph. A vertice $v \in V$ is *incident* with edge $e \in E$ iff $v \in e$. Two vertices $v, v' \in V$ are *adjacent* iff $\{v, v'\} \in E$. The *degree* of a vertex is the number of edges containing this

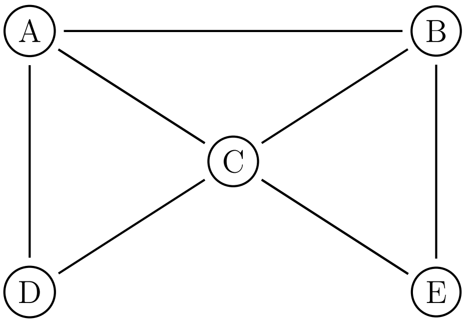

Figure 2.1: Graph with 5 vertices

vertex. For edges $\{e_1, e_2, \ldots, e_n\} \subseteq E$ we call $e_1e_2 \ldots e_n$ a *path* of G iff $e_1e_2 \ldots e_n$ is of the form $\{v_1, v_2\}\,\{v_2, v_3\} \cdots \{v_n, v_{n+1}\}$ with pairwise different vertices $v_j \in V$ for all $1 \leq j \leq n + 1$. The *length* of this path is given by n (the number of edges). The path if *from* vertex a *to* vertex b if a and b are both elements of exactly one of the edges the path consists of. Vertices a and b are then called the *endpoints* of the path.

Remark 2.4.4. Note, that a vertex v of a graph cannot be adjacent to itself because $\{v, v\}$ is not a 2-set. A path can be seen as a walk on edges (speaking loosely) from one vertex to another where each edge and each vertex is only traversed once (the vertices of the path are pairwise different thus so are the edges).

Example 2.4.5. When we picture a graph in this thesis we use a representation like in Fig. 2.1 where a vertex is depicted as a dot or a circled item (here the vertices are $\{A, B, C, D, E\}$ and the edges are lines drawn between adjacent vertices, *e. g.,* $e := \{A, B\}$ is an edge in this example but $e' := \{A, E\}$ is not an edge of the graph as A and E are not adjacent. The degree of A is 3. We can find a path $\{B, E\}\,\{E, C\}\,\{C, D\}$ in this graph.

Definition 2.4.6. (subgraph, connected) Let $G = (V, E)$ be a graph. A *subgraph* $G' = (V', E')$ of G is a graph with the property $V' \subseteq V$ and $E' \subseteq E$.

The graph G is *connected* iff $G \neq \emptyset$ and for every 2-sub-set $\{v, v'\} \subseteq G$ there exists a path $\{v, v_1\}\{v_1, v_2\} \cdots \{v_n, v'\}$ in G, with vertices $v_j \in V$ for all $1 \leq j \leq n \in \mathbb{N}$. Likewise, two vertices $\{v, v'\}$ are *connected* iff such a path exists.

Definition 2.4.7. (cycle) Let $G = (V, E)$ be a graph. For every path $\{v_1, v_2\}\{v_2, v_3\} \cdots \{v_{n-1}, v_n\}$ of G with $n > 2$ and $\{v_1, v_n\} \in E$ we call $\{v_1, v_2\}\{v_2, v_3\} \cdots \{v_{n-1}, v_n\}\{v_n, v_1\}$ a *cycle* of G. The *length* of this cycle is given by $n + 1$ (the number of edges).

Definition 2.4.8. (forest, tree, subtree, root, leaf) A graph G with the property that no cycles of G exist is called a *forest*.

A connected forest is called a *tree*. A subgraph of a tree T being a tree it self is called a *subtree* of T.

In a tree we can label exactly one node as the *root* of the tree. The tree is then called a *rooted tree*.

We call all nodes with degree 1 of a rooted tree the *leaves* of the tree except the root node that is called a leaf if it is the only node of a tree.

Remark 2.4.9. The trees that we use for our purposes in the following sections are mostly binary trees constructed from relationships between nodes that are regarded as parent and child of each other. A definition of these objects are usually given by using directed graphs. In order to keep the collection of definitions brief we define the terms on (not directed) trees by using the path lengths and do not use directed graphs in this work.

Definition 2.4.10. (parent, child) Let be $G = (T, E)$ be a rooted tree. If $\{v, v'\} \in E$ and there exists a path from root to v and a path from root to v' where the first path's length is smaller than the latter path's length then we call v the *parent* of v' and v' a *child* of v.

Pairwise different children having the same parent are called *siblings*.

A node is called the *ancestor* of a different node if there exists a path from the first node to a leaf containing an edge that is incident with the second node.

A node with ancestor v is called a *descendant* of v.

Remark 2.4.11. To see that the terms of the last definition are well defined we can use some of the fundamental assertions of graph theory that are collected in the following theorem.

Theorem 2.4.12. *Let v be a node in a rooted tree not being the root node. Let v' be a node adjacent to v and let v'' be any node different from v'.*

(i.) *There is exactly one path from root to v.*

(ii.) *The path lengths of v and v' differ by exactly one.*

(iii.) *Either v is a child of v' or vice versa.*

(iv.) *v is a descendant of root.*

(v.) *The root node is an ancestor of all other nodes.*

(vi.) *If v and v'' are connected than there is exactly one path from v to v''.*

(vii.) *v has a parent. The root node does not have a parent.*

(viii.) *If v is not an ancestor of v'' and vice versa then there is exactly one node that is an ancestor of both nodes and has a maximum path length to root (called the* least common ancestor *of the nodes).*

Proof. These basic properties of rooted trees are shown in many textbooks on graph theory. See [65] as textbook reference. □

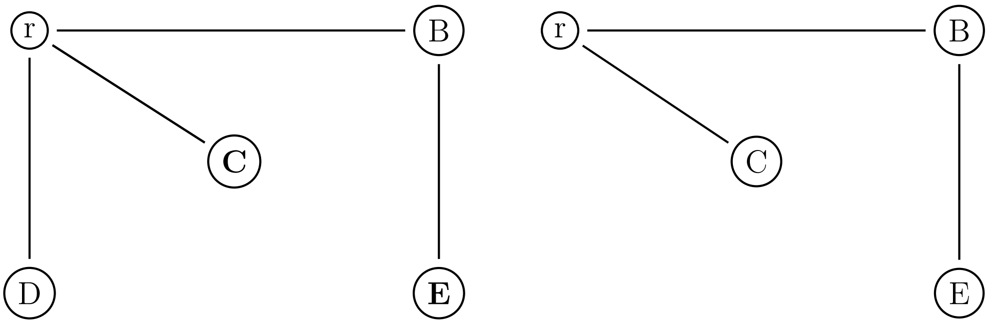

Figure 2.2: Tree with root r and corresponding Steiner tree $ST(\{C, E\})$

Definition 2.4.13. (binary tree, full binary tree, depth) A *binary tree* is a rooted tree in which all nodes have at most two children.

A *full binary tree* is a binary tree where all paths from root to a leaf have the same length and all nodes except the leaves have exactly two children.

The depth of a tree is the maximum length of all paths of the tree or is 0 if no paths exists.

Lemma 2.4.14. *A binary tree with* n *leaves has at least depth* $\log n$*. Equality is reached iff the binary tree is a full binary tree.*

Proof. See [65] as textbook reference. □

Definition 2.4.15. (Steiner tree) Let $T = (V, E)$ be a rooted tree with root node r. For a sub-set $V' \subseteq V$ we define the *Steiner tree* of V', denoted by $ST(V')$, as the unique subtree T' of T with $V(T') = V' \cup \{r\}$.

Remark 2.4.16. We define Steiner trees as sub-sets of trees but they could also be defined for any vertex sub-sets of graphs. In the latter case it is a difficult problem to find a (minimum) Steiner tree that fulfills the requirement. In our case (regarding trees) the Steiner tree is unique so there is no minimum requirement and it can easily be found by examining all paths that each connect a vertex of V' with root and take the set of all edges contained in these paths as $E(ST(V'))$ and the union of these edges as $V(ST(V'))$. To see that the Steiner tree is well defined in the last definition and learn more about the general Steiner tree problem for graphs we refer the reader to [88].

Example 2.4.17. See Fig. 2.2 for a Steiner tree example. The node D is not contained in $ST(\{C, E\})$ as it is not on a path to root. The node B is on E's path to root so it is an internal node in the Steiner tree.

3 Related Work

3.1 Motivation

Broadcast encryption allows a broadcast center to send an encrypted message to all receivers in a way that only legitimate receivers are able to decrypt it. The broadcast can be performed as a radio broadcast (*e. g.,* using satellites) or be recorded as a data stream on storage media (*e. g.,* Digital Versatile Disc).

In the literature one can find very efficient *revocation schemes* which are suitable for a small set $\mathcal{R}$ with $|\mathcal{R}| \ll |\mathcal{N}|$ of revoked receivers (*e. g.,* pirate receivers) compared to a huge number of total users $\mathcal{N}$ so that the broadcast communication can only be decrypted by the users in $\mathcal{T} := \mathcal{N} - \mathcal{R}$. Revocation schemes are therefore specific broadcast encryption schemes being most efficient in the setting where only few receivers are revoked.

The asymptotically most efficient known revocation schemes [43, 49, 64, 82] require a message header of length $O(|\mathcal{R}|)$ and user's individual private key size of $O(\log(|\mathcal{N}|))$. However, these revocation schemes are **not intended nor suitable** for a general subset case, *e. g.,* cases where $|R| \approx \frac{1}{2}|\mathcal{N}|$.

In this dissertation we will consider the case where **arbitrary sub-sets** are addressed by the sender. For this setting the trivial broadcast encryption scheme, addressing exactly all the users in the target set $\mathcal{N} - \mathcal{R}$, sends the message encrypted individually for each user yielding a total number of $|\mathcal{N} - \mathcal{R}|$ messages to be sent via broadcast and only $O(1)$ keys to be stored by a user.

This trivial scheme imposes severe shortcomings. Consider the following example: Assuming that any sub-set of $\mathcal{N}$ is chosen with equal probability for a transmission, an average number of $\frac{1}{2}|\mathcal{N}|$ messages needs to be sent via the broadcast channel for every transmission if the trivial scheme is used. In order to reduce this number, it is possible to assign keys to certain or all sub-sets of $\mathcal{N}$ and make these keys known only to the members of the sub-set. But even in the best (and not realistic) case where each user is provided with a key for all $2^{|\mathcal{N}|-1}$ sub-sets it belongs to, the numbers of bits needed to encode the sub-set key identifier is approximately $|\mathcal{N}|$ so any scheme which addresses the exact sub-sets would need to send $O(|\mathcal{N}|)$ message bits. Apart from that lower bound, a tradeoff between the number of keys stored by each user and the number of messages to be sent to establish a transmission session key needs to be considered. The number of colluders (users outside the target group cooperating to break the scheme) the system can tolerate is another major parameter.

We are also interested in the level of security (existence of one-way functions, number-theoretic or information-theoretic security) we can establish by using certain broadcast encryption schemes.

In the remaining sections of this chapter we assert the formal definitions of the regarded objects and compile related work and standards. We also provide comprehensive proofs of certain related published assertions which are not given or only sketched in the printed proceedings of conferences where the idea was presented.

3.2 Definitions

In 1993 Fiat and Naor introduced the notion of broadcast encryption [36] and presented several solutions to the problem of dynamically handling membership changes in secure one-way communication groups. Before, Berkovits [9] introduced the problem to securely broadcast to a small sub-set of the user base.

We will first give some formal definitions and introduce the notations about broadcast encryption schemes. In the literature there is no consistent terminology regarding the presentation of broadcast encryption schemes, we will loosely follow the terms introduced by Naor *et al.* [82], but use our variant consistently throughout this work.

Definition 3.2.1. (broadcast system) A *broadcast system* is made of a sender `BC` (the *broadcast center*) and a finite non-empty set $\mathcal{N}$ of *receivers* that we also call *users.*

Notation: The sender `BC` can only send to all receivers simultaneously (*broadcast message*) and there is no return-channel (*e. g.,* for acknowledgments or handshake protocol items). A sender can address certain users by using a *message header* containing explicit *user ID*s or by identifying a sub-set $\mathcal{T} \subseteq \mathcal{N}$ with a *sub-set ID.* To ease notation we will often use the term user when formally the user's ID is referred to. The addressees can also implicitly be addressed if the message is encrypted and only the users in the target set $\mathcal{T}$ can decrypt the message; in any case we will call these users the *privileged* set of users.

In the majority of cases a scheme does require both criteria, *i. e.,* the message header will contain a description of the sub-set $\mathcal{T}$ as well as the body is encrypted and only the privileged users can decrypt it. Moreover, the message body is encrypted by a *session key* (a *media key* or *disk key* in the case of storage media) and the session key itself can be derived by the privileged users by processing the header information. This is done for performance reasons as the message itself is typically much greater than a symmetric cryptographic key. In some cases, though, we might not want to make use of a session key (*e. g.,* when the goal is unconditional security) but then the same schemes can still be used by having a formal zero-length message body indicating that the session key is regarded as the message itself. Thus, we will always use a session key in our constructions without limiting security levels. This consideration motivates the following definition.

Definition 3.2.2. (broadcast encryption system) Let $\mathcal{K}$ be a set of keys and $\mathcal{B}$ a set of broadcast messages. A *broadcast encryption system* is a broadcast system with an associated triple of algorithms (Setup, Broadcast, Decrypt) such that

(i.) The setup algorithm (Setup) takes a user $u \in \mathcal{N}$ and computes the users's private key information $P_u \subseteq \mathcal{K}$.

(ii.) The broadcast algorithm (Broadcast) takes the set of privileged users $\mathcal{T} \subseteq \mathcal{N}$ and a session key $K \in \mathcal{K}$ and outputs a broadcast message $B \in \mathcal{B}$.

(iii.) Any user $u \in \mathcal{N}$ can run the decryption algorithm $\textsc{Decrypt}(B; P_u; u)$ that will output K if $u \in \mathcal{T}$ and fail if $u \notin \mathcal{T}$.

We will also define a modification of a broadcast encryption system. Let $\mathcal{T}_0 \subseteq \mathcal{N}$ with $\mathcal{T} \subset \mathcal{T}_0$. A broadcast encryption system *tolerates* free-riders $\mathcal{T}_0 \setminus \mathcal{T}$ iff *(i.)* and *(ii.)* holds and

(iii-a.) Decrypt outputs K if $u \in \mathcal{T}_0$ and fails if $u \notin \mathcal{T}_0$

is fulfilled.

Remark 3.2.3. We will extend this definition to systems that consider user locations (Definition 5.3.3) but initially we focus on the related research work.

The set of keys will always be a set of bit-strings of equal length if not specified further.[1]

A key will sometimes be regarded as a pair $(\textsc{id}_K, K)$ so that it is easier to describe algorithms that choose a certain key (by its key identification $\textsc{id}_K$). The definition does not cover the case where several non-privileged users pool their information and run Decrypt on parameters calculated from this pooled information. In fact the security of some schemes is dependent on the number of users being able to collaborate. This is formally treated in the next definitions.

Definition 3.2.4. (resiliency) Given a sub-set $\mathcal{S} \subseteq \mathcal{N}$, a broadcast encryption scheme is said to be *resilient* to a set $\mathcal{S}$ if, for every privileged set $\mathcal{T} \subseteq \mathcal{N} - \mathcal{S}$ a user $u \in \mathcal{S}$ knowing all keys in the set $\bigcup_{u' \in \mathcal{S}} P_{u'}$ cannot run the decryption algorithm Decrypt on values computed from this set of keys to output K.

Definition 3.2.5. (k-resiliency) A broadcast encryption scheme is said to be k-resilient if it is resilient to every set $\mathcal{S} \subseteq \mathcal{N}$ with $|\mathcal{S}| \leq k$.

The resiliency is one of the comparable parameters or properties of all schemes presented in this work.

3.3 Naive Approach

The first trivial scheme to address exactly all the users in the target set $\mathcal{T} = \mathcal{N} - \mathcal{R}$ is to send the message encrypted individually for each user yielding a total number of

[1] The keys from Definition 3.2.2 can be divided into two types: *long-lived keys* that are used throughout the lifetime of a broadcast encryption system and *short-lived keys* that are used only once, *e. g.,* session keys valid for one transmission only. In the majority of cases, both key types are randomly chosen bit-strings of the same length, so the set $\mathcal{K}$ is the set of all possible bit-strings and covers both types. Sometimes we might use shorter bit-strings for short-lived keys in order to decrease header lengths, then $\mathcal{K}$ could contain bit-strings of different lengths but the definition is still valid.

$|\mathcal{T}|$ messages to be sent via broadcast and only $O(1)$ keys (*e. g.,* exactly one key) to be stored by a user. This scheme can be a reasonable approach if the broadcast channel is sufficiently large or the target set is small.

Another trivial scheme is to assign a key for every possible sub-set of $\mathcal{N}$ and to provide each user with all keys assigned to a sub-set she belongs to. This way only one message needs to be sent via the broadcast channel and $2^{|\mathcal{N}|-1}$ keys are to be stored. Storage is the limiting factor then and the scheme is only suitable for very small sets $\mathcal{N}$.

Both trivial schemes yield a boundary for the two pivotal parameters, *i. e.,* the number of messages to be sent so that the target group can derive the broadcast key and the key storage required by each user. The number of messages is also called the message header length since a *header* of the broadcast message can be regarded as meta-information necessary for key-agreement and the message *body* is the transmission itself. Thus, each user does need at least one key (if every broadcast shall be encrypted) and she will not need more than $2^{|\mathcal{N}|-1}$ keys; the message header length is between 0 (no header necessary) and $|\mathcal{T}|$ (one message for each user). The literature is inconsistent whether this 0 is the real lower bound since we could argue that a message header is always necessary in order to specify the sub-set itself and $\log(|\mathcal{N}|)$ bits are needed to do that. We will consider this detail more deeply when certain schemes are compared regarding their header length complexities. To avoid ambiguities we define the term (zero-header length) and use it in the same way as the majority of the referenced literature.

Definition 3.3.1. (zero-header length) A broadcast encryption scheme is said to have a *zero-header length* if the broadcast algorithm (Broadcast) only outputs the necessary information to identify the set of privileged users $\mathcal{T}$.

3.3.1 Non-trivial constructions

First note, that a set of users could in general be split up into small batches and the trivial scheme to assign a key to each possible sub-set can be run for every batch serially in order to apply the scheme for a huge number of users. This batch split has been suggested in [36] but not formally defined which we will do next.

Definition 3.3.2. (batch split) A broadcast encryption scheme with user set $\mathcal{N}$ and associated algorithms Setup, Broadcast and Decrypt is *split up into b disjoint batches* (of users)

$$\mathcal{U}_1 + \mathcal{U}_2 + \cdots \mathcal{U}_b = \mathcal{N}$$

if all the $\mathcal{U}_i$ are non-empty and the following construction is used: The Setup algoritm is applied for each of the $\mathcal{U}_i$ to assign the users' private keys for b independent schemes. For a target set $\mathcal{T} \subseteq \mathcal{N}$, Broadcast is run serially for $i = 1 \ldots b$ on $\mathcal{T} \cap \mathcal{U}_i$ and the outputs are concatenated and sent via the broadcast channel (each sent output is prefixed by the batch number i so the ith batch can be identified by a receiver). User $u \in \mathcal{U}_i$ then runs Decrypt on the ith output part to decrypt the secret using the scheme for user set $\mathcal{U}_i$.

Remark 3.3.3. The usage of batches offers the following advantages:

- The size of private key space for each user is (for most schemes) increasing with the total number of users. Moreover, some schemes have a hard limit on the maximum number of users (see example).

- The batch size can be chosen advantageously (*e. g.,* choose powers of 2 if required by the scheme).
- Users belonging to different batches are unable to collude since their respective schemes' keys are independent from each other.

The downside is obvious:

- For each batch a transmission header is generated so (in the worst case) the header length is multiplied by b

Example 3.3.4. The trivial scheme to assign a key to every possible sub-set of $\mathcal{N}$ and provide each sub-set member with this key causes a private key space of $2^{|\mathcal{N}|}$. In order to overcome the the exponential growth of the user key size[2] we can use batches of the size $|\mathcal{U}_i| = 21$ and assign each user with $\frac{1}{2}2^{21} = 1,048,576$ keys (each user is contained in half of the sub-sets) so we have roughly one million keys per user and a header length of $\frac{1}{21}|\mathcal{N}|$ encrypted session keys (plus identifiers for sub-sets and batches).

There is obviously a tradeoff between the parameters key-space and header length. The private key space per user can be reduced by increasing the transmission header length and vice versa. We present a number of broadcast encryption schemes in the following sections that have these tradeoff properties and also bring along further parameters to be taken into consideration.

3.3.1.1 Early Results

Berkovits [9] studied the problem to broadcast a secret to some sub-sets of receivers and proposed to convert secret sharing schemes into secure broadcasting schemes. These

[2] When (as an example) 1 billion keys per user is considered the maximum storage of a hardware device then a user set size of 31 does already exceed this limit.

schemes can then be used only **one time** for broadcasting a secret, thus the set-up phase needs to be repeated after a broadcast. The resulting schemes are not efficient (compared to the Fiat-Naor schemes presented in the next section) and they do not fit into our definition of broadcast encryption so we do not elaborate on them further.

Before, Blom [11] studied symmetric key generation systems that could be considered as key pre-distribution systems and are thus partially applicable to the broadcast encryption scenario. The idea is to apply maximum distance seperable (MDS) codes in a way that k users need to to collude to "break" a code with a minimum distance of $n - k + 1$. This scheme was improved by Blundo *et al.* [13] in the context of key distribution schemes for dynamic conferences which solves the problem that (after an initial set-up phase) any sub-set shall be able to compute a secure group key for private group communication. They consider a non-interactive model which is applicable to the broadcast encryption problem since one group member could take the role of the broadcast center. Both schemes consider stronger requirements[3] than necessary for the broadcast situation so the resulting schemes are not as efficient as the dedicated broadcast encryption schemes presented in the next section.

3.4 The (Fiat, Naor)-Schemes

3.4.1 Fiat-Naor: Basic Scheme

The basic scheme is a straightforward construction of a broadcast encryption scheme for a given (small) resiliency parameter. The idea is to assign keys to users and sets of

[3] Broadcast encryption can be regarded as one use-case of group communication where the whole group passively listens to one member. If the requirement that every member can take over the role of the broadcast center is added then a non-interactive group communication scheme is constituted

users up to k elements in a way that all other users are provided with the key but no user in the sub-set knows the sub-set key. The non-empty set of keys known by every member of the target group are easily combined (XORed) to derive a session key that can be computed by all users in the target group but not using the combined knowledge of up to k non-privileged users as they will always miss at least one sub-set-key (*i. e.,* the key assigned to them as a k-sub-set of attackers). The exact scheme and its major properties are detailed by proving the following theorem.

Theorem 3.4.1. { Fiat and Naor [36] } *There exists an information-theoretically secure k-resilient broadcast encryption scheme for n users that requires each user to keep $\sum_{i=0}^{k} \binom{n-1}{i}$ keys and has a zero-header length.*

Proof. Let k be the desired resiliency parameter with $1 \leq k < n := |\mathcal{N}|$. We construct a broadcast encryption scheme and show that the requirements are met. Let the algorithm SETUP randomly and uniformly choose a key $K_{\mathcal{S}} \in \mathcal{K}$ for each sub-set $\mathcal{S} \subseteq \mathcal{N}$ with $|\mathcal{S}| \leq k$ and provide each user with the private key information $P_u := \left\{(\mathcal{S}_j, K_{\mathcal{S}_j}) : |\mathcal{S}_j| \leq k, u \notin \mathcal{S}_j\right\}$. As the number of i-sub-sets of $N-u$ is given by $\binom{n-1}{i}$ for every user u the number of keys per user can be added up to $\sum_{i=0}^{k} \binom{n-1}{i}$ keys. BROADCAST does output the zero length message header. The session key for target set $\mathcal{T}$ is given by

$$K := \bigoplus_{\mathcal{S} \subseteq (\mathcal{N}-\mathcal{T}) \ \wedge \ |\mathcal{S}| \leq k} K_{\mathcal{S}} \quad .$$

Every user $u \in \mathcal{T}$ can compute the $\oplus$-operation as she knows all the $K_{\mathcal{S}}$ being used in the computation. Any coalition $\mathcal{S}' \subseteq (\mathcal{N} - \mathcal{T})$ of at most k attackers is missing the key $K_{\mathcal{S}'}$ and cannot compute the session key. All keys are bit-strings of equal length so a missing key does provide information-theoretic security for the computation result. As this observation holds for any sub-set containing up to k users we have shown

k-resiliency. □

This scheme is suitable for small values of k but as the binomial coefficients grow (roughly) exponentially, the practical relevance is limited in general cases where arbitrary or maximum resiliencies are required. Regarding information-theoretic security and small values of k the scheme does provide a first bound of private user key space, though, and the scheme with parameter $k = 1$ can be used as a building block to construct schemes with higher resiliency.

3.4.2 Fiat-Naor: One-way Function Based Scheme

When information-theoretic security is not a requirement and cryptographic assumptions are possible it is possible to construct an advanced scheme that reduces the private user key space significantly. The idea from Fiat and Naor [36] is to use one-way functions for key generation. A user only needs to store some key information and can derive further keys needed to decrypt a broadcast transmission by iteratively evaluating a one-way function on these values. More precisely: For this scheme, we will need a length-doubling pseudo-random-number generator that computes keys assigned to a tree structure where the left and right successor's key is given by half of the PRNG output respectively.

Theorem 3.4.2. {Fiat and Naor [36]} *Assuming the existence of one-way functions there exists a* 1*-resilient broadcast encryption scheme for* n *users that requires each user to keep* $\lceil log\ n \rceil$ *keys and that broadcasts a zero-length header.*

Proof. We construct a broadcast encryption scheme and show that the requirements are met. Let $f : \mathcal{K} \to \mathcal{K}^2$ be a publicly known length-doubling pseudo-random-number generator where $\mathcal{K}^2$ is defined as the set of all bit-strings with the double length of the strings in $\mathcal{K}$. Setup creates a binary tree with $2^{\lceil \log(n) \rceil}$ leaves. Each user in $\mathcal{N}$ is mapped to a unique leaf, the remaining leaves (*e. g., n* is not a power of 2) are mapped to $\emptyset$ (but could be used for future users after implementation).

Beginning with the root node of the binary tree Setup recursively assigns a key and a sub-set of $\mathcal{N}$ to each node. The sub-set will either be $\{u\}$ ($\emptyset$, respectively) if the node is a leaf which is mapped to user u ($\emptyset$, respectively) or it will be the union of the sub-sets of its descendants. The key will be randomly and uniformly selected from $\mathcal{K}$ for the root node and be computed for all other nodes using the PRNG f by assigning the left half of the output of $f(k_{\mathcal{S}})$ to the left descendant and the right half to the right descendant. The input to f is the key $k_{\mathcal{S}}$ of the node assigned to the sub-set $\mathcal{S} \subseteq \mathcal{N}$ being the union of the sets assigned to the node's descendants.

A user u will get the keys assigned to the roots of all new trees that emerge when the nodes on the route from the user to the root is removed (including the root node and u). As this yields one new tree per level (except the leaf level) a forest consisting of $\lceil log\ n \rceil$ trees is constructed and each user does get the same number of keys. As we removed exactly the route of sub-sets that include the user it is easy to see that the user does now posses the key to each sub-set she does not belong to or can generate such a key by iteratively evaluating f from one new root down to the sub-set node.

Broadcast does output the zero length message header. The session key for target set $\mathcal{T}$ is given by

$$K := \bigoplus_{u \in (\mathcal{N} - \mathcal{T})} K_u \quad .$$

Notice that we are not achieving information-theoretic security here as the one-way assumption of the PRNG f does not have this property. □

Remark 3.4.3. The Setup algorithm's execution time can be reduced by not processing nodes with empty assigned sub-sets. An optimality condition is met when n is a power of 2 so we have $2^{\lceil \log(n) \rceil} = n$ and no "empty" leaf is left. In a real-word implementation it would be reasonable to choose a power of 2 that is great enough to exceed the number of users during the lifetime of the implementation and fill up empty leaves with new users subsequently.

3.4.3 Fiat-Naor: Number-theoretic Assumption Based Scheme

The scheme presented in the proof of Theorem 3.4.2 was based on the assumption that one-way functions (or pseudo-random-number generators) exist and it requires each user to store $\lceil log\ n \rceil$ keys in total. Fiat and Naor presented a further advanced scheme that reduces the storage to one key per user if a different cryptographic assumption is used: *the extraction of roots modulo an RSA composite is hard.* We put the formal requirement into the following theorem and present the scheme in the proof of the theorem.

Theorem 3.4.4. { Fiat and Naor [36] } *Assuming that root extraction modulo an RSA composite is a hard problem there exists a* 1*-resilient broadcast encryption scheme for* n *users that requires each user to keep* 1 *key and that broadcasts a zero-length header.*

Remark 3.4.5. Note, that 1 key in this setting is a number in long integer arithmetic that can still be represented by a bit string but may require a much higher string length than a symmetric key in the preceding schemes so we need to be careful when comparisons are drawn between the three schemes published by Fiat and Naor.

Proof of Theorem 3.4.4. We construct a broadcast encryption scheme and show that the requirements are met.

The algorithm SETUP selects two large primes P and Q and randomly and uniformly chooses $g \in \mathbb{Z}_{PQ}^*$. These three numbers are kept secret to all users. Each user u then receives as a public key a chosen prime p_u and as secret key information $k_u := g^{p_u}$. All public keys are issued to all users. The algorithm BROADCAST does output the zero length message header.

The session key for the target set $\mathcal{T}$ is given by

$$K := g^{\left(\prod\limits_{u \in \mathcal{T}} p_u\right)} \quad .$$

Every user u in $\mathcal{T}$ can compute

$$k_u^{\left(\prod\limits_{u' \in (\mathcal{T}-\{u\})} p_u{}'\right)} = g^{\left(\prod\limits_{u' \in (\mathcal{T}-\{u\})} p_{u'}\right) p_u} = g^{\left(\prod\limits_{u' \in \mathcal{T}} p_{u'}\right)} = K$$

and derive the session key by this computation. In order to show that users outside $\mathcal{T}$ cannot compute K we will use a contradiction to the number-theoretic assumption: a user outside $\mathcal{T}$ who is able to compute K can compute g. Our first observation is that a user $u \notin \mathcal{T}$ can compute $p_{\mathcal{T}} := \prod\limits_{u' \in \mathcal{T}} p_{u'}$ because the primes $p_{u'}$ are public information. As this product $p_{\mathcal{T}}$ is relatively prime to p_u the greatest common divisor of the two numbers is 1. Thus, the user u can calculate coefficients $a, b \in Z_{PQ}^*$ with the property

$$ap_{\mathcal{T}} + bp_u = 1$$

and when the user knows K she can also compute

$$K^a k_u{}^b = g^{ap_{\mathcal{T}}} g^{bp_u} = g^{ap_{\mathcal{T}} + bp_u} = g$$

and we have the desired contradiction to our hardness assumption. The feasibility of the coefficients computation for the relatively prime numbers is shown in [92]. □

Remark 3.4.6. The scheme only assigns one secret key per user but there are some serious issues regarding practical implementation. A user in the target group $\mathcal{T}$ needs to compute a product of $|\mathcal{T}|$ primes in long integer arithmetic; this could be a difficult task for a powerful device when millions of users are in a target group and it could easily be beyond the potentials of a smartcard controller. Moreover, the users need to store all the public keys also being long integers (one per user); this task could be a hard challenge for implementation into small devices.

We also need to be cautious when comparing the scheme used in Theorem 3.4.2 with that of Theorem 3.4.4: when focusing on private key space the number-theoretic approach used in the latter is apparently superior but when computation performance or public key space is included into these considerations the one-way function approach does catch up quickly. It is therefore of key importance to exactly name performance and space properties when different broadcast encryption schemes are compared and we will see more properties to come in other schemes.

We summarize the different schemes presented in this section in a corollary.

Corollary 3.4.7. *The Fiat-Naor construction provide broadcast encryption schemes with the following parameters (with $n := |\mathcal{N}|$) :*

Basic scheme: $\sum\limits_{i=0}^{k} \binom{n-1}{i}$ *keys per user for a k-resilient scheme. The center BC needs to store* $\sum\limits_{i=0}^{k} \binom{n}{i}$ *keys. Broadcasts are sent with a zero-length header.*

Scheme based on one-way functions: $\lceil log\ n \rceil$ *keys per user. BC needs to store* 1 *key (to generate a total of* $2n-1$ *keys mapped to the nodes of a tree). Broadcasts are* *sent with a zero-length header.*

Scheme based on root extraction problem: 1 *secret key (and* n *public keys) per user. In total* $n+3$ *keys are stored by* BC*. Broadcasts are sent with a zero-length header.*

The last two schemes are 1*-resilient. The term* zero length *refers to Definition 3.3.1.*

3.4.4 Fiat-Naor: General Construction of k-resilient Schemes

Our next task is to construct a k-resilient scheme using a 1-resilient scheme. This construction is sketched in [36] and some claims around the construction are proven in the same paper but no comprehensive proof of the assertions on the construction is given. We will do the construction and prove its properties using the notations of the preceding schemes.

Theorem 3.4.8. { Fiat and Naor [36] } *For a given* 1*-resilient broadcast encryption scheme with zero-header length and* N *users where every user is assigned* z *keys there exists a* k*-resilient broadcast encryption scheme assigning* $O(kz \log N)$ *keys with header length* $O(k^3 \log N)$*.*

Proof. Let l, m be positive integers. Let $\{f_i\}_{i=1}^{l}$ be a family of functions with $f_i : \mathcal{N} \rightarrow \{1, 2, \ldots, m\})$ with the property that for all sub-sets $\mathcal{S} \subseteq \mathcal{N}$ with $|\mathcal{S}| \leq k$ there exists one i with $u \neq u' \Rightarrow f_i(u) \neq f_i(u')$ for all $u, u' \in \mathcal{S}$ (in other words: f_i is injective on $\mathcal{S}$).

For (i, j) with $1 \leq i \leq l$ and $1 \leq j \leq m$ we will use an independent instance of the 1-resilient scheme and denote it by the triple $(\textsc{Broadcast}^{(i,j)}, \textsc{Setup}^{(i,j)}, \textsc{Decrypt}^{(i,j)})$. Let $K_u^{(i,j)}$ bet the secret key output of algorithm $\textsc{Setup}^{(i,j)}$ for user u.

Our next goal is to construct a new k-resilient scheme denoted by $(\textsc{Broadcast}, \textsc{Setup}, \textsc{Decrypt})$. Let $\textsc{Setup}$ assign to user u the secret key output $K_u := \left\{K_u^{(i,f_i(u))} : 1 \leq i \leq l\right\}$.

The new scheme's $\textsc{Broadcast}$-algorithm selects randomly and uniformly session key $K \in \mathcal{K}$ and other keys $K_1, K_2, \ldots K_{l-1} \in \mathcal{K}$ and computes $K_l := (\bigoplus_{i=1}^{l-1} K_i) \oplus K$. Let $B^{(i,j)}$ be the output of $\textsc{Broadcast}^{(i,j)}(\mathcal{T}^{(i,j)}, K_i)$ with $\mathcal{T}^{(i,j)} := \mathcal{T} \backslash \{u \in \mathcal{N} : f_i(u) \neq j\}$ and K_i as the respective session key. Then $\textsc{Broadcast}$ outputs the concatenated $B^{(i,j)}$ (being a total of lm items).

A user $u \in \mathcal{T}$ can decrypt K by computing

$$\bigoplus_{i=1}^{l} \textsc{Decrypt}^{(i,f_i(u))}(B^{(i,f_i(u))}, K_u^{(i,f_i(u))}, u) = K \tag{3.1}$$

which is possible for her because for all i we have: $u \in \mathcal{T}^{(i,f_i(u))}$.

We will now show that a coalition $\mathcal{S} \subseteq \mathcal{N} - \mathcal{T}$ cannot compute the session key K. Let i' be a parameter with the property that $f_{i'}$ is injective on $\mathcal{S}$. We will show that

$$K_{i'} := \textsc{Decrypt}^{(i',f_{i'}(u))}(B^{(i',f_{i'}(u))}, K_u^{(i',f_{i'}(u))}, u)$$

cannot be computed by the combined users in $\mathcal{S}$ and thus K cannot be computed as $K_{i'}$ is one of the l items in the $\oplus$-operation of equation 3.1.

First note, that for users in $\mathcal{S}$ the key $K_{i'}$ cannot be computed from knowing all messages $B^{(i,j)}$ with $i \neq i'$ because we have used independent instance of the 1-resilient scheme. Next, we will show that for each j that $K_{i'}$ can still not be computed when additionally the $B^{(i',j)}$ are known to the users in $\mathcal{S}$ and having proved that we can see that all $B^{(i',j)}$ are not sufficient for calculating $K_{i'}$ as the property is provided by the independent 1-resilient schemes.

Let $j \in \{1, 2, \ldots, m\}$. The function $f_{i'}$ is injective (when restricted to $\mathcal{S}$) so there is one or no $u' \in \mathcal{S}$ with $f_{i'}(u') = j$ but not more than one. We start with the case where the u' does exist. Every user $u \in \mathcal{S}$ has received for every i the keys from $\textsc{Setup}^{(i, f_i(u))}$ in the setup-phase so there is exactly one user who has received the keys from $\textsc{Setup}^{(i',j)}$ and that is user u'. In the other case where u' does not exist no user in $\mathcal{S}$ has received these keys. Thus, regarding the keys from $\textsc{Setup}^{(i',j)}$ at most one user in $u \in \mathcal{S}$ has the keys and as $\textsc{Setup}^{(i',j)}$ belongs to a 1-resilient scheme $K_{i'}$ cannot be computed by the users in $\mathcal{S}$.

We can apply Lemma 2.3.11 to select the values for l and m. Set $m := 2k^2$ and $l := \lceil k \log n \rceil$ and we have a reasonable assumption that the family $\{f_i\}_{i=1}^{l}$ does indeed contain a perfect hash function even when each f_i is an independent PRNG. For the proof, though, we will only use the fact that there exists such a family containing a perfect hash function respecting these values regardless how it is represented because we do not want to use the existence of PRNGs as a prerequisite for now.

Using the parameter setting for l and m we can verify the header length $O(k^3 \log N)$ as the message length is lm zero-header messages and the number of keys per user being $O(kz \log N)$ because each user u receives the key output $K_u := \left\{ K_u^{(i, f_i(u))} : 1 \leq i \leq l \right\}$ that is z keys per 1-resilient scheme and a total of $\lceil k \log n \rceil$ 1-resilient schemes per user. □

We can now apply the explicit construction of k-resilient broadcast encryption schemes to the 1-resilient schemes we have presented so far and get a new collection of k-resilient schemes.

Corollary 3.4.9. (i.) *There exists an information-theoretically secure k-resilient* *broadcast encryption scheme for n users that requires each user to keep $O(kn \log n)$ keys and has a header length of $O(k^3 \log n)$.*

(ii.) *Assuming the existence of one-way functions there exists a k-resilient broadcast encryption scheme for n users that requires each user to keep $O(k \log^2 n)$ keys and that has a header length of $O(k^3 \log n)$.*

(iii.) *Assuming that root extraction modulo an RSA composite is a hard problem there exists a k-resilient broadcast encryption scheme for n users that requires each user to keep $O(k \log n)$ keys and that has a header length of $O(k^3 \log n)$.*

Proof. The three claims hold by applying Theorem 3.4.8 to the three schemes laid out in in (i.) Theorem 3.4.1, (ii.) Theorem 3.4.2 and (iii.) Theorem 3.4.4 respectively where each scheme is used as a 1-resilient scheme for the construction. □

3.5 Revocation Schemes CS, SD and LSD

Naor, Naor and Lotspiech [82] introduced two new broadcast encryption schemes and some variants of it. The *Complete Subtree Method* (CS) and the *Sub-set Difference Method* (SD) are efficient schemes regarding the situation where the target set includes almost all users. This use case is of important practical relevance and other broadcast encryption schemes known at this time were rather inefficient in this scenario. We present the results of [82] using a notation consistent with the other schemes. We will also prove the asserted properties of the schemes; in the original work [82] some relevant assertions (*e. g.,* the correctness of the sub-set construction algorithm) are not proved and some subtleties regarding the number of keys were not treated exhaustively. We will put the framework concepts into a proper definition so that we can prove properties (*e. g.,* header length, user key size) of the to be used algorithms Setup and

Broadcast in the respective schemes in one comprehensive theorem. We will also evaluate the time complexity of the *Sub-set Difference Method* which was not part of the original work as well.

3.5.1 Subtree and Sub-set Revocation

Definition 3.5.1. (sub-set-cover) Let $\mathcal{N}$ be the set of users of a broadcast encryption system with target set $\mathcal{T}$. We will use the notion of set $\mathcal{R} := \mathcal{N} \setminus \mathcal{T}$ as the *revoked set* of users.

A *sub-set-cover* of $\mathcal{N}$ is defined as a collection of sub-sets $S_1, S_2, \ldots, S_w$ with $S_j \subseteq \mathcal{N}$ for all $1 \leq j \leq w$ where

(i.) each sub-set S_j is assigned a key L_j which can be derived by a user $u \in S_j$ from its secret information k_u and

(ii). for every revoked set $\mathcal{R} \subseteq \mathcal{N}$ there exists a collection of disjoint sub-sets $S_{i_1}, S_{i_2}, \ldots, S_{i_m}$ so that

$$\mathcal{N} \setminus \mathcal{R} = \bigcup_{j=1}^{m} S_{i_j}$$

Remark 3.5.2. The notion of revoked users instead of a target group is motivated by the idea that it is of practical relevance to construct schemes that are not required to support arbitrary target sets but shall be able to provide a broadcast key efficiently for the case were almost all users are in the target set but a rather small amount of users is revoked so the broadcast is generally intended for the whole user base but shall not be decryptable for users having been identified as pirates or traitors. When we implicitly assume that $|\mathcal{R}| \ll |\mathcal{N}|$ we use the term *revocation scheme* as a distinct broadcast encryption scheme.

Definition 3.5.3. (Complete Subtree Method) Let $\mathcal{N}$ be the set of users of a broadcast encryption system with revoked set $\mathcal{R}$. The number of users $n := |\mathcal{N}|$ shall be *w.l.o.g.* a power of 2.

The *Complete Subtree Method* specifies the algorithms Setup, Broadcast, Decrypt in the following way.

The algorithm Setup builds a full binary tree with the users as leaves with n leaves, $2n-1$ nodes and $\log n + 1$ path length from root to any leaf. Let v_i with $1 \leq i \leq 2n-1$ denote the nodes of the tree including the leaves. For each v_i we define a sub-set S_i as the set of all users having v_i as ancestor. To every node v_i a random and uniformly chosen key L_i is assigned and each user is provided with all node keys along the path from root to herself.

The algorithm Broadcast selects a session key K and takes $\mathcal{R}$ as input. Let $u_1, u_2, \ldots, u_r$ be the users in $\mathcal{R}$. Next, the algorithm sets up a Steiner tree $ST(\mathcal{R})$ according to Definition 2.4.15, yielding the minimal subtree that connects the root node to all revoked users. The sub-sets $S_{i_1}, S_{i_2}, \ldots, S_{i_m}$ are defined as the family of all subtrees of the full binary tree whose roots $v_1 \ldots v_m$ are not in $ST(\mathcal{R})$ and are adjacent to nodes of outdegree 1 in $ST(\mathcal{R})$.

Then Broadcast encrypts K using a key encryption algorithm E with keys $L_{i_1}, L_{i_2}, \ldots, L_{i_m}$ respectively and sends the tuple $(i_1, i_2, \ldots, i_m, E_{L_{i_1}}(K), E_{L_{i_2}}(K), \ldots E_{L_{i_m}}(K))$ on the broadcast channel.

The algorithm Decrypt of user u receives the tuple $(i_1, i_2, \ldots, i_m, C_1, C_2, \ldots C_m)$ where the C_i represent the encrypted session key and finds the i_j with $u \in S_{i_j}$. By using the key L_{i_j} the session key is decrypted using $K = E^{-1}_{L_{i_j}}(C_j)$ or the executions stops when $u \notin S_{i_j}$ (for all $1 \leq j \leq m$) because the user is revoked and no decryption

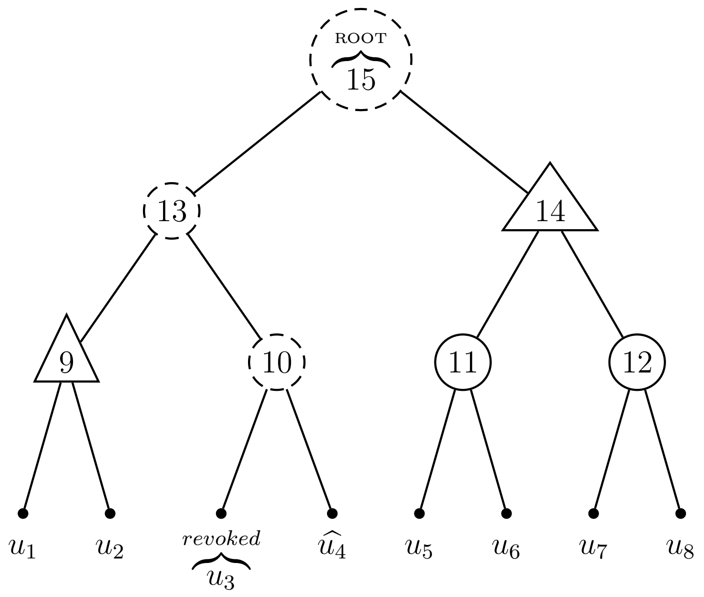

Figure 3.1: Complete Subtree Construction Example

key is available.

Example 3.5.4. See Figure 3.1 as an example of the construction for one revoked user u_3 of a total of 8 users thus generating a complete binary tree with 15 nodes and the corresponding Steiner tree consisting only of the path from the root node to u_3 and yielding subtrees with the following roots: user u_4 (representing its own sub-set $\{u_4\}$), node 9 (with sub-set $\{u_1, u_2\}$) and node 14 (with sub-set $\{u_5, \ldots, u_8\}$).

Theorem 3.5.5. { Naor, Naor and Lotspiech [82] } *The Complete Subtree Method provides a sub-set-cover of the users $\mathcal{N}$ and a broadcast encryption scheme (here: revocation scheme) with a header length of $O(r \log \frac{n}{r})$ and user key size of $O(\log n)$.*

Remark 3.5.6. For the sake of completeness we have to consider the case where no user is revoked ($r = 0$) and we have a zero header length as the sub-set key of $\mathcal{N}$ can be used. We will disregard this case of division by zero because it is obvious how the header looks like in the $(r = 0)$ case from Definition 3.5.3 where the Steiner tree is empty and only one sub-set is in the family.

Proof of Theorem 3.5.5. We show first that each user can derive the key encryption key for the session key from its private information and that the sub-sets are a disjoint cover to prove that a sub-set-cover according to definition 3.5.1 is achieved.

Because each user u is provided with all node keys along the path from root to herself and the sub-sets are exactly the leaves of the subtrees whose root nodes are ancestors of u it is obvious that the user knows all assigned keys to the sub-sets she belongs to. Thus she can "derive" the key encryption key for session key K from the private information as this key encryption key is one of the known sub-set keys. A revoked user, though, is not a member of any of the sub-sets $S_{i_1}, S_{i_2}, \ldots, S_{i_m}$ and thus does not know the assigned keys. As the sub-set keys are chosen independently the revoked user cannot deduce information about the keys from his private key information and thus cannot decrypt the session key.

To see that the sub-sets are a disjoint cover of $\mathcal{N} \setminus \mathcal{R}$ we first note, that all revoked users are leaves in the Steiner Tree and therefore cannot be members of one of the sub-sets because every revoked user and every ancestor of a revoked user cannot be a root of a subtree defining a sub-set. Regarding disjointness first note, that two subtrees of a binary tree are either disjoint or the root of one tree is a node in the other tree. By definition we only selected subtrees whose roots were adjacent to nodes of outdegree 1 in $ST(\mathcal{R})$. When we fix such a subtree we can see that a node other than its root cannot be adjacent to nodes of outdegree 1 in $ST(\mathcal{R})$ as it is either a leaf or its two children are not adjacent to such a node and we can repeat the argument recursively.

To show that the header length is $O(r \log \frac{n}{r})$ we consider that the number of encrypted keys in the message header is equal to the number of sub-sets or subtrees generated by the algorithm and thus is equal to the number of degree 1 nodes of $ST(\mathcal{R})$. We prove that the number of degree 1 nodes is at most $r \log \frac{n}{r}$ by induction on the tree depth. A tree with 1 node has no vertices so the number of degree 1 nodes of $ST(\mathcal{R})$ is 0. Next, we assume that the assertion holds for a tree of depth t (*i. e.,* $n = 2^t$) and we consider a tree of depth $t + 1$.

We regard two cases:

Case 1: All leaves in $\mathcal{R}$ are descendants from only one child of the root node. We can use the inductive assertion then for the subtree whose root is either the left or the right child of the root node as is has subtree depth t. The maximum number of degree 1 nodes is then 1 (root node) plus at most $r \log \frac{2^t}{r}$ and we have

$$1 + r \log \frac{2^t}{r} \leq r \log \frac{2^{t+1}}{r} = r \log \frac{n}{r}$$

and the induction is complete for case 1.

Case 2: The leaves in $\mathcal{R}$ are descendants from both children of the root node and each child is ancestor of at least one revoked user. For depth t we separate the number r of revoked users by $r = r_{\swarrow} + r_{\searrow}$ being the numbers of revoked users in the left ($r_{\swarrow}$) and right ($r_{\searrow}$) subtree of the root node. Note, that the number of degree 1 nodes are also the sum of the degree 1 nodes in the left and right subtree. The number of degree 1 nodes in the left subtree is by induction at most $r_{\swarrow} \log \frac{2^t}{r_{\swarrow}}$ as well as for the right subtree $r_{\searrow} \log \frac{2^t}{r_{\searrow}}$. We have

$$r_{\swarrow} \log r_{\swarrow} + r_{\searrow} \log r_{\searrow} + r \geq r \log r$$

and we can conclude that the overall number of degree 1 nodes is at most $r \log \frac{2^{t+1}}{r} = r \log \frac{n}{r}$ and the induction is complete for both cases.

The user key size is given by the tree depth because a user is provided with one key per sub-set along the path from the user to the root node getting $\log n + 1$ keys in total. □

Remark 3.5.7. Note, that information-theoretic security (for the encrypted session key) is achieved by construction in the proof of Theorem 3.5.5 because the sub-set keys given to each user are the node keys that were chosen randomly and uniformly and no other keys derived from this or other information were used.

Definition 3.5.8. (The Sub-set Difference Method) Let $\mathcal{N}$ be the set of users of a broadcast encryption system with revoked set $\mathcal{R}$. The number of users $n := |\mathcal{N}|$ shall be a power of 2 The *Sub-set Difference Method* specifies the algorithms Setup, Broadcast, Decrypt in the following way.

The algorithm Setup builds a full binary tree with the users as leaves generating a tree with n leaves.

Let v_i with $1 \leq i \leq 2n - 1$ denote the nodes of the tree including the leaves. For each pair (v_i, v_j) where v_j is a descendant of v_i we define a sub-set $S_{i,j}$ as the set of all users having v_i as ancestor and not having v_j as ancestor thus depicting the *difference* of two subtrees and the difference of their assigned set of leaves being sub-sets of $\mathcal{N}$ respectively. Setup assigns to each sub-set $S_{i,j}$ a key $L_{i,j}$ and each user is provided with private key information making it possible for her to derive the key $L_{i,j}$ iff she is a member of $L_{i,j}$.

The algorithm Broadcast selects a session key K and takes $\mathcal{R}$ as input. Let $u_1, u_2, \ldots, u_r$ be the users in $\mathcal{R}$. Next, the algorithm selects a family of disjoint sub-sets $S_{i_1,j_1}, S_{i_2,j_2}, \dots, S_{i_m,j_m}$ whose union is the set $\mathcal{N} \setminus \mathcal{R}$.

Then BROADCAST encrypts K using a key encryption algorithm E with keys $L_{i_1,j_1}, L_{i_2,j_2}, \dots, L_{i_m,j_m}$ respectively and sends the tuple $((i_1, j_1), (i_2, j_2), \dots, (i_m, j_m), E_{L_{i_1,j_1}}(K), E_{L_{i_2,j_2}}(K), \dots E_{L_{i_m,j_m}}(K))$ on the broadcast channel.

The algorithm DECRYPT of user u receives the tuple $((i_1, j_1), (i_2, j_2), \dots, (i_m, j_m), C_1, C_2, \dots C_m)$ where the C_i represent the encrypted session key and finds k with $u \in S_{i_k,j_k}$. By using the key L_{i_k,j_k} the session key is decrypted using $K = E^{-1}_{L_{i_k,j_k}}(C_k)$ or the executions stops when $u \notin S_{i_k,j_k}$ (for all $1 \leq k \leq m$) because the user is revoked and no decryption key is available.

Theorem 3.5.9. *Let $\mathcal{N}$ be the set of users of a broadcast encryption system with revoked set $\mathcal{R}$ and let $n := |\mathcal{N}|$. By using length-tripling PRNGs it is possible to assign $\frac{1}{2}\log^2 n + \frac{1}{2}\log n + 1$ keys per user and to fulfill the requirements for the algorithm* SETUP *of the Sub-set Difference Method.*

Proof. We describe a SETUP algorithm that meets the requirements.

By definition of the Sub-set Difference Method each sub-set $S_{i,j}$ contains the users having node v_i as ancestor but not node v_j as ancestor.

SETUP first assigns to each of the $n - 1$ nodes not being a leaf a meta key ℓ_i (called a *label* of node v_i) which is chosen randomly and uniformly for all $1 \leq i \leq n - 1$.

Let G be a PRNG taking bit-strings of a fixed length and outputting bit-strings with triple length. We split the output in three equal length parts from left to right: Let $G_L(x)$ be the left part of the output string on input x, $G_M(x)$ the middle part and $G_R(x)$ the right part.

Next SETUP assigns recursively a meta key $\ell_{i,j}$ to each sub-set $S_{i,j}$. For each node

v_i not being a leaf we assign to all sub-sets $S_{i,j}$ meta key $\ell_{i,j}$ by

- $\ell_{i,j} := G_L(\ell_i)$ in case v_j is the left child of v_i
- $\ell_{i,j} := G_R(\ell_i)$ in case v_j is the right child of v_i
- $\ell_{i,j} := G_L(\ell_{i,k})$ in case $v_k \neq v_i$ is parent of v_j and v_j is v_k's left child
- $\ell_{i,j} := G_R(\ell_{i,k})$ in case $v_k \neq v_i$ is parent of v_j and v_j is v_k's right child

The sub-set key $L_{i,j}$ for sub-set $S_{i,j}$ is then given by $L_{i,j} := G_M(\ell_{i,j})$.

Each user u is given all meta keys $\ell_{i,j}$ having all three properties:

i. v_j is descendant of v_i.

ii. u is descendant of v_i.

iii. u is descendant of **the sibling of** v_j.

We need to show now that each user can derive the key $L_{i,j}$ iff she is a member of $S_{i,j}$.

A user $u \in S_{i,j}$ is by definition a descendant of v_i but not of v_j. If u is a descendant of the sibling of v_j then she received $\ell_{i,j}$ directly and can compute the sub-set key $L_{i,j} := G_M(\ell_{i,j})$ and we are finished. Otherwise we can follow the path from v_j up to v_i and stop at a node v_x whose sibling $v_{x'}$ is an ancestor of u (existence is given by the fact that v_i's children at the latest are such nodes if no other on the way were.) User u has received the meta key $\ell_{i,x}$ and can now follow the path down from v_x to v_j and compute on the way for every node v_y with $G_L(\ell_{i,y})$ or $G_R(\ell_{i,y})$ the meta key for a left or a right child one after another until $\ell_{i,j}$ and then $L_{i,j} := G_M(\ell_{i,j})$ is computed.

A user $u \notin S_{i,j}$ is either not a descendant of v_i (and has then no meta keys that were computed from the random meta key ℓ_i) or it is a descendant from v_j. In the latter case we can look at the path from v_i to v_j that follows the computation of meta keys $\ell_{i,y}$ for nodes v_y on this way and we can see that none of these keys are provided to the user as no sibling of the nodes on the path is an ancestor of u thus u is not given any of the meta keys that are needed to recursively calculate $\ell_{i,j}$. Note, that also ℓ_i is not known to u as it is not given to any user.

We will now count the number of keys provided to each user. First, there is one key for the case $\mathcal{R} = \emptyset$. Next, for a node v_i with δ nodes between the leaf u and v_i there are δ meta keys ℓ_{i,j_k} with $1 \leq k \leq \delta$ provided to u. For every such δ that can take values from 0 to $\log n$ there exists exactly one node v_i per user u used for key assignment. Thus the number of keys $|I_u|$ per user is given by

$$|I_u| = 1 + \sum_{\delta=0}^{\log n} \delta = 1 + \frac{1}{2}(\log n)(1 + \log n) = \frac{1}{2}\log^2 n + \frac{1}{2}\log n + 1$$

and all assertions of the theorem are proved. □

Remark 3.5.10. With Theorem 3.5.9 we have shown that an algorithm Setup for the Sub-set Difference Method does exist and we can calculate the number of keys per user for this algorithm. We do not need to show that a disjoint cover of $\mathcal{N} \setminus \mathcal{R}$ exists for any Broadcast algorithm of this method because by definition of the Sub-set Difference Method for each pair (v_i, v_j) where v_j is a descendant of v_i a sub-set $S_{i,j}$ is defined and we can use the special sub-sets where v_j is a child of v_i to get sub-sets that include exactly the nodes of a sub**tree**. These subtrees are constructed by rooting a larger subtree at the parent node and then removing the sibling's subtree. Moreover, this way we can construct the set of nodes for all subtrees of the full binary tree except the full tree itself having no parent node (but we assigned a key to the sub-set $\mathcal{N}$ anyway). So we can use the same construction as in the Complete Subtree Method to get a disjoint cover of the set $\mathcal{N} \setminus \mathcal{R}$.

There are more sub-sets available compared to the Complete Subtree Method so we can expect that in many cases we will need fewer sub-sets to cover the target set.

Next, we are interested in the number of sub-sets of the form $S_{i,j}$ that are required by a Broadcast algorithm to cover the set $\mathcal{N} \setminus \mathcal{R}$ as this is also the number of ciphertexts representing the session key in the broadcast message header. It is also of interest how Broadcast can find such a cover in a constructive way. This is considered in the next theorem. Before we prove a lemma being helpful in the proof of the theorem.

Lemma 3.5.11. *Let T be a full binary tree with where each leaf is either marked or unmarked and at least two leaves are marked. There exists a node that is ancestor of exactly two marked nodes.*

Proof. We show how to find a desired node in the tree. Let each node be assigned the number of marked leaves being descendants of the node. We can walk along a path from root to the desired node. When root is assigned the number 2 we are finished otherwise we examine the two children of root: again we are finished if a number 2 is found; if not then we walk to one child being assigned 3 or higher and repeat the last step until a node with 2 is found. Existence is given by the fact that a visited node's number is equal or smaller than the parent's number and that a marked node's parent is assigned a 1 or a 2 and that the number 2 cannot be skipped either so we will always stop on the way. □

Theorem 3.5.12. *Let $\mathcal{N}$ be the set of users of a broadcast encryption system with revoked set $\mathcal{R}$ and let $r := |\mathcal{R}|$. There exists an algorithm* Broadcast *that finds a disjoint family of $2r - 1$ sub-sets in $O(r \log n + r)$ time and meets the requirements of the Sub-set Difference Method.*

Proof. We show how Algorithm Broadcast will construct disjoint sub-sets $S_{i_1,j_1}, S_{i_2,j_2}, \dots, S_{i_m,j_m}$ whose union is the set $\mathcal{N} \setminus \mathcal{R}$.

Let $ST(\mathcal{R})$ be the Steiner tree of $\mathcal{R}$ being the tree that is derived from the full binary tree containing all users by first deleting all vertices and then adding all paths from root to users in $\mathcal{R}$ and removing the unconnected nodes. Broadcast uses a tree T as working space. It is first set equal to $ST(\mathcal{R})$ and then reduced iteratively by one or more nodes in a loop structure but ever remains a connected tree; Broadcast leaves this loop when T is reduced to a single node.

Loop:

1. We can apply Lemma 3.5.1 and find two leaves λ_1 and λ_2 such that the nearest common ancestor Λ roots a subtree of the full binary tree that does not contain any other leaf of T beside the leaves λ_1 and λ_2. The child of Λ being the ancestor of λ_1 is denoted by λ'_1. If λ_1 is itself a child of Λ we set $\lambda'_1 := \lambda_1$. The same is done for a child λ'_2 that is either an ancestor or λ_2 itself. If no two leaves λ_1 and λ_2 can be found because only one leaf λ_0 is left then we set $\lambda'_1 := \lambda'_2 := \lambda_1 := \lambda_2 := \lambda_0$ and Λ be the root of T (note, that Λ could also be equal to the single leaf when only one node was in T in the very beginning).

2. If $\lambda'_1 \neq \lambda_1$ we add sub-set $S_{a,c}$ to the family; if $\lambda'_2 \neq \lambda_2$ we add $S_{b,d}$. If $\lambda'_1 = \lambda'_2$ and $\lambda'_1 \neq \Lambda$ then add $S_{0,a}$ to the family.

3. We remove the subtree rooted at Λ from T but add Λ again at its old position as a new leaf.

Until: there is only one node left in T.

Let m be the number of sub-sets in the family. Finally, Broadcast encrypts the session key K using a key encryption algorithm E with keys $L_{i_1,j_1}, L_{i_2,j_2}, \ldots, L_{i_m,j_m}$ respectively and constructs the output tuple $((i_1, j_1), (i_2, j_2), \ldots, (i_m, j_m), E_{L_{i_1,j_1}}(K), E_{L_{i_2,j_2}}(K), \ldots E_{L_{i_m,j_m}}(K))$ as required in Definition 3.5.8.

We can see from Lemma 3.5.1 that once a data structure is set up that assigns to each node the number of revoked users being descendants of the node in the full binary tree we can find the two leaves λ_1 and λ_2 in step 1 in at most $\log n$ steps.

In order to find an upper bound for m we observe that in step 1 a maximum of two sub-sets is added to the family because the third condition can only hold if the first and second does not hold. In the last iteration the third condition can hold and then a maximum of one sub-set is added before the loop is left. The number of leaves in T is reduced by one in each iteration (two leaves are removed and one is added); only in the final step it can happen that one leaf is in T before and after the iteration while exactly one sub-set is added. So $r-1$ times up to 2 sub-sets are added as the loop starts with r leaves and finishes with 1 leaf and one time another sub-set may be added. Adding this up we have a maximum of $2r-1$ sub-sets in the family, hence, the header length is $O(2r-1)$ because the output tuple contains $m \leq 2r-1$ encrypted session keys.

Broadcast needs at most $(r-1)\log n$ operations to find all pairs of leaves λ_1 and λ_2 in step 1 plus up to $2r-1$ operations to set up the output tuple so we get $O(r\log n+r)$ for the processing time of the algorithm.

It remains to show that the constructed sub-sets are disjoint and that we have a proper partition, *i. e.*,

$$\mathcal{N} \setminus \mathcal{R} = \bigcup_{k=1}^{m} S_{i_k,j_k}$$

shall hold. We will show this in the following. It might be helpful for the reader to use the example Fig. 3.2 while going through the next steps in the proof.

Regarding a user $u \in \mathcal{R}$ we observe that u is a leaf of T in the beginning and that for every iteration in step 2 while u is still in T a sub-set is added that does either exclude

u explicitly when u is equal to a node λ_1' or λ_2' (or both) or u is not a descendant of Λ as no other leaves than λ_1' and λ_2' could be descendants so no sub-set containing u is ever added. After u is removed from T an ancestor of u is a new leaf in T and any new sub-set being added afterwards does exclude this ancestor and all descendants by the same argument.

Regarding a user $u \notin \mathcal{R}$ we have three cases. 1: an iteration before the subtree containing u is cut, 2: an iteration while the subtree is cut and 3: an iteration after the subtree is cut. As the algorithm stops the loop with only the root node left and the user leaf can only once get cut out so we can see that case 2 happens exactly once while the other cases may happen never, once or more than once.

Case 1: When sub-sets are added while a subtree is cut out that (in the full binary tree) does not contain u it is clear that u cannot be a member of these sub-sets as they only contain leaves being descendants of the cutting point.

Case 2: Lets assume without loss of generality u is (in the full binary tree) a leaf in the left child's subtree of the cutting point. The left child of the cutting point in T cannot be a leaf because a subtree containing u has not been cut out yet. Thus the condition ($\lambda_1' \neq \lambda_1$) is met and a new sub-set is added containing the leaves of the subtree rooted at this left child but not containing the leaves in the subtree of the other selected node being either a revoked user or the leaves of another subtree that was cut out in an iteration before. Hence u is included in the first new added sub-set but cannot be included in a second new sub-set.

Case 3: When sub-sets are added after u is cut out and u is a descendant of the cutting point we observe that the ancestor of u currently being a leaf must be one selected leaf and is explicitly excluded together with its descendants from the sub-set

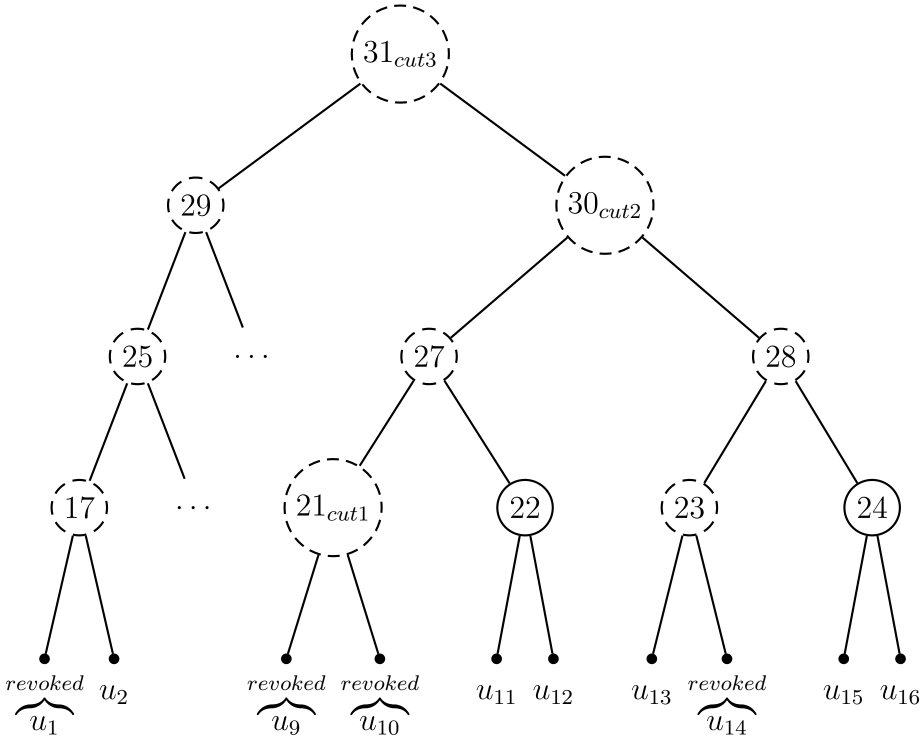

Figure 3.2: Sub-set Difference: full binary tree vs. Steiner tree

and thus u is excluded as well. When u is not a descendant of the cutting point it cannot be a member of the sub-set anyway.

Disjointness has been shown implicitly because we have seen that each user is a member of exactly one sub-set that was added to the family so an intersection of any two different sub-sets cannot contain any user. □

Example: We see in Fig. 3.2 an example construction of the sub-sets according to the proof of Theorem 3.5.12. We have a user set $\mathcal{N} := \{u_1, u_2, \ldots, u_{16}\}$ and revoked users $\mathcal{R} := \{u_1, u_9, u_{10}, u_{14}\}$. The Steiner tree being the tree T in the algorithm is denoted by the dashed line nodes and the revoked users' leaves.

Iteration 1: The algorithm selects u_9 and u_{10} in the first round and will cut the

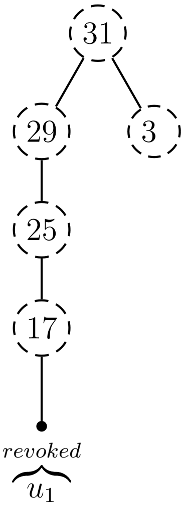

Figure 3.3: Subtree Difference: tree T after iteration 2

subtree at node 21 which becomes a new leaf. No new sub-sets are added to the family because the left and right child of node 21 is equal to the selected leaf (conditions $\lambda'_1 \neq \lambda_1$ and $\lambda'_2 \neq \lambda_2$ are not met) and the two leaves are not equal (condition $\lambda'_1{=}\lambda'_2$ is not met).

Iteration 2: The algorithm selects nodes 21 and u_{14} and adds the two sub-sets $S_{27,21}$ and $S_{28,14}$ to the family, node 30 is the cutting point and becomes a new leaf and we have a tree T as pictured in Fig. 3.3.

Iteration 3: The algorithm selects nodes u_1 and 30 and add the sub-set $S_{29,1}$ to the family (the other child of common ancestor 31 is the selected node 30 so the condition $\lambda'_2 \neq \lambda_2$ is not met), root node 31 is the cutting point and becomes a new leaf and the single left node in the tree T so the loop processing is stopped.

The constructed family is $\{S_{27,21}, S_{28,14}, S_{29,1}\}$ and the output tuple is

$$\textsc{Broadcast} \rightarrow ((27,21),(28,14),(29,1),E_{L_{27,21}}(K),E_{L_{28,14}}(K),E_{L_{29,1}}(K))$$

accordingly.

We summarize the schemes presented in this section:

Corollary 3.5.13. *The Complete Subtree (CS) and Sub-set Difference (SD) method provide broadcast encryption schemes with the following parameters (with* $n := |\mathcal{N}|$ *and* $r := |\mathcal{N}| - |\mathcal{T}|$*) :*

CS*: There are* $\lceil \log\ n \rceil$ *keys assigned per user, BC needs to store a total number of* $2n - 1$ *keys and the broadcast header consists of (at most)* $\lceil r \log\ \frac{n}{r} \rceil$ *messages.*

SD*: There are* $\left\lceil \frac{1}{2} \log^2 n\ \frac{1}{2} \log\ n \right\rceil + 1$ *keys assigned per user, BC stores* $n - 1$ *(label) keys, and the broadcast header consists of (at most)* $2r - 1$ *messages.*

Both schemes offer maximum resilience.

3.5.2 Layered Sub-set Difference

The major improvement of the Sub-set Difference method over the Complete Subtree method is the increase in the number of available sub-sets of the set of all users (with designated corresponding sub-set keys) performed in a way that a single user can derive sub-set keys for a larger number of sub-sets she belongs to. Thus the most efficient cover of a given privileged set can be constructed with fewer sub-sets because the appropriate sub-sets can be chosen from a larger pool. This advantage is paid for by the number of keys to be stored by each user that is increased from (roughly) $\log n$ to $\log^2 n$ (meta)keys per user. We realized a tradeoff between user private key size and broadcast message header length.

The quadratic increase of keys per user might be unacceptable for certain applications, particularly with regard to smartcard based Pay-TV the size of user key space is

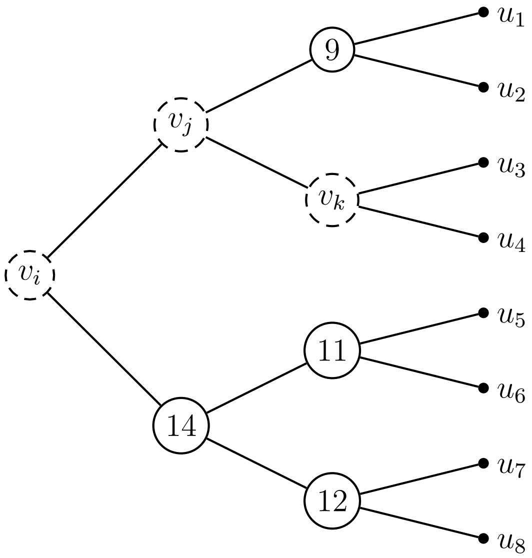

Figure 3.4: Redundant Sub-set $S_{i,k} = S_{i,j} \bigcup S_{j,k}$ of the SD Method

generally a hard limit for a smartcard and cannot be exceeded because the smartcard is normally the only secure key container in this application of broadcast encryption.

Halevy and Shamir [49] proposed a modification of the Sub-set Difference method that addresses the user's private key size. The improvement is motivated by the fact that the (large) number of sub-sets that is provided by the SD method contains some sub-sets that are never needed in a cover construction so these sets are regarded as superfluous (or at least be seen as redundant when other sets are available that could be used alternatively). The goal is to reduce the pool of available sub-sets in a way that predominantly only superfluous and redundant sub-sets are eliminated while at the same time the number of meta keys needed to derive the remaining sub-set's keys could be decreased. The remaining family of sub-sets is called the family of *useful* sub-sets.

An example for superfluous sets are all sets of the form $S_{i,i} = \emptyset$ being the empty set assigned a different sub-set key for each vertex i. A redundancy case is given by a collection of three sub-sets $S_{i,j}$, $S_{j,k}$, $S_{i,k}$ for pairwise different indices i, j, k as such an $S_{i,k}$ can obviously be expressed by the disjoint union of $S_{i,j}$ and $S_{j,k}$. See Fig. 3.4 for an example where $S_{i,k} = \mathcal{N} \setminus \{u_3, u_4\} = S_{i,j} \bigcup S_{j,k}$.

The LSD scheme's basic idea is to restrict the sub-sets used in the cover constructions to those $S_{i,j}$ where v_i and v_j are close to each other and the nodes are grouped in *layers* of close nodes.

Next, we give a formal description of Halevy and Shamir's scheme following the same notation used for the other broadcast encryption schemes presented in this work.

Definition 3.5.14. (Layered Sub-set Difference scheme) Let $\mathcal{N}$ be the set of users of a broadcast encryption system with revoked set $\mathcal{R}$. The number of users $n := |\mathcal{N}|$ shall be a power of 2 such that $\sqrt{\log n}$ is an integer. The *Layered Sub-set Difference scheme* specifies the algorithms Broadcast and Decrypt in the same way as it is done in Definition 3.5.8 for the Sub-set Difference Method. Regarding the Setup algorithm the following construction is used.

A set $S_{i,j}$ with $i \neq j$ is tagged *useful* if $\sqrt{\log n}$ is a divisor of the path length from root to node v_i or if the path lengths from root to v_i and from root to v_j each divided by $\sqrt{\log n}$ result in the same round up integer value. The path length is counted as the number of edges between the nodes.

The algorithm Setup assigns to each useful sub-set $S_{i,j}$ a key $L_{i,j}$ and each user is provided with private key information making it possible for her to derive the key $L_{i,j}$ iff she is a member of $L_{i,j}$.

Remark 3.5.15. The definition identifies the sub-sets that are sufficient for covering the privileged set of users in every transmission. It is not obvious that the family of these useful sub-sets has the same properties as the larger family of sub-sets used for the Sub-set Difference method so this has to be proved. Moreover, it is not clear from the definition how a smaller number of meta keys is to be selected and provided to each user so that a significant reduction of key space can be achieved while each user can derive at least all assigned sub-set keys of useful sets she belongs to. The latter problem is solved by the next theorem.

Theorem 3.5.16. *Let $\mathcal{N}$ be the set of users of a broadcast encryption system with revoked set $\mathcal{R}$ and let $n := |\mathcal{N}|$ be a power of 2 such that $\sqrt{\log n}$ is an integer. Then it is possible to assign $O(\log^{3/2} n)$ keys per user and to fulfill the requirements for the algorithm* Setup *of the Layered Sub-set Difference method.*

Proof. We describe a Setup algorithm that meets the requirements. The algorithm shall be the same as the Setup algorithm of the Sub-set Difference method (described in the proof of Theorem 3.5.9) apart from the following differences regarding user private key assignment.

Fixing a user u we count the number of useful sets $S_{i,j}$ the user belongs to. In the case where $\sqrt{\log n}$ is a divisor of the path length from root to node v_i we can regard the path from root to u with length $\log n$ and see that each v_i must be a node in this path and there is a total of $\sqrt{\log n}$ possible v_i's in this case. For each of these v_i's there are at most $\log n$ nodes below that could become a possible v_j so we count $O(\sqrt{\log n} \log n) = O((\log^{1/2} n) \log n) = O(\log^{3/2} n)$ different useful $S_{i,j}$ the user belongs to in this case.

In the other case where the path lengths from root to v_i and from root to v_j each divided by $\sqrt{\log n}$ result in the same round up integer value we notice that these pairs (v_i, v_j) are located in distinct layers of close nodes with path lengths of up to

$\sqrt{\log n}$ inside a layer and $\sqrt{\log n}$ layers in total. Inside a layer the number of possible (v_i, v_j) for user u is then $O(\sqrt{\log n}\sqrt{\log n}$ and adding this up for all layers we get $O(\sqrt{\log n}\sqrt{\log n}\sqrt{\log n}) = O(\log^{3/2} n)$ different useful $S_{i,j}$ the user is member of.

Combining the numbers of both cases we have $O(\log^{3/2} n)$ useful sub-sets in total for each user and Setup can provide each user with the corresponding sub-set keys and the assertion is proved. □

The next task is to show that the useful sets are really useful, *i. e.,* that these sets are sufficient to find a cover.

Theorem 3.5.17. *Let $\mathcal{N}$ be the set of users of a broadcast encryption system with revoked set $\mathcal{R}$ and let $n := |\mathcal{N}|$ be a power of 2 such that $\sqrt{\log n}$ is an integer. There is an algorithm* Broadcast *that generates messages with header length $4r - 2$ with $r := |\mathcal{R}|$ that fulfills the requirements of the Layered Sub-set Difference method.*

Remark 3.5.18. Note, that the Decrypt algorithm is defined for the Layered Sub-set Difference method in the same way as in Definition 3.5.8 (for the SD method) and does not need to be specified further because there is no ambiguity how to decrypt the message after receiving the broadcast message tuple outputted by the Broadcast algorithm. So, once we have presented a Setup and a Broadcast fulfilling the requirements we have shown the existence of a (complete) broadcast encryption scheme that is an instance of a Layered Sub-set Difference scheme.

Proof of Theorem 3.5.17. We re-use the results of the Sub-set Difference Scheme and show that given a cover $\mathcal{N} \setminus \mathcal{R} = \bigcup_{k=1}^{m} S_{i_k,j_k}$ (with $m \leq 2r - 1$) of not necessarily useful sets S_{i_k,j_k} we can construct a cover of useful sets because we will show that for all $1 \leq k \leq m$ the set S_{i_k,j_k} is either useful or the union of two useful sets so it could be replaced by useful sets in the construction.

Given any non-useful set $S_{i,k}$ we observe that v_i and v_k cannot be in the same layer (because otherwise the set would be useful). We can construct a partition $S_{i,k} = S_{i,j} \bigcup S_{j,k}$ by selecting a v_j where v_j is in the same layer as v_i and where the path from root to v_j is divided by $\sqrt{\log n}$. The node v_j is on the lower border of the layer so the value given by path length divided by $\sqrt{\log n}$ is already an integer and does not change after being rounded up. It is clear that then both $S_{i,j}$ and $S_{j,k}$ are useful as each set fulfills one of the cases in the definition of usefulness and the replacement can be done.

We have shown the existence of a partition of useful sets; finally, we want to count the number of these sets. Applying Theorem 3.5.12 we know that the SD method requires up to $2r - 1$ sub-sets. Each of those is either useful or to be replaced by 2 useful sets so a total of $4r - 1$ sets are needed and the message header length is likewise given by this value. □

Example 3.5.19. See Fig. 3.5 as an example for the construction of useful sets as depicted in the proof of Theorem 3.5.17. There are $n = 512$ users, thus $\sqrt{\log n} = 3$, we fix user u_1 and are given the not useful set $S_{i,k}$. The selected node v_j is then the left child of v_i as the path length from root to v_j is 3 (which is dividable by 3) and we have two new useful sets $S_{i,j}$ and $S_{j,k}$ that can replace $S_{i,k}$.

Remark 3.5.20. The improvements of the Layered Sub-set method can be generalized by reducing the family of available sub-sets further. The line of improvement is among other things based on the fact that a sub-set elimination technique like

$S_{i,k}$ is removed if $S_{i,k} = S_{i,j} + S_{j,k}$ and $S_{i,j}$,$S_{j,k}$ are available

can be iterated so sub-sets could also be removed when they are expressed by a union of three or more other sub-sets. Following this approach we can strike some of the

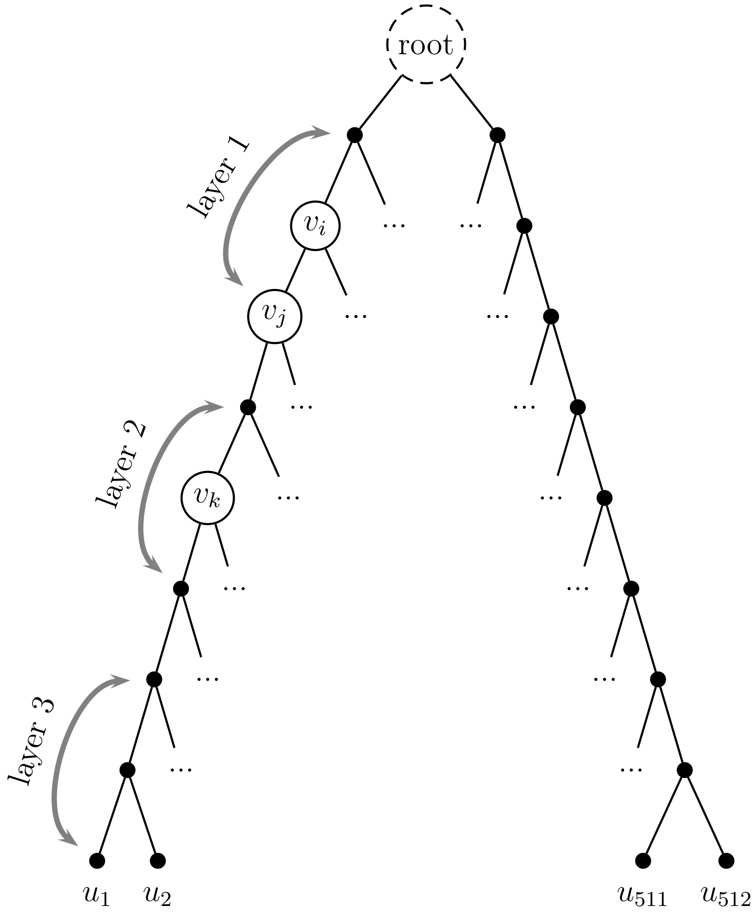

Figure 3.5: Layered Sub-set Difference for 512 users

useful sub-sets but could still find a cover for any privileged set. Halevy and Shamir [49] considered this and showed that the number of keys per users can be theoretically reduced to $O(\log^{1+\epsilon} n)$ for any given $\epsilon > 0$. However, the minimum number of users is quickly becoming astronomical in a scheme for small values of ϵ so the authors could not see a realistic use case. There are also other published improvements to LSD [7, 42, 79]. Since one goal of this dissertation is to compare different known and new schemes for realistic scenarios we do not elaborate on the the generalization of the LSD method further.

3.5.3 Logical Key Hierarchy

The *Logical Key Hierarchy* (LKH) scheme was proposed independently by Wong, Gouda and Lam (University of Texas at Austin) [103] and by Wallner, Harder and Agee (National Security Agency) [101] both in 1997. The NSA group published the concept as a proposed Internet standard (*Request for Comments 2627*) and LKH is nowadays known in many variants (*e. g.,* [17], [87]). In this thesis we will refer to the concept as it was laid out by Wallner *et al.* [101] when naming the LKH scheme.

The LKH scheme provides a solution for the problem of key management for multicast communication groups. It specifies two phases of operation: a setup phase that issues keys to each group member enabling a secure communication channel for messages sent by a center to the whole group – and a rekeying phase allowing to exclude members from the group or to add new members to the group by issuing new keys that are sent (encrypted) via multicast.

The architecture of LKH is focused on multicast but can also be applied to the broadcast encryption scenario and we can constitute algorithms Setup, Broadcast and Decrypt that implement the LKH scheme and follow Definition 3.2.2 regarding a broadcast encryption system when we allow the receivers to be stateful (*i. e.,* store and update keys) or when we bundle all the rekeying information history that is generated after system setup together and use it as message header for every broadcast message. The latter idea could be practical but the header length would be growing monotonously during the lifetime of the system which is not always desirable.

We give a simplified definition of the LKH scheme as a broadcast encryption scheme and state the important properties so a comparison to other schemes is possible. In the original description further variants are possible (*e. g.,* non-binary or unbalanced trees).

Definition 3.5.21. (LKH scheme in the broadcast scenario) Let $\mathcal{N}$ be the set of users of a broadcast encryption system. An LKH scheme consists of algorithms Setup, Rekey, Broadcast and Decrypt.

Setup initializes a full binary tree with a number of leaves that is the maximum number of allowed users during the lifetime of the system. Each node v_i in the tree is assigned a randomly and uniformly chosen key k_i and each user is mapped injectively to a leaf of the tree and is provided with all the keys assigned to nodes on the path from the mapped leaf to root.

Broadcast and Decrypt encrypt / decrypt the broadcast message using the key k_{root}.

The input for Rekey is a current user set $\mathcal{N}_t$ for a time t and a user u. If $u \in \mathcal{N}_t$ let $\mathcal{N}_{t+1} := \mathcal{N}_t \setminus \{u\}$ and the nodes on the path from leaf u to root are assigned new keys. Rekey outputs a broadcast message that encrypt every new node key for all users being leaves of a subtree whose root is a child of such a rekeyed node using the key assigned to each of these subtrees' root nodes. If $u \notin \mathcal{N}_t$ let $\mathcal{N}_{t+1} := \mathcal{N}_t \bigcup \{u\}$ and the new user (who brings along a private key k_u) is mapped to an unused leaf that is assigned the key k_u, and the keys on the path from u to root (except k_u) are changed in the same way as being done for the case $u \in \mathcal{N}_t$.

Example 3.5.22. See Fig. 3.6 for an example tree having 8 users mapped to the 8 leaves of the full binary tree before user u_3 is removed. The arrows denote a broadcast encrypted for users being a subtree's leaves. All keys along the path from root to u_3 are renewed and messages are generated to send the new key information of k_{13} and k_{15} to the set $\{u_1, u_2\}$ using the current key k_9 of node 9, to $\{u_5, \ldots, u_8\}$ the key k_{15}

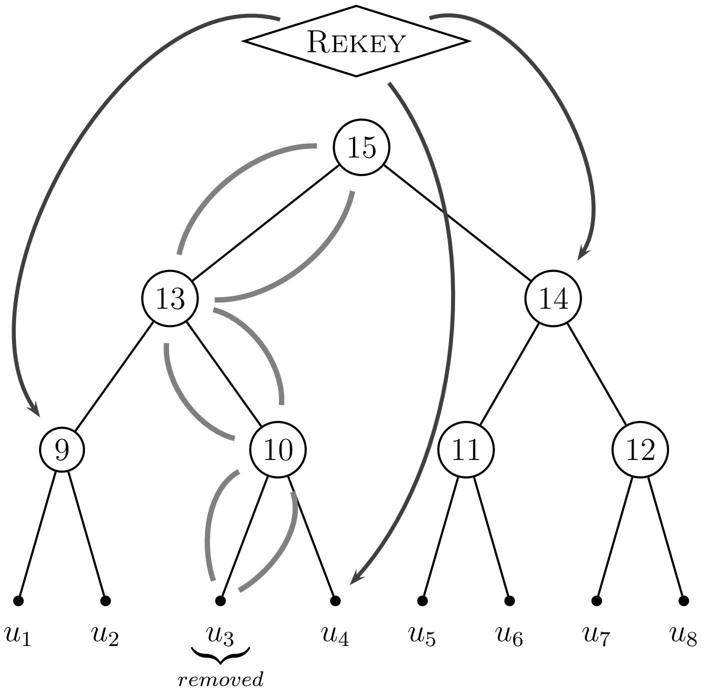

Figure 3.6: LKH rekeying example after removal of user u_3.

is sent using k_{14} as key encryption key and keys k_{10}, k_{13} and k_{15} are sent encrypted to $\{u_4\}$ using the user's private key k_4.

Remark 3.5.23. The reason for rekeying the path in case a new user is added is given by the requirement that new users shall not be able to decrypt broadcast messages sent before they were added to the group. Depending on the broadcast scenario (consider the value of real-time transmissions) this requirement can be dropped and the scheme could be simplified to some extend.

The definition specifies an additional algorithm (Rekey) and requires the receivers to maintain a state between transmissions thus it violates Definition 3.2.2 and we do **not** formally have a broadcast encryption scheme. However, since we can use the entire output history of Rekey as a message header sent by Broadcast this violation is easily healed and the scheme is transformed into a broadcast encryption scheme (for stateless receivers).

Next, we are interested in the efficiency of the LKH scheme (in the broadcast scenario). This is considered in the following lemma.

Lemma 3.5.24. *Let n be the maximum number of users for the LKH scheme. The number of operations for rekeying the user base in case a new user is added is bounded by* $2\log n$*; in case a user leaves the group it is* $\log n$*. Both case require a broadcast message of maximum length* $2\log n$*. Each user stores* $\log n$ *keys.*

Proof. Adding a new user requires REKEY to change all the keys on the path from a leaf to root, hence a total of $\log n$ nodes are rekeyed. Each of those node keys is encrypted with a child's key plus it is encrypted with the new user's private key so we count $2\log n$ encryption operations in total.

Removing a user follows the same concept as adding a user but the second encryption for a specific is not necessary so we have a total of $\log n$ encryption operations.

The message length counted as the number of encrypted keys is given by the same values as each encrypted key is part of the message.

The number of keys per user is given by the depth of the tree. □

Remark 3.5.25. We are a bit sloppy in the last lemma's proof as we counted encryptions as operations and used the encrypted keys' lengths as length units. There is no compelling logical reason to use exactly these items although in practical implementations the number of encryptions and key-size bit strings could be an apparent choice. For comparison purposes we are always safe when we use $O(\log n)$ as the value for all three properties. In the case when we use the REKEY output history as message header for a (stateless) broadcast encryption scheme we can conclude from the lemma that for r revoked users a header length of $O(r\log n)$ is achieved.

3.5.4 One-Way Function Tree

An enhancement of the LKH scheme was proposed by McGrew and Sherman [94] and the term *One-Way Function Tree* was proposed for their enhanced scheme. As this term is also used in other applications (*e. g.,* the Complete Subtree scheme also uses one-way functions for generating the keys in the tree structure) we will denote the enhanced scheme by LKH-OFT to avoid ambiguities. The LKH-OFT scheme is like LKH a key management scheme for for multicast communication groups that could be adjusted to the broadcast encryption case.

Definition 3.5.26. (LKH-OFT scheme in the broadcast scenario) Let $\mathcal{N}$ be the set of users of a broadcast encryption system. An LKH-OFT scheme consists of algorithms Setup, Rekey, Broadcast and Decrypt.

Setup initializes a full binary tree with a number of leaves that is the maximum number of allowed users during the lifetime of the system. Each leaf u_i in the tree is assigned a randomly and uniformly chosen key k_{u_i} and each user is mapped injectively to a leaf of the tree.

For each node v_j of the tree not being a leaf a key is assigned recursively by $k(v_j) := f(k(v_{j\swarrow})) \oplus f(k(v_{j\searrow}))$ where f is a length-preserving PRNG and $v_{j\swarrow}$ / $v_{j\searrow}$ denotes the left / right child of node v_j.

A *blinded* node key is given by f's output on the node's assigned key. Each user is provided with all the blinded keys assigned to nodes being a sibling of a node on the path from the mapped leaf to root.

Broadcast and Decrypt encrypt / decrypt the broadcast message using the key k_{root}.

The algorithm Rekey operates in the same way as Rekey of the LKH scheme (Def.

3.5.21) except the following modifications. In case a key is renewed and communicated to the users mapped to the subtree's leaves, the blinded keys are also communicated to all users who were provided with the blinded old key.

Example 3.5.27. See Fig. 3.6 for an example for the LKH-OFT scheme. User u_4 is initially provided with its private key and the blinded node keys $f(k(u_3))$, $f(k(9))$ and $f(k(14))$. After u_3 is removed all node keys on the path from u_3 to root are renewed and user u_4 will receive a new blinded key $f(k'(u_3))$ while user u_1 will receive a renewed blinded node key $f(k(10))$.

Lemma 3.5.28. *Let n be the maximum number of users for the LKH-OFT scheme. The number of operations for rekeying the user base in case a new user is added or removed is bounded by* $\log n$. *Both case require a broadcast message of maximum length* $\log n$. *Each user stores* $\log n - 1$ *keys and can derive the root node's key at any time.*

Proof. Adding or removing a user requires Rekey to change all the keys on the path from a leaf to root, thus a total of $\log n$ nodes are rekeyed. Each of those node keys except the root node is a sibling of a subtree's root node and was blindedly known to exactly the users mapped to the subtree's leaves so each new blinded node key is encrypted with the corresponding subtree's root key for distribution and a total of $\log(n)$ encryption operations are performed.

The broadcast length is the number of encrypted keys multiplied by the encrypted key size so the same value is calculated when we assume a fixed length and take this as a unit.

The number of keys per user is a private key plus a sibling's blinded key on the path to root (but not for root itself) so it is given by the depth of the tree minus 1.

By definition of the rekeying algorithm it is clear that a user always knows its private key plus its sibling's blinded private key and all siblings' blinded keys on the path to one of root's children. Thus, it is possible for a user to calculate the root key by recursively applying f to children of root as the key for a child is either known or could be expressed by f. Note, that the recursion stops at the leaf level. □

Remark 3.5.29. The improvement of the LKH-OFT scheme over the original LKH scheme is mainly noticeable in the case of a new user added to the system as the broadcast message header is reduced to half then. The key storage is decreased as well (by one key) but this advantage is at least compensated in a realistic practical implementation by the fact that the root key (also being the transmission key or session key encryption key) needs to be calculated for each transmission by $\log n$ PRNG operations and kept in secure memory during usage. Hence, we anticipate that the root key would be rather stored as well to avoid these operations and the small improvement regarding user key space would not be exploited.

The LKH-OFT scheme violates the Definition 3.2.2 of a broadcast encryption scheme since a state is maintained and an additional algorithm (Rekey) is introduced. As explained in Remark 3.5.23 to the LKH scheme we can transform the scheme into a a broadcast encryption scheme for stateless receivers by collecting the output of Rekey and sending it as a message header for each transmission.

3.6 Further Results Regarding Different Requirements on BE Schemes

There are a number of publications considering the problem to design efficient solutions for broadcast encryption *i. e.,* to minimize the communication overhead, while keeping the storage required by users and broadcasters small (*e. g.,* polynomial in $|\mathcal{N}|$ and / or the resiliency parameter).

Boneh and Silverberg [14] showed that by using n-linear maps a collusion secure scheme with a fixed size public key and message header length can be achieved; Boneh and Waters [15] improved this by limiting a modified scheme to bilinear maps. Both schemes do not provide information-theoretic security.

Dodis and Fazio [28] extended the schemes CS, SD and LSD to the public key setting.

Luby and Staddon [75] considered the information theoretic security case and give general lower bounds for revocation schemes. Applying these bounds to the general case (*i. e.,* not assuming $|\mathcal{R}| \ll |\mathcal{N}|$) shows that broadcast schemes with unconditional security are never efficient in the sense that one tradeoff obstacle is insurmountable: either the message header length is large or the user key size is large. Blundo *et al.* [12] are able to fix $O(1)$ as lower and upper bound for the header length as well as the user private key size by imposing requirements on the length of the broadcast message of a one-time revocation scheme.

Another result on the information theoretic case by Kumar *et al.* [69] presents a one-time revocation scheme (where the secure broadcasting can only be done once) removing r users with header length $O(r \log |\mathcal{N}|)$ and user key size $O(r^2)$ (not depending on $|\mathcal{N}|$). The efficiency of one-time revocation was improved by Pinkas [84] to a header length of $O(r)$ and a single private key per user.

Goodrich *et al.* [42] introduced the stratified sub-set difference method (SSD method) for revocation schemes achieving for the computational security setting a transmission header length of $O(|\mathcal{R}|)$, user computation overhead of $O(|\mathcal{N}|^{1/d})$ and $O(\log |\mathcal{N}|)$ user storage (where d is a predetermined constant being fixed before the scheme's set-up).

For the same setting Jho *et al.* [64] presented the *one-way chain* based revocation scheme resulting in less than $O(|\mathcal{R}|)$ header length but with user worst case computation cost of $O(|\mathcal{N}|)$ and large key storage (a binomial of $|\mathcal{N}| - 1$).

An efficient approach to transform an arbitrary broadcast encryption scheme to a scalable scheme while preserving the security of the underlying scheme was presented by Hwang *et al.* [56]. In their work a compiler is constructed that processes an original BE scheme and outputs a new scheme for a large number of group users that maintains transmission overhead of the original scheme asymptotically but gains reduction in user key size and / or user computation overhead.

3.7 Matrix-Based Schemes

Among the tree-based schemes depicted in the last sections and combinatorial revocation schemes [1, 38, 70, 75], there have been several proposals for matrix-based schemes, especially as part of technical industrial standards.

3.7.1 Digital Versatile Disk, Content-Scrambling System

The Digital Versatile Disc (DVD) is an optical storage medium for data storage, including multimedia content in MPEG format (see section 3.9.1 for details on MPEG). The DVD specification [30, 31, 32] was published (in its first version) September 1996. Initially, the DVD was designed for video storage only and the acronym was a short cut for *Digital Video Disk*. To differentiate between pre-recorded DVDs for video storage from those used for data storage, sometimes the terms *DVD-Video* and *DVD-ROM* – and also *DVD-Audio*, a format for delivering high-fidelity audio content are used. A DVD is able to store data amount of up to 17 GB (depending on disk type). There are several standards for recordable [53] and rewritable [33, 52] DVDs: DVD+RW (4.7 GB rewritable capacity per side), DVD+R (4.7 GB write-once c. p. s.), DVD-RAM (2.6 GB rewritable c. p. s.), DVD-RW (4.7 GB rewritable c. p. s.). A DVD-Player does not necessarily support all these formats, *e. g.,* DVD-RW is compatible with about 70 percent of the existing players by the end of 2005 [21]. Apart from the issue of competing and incompatible standards, it can be said that it is generally possible for a consumer to make a physical copy from an unprotected DVD to a writable medium.

The standardized content protection for DVD-Video is the Content-Scrambling System (CSS). The proprietary stream cipher algorithm for encrypting the content as specified by CSS turned out to be vulnerable – not only because the key length (40 bits) was rather short, also design weaknesses were identified [95]. Descrambling the content requires a pair of keys. One of the keys (*disk key*) is unique to the DVD, while the other is unique to the MPEG-2 file (*title key*) being descrambled. The disk key is stored on the "hidden" area of the disk to prevent byte-for-byte copies of the DVD (because a compliant device is unable to copy this area). Moreover, the disk key is only stored encrypted on the disk: multiple times, each instance encrypted with a different *player key.* Each player manufacturer is allocated at least one of 409 player keys for incorporation in its players. Disk and title keys can be passed from a drive to a software descrambler on a PC using a handshake protocol that enables an encrypted communication channel for the disk key transfer.

Refraining from the security weaknesses, the rights holders can use the DVD content protection to enforce two acts:

- Player keys can be revoked by not supporting them for future DVD productions.

- DVDs can be assigned to a world region.

Code	Region
0	No region coding
1	USA, Canada
2	Europe, Japan, South Africa
3	Korea, Thailand, Vietnam, Borneo and Indonesia
4	Australia and New Zealand, Central and South America
5	India, Africa, Russia and former USSR countries
6	China
7, 8, 9	unused, special use

Table 3.1: Region Codes

The latter act is technically enforced in the way that a DVD-Video carries a code that specifies a *region* of the world (one or more regions out of six, see Tab. 3.1) where the media can be played. The policy is that a compliant DVD player bought in one of such regions will only play media dedicated for this region. The rights holders can use the region coding to define different prices and release dates for each region.

The region codes were partly rendered useless after computer DVD drives could be used to play DVD content disregarding the region code information or by altering the player region code, if necessary by applying freely available patches to the player software.

3.7.1.1 Formal Treatment

The DVD content protection can be regarded as a simple revocation scheme when we formalize the player key assignment and key usage as described by the DVD specification [30, 31, 32] and related documents [96].

Definition 3.7.1. (DVD content protection) Let $\mathcal{K}$ with $\mathcal{K} = \{k_1, k_2, \ldots, k_{409}\}$ be a set of 409 keys (called the player keys), $\mathcal{N}$ be the users of a broadcast system (called devices) and let the algorithms Setup, Broadcast and Decrypt be defined that

- Setup assigns each user a subset of the player keys, *i. e.,* $P_u \subseteq \mathcal{K}$
- Broadcast reads the target set $\mathcal{T} \subseteq \mathcal{N}$ where $\mathcal{T} := \mathcal{N} - \mathcal{R}$ is identified by a set of revoked player keys $\mathcal{R}$.

  Then Broadcast selects a session key k (called disk key) randomly and uniformly chosen from $\{0,1\}^{40}$, encrypts the disk key multiple times (once for each of m non-revoked player keys denoted by $i_1, \dots i_m$), *i. e.,*

  $$\{k_{i_1}, k_{i_2}, \dots, k_{i_m}\} \cap P_u = \emptyset \quad \forall u \in \mathcal{R}$$

  and outputs the concatenation of these encrypted keys:

  $$(\mathrm{ID}_{i_1}, k_{i_1}(k), \mathrm{ID}_{i_2}, k_{i_2}(k), \dots, \mathrm{ID}_{i_m}, k_{i_m}(k))$$

- Decrypt reads the concatenation until a matching ID is found and decrypts the disk key with the associated player key or outputs an empty string if no matching ID is found.

Theorem 3.7.2. *Let $\mathcal{N}$ be the set of users of a broadcast system and let the algorithms* Setup*,* Broadcast *and* Decrypt *be defined as in Definition 3.7.1. Then the resulting scheme constitutes a broadcast encryption system with resiliency bounded by* 409*.*

Proof. The scheme is straightforward: the session key is encrypted with each non-revoked device key and the different cryptograms are concatenated. A privileged user is by definition in possession of at least one player key and can thus decrypt the session key. A non privileged user misses all the m player keys and can thus not decrypt the session key.

A colluding set of non-privileged users can not derive one of the m player keys since all are independently chosen from all player keys that can be combined by the colluders, thus the scheme offers maximum resiliency. □

Remark 3.7.3. Since only a few or only one of the 409 player keys are copied to each produced player of a certain manufacturer, the users of the scheme are “cloned” in order to let the scheme serve millions of player instances. This is not a violation of the definition since all of these clones can be revoked only at once (which is intended by the standard as the aim was to revoke a manufacturer if its key is read out of one device).

3.7.2 CPPM, CPRM

The *Content Protection for Prerecorded Media Specification* (CPPM) defines a method for protecting content distributed on prerecorded (read-only) media types. The authors (four manufacturer companies) of the specification [60] claim that the protection method is “renewable”, *i. e.,* that by way of precaution there is a possibility to react to issues regarding compromised secret key information. The storage media which are focused on by CPPM are DVD disks and its successors [58], thus it can be regarded as one alternative to the unreliable CSS protection scheme, but there are also specifications regarding SD-memory cards [61]. CPPM is the chosen content protection standard regarding the DVD-Audio.

CPPM uses the following cryptographic functions based on the *C2 cipher algorithm* [59]: $C2\_E(k, m)$ and $C2\_D(k, c)$: C2-encryption and C2-decryption in electronic codebook mode mode with key k (56 bits) message m (64 bits) and ciphertext c (64

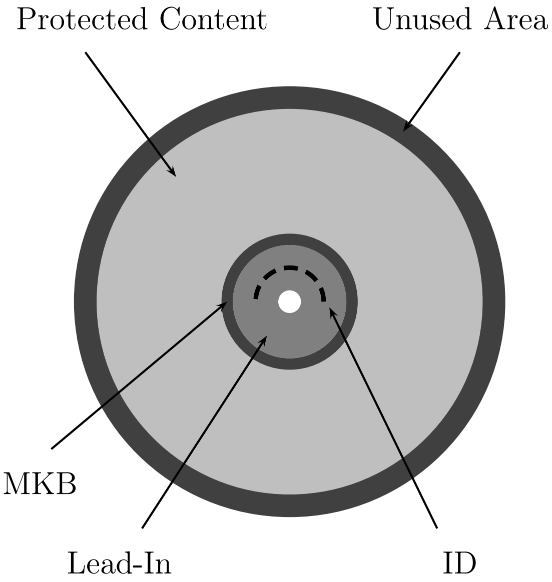

Figure 3.7: CPPM disk areas (simplified)

bits); analogously, $C2\_ECBC(k, m)$ and $C2\_DCBC(k, c)$: encryption and decryption in converted cipher block chaining (C-CBC) mode, where m and c are of arbitrary length. Finally, $C2\_G(d1, d2)$: one-way function (based on C2 cipher) mapping values $d1$ (56 bits) and $d2$ (64 bits) to a 64-bit value.

The *4C Entity*[4], the scheme's license authority, provides secret device keys to each playback device manufacturer for inclusion into the devices. The n device keys are referred to as $K_{d\_i} : (i = 0, \ldots, n-1)$. For each device key there is an associated column and a (confidential) row value, referred to as $C_{d\_i} < 2^8$ and $R_{d\_i} < 2^{16} : (i = 0, \ldots, n-1)$ respectively. While it is possible for a device to have more than one device key with the same associated row value, no two device keys share a column value.

[4] The name denotes the fact that 4 companies (IBM, Intel, Matsushita and Toshiba) developed this standard

3.7.2.1 Media Licenses

A manufacturer requests on behalf of the content owners for each medium title a unique identifier (ID), a media key block (MKB) and a corresponding media key K_m from the 4C Entity. The manufacturer also defines the copy control information (CCI) data field for the medium. The *Media Key Block* (MKB) is generated by the 4C Entity and stored together with the ID on the medium (see Fig. 3.7) by the manufacturer in a way that the ID cannot be copied to writable media. The MKB is processed by compliant devices to calculate a media key K_m by applying the device keys.

If a set of device keys is compromised (*e. g.,* leaked from the device and used for copyright infringement) it can be revoked by arranging that each new generated MKB cannot be processed with the help of the revoked key set for calculating the media key K_m.

The content key K_c is calculated from the triple (K_m, ID, CCI) through repetitive computation of the one-way function $C2\_G$. Details may vary depending on media type but it is essential that the content key is derived by a one-way computation process from all three input values. The key K_c is applied for encrypting the content.

3.7.2.2 Decryption and Playback

A playback device reads the MKB, which is formatted as a sequence of contiguous records, from the medium by processing records one-by-one, in order, from first to last. The algorithm is referred to by $Process\_MKB$. There are four types of records.

- Verify Media Key Record (VMKR): Is processed to check whether the correct media key K_m has already been computed.

- b) Calculate Media Key Record (CMKR): The device reads the column field d_i

of the record that specifies a device key K_{d_i}. If the device keys do not contain K_{d_i} then the record is discarded. Otherwise, the encrypted key data D_{ke_0}, D_{ke_1}, D_{ke_2}, etc. is read from the record and by using the row value R_{d_i} the encrypted key $D_{ke\_(R_{d\_i})}$ is selected and decrypted with K_{d_i}, obtaining a current (temporary) media key K_m.

- Conditionally Calculate Media Key Record (CCMKR): The CCMKR data fields are additionally encrypted with a session key that may match the current media key K_m. This checked by first decrypting a verification value. Upon successful decryption a column value d_i is read to select a device key K_{d_i} and to decrypt the encrypted[5] key $D_{ke\_(R_{d\_i})}$ selected by the respective row value R_{d_i}.

- End of Media Key Block Record (EMKBR): The algorithm $Process\_MKB$ is stopped and the current media key K_m becomes the actual media key for decrypting the content.

After processing all records the player either has computed a valid media key, or considers itself revoked and notifies the user[6].

The renewability paradigm is realized by the following mechanism: A MKB may contain several CCMKRs that can be grouped as chains. One of the chains is to be processed successfully by all devices that cannot derive the media key K_m by processing a CMKR. The chain's end contains the media key while each chain link contains a temporary key needed to process the next item. This mechanism provides a set intersection operation defining a sub-set of devices that each possess all required player

[5] This key is doubly encrypted by a device key as well as the computed session key

[6] The exact user notification method is not defined by the standard but a suggestion is made that a "device could exhibit a special diagnostic code, as information to a service technician" [60] in order to let the user know of a revocation.

keys to process the chain – and a set union operation because each chain's respective sub-set as well as each sub-set defined by a CMKR is united to become the set of all devices being able to decrypt the content.

3.7.2.3 CPPM / CPRM as a Broadcast Encryption Scheme

The algorithm $Process\_MKB$ together with the MKB generation process and the process assigning keys to each player defines a scheme to select certain sub-sets of the set of all players that are enabled to decrypt a medium's content. To be close to practical implementations we could use the term *player class* referring to a set of players that (due to large-volume production) share identical device keys and that are revoked together but as the standard on principle allows individual key assignment we stick to the term player in this context.

The process assigning the device keys to the players has not been published but it was shown [4] that there exist several suitable key-assignment strategies based on finite geometric structures so that more devices or device classes than ever will be needed can be specified.

Since neither the current key assignement strategy is publicly available nor the algorithm is published how the MKB is constructed by the 4C Entity on behalf of a media manufacturer, we are unable to formalize the scheme and to prove that it is a broadcast encryption scheme according to Definition 3.2.2 because we cannot as accurately describe the Setup and Broadcast algorithms of the scheme as it would be necessary for a rigorous treatment. However, since we know how the MKBs are processed by the players, reasonable assumptions on the parameters could be made. Naor et al. [83] describe a possible CPPM set-up achieving roughly a header length of $r \log |\mathcal{N}|$

but not always a perfect coverage of the target set. Based on this considerations, the CPPM scheme can be compared with the Complete Subtree method (see Definition 3.5.3) regarding efficiency.

3.8 Trusted Computing / Tamper Resistance

In 1999, Intel, Microsoft, IBM, Hewlett-Packard, and several other computer industry companies founded the *Trusted Computing Platform Alliance* (TCPA) which was later renamed to the *Trusted Computing Group* (TCG). The self-assigned scope of the organization is "to develop and promote open, vendor-neutral, industry standard specifications for trusted computing building blocks and software interfaces across multiple platforms" [97].

The TCG has developed specifications for the *Trusted Platform Module* (TPM) mainly used for PC technology and a software interface specification to enable application development for systems using the TPM. The TPM has gained particular interest as a possible technical prerequisite to enforce DRM on end-users' PC equipments but there are also other TPM use cases *e. g.,* protection against malicious code or user authentication and also further hardware architectures targeted by the TCG *e. g.,* servers, mobile phones and PDAs.

In June 2002, the leading PC operating system developer Microsoft announced *Palladium*, an enhanced version of its *MS Windows* product family supporting TPM's software interface specification due for release in 2004; the project was renamed in 2003 to *Next Generation Secure Computing Base* (NGSCB) but it was unclear what part of the Trusted Platform Module functionality would be supported by NGSCB. Microsoft made public that the operating system and hardware changes introduced to

MS Windows by NGSCB "offer a way to isolate applications (to avoid snooping and modification by other software) and store secrets for them while ensuring that only software trusted by the person granting access to the content or service has access to the enabling secrets. A DRM system can take advantage of this environment to help ensure that content is obtained and used only in accordance with a mutually understood set of rules." [78].

In principle the TPM specification [98] describes a computing platform that makes use of a *trusted component*. This trusted component is given by a built-in hardware and is the basis of trust for software processes. The security functions of the security hardware in a Trusted Platform are assumed to be always trustworthy. The hardware is a root of trust and is able to govern both the hardware and the software environment of a specific system. If the software is accepted to be trustworthy by the trusted component then it may operate as normal process. This root of trust is the core TPM capability.

3.8.1 Capabilities of the TPM

The capabilities offered by the TPM are cryptographic functions and secure storage of keys. We name the major items of the current version of the specification [98].

- Hash functions: The algorithm SHA-1 is available.
- Random number generator: available. Can be used for symmetric key generation.
- RSA key generation is available.
- RSA private key usage (for decryption and digital signatures).

- Symmetric encryption and decryption: 3DES is available, AES is announced for an upcoming version.
- Key storage for symmetric and asymmetric keys.
- Attestation capability (detailed in the following)

Regarding the attestation capability we can distinguish (following [71]) three different roles served by a trusted platform: The *user* of the computing platform is interested in the protection of private information (*e. g.,* private user keys), the *owner* of the platform define a certain policy being enforced by the platform (*e. g.,* usage of initially installed software only) while a *provider* of a digital good wants to ensure the integrity of the platform in a way that these goods are protected (*e. g.,* copy protection). The concept of verifying the platform state by a remote party (owner, provider) is known as *remote attestation*.

Attestation provides the authenticity of a platform's integrity, state and / or configuration. On a trusted platform, a running process can send attestation challenges to another process to verify its integrity or running state. This can be used for mutual attestation of software processes that need to co-operate to make a functionality available (*e. g.,* one application for playing digital content and another application for acquiring and moving the content).

There are a number of keys with fixed purpose:

- Endorsement Key Pair (EK): Each TPM generates an RSA key pair called an Endorsement Key that is bound to only one TPM. It shall never leave the module but it might be deleted and replaced by a new EK.

- Attestation Identity Keys (AIK): AIKs are used to *attest* the platform configuration and state to another party (without revealing the identity of the platform). A platform can be assigned an unlimited number of AIKs in order to achieve anonymity.
- Storage Root Key (SRK): In each platform there is one SRK that can be regarded as the root of a tree where all keys are protected by their ancestor keys.

For our purposes (Digital Rights Management for broadcasted digital goods) the TPM could be used as a multimedia PC's root of trust for secure remote key distribution as well as secure user localization regarding the enforcement of location-dependent licenses. The roles of owner and provider of the platform could then both be filled by the broadcaster / seller of digital transmissions (*e. g.,* a Pay-TV station).

3.8.2 PCRs and Remote Attestation

The TPM contains a set of Platform Configuration Registers (PCRs). A fingerprint of protected data or software represents the characteristics of the fingerprinted object such as integrity, states of a system or user process and configurations. The specific PCR value is updated by applying cryptographic hash functions on its present value concatenated with the new data. Thus, PCR values can record the integrity and state of a platform from the boot loader process to operating system start up and applications: After boot up, the integrity of the boot loader and the secure kernel (SK) are measured by the TPM in a platform and stored in PCRs respectively. The SK is assigned a key pair where the public key is certified by one AIK of the TPM. Before starting any process, the SK measures the integrity of the code and stores the hash in its private memory. The SK also generates a fresh key pair for the trusted reference monitor

(TRM), where the public key is itself certified by the SK

In case of a remote attestation request, the TPM signs the values in the PCRs with its AIK and the SK signs the integrity value of the TRM with its private key. The signed values are sent to the remote party that has requested the attestation. The remote party verifies both signatures and the public key certificates of the AIK and the SK. If the verification is successful the SK as well as the TRM are trusted.

3.8.3 Sealing

An important use case of a trusted platform is that an object that is TPM protected can be *sealed* to a fixed software state.

At the time the object is created, the creator specifies a software state that has to be in place when the secret data is to be exposed (*i. e.,* decrypted and processed further). Before decryption of the protected object, the TPM first checks if the measured software state equals the defined state. Only if the states match, the TPM grants access to the sealed object.

3.8.4 TPM Command Set

We will use some of the TPM commands and data structures as specified by the Trusted Computing Group [99].

The TPM_CreateWrapKey command both generates and creates a secure storage bundle for asymmetric keys, the TPM_CMK_CreateKey command does the same but the keys' migration is controlled by a migration authority. The TPM_CertifyKey operation allows one key to certify the public portion of another key. The TPM_Quote operation provides cryptographic reporting of *Platform Configuration Register* (PCR) values, a PCR being a 160-bit storage location for discrete integrity measurements.

Finally, TPM_Seal allows software to explicitly state the future trusted configuration that the platform must be in for a certain secret to be revealed.

3.9 Industrial Standards

3.9.1 DVB, MPEG

The standards DVB and MPEG are of special importance for this work as our goal in the considerations of a global broadcast service (see Chapter 5) is to re-use conventional conditional access technology for securing multimedia transmissions, at least we aim to re-use existing standards as far as reasonable. In this section the established standards in this area are presented.

The broadcasted digital multimedia content we encounter in real-life applications is often coded in MPEG [63] format and transported by DVB technology.

MPEG (Moving Picture Experts Group) standardizes the content bitstream format and how a decoder shall interpret a bitstream. Actually, MPEG is a family of several standards; of major relevance to this work are the following items:

- **MPEG-1**: Video and audio compression standard, used as the standard for the *Video CD*, includes the MP3 audio compression format.

- **MPEG-2**: Transport, video and audio format for television broadcast. Supported transport technology is DVB (among others) and DVD video as storage medium.

- **MPEG-4**: Introduces Digital Rights Management.

- **MPEG-21**: Future standard that incorporates the older standards by defining an open framework for multimedia applications and defines a machine-readable *Rights Expression Language* (REL).

DVB (Digital Video Broadcasting), is a collection of internationally agreed, open standards for digital television maintained by the DVB Project, where broadcasters, manufacturers and regulatory bodies (*e. g.,* ETSI) form a consortium. DVB defines the physical layer and data link layer of the signal that can be broadcasted via satellite (DVB-S), terrestrial emitters (DVB-T) and cable (DVB-C). All data (not only the multimedia content) is transmitted in MPEG-2 transport streams. DVB-T has become the successor of terrestrial analog television in Germany, in 2003 the city of Berlin was the first area to completely stop broadcasting analog TV signals and solely support DVB-T as terrestrial television broadcasting technology [100].

3.9.2 Common Scrambling, Common Interface

Currently, most Pay-TV subscribers own a set-top box or set-top terminal (STT) equipped with a *Common Interface* [20], an established standard used in digital video broadcasting. This Common Interface is connected with a *CI module* that is incorporating a smartcard reader where the user will put in a smartcard issued by his Pay-TV provider. The standard format for a CI module is a PCMCIA card. There are also a less expensive variants of CI modules that include the smartcard's functionality and do not provide a smart-card reader.

Most currently available set-top boxes and DVB interface cards for Personal Computers provide at least one or more than one Common Interface slot.

Different CI modules facilitate different cryptographic protocols and algorithms that the Pay-TV service providers use and implement on the module. All STTs are applying the same content descrambler (specified by the *Common Scrambling Algorithm*). During a secured transmission the STTs continuously receive so-called *Control Words* via the Common Interface that they need to descramble the secured content. These Control Words are short-lived session keys only used for small parts of one transmission.

4 Two Pseudo-Probabilistic Broadcast Encryption Schemes

In this chapter we will propose two new broadcast encryption schemes operating in a pseudo-probabilistic way. Both schemes realize their efficiency by accepting an adjustable ratio of free-riders. The first scheme is computationally secure, but puts certain undesirable constraints on the abilities of attackers; the improved scheme is information-theoretically secure and lacks these constraints. We will give a calculation of the parameter tradeoffs of our schemes and discuss the collusion resiliency.

4.1 Idea: Relaxed Requirements, New Tradeoffs

In order to set up schemes that are more efficient than sending $|\mathcal{N}| - |\mathcal{R}|$ messages we are relaxing the requirement that only the users in the target group $\mathcal{T} := \mathcal{N} - \mathcal{R}$ can decrypt the message by allowing a certain (small) number of users in $\mathcal{R}$ to decrypt the transmission as long as every user in $\mathcal{T}$ can receive the transmission. In this case new requirements on a relaxed scheme are to be considered: The number of users who can receive a transmission they have not subscribed to, *i. e.,* the number of *free-riders*, shall be minimized and—following economic, game-theoretic requirements (see *e. g.,* [93])—a user shall not gain any information whether she might be a free-rider for a future transmission by examining the past transmissions.

For example in a pay-TV scenario, we want to avoid a situation where two users $u_1, u_2 \in \mathcal{N}$ are put in one subscription set so that each time user u_1 subscribes to a transmission the user u_2 becomes a free-rider. The user u_2 might learn that he often becomes a free-rider for a certain kind of transmission preferred by u_1 (*e. g.,* , *Tarantino* movies) and will stop subscribing for these transmissions to avoid *unnecessary* payment.

The main area of tradeoff parameters to be considered in this relaxed notion of broadcast encryption is the number (or ratio) of free-riders versus the message header length versus the user key size. Other major requirements on a scheme are collusion resiliency (*i. e.,* the number of non-subscribers that may collude without being able to access the secured transmission) and underlying security assumptions (*e. g.,* unconditional security versus computational security).

4.2 The Bit-wise Biased Sub-set Scheme

4.2.1 Notations, Definitions and Basic Idea

Let $\mathcal{N}$ be the set of all users of a broadcast scheme and $\mathcal{T}$ be the set of users which shall receive a certain transmission.

Each user $u \in \mathcal{N}$ is provided with a fixed set of secret keys K_u which are assigned to him before receiving any transmission. Each user owns at least one individual key $k_u^{indv} \in K_u$ only known to him and the sender; the other keys might also be shared between several users, which is not known to the users sharing a key. During a transmission any user might receive further one-time usage keys (session keys, key encryption keys) which are not re-used and do not need to be stored after the transmission (thus we have a *stateless receiver*).

The basic idea of our scheme is to transmit the session key for a certain transmission

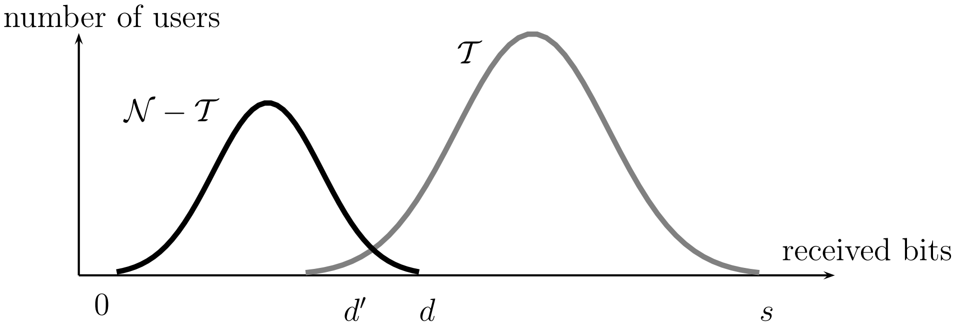

Figure 4.1: Distribution of received key bits

bit-wise in a probabilistic way to all users in $\mathcal{N}$ so that the users in $\mathcal{T}$ receive on average more key bits than the users in $\mathcal{N}-\mathcal{T}$, thus only a small fraction of the users in $\mathcal{N}-\mathcal{T}$ is able to decrypt the transmission. Most users in $\mathcal{T}$ are provided with enough bits of the session key to derive the full key after exhaustive search. For the great majority of the users in $\mathcal{N}-\mathcal{T}$ it is infeasible to derive the session key in due time (*e. g.,* before the transmission starts or before the transmitted data becomes outdated).

For each transmission we choose a security parameter s and the generated session key k_S consists of $|k_S| = s$ bits. This key is valid for one transmission only. For the users in $\mathcal{T}$ a minimum of $d < s$ bits is needed to derive k_S (d is chosen according to the computation power of the users). We assume potential attackers could be more powerful than the ordinary users, so they only need $d' \leq d < s$ bits to derive k_S in due time. The goal of the scheme is then that after a protocol run, the great majority of users in $\mathcal{T}$ has received more than d bits when at the same time only a small minority of users in $\mathcal{N}-\mathcal{T}$ has received d' or more bits (see Figure 4.1).

Our scheme works in two phases: First a number of messages each carrying one key bit of k_S is broadcasted (each message can only be decrypted by a different sub-set of $\mathcal{N}$ provided with the right sub-set key) so that a certain number of the users $\mathcal{T}' \subset \mathcal{T}$

has received at least d bits (*e. g.,* targeting $\frac{|\mathcal{T}'|}{|\mathcal{T}|} > 0.95$). In the second phase each user in $\mathcal{T} - \mathcal{T}'$ is provided individually with the full session key using the keys k_u^{indv} for all $u \in \mathcal{T} - \mathcal{T}'$.

Remark 4.2.1. Our approach broadcasts a secret by gradually broadcasting parts (bits or shares) of the overall secret to certain sub-sets so that any party having enough bits (or shares) can compute or recover the complete secret, *e. g.,* the session key of a pay-TV broadcast transmission. The gradual transmission of secrets has been previously applied in the area of fair exchange [24]. In this context there is an additional "verifiability" requirement, as released parts of the secret have to be verifiable, such that a cheating party cannot gain valid parts of an honest party's secret, while sending random bits to this party. In the broadcast encryption setting the verifiability requirement can be neglected as the sender is trusted and it is only a unidirectional release of secrets. This gradual probabilistic broadcast of secrets represents, to the best of our knowledge, a new probabilistic approach to broadcast encryption, which may foster further advance in broadcast encryption.

4.2.2 Setting Up the Scheme

The algorithm Setup reads parameters M and selects M sub-sets $N'_1 \dots N'_M \subset \mathcal{N}$, which are chosen randomly and uniformly from the set of all sub-sets of $\mathcal{N}$ with $\frac{1}{2}|\mathcal{N}|$ elements, so $|N'_i| = \frac{1}{2}|\mathcal{N}|$ for all $i = 1, \dots, M$. For each sub-set N'_i a *sub-set key* $k_{N'_i}$ is chosen randomly and uniformly and let $k_{N'_i} \in P_u \Leftrightarrow u \in N'_i$. So each user knows the key assigned to each sub-set she belongs to, but she does not know any other sub-set keys, so she stores $\approx \frac{1}{2}M$ sub-set keys in total (note, that key storage could be reduced heavily if a PRNG-based algorithm is used to generate keys before usage, but our aim is to reach unconditional security for a variant of this algorithm).

Following Definition 3.3.2 we do intend to split the users $\mathcal{N}$ in equal-sized batches of users since we can take advantage of the properties of the batch split for implementation (*e. g.,* reduce the size of the sub-sets $N'_1 \dots N'_M \subset \mathcal{N}$). However, we will keep the notion $\mathcal{N}$ of the set of all users because the scheme's definition does not rely on the batch split and is valid for user sets of any size.

4.2.3 Broadcasting

For a transmission a set $\mathcal{T} \subset \mathcal{N}$ of valid subscribers is given as input to the BROADCAST algorithm that is specified in the following. First, parameters M, d, s are read and a session key k_S having a length of s bits is chosen randomly and uniformly for the transmission.

In order to broadcast k_S to the set $\mathcal{T} \subset \mathcal{N}$ of users,BROADCAST sorts the sub-sets N'_i so that we can assume for the sorted sub-sets $N_i := N'_{\pi(i)}$ that

$$N_i \geq_{\mathcal{T}} N_{i+1} \qquad \forall i : 1 \leq i < M \tag{4.1}$$

where for arbitrary sub-sets N_a, N_b we define

$$N_a \geq_{\mathcal{T}} N_b :\Leftrightarrow |N_a \cap \mathcal{T}| \geq |N_b \cap \mathcal{T}| \tag{4.2}$$

using a suitable permutation π for representing the sorted sub-sets. Loosely speaking, we let N_1 be the sub-set containing the highest number of subscribed users, N_2 is next with the second greatest bias towards the number of subscribers, so for small indices i, the bias of the sub-sets N_i towards $\mathcal{T}$ is high, see Figure 4.2 for illustration.

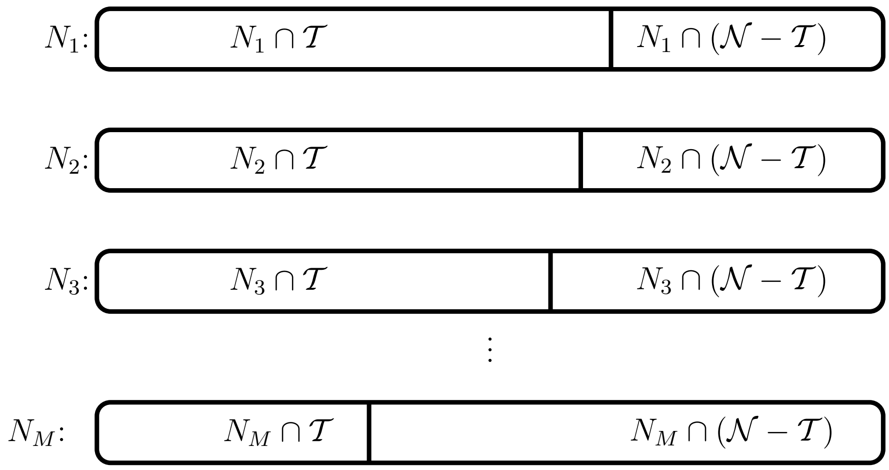
N_1:
$N_1 \cap \mathcal{T}$
$N_1 \cap (\mathcal{N} - \mathcal{T})$
N_2:
$N_2 \cap \mathcal{T}$
$N_2 \cap (\mathcal{N} - \mathcal{T})$
N_3:
$N_3 \cap \mathcal{T}$
$N_3 \cap (\mathcal{N} - \mathcal{T})$
$\vdots$
N_M:
$N_M \cap \mathcal{T}$
$N_M \cap (\mathcal{N} - \mathcal{T})$

Figure 4.2: Biased sub-sets

The Broadcast run consists of two phases.

4.2.3.1 Phase 1

The session key k_S will be transmitted bit-wise: Let $k_{S,1}$, $k_{S,2}$, $\ldots$, $k_{S,s}$ denote the session key bits. First bit $k_{S,1}$ is sent to sub-set N_1 using sub-set key k_{N_1}, then $k_{S,2}$ is sent to N_2 etc. until all bits are sent via broadcast channel, each preceded by the sub-set key number that is given by the permutation function. The transmission can be represented by a tuple

$$\big((\pi^{-1}(1), E_{k_{N_1}}(k_{S,1})), (\pi^{-1}(2), E_{k_{N_2}}(k_{S,2})), \ldots, (\pi^{-1}(s), E_{k_{N_s}}(k_{S,s}))\big)$$

of pairs; each pair consists of a sub-set key number and an encrypted session key bit.

4.2.3.2 Phase 2

For all $1 \leq j \leq s$ let $\mathcal{T}_j' \subset \mathcal{T}$ be the set of users in $\mathcal{T}$ that have received at least d session key bits after $k_{S,j}$ is broadcasted.

For each user in $u \in \mathcal{T} - \mathcal{T}'_s$ BROADCAST provides the full session key by using her unique secret key $k_u^{indv} \in P_u$ to encrypt an individual message for her and send it via the broadcast channel so without loss of generality users $u_1, u_2, \ldots, u_g$ be the elements of $\mathcal{T} - \mathcal{T}'_s$ and a tuple

$$\Big((u_1, E_{k_{u_1}^{indv}}(k_S)), (u_2, E_{k_{u_2}^{indv}}(k_S)), \ldots, (u_g, E_{k_{u_g}^{indv}}(k_S))\Big)$$

is sent in phase 2.

4.2.4 Decryption

The algorithm DECRYPT of user u reads parameters M, d, s and receives the first tuple

$$\big((\pi^{-1}(1), C_1), (\pi^{-1}(2), C_2), \ldots, (\pi^{-1}(s), C_s)\big)$$

in order to determine the sub set keys that the users possesses. Then the respective session keys bits are decrypted from the C_i. If d or more key bits can be decrypted then the remaining bits are found by exhaustive key search.

If less than d bits can be decrypted the second tuple

$$\left((u_1, E_{k_{u_1}^{indv}}(k_S)), \ldots, \underbrace{(u, E_{k_u^{indv}}(k_S))}_{\text{matching pair for user } u}, \ldots, (u_g, E_{k_{u_g}^{indv}}(k_S))\right)$$

is parsed and the session key can be decrypted by DECRYPT for user u by using the individual key k_u^{indv} after a matching pair is found.

4.2.5 Security of the Scheme

The desired property of the specified algorithms is the suitability for a (secure) broadcast encryption scheme. The definition of such a scheme includes the case where free-riders are accepted in a way that the scheme could address a set of targets $\mathcal{T}_0 \subseteq \mathcal{N}$) containing the privileged set augmented with the respective free-riders. These considerations are detailed in the following theorem.

Theorem 4.2.2. *Let $\mathcal{N}$ be the set of users of a broadcast system and let the algorithms* Setup, Broadcast *and* Decrypt *be defined as in the Biased Sub-set Scheme with parameters M, d, d' and s.*

For each sub-set $\mathcal{T}$ of $\mathcal{N}$ let $\mathcal{T}_0$ be the union of $\mathcal{T}$ and the non-privileged users who received d' or more session key bits after the phase 1 tuple is broadcasted (these users are called free-riders).

Then the resulting scheme constitutes a broadcast encryption system tolerating free-riders (according to Definition 3.2.2).

Proof. Let $\mathcal{N}$ be the set of users and d, d', s be the scheme's parameters.

Every privileged user $u \in \mathcal{T}$ does receive the session key k_s of a transmission either in phase 1 by exhaustive search of the remaining $s-d$ key bits or in phase 2 by individual decryption of the full encrypted session key k_s. Thus, every privileged user can decrypt the broadcast message.

Following the specification of the Broadcast algorithm we either have for a target $\mathcal{T}$ no free-riders or we have by definition an augmented set $\mathcal{T}_0$ with $\mathcal{T} \subset \mathcal{T}_0$ and Decrypt outputs session key k_s if $u \in \mathcal{T}_0$ and fails if $u \notin \mathcal{T}_0$ because a non-privileged user either becomes a free-rider and receives by definition d' bits or more or he receives less than d' bits so he is not a member of the augmented target set and cannot decrypt the session key by exhaustive key search. □

Remark 4.2.3. Theorem 4.2.2 uses a convenient construction of the augmented set of targets $\mathcal{T}_0$ to fulfill the definition of a broadcast encryption system. One major outstanding open problem is to determine the size of $\mathcal{T}_0$ or to ensure by selecting appropriate paranmeters M, d and s that the ratio $\frac{|\mathcal{T}_0 - \mathcal{T}|}{|\mathcal{N} - \mathcal{T}|}$ is reasonably small. This is subject of the next section and Theorem 4.2.4.

The agreement that $s - d'$ bits cannot be searched exhaustively by an attacker is a delicate matter for practical considerations. An attacker might be more interested in the end of a transmission than in the first parts so he could try to finish the exhaustive search before a long transmission is over; our security considerations do not take such a scenario into account so the value d' might be reduced further in practice to be on the safe side. Fortunately, we are able to omit this parameter in the improved scheme (presented in section 4.3) so we do not need to dwell on that subject further.

4.2.6 Efficiency of the Scheme

We are now interested in the number of messages sent in the two phases of the broadcast period and the number of free-riders. These values are dependent on the number of pre-distributed keys (M) and the length of the session key (s); the practical relevance of the scheme cannot be assessed without knowing how large the number of keys to be stored by each user under certain parameters is. We will use statistical approximations to derive a formula for the calculation of the upper bound of the parameters. In the annex some simulation results are provided that substantiate these approximations.

Theorem 4.2.4. (i.) *The ratio of free-riders* $FR_{rat} := \frac{|\mathcal{T}_0 - \mathcal{T}|}{|\mathcal{N} - \mathcal{T}|}$ *can be approximated by*

$1 - \Phi_{\mathcal{N}-\mathcal{T}}(d')$ *where* $\Phi_{\mathcal{N}-\mathcal{T}}$ *is the distribution function of the normal distribution*

$$N\left(\frac{\overline{n_s}\cdot s}{|\mathcal{N}-\mathcal{T}|}, \frac{\overline{n_s}\cdot s}{|\mathcal{N}-\mathcal{T}|}(1-\frac{\overline{n_s}}{|\mathcal{N}-\mathcal{T}|})\right)$$

and where $\overline{n_s}$ *is the average number of non-privileged users in the sorted sub-set* N_s *given by*

$$\overline{n_s} := \frac{1}{2}|\mathcal{N}| - \overline{t_s}$$

and where $\overline{t_s}$ *is the average number of privileged users in the sorted sub-set* N_s *and* $\frac{\overline{t_s}}{|\mathcal{T}|}s$ *is the average key-bit information received by the subscribed users per transmission of s bits:* $\overline{t_s}$ *can be approximated by* $\Phi^{-1}(1-\frac{s}{M})$ *where* Φ^{-1} *is the quantile function of the Gauss distribution*

$$N\left(\frac{|\mathcal{T}|}{2}, \frac{|\mathcal{T}|}{4}(1-\frac{|\mathcal{T}|}{|\mathcal{N}|})\frac{|\mathcal{N}|}{|\mathcal{N}|-1}\right) .$$

(ii.) *The ratio of users receiving at least d bits in phase 1 is*

$$SUC_{rat} \approx 1 - \Phi_{\mathcal{T}}(d)$$

where $\Phi_{\mathcal{T}}$ *is the distribution function of the the distribution*

$$N\left(\frac{\overline{t_s}}{|\mathcal{T}|}s, \frac{\overline{t_s}}{|\mathcal{T}|}s(1-\frac{\overline{t_s}}{|\mathcal{T}|})\right) .$$

(iii.) *The number of messages to be sent in phase 2, which is the the number of users not having received at least d bits in phase 1, can be approximated by* $(1-SUC_{rat})|\mathcal{T}| = \Phi_{\mathcal{T}}(d)|\mathcal{T}|$*.*

Proof. First, we determine the distribution of the values $t'_i := |N'_i \cap \mathcal{T}|$ (*i. e.*, the number of privileged users per sub-set) for the unsorted sub-sets N_i'.

The probability p_t for the event that a randomly chosen sub-set of $\mathcal{N}$ with $\frac{1}{2}|\mathcal{N}|$ elements contains exactly t privileged users is given by

$$p_t = \frac{\binom{|\mathcal{T}|}{t}\binom{|\mathcal{N} \setminus \mathcal{T}|}{\frac{1}{2}|\mathcal{N}| - t}}{\binom{|\mathcal{N}|}{\frac{1}{2}|\mathcal{N}|}} \tag{4.3}$$

for all $0 \leq t \leq max(\frac{1}{2}|\mathcal{N}|, |\mathcal{T}|)$ because we count $\binom{|\mathcal{T}|}{t}$ possible sub-sets with t elements of $\mathcal{T}$ and $\binom{|\mathcal{N}\setminus\mathcal{T}|}{\frac{1}{2}|\mathcal{N}|-t}$ sub-sets for the remaining $\frac{1}{2}|\mathcal{N}| - t$ non-privileged users (out of $|\mathcal{N} \setminus \mathcal{T}|$) considering a total of $\binom{|\mathcal{N}|}{\frac{1}{2}|\mathcal{N}|}$ possible sub-sets containing exactly half of the users.

The discrete distribution given by 4.3 is the hypergeometric distribution with parameters *sequence of draws* $\frac{1}{2}|\mathcal{N}|$ *out of* $|\mathcal{N}|$ with number of *good* items $|\mathcal{T}|$. The mean μ_{p_t} of this distribution is thus given by

$$\mu_{p_t} = \frac{1}{2}|\mathcal{N}| \cdot \frac{|\mathcal{T}|}{|\mathcal{N}|} = \frac{|\mathcal{T}|}{2}$$

i. e., on average, half of the privileged users are in a sub-set N_i' and the variance $\sigma^2_{p_t}$ by

$$\sigma^2_{p_t} = \frac{1}{2}|\mathcal{N}| \cdot \frac{|\mathcal{T}|}{|\mathcal{N}|} \cdot (1 - \frac{|\mathcal{T}|}{|\mathcal{N}|}) \cdot \frac{|\mathcal{N}| - \frac{|\mathcal{N}|}{2}}{|\mathcal{N}| - 1} = \frac{|\mathcal{T}|}{4} \cdot (1 - \frac{|\mathcal{T}|}{|\mathcal{N}|}) \cdot \frac{|\mathcal{N}|}{|\mathcal{N}| - 1} \ .$$

This distribution can also be approximated to be Gaussian *i. e.,* $\forall i = 1 \ldots M : t_i' \sim N(\frac{|\mathcal{T}|}{2}, \frac{|\mathcal{T}|}{4}(1 - \frac{|\mathcal{T}|}{|\mathcal{N}|})\frac{|\mathcal{N}|}{|\mathcal{N}|-1})$.

After sorting the M sub-sets we have the most biased s values $t_i := |N_i \cap \mathcal{T}| > \frac{|\mathcal{T}|}{2}$ for $i = 1 \ldots s$ with average values

Notation	Meaning
$B(n,p)$	binomial distribution
FR_{rat}	free-rider ratio after phase 1
$i = 1 \dots M$	index of a sub-set
M	number of sub-sets N'_i
$N(\mu, \sigma^2)$	Gaussian distribution
$\|\mathcal{N}\|$	number of users
N'_i	sub-sets, half of all users each
$n := \frac{1}{2}\|\mathcal{N}\|$	number of choices, Bernoulli (p, n)
$\overline{n_s}$	average non-priv. users in sorted sub-set s
$p = \frac{\|\mathcal{T}\|}{\|\mathcal{N}\|}$	probability, Bernoulli (p, n)
p_t	probability: t users are privileged
s	number of transmitted bits / shares
SUC_{rat}	success ratio after phase 1
$\|\mathcal{T}\|$	number of privileged users
t'_i	priv. users in unsorted sub-set i
$\overline{t_s}$	average priv. users in sorted sub-set s

Table 4.1: Notations: Theorem 4.2.4

$$\overline{t_i} = \Phi^{-1}(1 - \frac{i}{M}) \tag{4.4}$$

where Φ^{-1} be the quantile function of the Gaussian probability distribution $N(\frac{|\mathcal{T}|}{2}, \frac{|\mathcal{T}|}{4}(1 - \frac{|\mathcal{T}|}{|\mathcal{N}|})\frac{|\mathcal{N}|}{|\mathcal{N}|-1})$. As we assume the number $s \ll M$ we approximate the value $t_i \approx \overline{t_s}$ for all $i < s$, so the bias' of all the sub-sets used for transmitting the key-bits are estimated to be equal to the bias of the last used sub-set in step s (note, that the scheme is more efficient than approximated here as the other bias' are higher, see Fig. 4.3 for illustration).

Using these approximations we can now calculate that each user in $\mathcal{T}$ has received every key-bit with probability $\frac{\overline{t_s}}{|\mathcal{T}|} > \frac{1}{2}$, thus he has received $\frac{\overline{t_s}}{|\mathcal{T}|}s$ key-bits on average

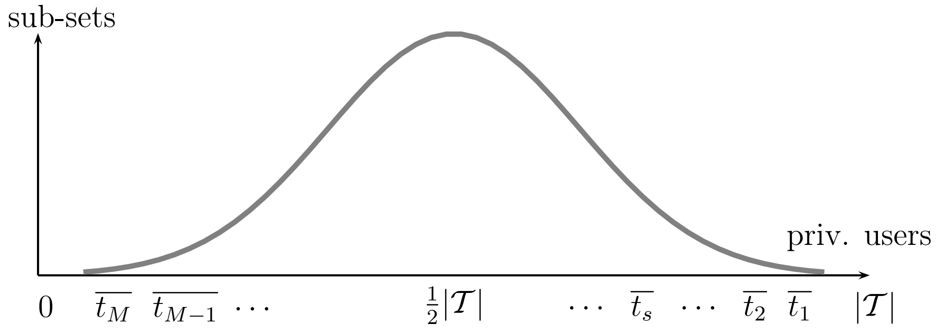

Figure 4.3: Distribution of privileged users (in sorted sub-sets)

and the number of key-bits received by each privileged user is (by approximation of the binomial distribution $B(s, \frac{\overline{t_s}}{|\mathcal{T}|})$) Gaussian distributed with parameters $N(\frac{\overline{t_s}}{|\mathcal{T}|}s, \frac{\overline{t_s}}{|\mathcal{T}|}s(1 - \frac{\overline{t_s}}{|\mathcal{T}|}))$.

For the users is $\mathcal{N} - \mathcal{T}$ we set $\overline{n_s} := \frac{1}{2}|\mathcal{N}| - \overline{t_s}$ as the remaining number of non-privileged users in the sub-set N_s and can thus determine the probability $\frac{\overline{n_s}}{|\mathcal{N}-\mathcal{T}|} < \frac{1}{2}$ for these users to receive a certain key bit. Analogously to the privileged users we approximate the binomial distribution $B(s, \frac{\overline{n_s}}{|\mathcal{N}-\mathcal{T}|})$ by the Gaussian distribution

$$N\left(\frac{\overline{n_s} \cdot s}{|\mathcal{N} - \mathcal{T}|}, \frac{\overline{n_s} \cdot s}{|\mathcal{N} - \mathcal{T}|}(1 - \frac{\overline{n_s}}{|\mathcal{N} - \mathcal{T}|})\right) \quad .$$

The ratios FR_{rat} and SUC_{rat} can then be calculated from the cumulative distribution functions by choosing d (and d') as parameters for the privileged (and non privileged) users. □

4.2.6.1 Regarding Approximations

See. Table 4.2 for a compilation of the approximations used for calculating the scheme's efficiency. In order to prove Theorem 4.2.4 we used a Gaussian distribution as approx-

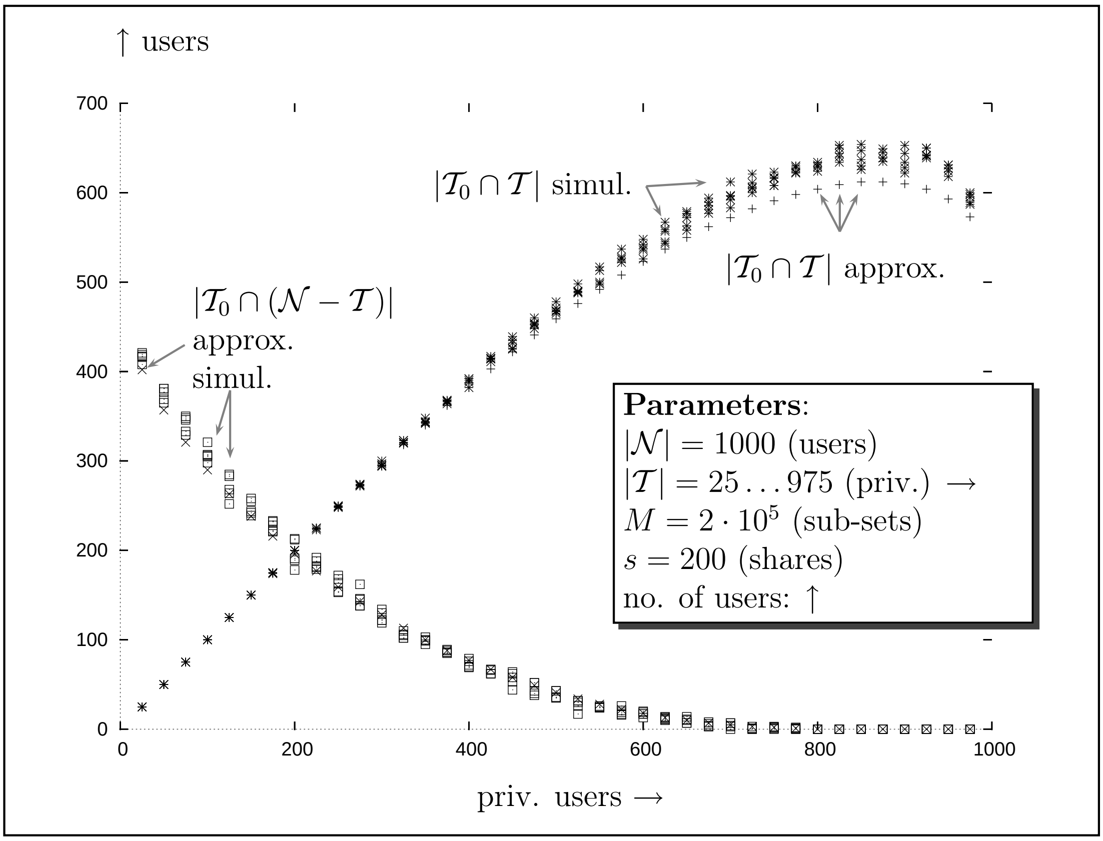

Figure 4.4: Approximation vs. Simulation Results

imation of the hypergeometric distribution of the values $t'_i := |N'_i \cap \mathcal{T}|$, being itself an estimated parameter used in two binomial distributions that are also approached using the normal approximation. At each step we introduced an error that is supposed to be *small* for *large* numbers $|\mathcal{N}|$ and $|\mathcal{T}|$ but the magnitude of this error can still be considerable because each partial error is carried over and accumulated to the next step. We take this into account and run the simulations (see Annex A) of our scheme without any approximated parameters. A comparison is drawn in Fig. 4.4 where the approximated values following Theorem 4.2.4 are denoted by the symbol × (for free-riders, *i. e.*, non-privileged users receiving $d' := s/2$ key bits) and + (for the privileged

value	**approximated by**
hypergeometric distribution of the values $t'_i := \lvert N'_i \cap \mathcal{T}\rvert$ (the number of privileged users per sub-set)	Gaussian distribution $N(\frac{\lvert\mathcal{T}\rvert}{2}, \frac{\lvert\mathcal{T}\rvert}{4}(1-\frac{\lvert\mathcal{T}\rvert}{\lvert\mathcal{N}\rvert})\frac{\lvert\mathcal{N}\rvert}{\lvert\mathcal{N}\rvert-1})$
sorted average value $\overline{t_i}$	$\overline{t_s}$ for $i \leq s$
binomial distribution $B(s, \frac{\overline{t_s}}{\lvert\mathcal{T}\rvert})$	Gaussian distribution $N(\frac{\overline{t_s}}{\lvert\mathcal{T}\rvert}s, \frac{\overline{t_s}}{\lvert\mathcal{T}\rvert}s(1-\frac{\overline{t_s}}{\lvert\mathcal{T}\rvert}))$
binomial distribution $B(s, 1-\frac{\overline{t_s}}{\lvert\mathcal{T}\rvert})$	Gaussian distribution $N((1-\frac{\overline{t_s}}{\lvert\mathcal{T}\rvert})s, \frac{\overline{t_s}}{\lvert\mathcal{T}\rvert}s(1-\frac{\overline{t_s}}{\lvert\mathcal{T}\rvert}))$

Table 4.2: Approximations used for efficiency calculations

users receiving $d := s/2$ key-bits). The real values are dithering, *i. e.,* show different results on the y-axis since they are outcomes of a probabilistic algorithm, the approximated values (calculated with the script of Fig. A.1 that implements the formulae of Theorem 4.2.4) provide accurate results.

Example 4.2.5. Let us assume $|\mathcal{N}| = 10000$ and $|\mathcal{T}| = 4000$. We want to calculate the probability P_t that a randomly chosen sub-set with $\frac{1}{2}|\mathcal{N}| = 5000$ elements contains 1950 or less privileged users (note: the expected or mean value is 2000 users).

The exact value of P_t is (by using equation 4.3) given by

$$P_t = \sum_{t=0}^{1950} p_t = \sum_{t=0}^{1950} \frac{\binom{|\mathcal{T}|}{t}\binom{|\mathcal{N}\setminus\mathcal{T}|}{\frac{1}{2}|\mathcal{N}|-t}}{\binom{|\mathcal{N}|}{\frac{1}{2}|\mathcal{N}|}} = \sum_{t=0}^{1950} \frac{\binom{4000}{t}\binom{6000}{5000-t}}{\binom{10000}{5000}} = 0.0216456 \quad (4.5)$$

while the normal approximation provides the value

$$
\begin{aligned}
P_t &\approx \frac{1}{\sigma \cdot \sqrt{2\pi}} \int_{-\infty}^{1950} e^{-\frac{1}{2}\left(\frac{t-\mu}{\sigma}\right)^2} \mathrm{d}t \\
&= \left(\frac{1}{\sqrt{\frac{|\mathcal{T}|}{4}(1 - \frac{|\mathcal{T}|}{|\mathcal{N}|})\frac{|\mathcal{N}|}{|\mathcal{N}|-1}}\sqrt{2\pi}} \right) \int_{-\infty}^{1950} e^{-\frac{1}{2}\left(\frac{t-\frac{|\mathcal{T}|}{2}}{\sqrt{\frac{|\mathcal{T}|}{4}(1-\frac{|\mathcal{T}|}{|\mathcal{N}|})\frac{|\mathcal{N}|}{|\mathcal{N}|-1}}}\right)^2} \mathrm{d}t \\
&= \left(\frac{1}{\frac{1000}{3333}\sqrt{2 \cdot 6666 \cdot \pi}} \right) \int_{-\infty}^{1950} e^{-\frac{1}{2}\left(\frac{t-2000}{\frac{1000}{3333}\sqrt{6666}}\right)^2} \mathrm{d}t \\
&= 0.0206185 \qquad (4.6)
\end{aligned}
$$

so we can observe that this parameters, which are relevant for practical use cases, show a rather sound approximation of the hypergeometric distribution by the normal distribution.

4.2.6.2 Parameter results of the theorem

The scheme can be parameterized with the values s and M (and with $|\mathcal{N}|$ if we use a batch split of the user set where we can select the batch size). From these values we determine d and d' (from s and the computation power of the users) and calculate (by the approximations from Theorem 4.2.4) the free-rider ratio FR_{rat} and success-ratio SUC_{rat} for each target size $|\mathcal{T}|$, hence the total number of messages: $(1-SUC_{rat})|\mathcal{T}|+s$ (Phase 1: s messages, Phase 2: one message per privileged user that was not successfully provided with enough key bits).

We summarize the scheme's parameters:

Corollary 4.2.6. *The Biased Sub-set Scheme provides a broadcast encryption scheme* *with the following parameters (with $n := |\mathcal{N}|$ and $t := |\mathcal{T}|$).*

There are $\frac{1}{2}M$ keys assigned per user, BC needs to store a total number of M keys and the broadcast header consists of $s + \lceil (1 - SUC_{rat}) \cdot t \rceil$ messages (where SUC_{rat} is dependent on n, t, s and M).

The scheme is 1-resilient and tolerates $\lceil FR_{rat} \cdot (n - t) \rceil$ free-riders.

4.2.7 Batches of Users

As mentioned before we intend (following Definition 3.3.2) to divide the user set into batches of a certain size which we still denote $|\mathcal{N}|$ to avoid unnecessary notations and run the scheme for each batch serially.

The reason to use smaller batches instead of running the scheme for the full set of users is the size of private key space for each user and an increase of the resiliency (see Theorem 4.4.1 for the latter aspect). The parameter M (number of sub-sets) determines the bias of a sorted sub-set towards the target set which is also dependent on the total number of users (see Fig. A.4 in the annex for illustration). The bias then determines the number of free-riders. Thus, in order to have a low free-rider ratio for a rather large set of users, either M needs to be increased or the number of users needs to be limited to a batch size and an independent scheme is applied on each batch.

Let the number of batches that make up the real user-base be denoted by m so our total number of users is $|\mathcal{N}|m$. We now face the problem of selecting the parameters during the set-up phase: batch size $|\mathcal{N}|$, number of sub-sets M – and for each transmission the parameters s, d and d'. There is obviously a tradeoff when we fix a number of free-riders: For a smaller batch size, we have better biases and need less sub-sets (and less keys to be stored by the users), but we need to run the whole scheme more often and increase the transmission length.

Example 4.2.7. For the parameters $|\mathcal{N}| = 1,000$, $|\mathcal{T}| = 700$, $M = 100,000$, $s = 110$ and $d' = 55$ the scheme results to a free-rider ratio $FR_{rat} = 0.06$ and 287 header messages.

In order to fulfill a requirement that the free-rider ratio shall not exceed 0.03 we have several options: we could either increase M to $1,000,000$ (then $FR_{rat} = 0.029$ and a header of 258 messages is achieved) or reduce with $m = 2$ the batch size to $|\mathcal{N}| = 500$ (then $|\mathcal{T}| = 350$, $FR_{rat} = 0.012$ and a header of $m \cdot 170 = 340$ messages is achieved).

Thus, we can choose between two alternatives: use a batch split and increase the total transmission header but keep the private key size or increase the private key size and keep (or reduce) the transmission header length.

In the improved scheme presented in the next section we can also increase the parameter s since $d' = d$ can be chosen freely then, thus it will be possible to reduce the free-riders by increasing the header length without using a batch split.

4.3 The Improved Biased Sub-set Scheme

4.3.1 Shortcomings of the Bit-wise Scheme

In the Bit-wise Biased Sub-set Scheme the session key is broadcasted bit after bit to the user set. This approach can be regarded as an inaccurate secret sharing scenario where the full message is obtained from some secrets and a fair amount of computation power.

This scenario does not only introduce security subtleties because every user receives some bits of the session key so it does not remain secret during the protocol run, it also requires each privileged user to perform an extensive key search that puts a considerable load on the equipment and consumes time as well as energy.

The bit-wise scheme's weaknesses are summarized below:

- The non-authorized users who are not free-riders do receive partial information as they receive a certain amount of key-bits.

- Authorized users have to perform an exhaustive search for up to $s - d$ bits of the session key. This could be costly.

- Each bit that an unauthorized user does not receive doubles his computational expense required for computing the full session key k_S. However, this still requires a rather large spread $d - d'$ between the number of bits received by authorized users in $\mathcal{T}$ and those received by unauthorized users in $\mathcal{N} \setminus \mathcal{T}$. Furthermore, estimating the computational power of adversaries is difficult, since exhaustive key-search can be easily parallelized and media content is sufficiently popular to attract many users in participating in a parallelized search for session keys.

4.3.2 Improvements Based on Secret Sharing

The bit-wise scheme introduced in the previous section uses a partial transmission of the session key k_S and security is based on the statistical property that unauthorized users receive on the average fewer bits (d' bits) of the session key than authorized users (d bits). We introduce a modification of the scheme by applying a cryptographic secret-sharing mechanism that introduces shared secrets to be used instead of partial information.

We will use the notion of a (k, n) *secret sharing scheme* consisting of two algorithms: `Share` and `Reconstruct`. Given a secret s the sharing algorithm `Share`(s) outputs n shares $s_1, \ldots, s_n$. Given shares $s_{i_1}, \ldots, s_{i_k}$, the reconstruction algorithm `Reconstruct`$(s_{i_1}, \ldots, s_{i_k})$ outputs the original shared secret s, so given any k of the n shares, the original secret s can be reconstructed, but knowledge of less than k shares does not reveal any information.

For our construction one of the first proposed schemes (Shamir's scheme: [91]) can be used. This scheme shares a secret $s \in F$ (*e. g., $F = Z_p$* with $p > n$) by choosing a random polynomial *pol* of degree $k-1$ and with constant term s (*i. e., $pol(0) = s$*) over F. The shares are defined as $s_i := (i, s(i))$, $i = 1, \ldots, n$, *i. e.,* each share is a point of the polynomial's graph. Given k different shares, the polynomial *pol* (and consequently the secret $s = pol(0)$) can be efficiently and uniquely reconstructed by performing a Lagrange interpolation.

The idea of applying secret sharing to overcome the limitations of the basic scheme is quite simple: The improvement is to replace the *bit-wise* broadcast by the transmission of shares of the secret to the sub-sets instead.

There is no modification necessary for the Setup algorithm but we will modify Broadcast and Decrypt.

4.3.3 Modification of the Broadcast Algorithm

In **Phase 1** of the improved scheme Broadcast applies a (d, s)-secret-sharing scheme to the session key k_S, which results in the shares $s_1, \ldots, s_s$. Instead of encrypting and broadcasting single bits of the key k_S, the sender encrypts the share s_1 with sub-set key k_{N_1} and broadcasts the encrypted share (which can only be decrypted by members of N_1). Afterwards s_2 is sent to N_2, etc., resulting in the tuple

$$\big((\pi^{-1}(1), E_{k_{N_1}}(s_1)), (\pi^{-1}(2), E_{k_{N_2}}(s_2)), \ldots, (\pi^{-1}(s), E_{k_{N_s}}(s_s))\big)$$

that is sent via the broadcast channel.

Regarding **Phase 2** we define analogously to the bit-wise scheme that for all $1 \leq j \leq s$ let $\mathcal{T}'_j \subset \mathcal{T}$ be the set of users in $\mathcal{T}$ that have received at least d shares after $E_{k_{N_j}}(s_j)$ is broadcasted and no further modifications need to be applied as the full session key can be sent to each user in $u \in \mathcal{T} - \mathcal{T}'_s$ with the same tuple

$$\Big((u_1, E_{k^{indv}_{u_1}}(k_S)), (u_2, E_{k^{indv}_{u_2}}(k_S)), \ldots, (u_g, E_{k^{indv}_{u_g}}(k_S))\Big)$$

as in the bit-wise scheme.

4.3.4 Modification of the Decrypt Algorithm

The algorithm Decrypt of user u reads parameters M, d, s and receives the first tuple

$$\big((\pi^{-1}(1), C_1), (\pi^{-1}(2), C_2), \ldots, (\pi^{-1}(s), C_s)\big)$$

and the respective shares are decrypted from the C_i for known sub-set keys k_i. Given at least d shares Decrypt can apply `Reconstruct` to efficiently reconstruct the secret k_S. Therefore, instead of performing an exhaustive search for the missing key bits, a receiver only performs a Lagrange interpolation to compute the complete session key.

If less than d shares can be decrypted then the second tuple is parsed in the same way as in the bit-wise scheme and the session key can be decrypted by Decrypt for user u by using the individual key k^{indv}_u.

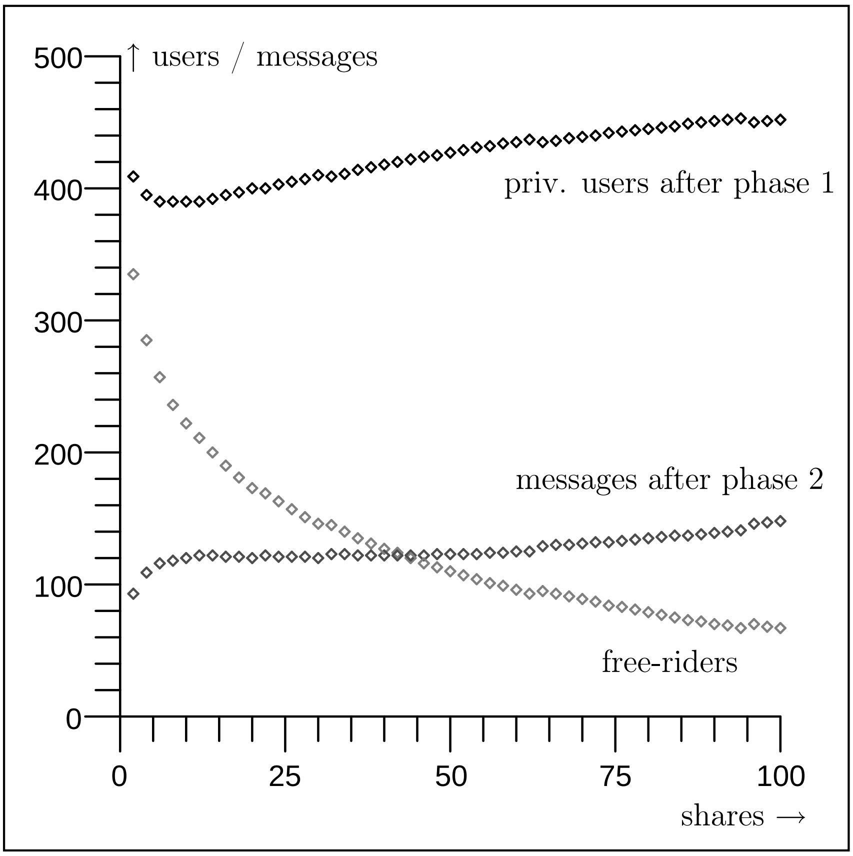

Figure 4.5: Number of shares as parameter

Example 4.3.1. See Fig. 4.5 for an example (simulation data) where the number of shares being sent in phase 1 is a parameter on the x-axis while the other parameters, number of users $|\mathcal{N}| = 1000$, number of privileged users $|\mathcal{T}| = 500$, number of sub-sets $M = 10^6$ are fixed. In phase 1 the number of messages is the number of shares and half of the shares are needed for secret reconstruction. The total number of messages after phase 2 is given by the number of shares sent in phase 1 plus the message per privileged user who receives the session key in phase 2. The number of free-riders can be decreased with a higher number of shares (and the cost of a higher number of total messages after both phases).

4.3.5 Security of the Improved Scheme

The goal of the modifications was to improve the security of the bit-wise scheme. While we still have to concede the existence of free-riders we can now use the same threshold parameter d for privileged users as well as attackers. Moreover, the modifications provide unconditional security in the sense that non-privileged users who do not become free-riders do not learn anything about the key k_s.

Theorem 4.3.2. *Let $\mathcal{N}$ be the set of users of a broadcast system and let the algorithms* Setup*,* Broadcast *and* Decrypt *be defined as in the Improved Biased Sub-set Scheme with parameters M, d and s.*

For each sub-set $\mathcal{T}$ of $\mathcal{N}$ let $\mathcal{T}_0$ be the union of $\mathcal{T}$ and the non-privileged users who received d or more shares after the phase 1 tuple is broadcasted (these users are called free-riders).

Then the resulting scheme constitutes a broadcast encryption system tolerating free-riders (according to Definition 3.2.2).

Proof. Let $\mathcal{N}$ be the set of users, $\mathcal{T}$ be the target set and M, d, s be the scheme's parameters.

We re-use the results from the proof of Theorem 4.2.2. Every user $u \in \mathcal{T}$ does receive the session key k_s of a transmission either in phase 1 by reconstructing the secret or in phase 2 by individual decryption of the full encrypted session key k_s.

A non-privileged user either becomes a free-rider and is a member of $\mathcal{T}_0 \setminus \mathcal{T}$ or does not become a free-rider and no information about k_s can be derived from the knowledge of less than d shares (also not if unlimited computation power was available) so the requirements of Definition 2.3.5 are fulfilled and unconditional security is achieved. □

Any unauthorized user in $\mathcal{N} - \mathcal{T}$ is unable to gain any information about the session key as long as he receives less than d shares. This is a significant improvement over the basic scheme, where an attacker could use extra time or extra computation power to derive more key-bits than ordinary users – the threshold value d in the improved scheme is a hard threshold.

4.4 Resiliency of the Schemes

The proposed schemes offer a major drawback: they are highly vulnerable to colluders being able to combine their respective set of sub-set keys.

Users who could combine their pre-distributed keys would receive more key-bits (or shares in the improved scheme) than any other user. In the case of two users sharing their sub-set key pool they would increase their portion of known sub-set keys from 0.5 to 0.75 each. This is higher than a reasonable bias being achieved by sorted sub-sets, thus the two users would become pirate free-riders for all transmissions! It is obvious that the schemes do not offer any resiliency for colluders being a member of the same user set.

However, for different reasons users may be grouped together in batches (see Definition 3.3.2) and users from different batches cannot gain anything from collusion when the scheme is run serially for each batch and different key encryption keys for the session key would be used. Note, that we do need to introduce keys for encrypting the session key in a way that for each batch the scheme is used to transmit a batch key encryption key and the encrypted session key is sent encrypted to the batch. This way, non-privileged users from different batches are unable to collude and partial key information or shares cannot be combined by users from different batches.

The resiliency is 1 from a worst-case point of view (if unfortunately two users from the same batch collude then they can determine the key k_s) or dependent on the number of batches from an average-case point of view. We will show in the next theorem that the birthday paradox can be appied here if the batch split is performed in a proper way. So the resiliency of our proposed schemes can be roughly approximated by the square root of the number of batches after the split.

Theorem 4.4.1. *Let a* 1*-resilient broadcast encryption scheme with user set* $\mathcal{N}$ *be split up into* b *disjoint batches*

$$\mathcal{N} =: \mathcal{U}_1 + \mathcal{U}_2 + \cdots \mathcal{U}_b$$

according to Definition 3.3.2, where the batches are of the same size, i. e., $|\mathcal{U}_1| = |\mathcal{U}_2| = \cdots = |\mathcal{U}_b|$ *and each user is assigned its batch randomly and uniformly. Then the probability that* $k < b$ *non-privileged users (that are chosen independently from the batch assignment) are successfully able to collude is given by*

$$P_C = 1 - \frac{b!}{(b-k)! \cdot b^k} \quad .$$

Proof. According to Definition 3.3.2 the associated algorithms Setup and Broadcast operate in the following way: Setup is applied to each of the $\mathcal{U}_i$ to assign the users' private keys for the b schemes independently. For each scheme a target set $\mathcal{T} \cap \mathcal{U}_i$ is then addressed by Broadcast. In order to collude successfully, at least two users must be in the same batch since we deal with independent 1-resilient broadcast encryption schemes with disjoint user sets. Thus, we have an application of the birthday paradox and can determine the size of the space of all selections of k batches by b^k, the number of k-permutations of the b batch numbers by $\frac{b!}{(b-k)!}$ and – since the batch assignments were chosen uniformly – we have the probability for a colliding assignment

$$P_C = 1 - \frac{\frac{b!}{(b-k)!}}{(b^k)} = 1 - \frac{b!}{(b-k)! \cdot b^k}$$

and the assertion is proved. □

Example 4.4.2. If we consider one million users and split them up into 1000 batches, each containing 1000 users or – alternatively – 100 batches with 10,000 users then we have the following dependencies between the number of colluders k and the probability of collusion success P_C.

b	k	P_C	b	k	P_C
1,000	5	0.00997	100	2	0.01
1,000	10	0.04414	100	5	0.09655
1,000	15	0.10014	100	10	0.37184
1,000	20	0.17407	100	12	0.49685
1,000	30	0.35554	100	14	0.61479
1,000	50	0.71227	100	16	0.71841

If a collusion success probability of less than 0.05 is considered insignificant in a practical application it could be stated that a split into thousand batches (almost) provides 10-resiliency for 1-reslient schemes.

Apart from the average-case analysis, the general construction from Fiat and Naor described in section 3.4.4 (Theorem 3.4.8) can be used to achieve higher resiliency for the two schemes. This approach is detailed in the next section.

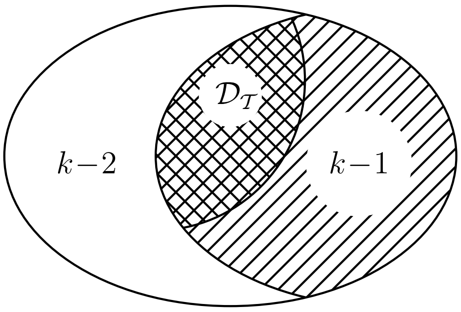

Figure 4.6: Resiliency Domains

4.4.1 Higher-Resiliency Construction

While the general construction of Fiat and Naor to achieve k-resiliency for broadcast encryption schemes by using 1-resilient schemes as building blocks is valid for any 1-resilient scheme, the header-length computations of the resulting scheme require the building block schemes to be zero-header schemes (following the Definition 3.3.1) which is not the case for the probabilistic schemes introduced in this chapter. Fortunately, it is possible to customize the general construction to exploit certain properties of the biased-sub-set scheme and to gain acceptable header-lengths for reasonable use-cases.

Before we apply Fiat and Naor's general construction to the Improved Biased Sub-Set Scheme and examine the resulting scheme we introduce a relaxed notion of resiliency before.

Definition 4.4.3. (k-resiliency on a resiliency domain) A broadcast encryption scheme with user set $\mathcal{N}$ and target set $\mathcal{T}$ is said to be k-resilient on a resiliency domain $\mathcal{D}_\mathcal{T} \subseteq \mathcal{N}$ if it is resilient to every set $\mathcal{S} \subseteq \mathcal{D}_\mathcal{T}$ with $|\mathcal{S}| \leq k$.

The background of this definition is the existence of free-riders in the 1-resilient schemes. While we could use the schemes as building blocks for k-resilient schemes, the construction of Fiat and Naor does not consider the existence of free-riders and induces a possibly complex structure on the set of users (see Fig. 4.6 as an example). We have a resiliency domain $\mathcal{D}_\mathcal{T}$ of which any k members are unable to collude while there could be other domains being (loosely speaking) itself $(k-1)$- or $(k-2)$-resilient as long as members of the same domain do collude internally or with higher-resilient domains. Of pivotal interest is the existence of a domain $\mathcal{D}_\mathcal{T}$ providing k-resiliency according to Definition 4.4.3 which is addressed in the next theorem and the size of $\mathcal{D}_\mathcal{T}$ that is determined by Theorem 4.4.6.

Theorem 4.4.4. *For a given Improved Biased Sub-Set Scheme with user set $\mathcal{N}$ and target set $\mathcal{T}$ where every user is assigned z keys, there exists a k-resilient broadcast encryption scheme on a resiliency domain $\mathcal{D}_\mathcal{T} \subseteq \mathcal{N}$ assigning $\lceil kz \log N \rceil$ keys .*

Proof. We follow the construction in the proof of Theorem 3.4.8. Let l, m be positive integers and $\{f_i\}_{i=1}^{l}$ be a family of functions with $f_i : \mathcal{N} \to \{1, 2, \ldots, m\}$ with the property that for all sub-sets $\mathcal{S} \subseteq \mathcal{N}$ with $|\mathcal{S}| \leq k$ there exists one i with $u \neq u' \Rightarrow f_i(u) \neq f_i(u')$ for all $u, u' \in \mathcal{S}$.

For (i, j) with $1 \leq i \leq l$ and $1 \leq j \leq m$ we will use an independent instance[1] of the Improved Biased Sub-Set Scheme (as laid out in Section 4.3) and denote it by the triple $(\textsc{Broadcast}^{(i,j)}, \textsc{Setup}^{(i,j)}, \textsc{Decrypt}^{(i,j)})$. Let $P_u^{(i,j)}$ be the secret key output of algorithm $\textsc{Setup}^{(i,j)}$ for user u thus

$$k_{N'_\mu}^{(i,j)} \in P_u^{(i,j)} \Leftrightarrow u \in {N'_\mu}^{(i,j)}$$

i. e., each user knows the key assigned to sub-set ${N'_\mu}^{(i,j)}$ of the scheme (i, j) when the user is a member of the sub-set.

[1] Each of those instances is a self-contained scheme with its own set of pre-distributed keys and chosen sub-sets.

Analogously to the proof of Theorem 3.4.8 we can construct a new scheme denoted by $(\textsc{Broadcast}, \textsc{Setup}, \textsc{Decrypt})$ where $\textsc{Setup}$ assigns to user u the secret key output $P_u := \left\{P_u^{(i,f_i(u))} : 1 \leq i \leq l\right\}$ and the $\textsc{Broadcast}$-algorithm selects the session key $K \in \mathcal{K}$ and other keys $K_1, K_2, \ldots K_l \in \mathcal{K}$ with

$$K = \left(\bigoplus_{i=1}^{l} K_i\right)$$

after the key K_l is computed appropriately.

Let $B^{(i,j)}$ be the output of $\textsc{Broadcast}^{(i,j)}(\mathcal{T}^{(i,j)}, K_i)$ where the set $\mathcal{T}^{(i,j)}$ contains all users of the target set that are mapped to the value j by the function f_i. $\textsc{Broadcast}$ outputs all the $B^{(i,j)}$ as one concatenated string.

The resiliency domain $\mathcal{D}_\mathcal{T} \subseteq \mathcal{N}$ is given by the set of all non-privileged users who do not become free-riders in any of the l building block schemes:

$$\mathcal{D}_\mathcal{T} := \{u \in (\mathcal{N} \setminus \mathcal{T}) : \forall i \in \{1, \ldots, l\} : u \text{ is not freerider of scheme } (i, f_i(u))\}$$

Each privileged user is then able to decrypt all the keys $K_1, K_2, \ldots K_l$ and to compute the session key K while any k-set of colluding not-privileged users (in the resiliency domain) misses at least one of the keys since there exists by requirement an index i with $u \neq u' \Rightarrow f_i(u) \neq f_i(u')$ for all u, u' of the k-set and K_i cannot be decrypted by any of the colluders who can also not combine their keys as they belong to different independent broadcast encryption schemes regarding the decryption of K_i. This reasoning is detailed in the proof of Theorem 3.4.8 where also the parameter dependencies $m = 2k^2$ and $l = \lceil k \log n \rceil$ are established. □

Remark 4.4.5. Despite the fact that we have a valid construction, the header length of the resulting scheme can be seen as a “problematic” parameter since we concatenate $lm \approx 2k^3 \log n$ individual scheme headers to one resulting header so we end up with large numbers even for small resiliency parameters k.

Moreover, we can observe that the independent Improved Biased Sub-Set Schemes that are grouped together in the construction operate redundantly in one important facet: Every scheme incorporates the Phase 2 where, for each user who has not received enough shares, the key K_i is sent individually encrypted by using the user’s unique secret key $k_u^{(i,f_i(u))\ \ indv} \in P_u^{(i,f_i(u))}$. As each user is part of a total of $l = \lceil k \log n \rceil$ different Improved Biased Sub-Set Schemes, it is possible that one user receives several individual messages, each belonging to different building block schemes. This is obviously inefficient because the full session key K could be delivered with a message of the same size and none of the K_i is then needed by this user.

Another observation is that in order to become a free-rider (disregarding collusions now) a user needs to be a free-rider in all the $l = \lceil k \log n \rceil$ different independent schemes at once which can be seen as rather unlikely, even for small resiliency parameters k. In order to increase efficiency it appears to be reasonable to allow rather great free-rider ratios in the building block schemes as long as the free-rider ratio of the resulting scheme is tolerable.

Finally, it shall be noted that we there is no minimum requirement on the resiliency domain in the claim of Theorem 4.4.4, so it is possible to obtain a trivial domain (*e. g.*, an empty set) but achieve misleading high resiliency parameters.

We will modify the construction in the proof of the next theorem in a way that the properties of the internal constructions of the Improved Biased Sub-Set Scheme are utilized more efficiently. The strategy of the construction is to use only the phase 1 of each building block scheme and run a phase 2 afterwards for all remaining privileged users.

Theorem 4.4.6. *For a given Improved Biased Sub-Set Scheme with user set $\mathcal{N}$, target set $\mathcal{T}$ and parameters: number of sub-sets M, number of transmitted shares s and threshold value d, there exists a k-resilient broadcast encryption scheme on a resiliency domain $\mathcal{D}_\mathcal{T} \subseteq \mathcal{N}$ of approximated size $|\mathcal{D}_\mathcal{T}| \approx |\mathcal{N} - \mathcal{T}|(1 - FR_{rat}^{(i,j)})^{\lceil k \log n \rceil}$ assigning $Mk\log(|\mathcal{N}|)$ keys to each user and an average header length of $(1 - SUC_{rat})|\mathcal{T}| + s\lceil k \log n \rceil$ messages, where the free-rider ratio $FR_{rat} = (FR_{rat}^{(i,j)})^{\lceil k \log n \rceil}$ is determined by the building block free-rider ratios $FR_{rat}^{(i,j)} = 1 - \Phi_{\mathcal{N}-\mathcal{T}}(d)$ and where $\Phi_{\mathcal{N}-\mathcal{T}}$ is the distribution function of the normal distribution $N\left((1 - \frac{\overline{t_s}}{|\frac{|\mathcal{T}|}{2k^2}|})s, \frac{\overline{t_s}}{|\frac{|\mathcal{T}|}{2k^2}|}s(1 - \frac{\overline{t_s}}{|\frac{|\mathcal{T}|}{2k^2}|})\right)$ and $\overline{t_s}$ can be approximated by $\overline{t_s} = \Phi^{-1}(1 - \frac{s}{M})$ where Φ^{-1} is the quantile function of the Gauss distribution $N\left(\frac{|\mathcal{T}|}{4k^2}, \frac{|\mathcal{T}|}{4k^2}(1 - \frac{|\mathcal{T}|}{2k^2|\mathcal{N}|})\right)$ and $SUC_{rat} = (SUC_{rat}^{(i,j)})^l$ with $SUC_{rat}^{(i,j)} = 1 - \Phi_\mathcal{T}(d)$ where $\Phi_\mathcal{T}$ is the distribution function of the the distribution $N\left(\frac{2\overline{t_s}k^2 s}{|\mathcal{T}|}, \frac{2\overline{t_s}k^2 s}{|\mathcal{T}|}\left(1 - \frac{2\overline{t_s}k^2}{|\mathcal{T}|}\right)\right)$.*

Proof. We follow Fiat and Naor's construction as laid out in the proof of Theorem 3.4.8. Let l, m be positive integers and $\{f_i\}_{i=1}^{l}$ be a family of functions with $f_i : \mathcal{N} \to \{1, 2, \dots, m\}$ with the property that for all sub-sets $\mathcal{S} \subseteq \mathcal{N}$ with $|\mathcal{S}| \leq k$ there exists one i with $u \neq u' \Rightarrow f_i(u) \neq f_i(u')$ for all $u, u' \in \mathcal{S}$.

For (i, j) with $1 \leq i \leq l$ and $1 \leq j \leq m$ we will use an independent instance of phase 1 of the Improved Biased Sub-Set Scheme and denote it by the triple $(\textsc{Broadcast}^{(i,j)}, \textsc{Setup}^{(i,j)}, \textsc{Decrypt}^{(i,j)})$. Let $P_u^{(i,j)}$ be the secret key output of algorithm $\textsc{Setup}^{(i,j)}$ for user u thus

$$k_{N'_\mu}^{(i,j)} \in P_u^{(i,j)} \Leftrightarrow u \in {N'_\mu}^{(i,j)}$$

i. e., each user knows the key assigned to sub-set ${N'}_{\mu}^{(i,j)}$ of the scheme (i, j) when the user is a member of the sub-set.

Phase 1: We can now construct a new scheme denoted by $(\textsc{Broadcast}, \textsc{Setup}, \textsc{Decrypt})$ where $\textsc{Setup}$ assigns to user u the secret key output $P_u := \left\{P_u^{(i,f_i(u))} : 1 \leq i \leq l\right\}$ and the $\textsc{Broadcast}$-algorithm selects the session key $K \in \mathcal{K}$ and other keys $K_1, K_2, \ldots K_l \in \mathcal{K}$ with

$$K = \left(\bigoplus_{i=1}^{l} K_i\right)$$

after the key K_l is computed appropriately.

Let $B^{(i,j)}$ be the output of $\textsc{Broadcast}^{(i,j)}(\mathcal{T}^{(i,j)}, K_i)$ where the set $\mathcal{T}^{(i,j)}$ contains all users of the target set that are mapped to the value j by the function f_i. $\textsc{Broadcast}$ outputs all the $B^{(i,j)}$ as one concatenated string.

Phase 2: Subsequently, $\textsc{Broadcast}$ outputs individual messages to all users in $\mathcal{T}$ who have not received enough shares in at least one of the l building block schemes.

The resiliency domain $\mathcal{D}_{\mathcal{T}} \subseteq \mathcal{N}$ is given by the set of all non-privileged users who do not become free-riders in any of the l building block schemes:

$$\mathcal{D}_{\mathcal{T}} := \{u \in (\mathcal{N} \setminus \mathcal{T}) : \forall i \in \{1, \ldots, l\} : u \text{ is not freerider of scheme } (i, f_i(u))\}$$

Each privileged user is then either assigned the session key individually or is able to decrypt all the keys $K_1, K_2, \ldots K_l$ and to compute the session key K while any k-set of colluding not-privileged users (in the resiliency domain) misses at least one of the keys since there exists by requirement an index i with $u \neq u' \Rightarrow f_i(u) \neq f_i(u')$ for all u, u' of the k-set and K_i cannot be decrypted by any of the colluders who are unable to combine their keys as they belong to different independent broadcast encryption schemes regarding the decryption of K_i.

This reasoning is detailed in the proof of Theorem 3.4.8 where also the parameter dependencies $m = 2k^2$ and $l = \lceil k \log n \rceil$ are established.

In order to determine the number of free-riders in the independent building block schemes we first observe that each scheme (i, j) only regards a sub-set of $\mathcal{T}$ as its privileged user set we have

$$\mathcal{T}^{(i,j)} = \{u \in \mathcal{T} : f_i(u) = j\}$$

and we can approximate

$$|\mathcal{T}^{(i,j)}| \approx \frac{|\mathcal{T}|}{m} = \frac{|\mathcal{T}|}{2k^2}$$

by assuming that the function f has an equal distribution (in implementation cases a PRNG would be used, see the proof of Theorem 3.4.8). The other parameters of the building block schemes: number of sub-sets M, number of transmitted shares s and threshold value d are not changed by the construction. Thus, we can now apply Theorem 4.2.4 to compute the efficiency of each building block scheme.

$FR_{rat}^{(i,j)} = 1 - \Phi_{\mathcal{N}-\mathcal{T}}(d)$ where $\Phi_{\mathcal{N}-\mathcal{T}}$ is the distribution function of the normal distribution

$$N\left((1 - \frac{\overline{t_s}}{|\frac{|\mathcal{T}|}{2k^2}|})s, \frac{\overline{t_s}}{|\frac{|\mathcal{T}|}{2k^2}|}s(1 - \frac{\overline{t_s}}{|\frac{|\mathcal{T}|}{2k^2}|})\right)$$

and $\overline{t_s}$ can be approximated by $\overline{t_s} = \Phi^{-1}(1 - \frac{s}{M})$ where Φ^{-1} is the quantile function of the Gauss distribution

$$N\left(\frac{|\frac{|\mathcal{T}|}{2k^2}|}{2}, \frac{|\frac{|\mathcal{T}|}{2k^2}|}{2}(1 - \frac{|\frac{|\mathcal{T}|}{2k^2}|}{|\mathcal{N}|})\right) = N\left(\frac{|\mathcal{T}|}{4k^2}, \frac{|\mathcal{T}|}{4k^2}(1 - \frac{|\mathcal{T}|}{2k^2|\mathcal{N}|})\right) .$$

To be a free-rider of the resulting scheme, a non-privileged user has to become a free-rider of each of the l independent schemes she takes part in, thus we have $FR_{rat} = (FR_{rat}^{(i,j)})^l$ and a total number of $FR_{rat}|\mathcal{N} - \mathcal{T}|$ free-riders. Regarding the success parameter we can apply Theorem 4.2.4 again and have

$$SUC_{rat}^{(i,j)} = 1 - \Phi_{\mathcal{T}}(d)$$

where $\Phi_{\mathcal{T}}$ is the distribution function of the the distribution

$$N\left(\frac{\overline{t_s}}{\frac{|\mathcal{T}|}{2k^2}}s, \frac{\overline{t_s}}{\frac{|\mathcal{T}|}{2k^2}}s(1 - \frac{\overline{t_s}}{\frac{|\mathcal{T}|}{2k^2}})\right) = N\left(\frac{2\overline{t_s}k^2s}{|\mathcal{T}|}, \frac{2\overline{t_s}k^2s}{|\mathcal{T}|}\left(1 - \frac{2\overline{t_s}k^2}{|\mathcal{T}|}\right)\right)$$

and $SUC_{rat} = (SUC_{rat}^{(i,j)})^l$ so a total number of $SUC_{rat}|\mathcal{T}|$ users can compute the session key after phase 1. In phase 2, the remaining $(1 - SUC_{rat})|\mathcal{T}|$ users receive their individual message containing the encrypted key K. We can determine the resulting scheme's header length by adding the $sl = s\lceil k \log n \rceil$ shares sent in total.

The size of the resiliency domain can be approximated by counting the non-privileged users who have not become a free-rider in any of the $\lceil k \log n \rceil$ schemes, thus we have

$$|\mathcal{D}_{\mathcal{T}}| \approx |\mathcal{N} - \mathcal{T}|(1 - FR_{rat}^{(i,j)})^{\lceil k \log n \rceil}$$

and all assertions are proved. □

4.5 Further Improvements

4.5.1 Free-Rider Elimination

It is possible to avoid the existence of free-riders if the biased-sub-set schemes are combined with a revocation scheme (*e. g.,* with Naor *et al.*'s sub-set-difference scheme [82], described by Definition 3.5.8).

We take advantage of the following observations

- The set of free-riders is known by the sender. The set could be determined before the protocol is started in order to take reasonable precautions.

- Different broadcast encryption schemes can be combined, *i. e.,* they can be run consecutively in a way that the first scheme distributes a pre-session key that is used for encrypting all the communication of the second scheme, thus, only the privileged users from the first scheme are able to take part in the session key distribution of the second scheme and the result is a set intersection operation of the privileged user sets. The same result can be achieved if two independent pre-session keys are distributed with the two schemes and the session key results from the exclusive-or operation of both pre-session keys.

- The biased-sub-set schemes introduce with respect to suitable parametrization a *small* amount of free-riders while the revocation schemes are designed for the exclusion of a *small* amount of revoked users so there is a straightforward concept to let the different schemes complement one another.

Taking this into account we can combine a revocation scheme with our scheme by first identifying the free-riders of the biased sub-set scheme and then run the revocation scheme **before** the biased sub-set scheme in order to distribute the pre-session key that excludes the set of anticipated free-riders from the protocol communication. As both protocols are unidirectional there is no real impact in the decision which scheme is run first because a receiver could always (if capable) record the broadcast and choose by itself which transmission is parsed first.

Note, that the extension of the biased sub-set scheme with a revocation scheme does not support unconditional security for the combined scheme because the revocation schemes only offer computational security, but the construction is still useful for practical implementations, especially when the sender wants to avoid free-riders only for few transmissions.

4.5.1.1 Revocation vs. Biases Sub-Sets without Free-riders

Taking the consideration from section 4.5.1 into account a sender that does not tolerate any free-riders may choose one of three alternatives

1. Use a revocation scheme (*e. g.,* SD scheme of section 3.5.8)
2. Use the biased sub-set Scheme and combine it with a revocation scheme in order to eliminate free-riders
3. Use a general broadcast encryption scheme without free-riders (*e. g.,* one of the straightforward schemes from section 3.3).

If we assume that the sender has already pre-distributed all necessary keys to run either of the schemes the decision will depend naturally on the number of messages (the header length) necessary for both alternatives.

For given parameters $|\mathcal{N}|$ and $|\mathcal{T}|$ we know that (in the average case) for the first alternative the sender has to transmit $1.25(|\mathcal{N}| - |\mathcal{T}|)$ messages in order to revoke the users in $\mathcal{N} - \mathcal{T}$ when we choose the SD scheme as revocation scheme. The second alternative requires the sender to transmit messages of three kinds, *i. e.,*

(i.) the shares used in the biased sub-set scheme,

(ii.) the individual messages in phase 2 of this scheme and

(iii.) the messages of the revocation scheme in order to revoke the free-riders of the biases sub-set scheme.

While the revocation schemes are generally suited for a rather small amount of free-riders and become inefficient for large sets of revoked users (*i. e.,* small sets of target users) the biased sub-set scheme is designed for a general case for arbitrary target sub-sets so the goal is to determine the *break-even point* that is (if it exists at all) a number of revoked users that is to be exceeded for letting Alternative 2 having a smaller total amount of messages than Alternative 1.

Example 4.5.1. See Fig. 4.7 as an example where Alternative 1 is the sub-set difference scheme and Alternative 2 is the biased-sub-set scheme combined with the sub-set difference scheme for free-rider elimination. (Simulation data with parameters: number of users $|\mathcal{N}| = 1000$, number of privileged users on x-axis, number of sub-sets $M = 10^5$, number of shares $s = 50$ and the total number of messages on the y-axis).

In this example we can see that the number of messages in the revocation scheme follows the a straight line (1.25 messages per revoked user) while the total number of messages of three kinds (*i. e.,* shares of Phase 1, individual messages of Phase 2 and revocation header for free-rider elimination) of Alternative 2 is a more sophisticated graph showing an increase for a decreasing number of revoked users. When a number

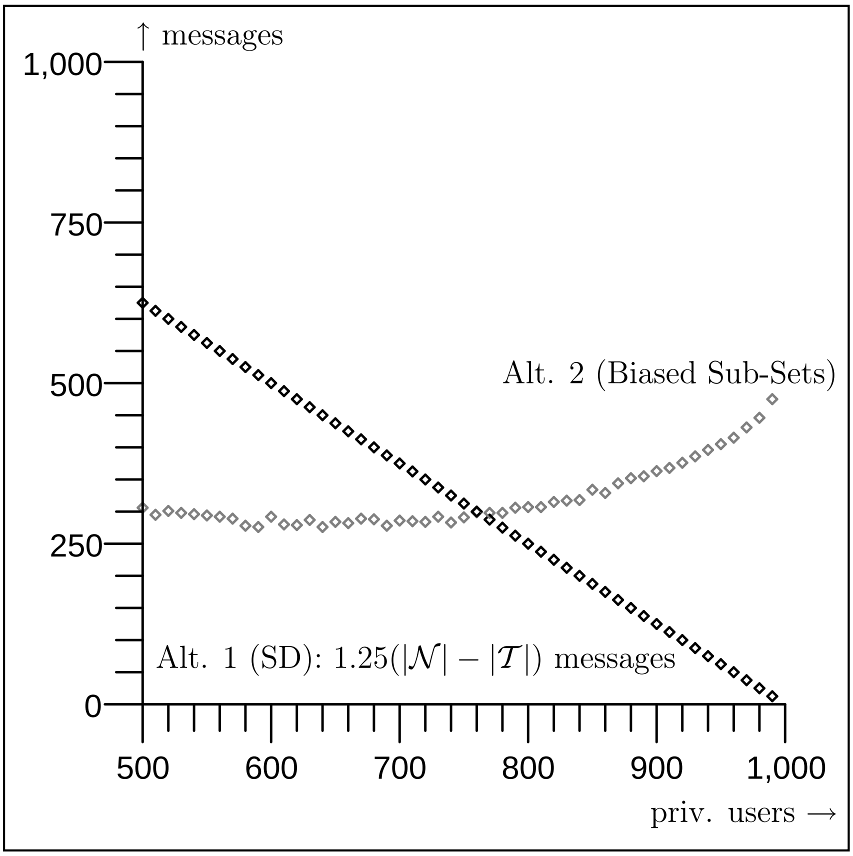

Figure 4.7: SD Revocation vs. Biases Sub-Sets without Free-riders

of 250 revoked users is exceeded, the Alternative 2 has passed the break-even point and becomes more efficient than the revocation scheme.

Note, that the existence of such a break-even point is not always guaranteed and dependent on the parameters used in the biased sub-set scheme. When we choose an extreme value, *e. g.,* let the number of shares sent in phase be higher than $1.25|\mathcal{N}|$ then the total number of messages of Alternative 2 is always higher than the $1.25(|\mathcal{N}| - |\mathcal{T}|)$ messages of Alternative 1 because the constant number of messages in phase 1 of the biased sub-set scheme is independent from the number of target users.

Definition 4.5.2. The *combined scheme* shall denote the broadcast encryption scheme that results from running the improved biased sub-set scheme and the sub-set difference scheme consecutively in a way that the free-riders that arise from the biased sub-set scheme are eliminated as revoked users by the sub-set difference scheme.

Remark 4.5.3. The idea to construct such a combination of schemes could have been based on any other revocation scheme as well. The reason to choose the sub-set difference schemes instead of a more advanced scheme (*e. g.,* the LSD scheme) are twofold. First, the scheme's average efficiency of 1.25 messages per revoked user is not substantially improved by the LSD scheme or its variants and the improved key storage per user is also less relevant as the biased sub-set scheme requires a comparable large key space for its own set-up. A second reason is given by the requirement of the LSD scheme that the number of users $n := |\mathcal{N}|$ shall be a power of 2 such that $\sqrt{\log n}$ is an integer. This requirement prohibits certain batch sizes being useful for practical combination of schemes, especially the values

$$2^9 = 512 < n < 65536 = 2^{16}$$

are relevant for batch sizes of the biased sub-set scheme, where choices in the range $1000 \ldots 10000$ users per batch show suitable results in practical settings. The SD scheme only requires powers of 2 so the choices 1024, …, 8192 are possible for scheme combination. Moreover, the good asymptotic efficiency results from the LSD schemes are realized only for very large numbers of users.

We fix the results or the considerations of this section in the following lemma.

Lemma 4.5.4. *The average number of messages sent for the combined scheme is given* *by*

$$s + (1 - SUC_{rat})|\mathcal{T}| + 1.25FR_{rat}(|\mathcal{N}| - |\mathcal{T}|)$$

using the notations from Table 4.1.

Proof. The transmission's session key can be constructed from two partial keys being added by the xor-operation. The first part is transmitted with the biased sub-set scheme while the second part is transmitted with the SD scheme. We calculate for this combined scheme the header lengths by using the notations from Table 4.1 and the results from Theorem 4.2.4. First, the sender transmits

$$s$$

messages to broadcast the shares then

$$(1 - SUC_{rat})|\mathcal{T}|$$

messages to provide the session key to the privileged users who have not received the necessary amount of shares to reconstruct the first key part and, finally,

$$1.25FR_{rat}(|\mathcal{N}| - |\mathcal{T}|)$$

to revoke the free-riders from the biased sub-set scheme as there is 1.25 messages per revoked user (average value) to transmit the second key part.

□

4.5.1.2 Break-Even Point of the Combined Scheme

In the annex (Fig. A.6) we provide some simulation data for a combined scheme consisting of the improved biased sub-set scheme and Naor *et al.*'s SD scheme.

In order to determine the break-even point, *i. e.,* to calculate the number of revoked users that let the combined scheme be more efficient (regarding the total number of messages) than the SD revocation scheme we can use the approximation from Theorem 4.2.4 and replace

$$SUC_{rat} \quad \text{by} \quad 1 - \Phi_{\mathcal{T}}(d)$$

as well as

$$FR_{rat} \quad \text{by} \quad 1 - \Phi_{\mathcal{N}-\mathcal{T}}(d)$$

to get an expression that needs to equal the

$$1.25(|\mathcal{N}| - |\mathcal{T}|)$$

messages necessary for only using the SD revocation scheme. Thus we can construct an equation

$$s + (1 - SUC_{rat})|\mathcal{T}| + 1.25FR_{rat}(|\mathcal{N}| - |\mathcal{T}|) = 1.25(|\mathcal{N}| - |\mathcal{T}|) \tag{4.7}$$

where $\Phi_{\mathcal{T}}$ is the distribution function of the the distribution

$$N\left(\frac{\overline{t_s}}{|\mathcal{T}|}s, \frac{\overline{t_s}}{|\mathcal{T}|}s(1 - \frac{\overline{t_s}}{|\mathcal{T}|})\right) .$$

and $\Phi_{\mathcal{N}-\mathcal{T}}$ is the distribution function of the normal distribution

$$N\left(\frac{\overline{n_s}\cdot s}{|\mathcal{N}-\mathcal{T}|}, \frac{\overline{n_s}\cdot s}{|\mathcal{N}-\mathcal{T}|}(1-\frac{\overline{n_s}}{|\mathcal{N}-\mathcal{T}|})\right)$$

both having the same variance but different mean values and by subtracting the right side of Equation 4.7 and using the approximation replacements above we get

$$\begin{aligned} 0 = &\\ & s + (1-(1-\Phi_{\mathcal{T}}(d)))|\mathcal{T}| + 1.25(1-\Phi_{\mathcal{N}-\mathcal{T}}(d))(|\mathcal{N}|-|\mathcal{T}|) - 1.25(|\mathcal{N}|-|\mathcal{T}|) \\ &= s + \Phi_{\mathcal{T}}(d)|\mathcal{T}| + 1.25\left((1-\Phi_{\mathcal{N}-\mathcal{T}}(d))(|\mathcal{N}|-|\mathcal{T}|) - |\mathcal{N}| + |\mathcal{T}|\right) \quad (4.8) \end{aligned}$$

and we can solve the equation for given values $|\mathcal{N}|$, s and d to find approximated break-even points (*i. e.,* solutions for a variable value $|\mathcal{T}|$).

4.5.1.3 Parameter Result of the Combined Scheme

We summarize the combined scheme's parameters:

Corollary 4.5.5. *The combined scheme provides a broadcast encryption scheme with the following parameters (with $n := |\mathcal{N}|$ and $t := |\mathcal{T}|$).*

There are $\frac{1}{2}M$ keys assigned per user, BC needs to store a total number of M keys and the broadcast header consists of (at most) $s + \lceil (1 - SUC_{rat}) \cdot t \rceil + 2\lceil FR_{rat} \cdot (n-t) \rceil - 1$ messages (where SUC_{rat} and FR_{rat} are dependent on n, t, s and M).

The scheme is 1-resilient.

4.5.2 Improvements for Stateful Receivers

The notion of a *stateless receiver* is used to compare schemes that can be based on the same low-end receiver capabilities. We have shown in the preceding sections that the biased sub-set schemes do operate within this minimum scenario. Having said that we are now interested in the possible improvements that can be based on the capability to persistently store some state by the receiver.

If the receivers are not stateless, the agreed session key for a certain transmission can be learned and be re-used as a sub-set key for a future protocol run. In practice it is likely that the target set of one transmission is very *similar* to the target set of a related transmission, so if the key is used as a future sub-set key the bias towards the target set will often be much higher than that of a normal sorted sub-set. Hence, the scheme will become more efficient for future runs when the receivers have stored their transmission session keys for each time the receiver was in the privileged set of users.

4.6 Key Compression and Key Assignment Strategies

4.6.1 Smartcard Properties

The current state-of-the-art conditional access technology relies on smartcards and their tamper-resistant properties. The smartcards used in the CI modules (see section 3.9.2) are issued by the Pay-TV provider and their size is of credit card dimensions so the user is able to put the card into the CI slot of his STT without the need of any technical expertise.

The smartcards store the secret key information in their tamper-resistant EEPROM-type memory while the firmware (that contains parts of the implementation of the broadcast encryption scheme) is stored in ROM. While the advantage of smartcard technology is the tamper-resistant property that allows to deliver secret key information to the user and also to exchange keys and also algorithms by delivering new cards to the user base the disadvantage is the limited key space in the EEPROM. If a symmetric key is considered to be in the length range of 64 – 160 bits and EEPROM memory sizes of current smartcards are 128 or 256 KBytes then roughly five to ten thousand secret keys can be stored in a secure memory environment including their respective key identifiers.

Smartcards support persistent mutable memory, thus they can be considered as stateful receivers and the stateless schemes considered throughout this work do not exploit this feature. However, in the Pay-TV use case it cannot be assumed that each receiver is switched on and is able to receive the broadcast signal permanently since the user might only power it up for a dedicated transmission. Therefore it is a sensible approach to use the statefulness feature only for rare events (*e. g.,* activation of subscriptions, customer service activities) but not for normal operation. Hence, the usage of stateless schemes for broadcast encryption is a reasonable even if smartcard based conditional access technology is available.

4.6.2 Generating Sub-set Keys

The Biased Sub-set schemes use the parameter M as the number of chosen sub-sets each associated with a unique symmetric sub-set key. Instead of using the available key space for storing sub-set keys (*e. g.,* ten thousand sub-set keys per user smartcard suggests to set $M := 20000$), the keys could be used as a seed to generate the sub-set keys in a way that each sub-set key can be computed from a collection of pre-keys

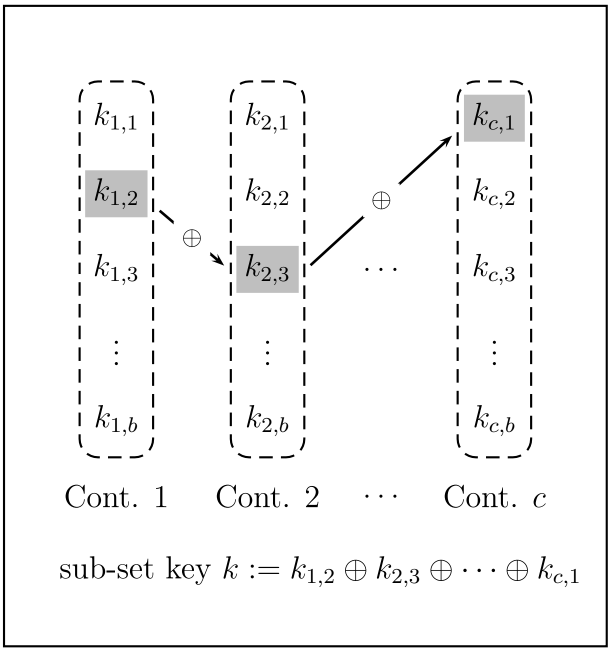

Figure 4.8: Key Containers with example sub-set key k

whereas only half of the users are assigned *all* of the necessary pre-keys needed to compute a sub-set key. The other half of the user base is missing at least on of the pre-keys so that the computed sub-set key does indeed match to a sub-set as required by the proposed scheme.

We take advantage of the following deliberation: A sub-set key can be computed by a number of b pre-keys; each pre-key is selected from a container of c pre-keys while there is a total number of b containers, and exactly one pre-key from each container is needed to compute a sub-set key. The computation shall be simple and be performed by the smartcard itself so we choose the XOR-operation for convenience; see Fig. 4.8 as illustration.

We will now calculate the number of pre-keys that each user needs to know in order to be able to compute half of the sub-set keys. First, we define the pre-key containers and map the users to the combined pre-keys.

Definition 4.6.1. Let $\mathcal{N}$ be a set of users of a broadcast system. A *key container* is a finite non-empty set of pre-keys which can be assigned to one or several users. Let $C_1, \dots, C_c$ be different key containers of b keys each.

A *b-c combination* denotes the function that maps a tuple $(k_1, k_2, \dots, k_c)$ where each k_i is an element of a container C_i to a sub-set $N' \subseteq \mathcal{N}$ in way that a user $u \in N'$ iff u is assigned all the k_i $(i = 1, \dots, c)$, *i. e.,*

$$(k_1, k_2, \dots, k_c) \mapsto \{u \in \mathcal{N} : u \text{ is assigned pre-key } k_i \ \ \forall i \in \{1, \dots, c\}\ \}$$

The key $k_{N'} := k_1 \oplus \cdots \oplus k_c$ is then the *combined sub-set key* of sub-set N'.

Lemma 4.6.2. *If, by following Def. 4.6.1, a user is assigned $b' < b$ pre-keys of each container of a b-c* combination *and if*

$$(\frac{b'}{b})^c = \frac{1}{2}$$

holds, then the user can compute exactly half of all the combined sub-set keys.

Proof. We count the number of combined sub-sets keys that result from the different tuples being combinations of c pre-keys so we have b^c combined sub-sets keys in total; each user can compute $(b')^c$ of the combined sub-set keys, so in total he can compute

$$(b')^c = \left(\frac{b'}{b}\right)^c (b^c) = \frac{1}{2} b^c$$

combined sub-set keys, *i. e.,* exactly half of all the combined sub-set keys. □

Example 4.6.3. Assuming that roughly ten thousand keys can be stored in a smartcard's EEPROM, we could use the following key configuration: 10 containers, each filled with 1024 pre-keys. Each user is (using Lemma 4.6.2) assigned randomly chosen

$$1024 \sqrt[10]{\frac{1}{2}} \approx 955$$

pre-keys from each container, then he is able to compute about half of the combined sub-set keys using the $\oplus$-operation with 10 items.

The sub-set identification can easily be done by concatenating $\log 1024 = 10$ bits per container, *i. e.,* 100 bits to denote one of the (up to) 2^{100} sub-sets in a broadcast header.

4.6.3 Choosing Generated Sub-set Keys Efficiently

In the last section the receiver's point of view (calculating the combined sub-set key from the pre-keys) is described. Next, we focus on the sender side; how can the correct sub-sets be identified that are most biased towards the target group. In the improved biased sub-set scheme (section 4.3) a key is chosen independently for every sub-set and each users stores half of the keys; the sender's task is then to sort the sub-sets in order to a find the suitable permutation π (introduced in section 4.2.3) for representing the sorted sub-sets.

While the task to sort a number of items that are in the order of magnitude of available user key space can be considered feasible for a sender (that can be assumed to have more computation power than a smartcard), the new task to sort all *possible* sub-sets defined by pre-key computation can easily become infeasible. Considering the numbers from Example 4.6.3, up to 2^{100} sub-sets are to be examined and sorted; this clearly exceeds the computation power of any sender, when the best known algorithm is to “visit” every sub-set individually.

While we may not be able find a the optimal bias of all sub-sets we can use a heuristic approach and apply a greedy algorithm. Greedy heuristics perform well in many situations, and have been well-studied (*e. g.,* [48]) in the algorithmic community.

To find suitable heuristics we first note, that the bias of a sub-set towards a target set is somehow dependent on the bias within a pre-key container because a user is in a sub-set iff she is assigned every pre-key (one per container) that by combination denote the respective sub-set. Thus, there is some evidence for a strategy to select those pre-keys of a container that are assigned to more member of the target set than the other pre-keys of that container. Secondly, we observe a similarity to the problem of optimizing a product of positive numbers as the quota of privileged users in a sub-set that is denoted by a combination of randomly chosen pre-keys can be regarded as the

Algorithm 1 Greedy set-up algorithm for choosing pre-keys

```
Require: $c > 0$, $b > 0$ {number of containers, pre-keys per container}
Require: $\mathcal{T} \subseteq \mathcal{N}$ {privileged user set}
  $max \leftarrow 0$
  $y \leftarrow 0$
  for $i = 1$ to $c$ do
    $max_i \leftarrow (-1)$
    for $j = 1$ to $b$ do
      if number of users in $\mathcal{T}$ knowing $k_{i,j}$ is greater than $max_i$ then
        set $max_i$ to this number
        $x_i \leftarrow j$ {stores the maximum position per container}
      end if
    end for
    if $max_i > max$ then
      $max \leftarrow max_i$
      $y \leftarrow i$ {stores the maximum container index}
    end if
  end for
```

product of quotas per container. So, after greedily identifying the first sub-set it is reasonable to select the greatest factor and replace it by the next second best factor. For example consider the product $4 \cdot 5 \cdot 3$ and the task is to repeatedly choose a factor and decrement it but maximize the product at the same time, then a reasonable choice would be: $4 \cdot 5 \cdot 3$ — $4 \cdot 4 \cdot 3$ — $4 \cdot 3 \cdot 3$ — $3 \cdot 3 \cdot 3$ etc. because a product is maximal when the factors are equal (assuming the sum of the factors is the limit).

Both of these heuristics are reflected in the following algorithms. See Algorithm 1 for the set-up phase of the greedy approach. The algorithm processes each pre-key in each container and determines the "maximum pre-key" of each container, *i. e.,* the pre-key that is assigned to the maximum number of privileged users. When several maximum values are encountered the first one is stored. In the same way the "maximum container" is selected by value y.

The Algorithm 2 describes the actual selection of sub-sets implicitly by choosing the combinations of pre-keys. The first tuple consisting of the all the maximum keys is outputted immediately as it is known from the set-up part (Alg. 1); then the second best pre-key of the maximum container becomes its new maximum pre-key while the other key is marked so it cannot be used again. It is possible (but considered to be unlikely) that a container is used up and there is no remaining pre-key. Even though this situation could be handled by a more sophisticated algorithm we choose the emergency stop as the heuristic approach has failed in this case.

Algorithm 2 Greedy send algorithm for choosing pre-keys

```
Require: $c$, $b$, $\mathcal{T}$ and values computed by set-up Algorithm 1
  while not enough sub-set tuples are generated do
    output sub-set tuple $(x_1, x_2, \ldots, x_c)$
    mark $k_{y,x_y}$ {pre-key not to be used again}
    $max_y \leftarrow (-1)$
    for $j = 1$ to $b$ do
      if number of users in $\mathcal{T}$ knowing $k_{y,j}$ is greater than $max_i$ AND $k_{y,j}$ is unmarked
      then
        set $max_y$ to this number of users
        $x_y \leftarrow j$ {stores the new maximum position in container $y$}
      end if
    end for
    if $max_y = (-1)$ then
      STOP {Emergency Stop. Considered unlikely: all pre-keys used up in container
      $y$}
    end if
    $max \leftarrow 0$
    for $i = 1$ to $c$ do
      if $max_i > max$ then
        $max \leftarrow max_i$
        $y \leftarrow i$ {stores the new maximum container index}
      end if
    end for
  end while
```

5 Global Multimedia Broadcasts with Regional DRM enforcement

5.1 Pay-TV / Pay-per-view Status Quo

These days, a Pay-TV provider is serving customers in a dedicated region (*e. g.,* a country). Its program offering (*e. g.,* language, interest) is tailored for the potential customers in this region [2, 6, 50]. However, the provider often sends the secured transmission to a super-set of the user base since transmission signals go beyond geographical borders.

The commercial scenario we consider in this chapter is that of content broadcast transmission in such a way that the transmissions are secured (encrypted, scrambled), so that only authorized users (*e. g.,* subscribers, pay-per-view customers) can receive (decrypt, descramble) the transmission. An important example is digital Pay-TV, where a digital signal is broadcasted via satellite, cable or a terrestrial radio connection (*e. g.,* DVB-T). But there are also other applications for pay-per-view services such as Internet multicasts for audio, video or data transmissions similar to the Pay-TV situation.

Considering satellite broadcasting, a transmission intended for one country could be received (but not decrypted) by users in many countries. Transmission costs per user would be lower if these users outside the region could participate in the Pay-TV scheme because the sender bears the cost for using a satellite data channel. The cost of a transmission is independent from the number of users receiving the transmission.

While the digital signal is broadcasted across borders, current Pay-TV systems are serving only a sub-region of the signal coverage area. We want to introduce an improved system that covers the whole area of a sender's footprint or extend a limited area to all STTs connected to the Internet with no further geographical limits but still enforce regional DRM policies on the delivered content.

First note, that digital multimedia content is already suited for cross-border delivery as current standards as MPEG2 [63] do support multiple audio streams and subtitle texts per video stream so that each user can choose the desired language of a transmission as long as it is part of the content.

5.2 The Global Content Provider

A content provider that is able to send the transmission through several networks (*e. g.,* more than one satellite or cable network) and cover most parts of the world can reduce transmissions costs per customer. When he is legally entitled by the rights-owners to act as a global service provider and enforces the digital rights in each region it becomes a super-regional or *global* provider transmitting global (*e. g.,* Pay-TV) transmissions.

Such a global multimedia transmission could be produced with several audio and subtitle text channels so that each user in some part of the world could select the correct channels for his needs and is able to consume the global transmission that is sent to many parts of the world. Global transmissions are technically possible (*e. g.,* through satellite networks) but global broadcasting to end-users has not emerged yet.

In order to send transmission to super-regions efficiently the STT technology is required to be identical in most of the regions. Currently, there are only few standards for digital television broadcasts that cover major parts of the world, America: ATSC standard, Japan: ISDB-T (similar to DVB-T), Europe, parts of Asia and Australia: DVB-S/T/C. The DVB standard family could easily grow to a globally used standard if global Pay-TV services became a reality. Apart from that, standard STTs and broadcast technologies are becoming already a reality [62, 57].

When the Internet is used for selling the pay-per-view licenses the global provider becomes an e-commerce service. An STT, which is temporarily connected to the Internet and being able to receive constantly broadband digital data via non-Internet sources (*e. g.,* satellite reception), would be the user's equipment in such an e-commerce scenario.

Currently, the digital rights holders sell the content on different terms to each regional Pay-TV provider and they enforce DRM differently in the regions. The global system portrayed in this chapter shall meet the rights holders' requirements regarding content protection and regional licensing politics.

In this dissertation we will not predict whether the digital rights holders may welcome such a global Pay-TV provider but we assume that they will most probably not accept a new system with the content being distributed in a way that infringes rules on regional licensing and eases piracy.

5.2.1 Technology and Business Parameters

A content broadcast system that covers multiple regions needs to be designed in a way that no feasible hardware or software manipulation to the STT does help an attacker to construct a super-terminal showing all transmissions from all regions *for free.* It is unlikely that the rights-owners would accept a global Pay-TV system that cannot assure this hard minimum requirement.

Unfortunately to the e-commerce approach, a global multimedia broadcast system cannot use the Internet for transmitting the (secured) content for performance reasons today. Also in the near future it is unlikely that a home Internet connection is so stable and swift that a high-quality multimedia transmission, which requires a stable bit-rate for several hours, is transmitted without deferments or connection disruptions (a Pay-TV transmission is easily exceeding 4GB of data). The data size of multimedia transmissions is growing as well as the bandwidth of home Internet connections but it is unforeseeable when the gap might be closed. Note, that there is a multicast future for IP-based communication [55] but IP multicast is not necessarily the right choice for distributing Pay-TV via Internet [85] and Internet-based TV can generally not provide satisfactory quality of the transmitted pictures [73].

However, while the Internet can be used to perform the Pay-Per-View sales transaction delivering a license to the user equipment, the same user equipment could link to other high-bandwidth broadcast data sources (*e. g.,* satellite, cable, digital radio broadcast) in order to receive the multimedia transmission. A global Pay-TV server could use several satellites and cable networks simultaneously to transmit to a user base distributed across different regions.

5.2.2 Regional Pricing

The rights-owners may require that for every region a different pricing and subscription model for transmissions can be set and also that some regions shall be blacked out for certain transmission but might participate in a later re-transmission when other regions

	Region 1	Region 2	Region 3
Transmission 1 (**date:** Jan 15) **audio:** E, F, DE **subtitle:** E, F	per-view 5$ audio: all subtitle: all	per-view 3$ audio: DE subtitle: E1, DE	black out
Transmission 2 (**date:** Feb 12) **audio:** E, F, DE, IT, CH, JP **subtitle:** E1, E2, F, DE1, DE2, ...	per-view 3$ audio: all subtitle: all	black out	per-view 5$ audio: E, JP subtitle: all
Transmission 3 (**date:** Jun 19) **audio:** E, F, DE **subtitle:** E1, E2	free for subscribers audio: all subtitle: all	per-view 1$ audio: DE subtitle: E1, DE	free for subscribers audio: E, JP subtitle: all

Table 5.1: Several transmissions of same content, example

are excluded. The different prices per region and respective black-out-rules are highly apparent for the transmission of popular events (*e. g.,* sport events [19]).

We illustrate these different regional distribution or business models with Tab. 5.1. The rights holder of a certain multimedia content (*e. g.,* a movie production) sells to regional Pay-TV providers. The pricing, release dates and supported languages (audio and subtitles) depend on the region. Each regional provider is able to diversify the digital content to meet the requirements and transmit the tailored content to his customers. However, if a global provider emerges that wants to serve all regions with *one* transmission covering all regional needs (and reducing transmissions costs) it has to ensure that the customers can only receive (*i. e.,* descramble) the content they are entitled to. For instance, a customer living in *Region 1* of Fig. 5.1 shall not be able to buy the Pay-Per-View package for *Transmission 1* in *Region 2* and receive it at home for a lower price or with more language options than entitled. Loosely speaking, the equipment shall know where it is located and change its behavior when moved to another region and follow local rules, regardless where it was bought.

The rights-owners will probably not accept a global system if there is a risk of pirate decoders that could circumvent the whole system and become super-terminals showing all content for free.

Currently, most Pay-TV providers focus on a country and the offering is sold in this country only [74]. A user could export a STT to another country, though, and use a subscription there if signal reception is technically possible. However, the transmission for one country do most likely not contain multiple audio tracks or subtitle information so that users in another country speaking a different mother-tongue will in general be less interested in the offering. This provides implicit DRM enforcement: a user outside a region cannot consume the content that is dedicated for the region because this content is simply unavailable outside.

From a rights-holder's point of view, a global Pay-TV provider could reduce its costs per transmission compared to a national service provider and Pay-TV as a sales channel for multimedia productions could return higher profits as costs are are cut back. A national Pay-TV provider being forced to go global might not favor globalization of his services and see the need to seek an international alliance or merger as supposably only a few global players will be able to compete on a global market [72].

5.2.3 Related Ideas

In the past there was some development to standardize multimedia data formats so that hardware media could be produced and used world-wide while the rights-holders' ideas of regional sales strategies are respected: The Digital Versatile Disc region codes (see section 3.7.1) are an example for this strategy. This technology cannot prevent that a player is exported to another region, thus it is not suitable for technically enforcing different media prices in the regions.

Another development are the Internet video initiatives: *CinemaNow* [22] was founded in 1999 by several companies and headed by Microsoft. Several thousand movies are offered on a pay-per-view basis. The system does not incorporate the full offering of a Pay-TV provider (*e. g.,* real time transmission of events) and the multimedia content does not meet the established digital television standards (this reduces the media file sizes though). *MovieLink* [80] was launched in 2002 by major digital rights holders (*i. e.,* Hollywood movie studios). Several hundred movies are offered but only US customers can participate in the service so the regional DRM issue was solved in a rather simple way that only one region is covered. Potential customers accessing the web-site from outside the USA are informed that their IP address indicate that they are outside the United States and thus cannot use the service. Movielink does not disclose information how accurate the localization based on IP addresses has proved to be in the last years since service set-up. Both initiatives might give the industries helpful experiences for future Internet based movie distribution channels but they are not intended to take over the role of Pay-TV stations.

5.3 License: Background and Definitions

Let us first fix what a license in our regional DRM context shall be. A user is entitled to certain content consumable under certain conditions that depend on the region once she bought a subscription and / or a Pay-Per-View product. This consumption right shall be her license and be expressed as a machine-readable license ticket shown in Fig. 5.1.

The license ticket is usually sent via the broadcast channel individually to each user and processed by the user equipment or an essential part thereof (*e. g.,* a smartcard). It carries the User ID so the STT or the smartcard knows whether it shall process the ticket, a Broadcast ID to map it to a transmission and a Region ID to specify the region in its regional licensing model. The restrictions are not necessarily a list of rules but could be a set of one-time keys which allow to decrypt certain parts of a transmission. This broad definition of restrictions allows us to be compatible with established technology standards (see implementation issues in Chapter 6). The Broadcast ID refers to a certain transmission and will change when a transmission is repeated. The time of the transmission is thus indirectly encoded with this ID. The ticket may be authenticated[1] by some cryptographic mechanism (*e. g.,* a MAC function) and may carry a unique number for differentiation.

Analogously to the license ticket issued for a single user (identified by a User ID) we also want the broadcast center to be able to issue licenses to a group of users, *i. e.,* a to a certain sub-set of the user base. This is depicted in Fig. 5.2 where a Sub−set ID is included in the ticket data structure to indentify the respective sub-set.

[1] Authentication is an option here since a ticket could be transmitted via a secure channel and the keys being necessary to decrypt a certain content can be part of the *restrictions* thus no further authentication is needed.

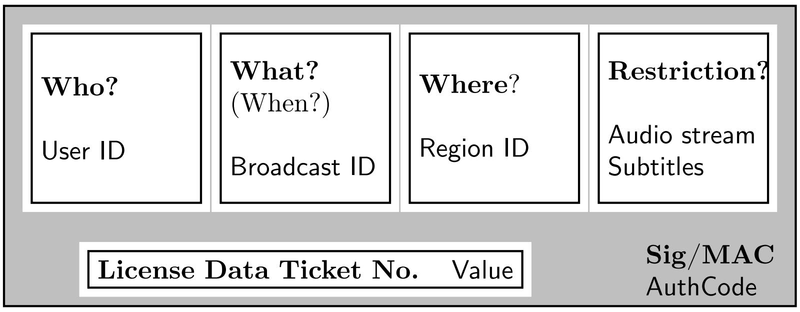

Figure 5.1: License Ticket

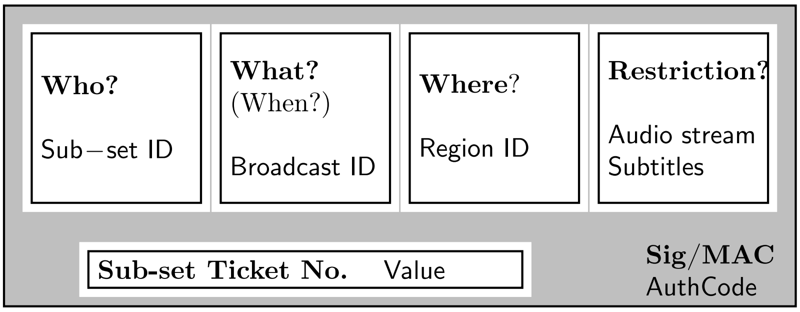

Figure 5.2: Group License Ticket

We formalize these considerations that are pictured in Fig. 5.1 and Fig. 5.2 in the following definition.

Definition 5.3.1. (license, license ticket, group license ticket) For a given broadcast system consisting of a sender BC, set $\mathcal{N}$ of users and a certain content to be transmitted by BC, we define a *license* as the right of a specified individual user to consume the content transmission under certain conditions (region, time, available languages).

A license can be expressed by a *license ticket* that is the concatenation (expressed by a mathematical tuple) of the following bit-strings

$$(\mathsf{User\ ID}, \mathsf{Broadcast\ ID}, \mathsf{Region\ ID}, \mathsf{AudioConf}, \mathsf{TicketNo.}, \mathsf{AuthCode}) \quad .$$

Moreover, a license being issued for a sub-set of the user base can be expressed by a *group license ticket* that is the concatenation of the following bit-strings

$$(\mathsf{Sub-set\ ID}, \mathsf{Broadcast\ ID}, \mathsf{Region\ ID}, \mathsf{AudioConf}, \mathsf{TicketNo.}, \mathsf{AuthCode}) \quad .$$

The bit-string User ID shall identify a user $u \in \mathcal{N}$; Broadcast ID shall identify a certain broadcast message header (and implicitly the time when this broadcast is transmitted); Region ID shall identify an element of a non-empty finite set $\mathcal{L}$ of regions. The bit-strings AudioConf, TicketNo. and AuthCode are place-holders in the ticket data structue for usage[2] by (technical implementations) of rights management or broadcast encryption schemes with no formal semantics within this definition.

Remark 5.3.2. The AudioConf-field could be generalized to express more complex detailed usage rights (not only related to the audio streams as the field name indicates) but we want to keep the definition brief so that a self-contained DRM realization with the support of existing CAS technologies is achievable for a pay-TV scenario.

After having established the location-aware licenses we can now formalize a location-based broadcast encryption system that extendes the Definition 3.2.2 of a broadcast encryption system in the way that the the set of privileged users $\mathcal{T}$ is determined by the set of users holding a certain license and being located in a matching region.

Definition 5.3.3. (location-dependent broadcast encryption system) Let $\mathcal{K}$ be a set of keys, $\mathcal{B}$ a set of broadcast messages and $\mathcal{L}$ be a set of Region ID-values each identifying a certain geographical region. A *location-dependent broadcast encryption system* is a broadcast system with an associated triple of algorithms (Setup, Broadcast, Decrypt) such that

[2] The TicketNo. string will be used to assemble tickets that are sent in parts via different radio channels while the AudioConf string will contain decryption keys of CAS based schemes.

- The setup algorithm (Setup) takes a user $u \in \mathcal{N}$ and computes the users's private key information $P_u \subseteq \mathcal{K}$.

- For a given Group License Ticket GT the set $\mathcal{T}_{GT} \subseteq \mathcal{N}$ denotes the sub-set of users that is identified by the Sub−set ID data field of GT.

- The set $\mathcal{T}_{Loc} \subseteq \mathcal{N}$ denotes the sub-set of users that is located in the region identified by the Region ID data field of GT during the broadcast phase.

- The broadcast algorithm (Broadcast) takes the set of privileged users $\mathcal{T} := \mathcal{T}_{Loc} \cap \mathcal{T}_{GT}$ and a session key $K \in \mathcal{K}$ and outputs a broadcast message $B_{r_{\mathsf{ID}}} \in \mathcal{B}$ for every region identified by r_{ID}.

- Any user $u \in \mathcal{N}$ can run the decryption algorithm $\text{Decrypt}(B; P_u; u; r_{\mathsf{ID}})$ that will output K if $u \in \mathcal{T}$ and if u is in the region identified by r_{ID} but fail if at least one of the conditions is not met.

Remark 5.3.4. The Definition 5.3.3 does include overlapping regions since no formal requirements were introduced that prevent a user from being located in several regions at once. The definition caters for two different "strategies" for regional license enforcement: the output of the broadcast algorithm Broadcast can be regionally diversified to enable location-dependent broadcasting (the goal would be, though, that the output is identical for most or all reasons) and the decryption algorithm Decrypt may explicitly process the region information r_{ID} in order to enable internal rights-enforcing decisions (*e. g.,* stop the decryption process if the user is located outside a valid region).

5.4 Requirements for a Global Multimedia Pay-Per-View System

In order to reflect the technological and business parameters laid out in the above sections we fix a set of requirements for a new global system for the transmissions of multimedia content that follows security and business considerations.[3]

- R1: The system shall provide a global offering to users who are able to receive high-bandwidth broadcast data.
- R2: There shall be no theoretical possibility to construct a system pirate device displaying all content of all regions without a license. Moreover, a pirate device shall never display content dedicated for a region it is not located in.
- R3: The multimedia coding technology and the STT architecture shall adopt existing standards. There shall be no need to roll out a new broadcasting and set-top technology - *e. g.*, latest-generation STTs already shipped to the user base shall be ready to participate in a new system.
- R4: The STTs may be exported but shall always enforce DRM policies in the region they are located in.
- R5: The STT technology shall adopt the same tamper-proof and system security requirements as existing Pay-TV systems. Established conditional access solutions shall be used.

[3] The process of fixing this list of requirements involved extensive discussions I had with participants of the conferences IEEE-CEC 2005 (Munich) and DRMtics 2005 (Sydney). I am deeply grateful to these colleagues because only when requirements reflect real-world problems, the corresponding solutions might have substantial significance.

- `R6`: An existing state-of-the-art Pay-TV provider shall be able to extend the regional coverage to a global coverage while the legacy user base can still be served with the same technology.

- `R7`: If the system security is compromised in one region the system security in other regions shall not be affected necessarily.

The goal is to construct a scheme meeting all of these requirements.

5.5 A Multimedia Pay-per-View System Approaching E-Commerce

The new system architecture that we want to sketch in this section shall fulfill the security and DRM requirements we have identified in the preceding section. It shall not be less secure or less efficient than current productive Pay-TV systems that operate in a dedicated region (*e. g.,* country), *i. e.,* the aggregation of several regional systems to one new global system shall not make it easier for an attacker to jeopardize the system security in one region only because the system is compromised in another region.

5.5.1 Architecture Goals and Definitions

Our proposed system (detailed in section 5.5.2) provides multimedia transmissions that are secured by encryption (as it is the case for current Pay-TV systems). Every user $u \in U$ owns a unique user key k_u that is stored on the smartcard inside the device. All content is encrypted at source and decrypted at the STT. To decrypt a transmission T one or more session keys $k_1^S \ldots k_n^S$ are needed that are derived by the user's STT by using the individual user key and information sent to the STT in a control message individually encrypted for each user who has obtained a license.

The session key(s) are valid for one transmission only and used to decrypt the video content as well as the audio content a user is entitled to according to his subscription contract and / or pay-per-view payment. We will call this collection of user rights the user's license (following Definition 5.3.1). Note, that a user may only be entitled to some sub-set of the transmitted video or audio content; this sub-set is dependent not only on the possession of a license ticket but also on the location of the user. In our proposed system architecture, the STT can only derive the session keys that make it possible to decrypt exactly the specified content parts.

Concept idea: If a user moves the STT to a different area before or after obtaining a license the STT is not able to decrypt more content than the user is entitled to in this different area. The system enforces the DRM policy with technical measures – we will not rely on a legal environment that prohibits the misappropriation of content by assigning penalties for wrong behavior. Note, that such a uniform legal requirement could not be easily established across borders.

5.5.2 A New System on Top of Existing Standards

We use conventional conditional access technology for securing the multimedia transmissions. The content is coded in MPEG [63] format and encrypted by the DVB encryption standard.

5.5.2.1 Common Interface

The user STTs is equipped with a *Common Interface* [20], an established standard used in digital video broadcasting. This Common Interface is normally connected with a *CI* *module* that is incorporating a smartcard reader where the user will put in a smartcard issued by the Pay-TV provider. Most currently available set-top boxes and DVB-cards for Personal Computers provide at least one or more than one Common Interface slot. Different CI modules facilitate different cryptographic protocols and algorithms used by the Pay-TV service providers and implemented on a CI module. All STTs are applying the same decryption algorithm (the *Common Scrambling Algorithm*), during a secured transmission the STTs continuously receive *control words* via the Common Interface that are needed to descramble the secured content. These Control Words are short-lived session keys only used for a part of one transmission. Our proposed solution will operate on top of this existing technology so today's standard STTs equipped with the common interface could be used in the new solution and the same level of cryptographic security is achieved.

5.5.2.2 DRM Device

The proposed solution will contain a new component: the *DRM device*. This device takes over the role of today's CI module and but still sends the Control Words to the STT and carries the provider's smartcard if a tamper proof key container is required.

The new functionality introduced here is that this device will also receive location dependent information so that the STT needs to be located in the correct region when a license obtained by the user is only valid for a named region. The idea is to utilize a different radio network that can send small amounts of information individually encrypted for a user to a rather small region so that this information would be missing in other regions where the broadcast signal is still available.

In order to use well established technology this location dependent information could be transmitted via the GSM [54] mobile phone network[4] using the service *cell broadcast* [51] which is a GSM network feature allowing messages of up to 82 bytes to be broadcasted to all mobile phones within a geographical area. This radio interface does require the DRM device to incorporate a basic non-voice GSM terminal card so that these cell broadcast messages can be received. The Pay-TV smartcard can be used to store the GSM network authentication data (IMSI, K_i) so that no further hardware is needed (see Fig. 5.3). In the case where a GSM mobile phone is a STT by itself, no extra GSM terminal card is needed. Regions without GSM coverage can also participate if another local radio network with cell addressability is available (*e. g.,* legacy analog mobile phone networks, pager networks – or newer digital networks: DVB-T or UMTS).

5.5.2.3 Local Key Information

The sender broadcasts some of the information that is necessary to derive the Control Words individually encrypted to each user addressing only the GSM cell area the STT is located in. By evaluating the location information given by the user while purchasing his license or obtained through outgoing GSM communication by the device. The exact way to obtain location information about the STT can be chosen according to user privacy regulations. The GSM cell is a rather small area with a diameter of less than 20 kilometers so it sets a sufficient hard limit regarding the mobility of the STT

[4] The GSM cellular network is subdivided into location areas, each consisting of several cells. A cell has a radius between 300 meters and 35 kilometers. On the network level, location information is the address of a particular location area. In oder to determine the exact location of a mobile phone, the entire location area must be paged. Each location area has a Visitor Location Register (VLR) that contains copies of the profiles of all mobile subscribers currently registered in the location area. Normally, the VLR is co-located with a Mobile services Switching Center (MSC), and both MSC and VLR cover the same location area. Moreover, each GSM network has a logically centralized subscription database, the Home Location Register (HLR). The HLR stores the network address of the most recently used VLR for each subscriber [34].

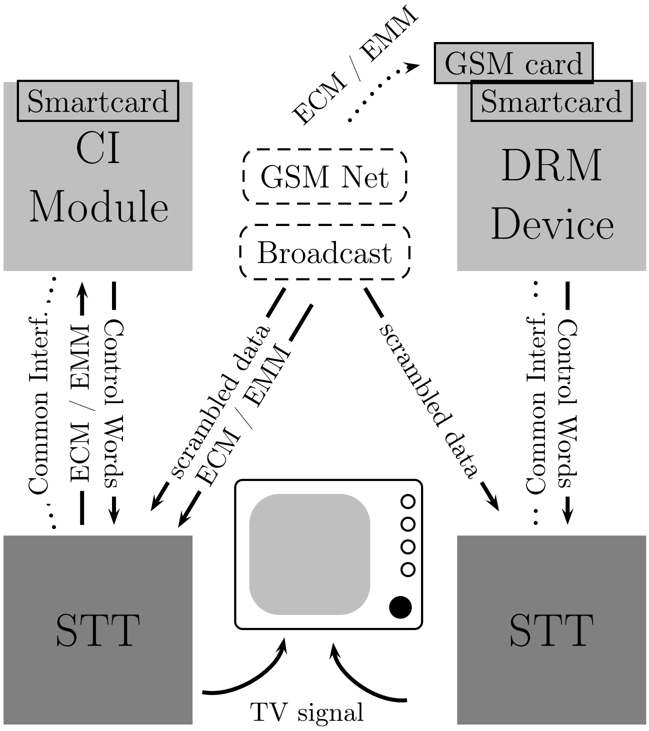

Figure 5.3: Standard Conditional Access Scheme — Proposed New Scheme

that shall not be moved outside a much greater area (*e. g.,* a country).

The key feature here is that this location dependent information cannot be received when the DRM device (connected to the STT) is moved outside the region. An attacker might be able to simulate localization information (*e. g.,* manipulate the unsigned GPS[5] data signal fed to an STT if GPS technology was used instead) but he will not be able to receive information that is not broadcasted at the STT location GSM radio cell.

Before a transmission starts the sender sends *Entitlement Control Messages (ECMs) and Entitlement Management Messages (EMMs)* to the STT as it is the case in current

[5] Satellite navigation is currently used widely through the *Global Positioning System* (GPS). GPS is a global, satellite based radio navigation system providing three-dimensional position, velocity, and highly-accurate time information to GPS receivers anywhere on or near the surface of the earth. The GPS consists of 24 satellites in six orbital planes with a 12-hour period being positioned in a way that between five and eight of the satellites are visible from any position on or above the earth surface [25, 102].

Pay-TV technology (for details refer to [20]). These messages are required by the CI Module to activate certain subscriptions and to generate the Control Words during the transmission. The difference is that in our proposed system the ECMs / EMMs are not sent via the broadband channel but via the GSM cell broadcast to the specific user location. Hence, the user can only exercise a license if the STT location matches the region attribute of the license so a rather inexpensive license obtained by pretending to be in a different region is useless as the DRM device will not receive the ECM / EMM messages and cannot generate the Control Words necessary to descramble the content. Because this location property is not achieved through cryptographic measures but through location dependent information the mechanism cannot be attacked with cryptanalytic strategies or by breaking tamper-resistant devices (*e. g.,* smartcards).

Note, that the business model of the digital rights holder might require that the content is received as free-TV in some regions, sent to a set of flat-fee subscribers in another region while in yet another region only Pay-Per-View is offered. All these different regional DRM policies can be honored by our proposed system although only one secured transmission is broadcasted globally.

5.5.3 Properties and Security Attributes of the Proposed System

The proposed system can assure the same security standard as existing regional Pay-TV systems if the sole technology change in this region is that Pay-TV CI modules are replaced by the proposed DRM devices and if the tamper resistance properties of the modules and the devices is comparable. The feasibility of an implementation depends on the global adoption of conditional access technology and DVB standards.

5.5.3.1 Relocating Control Words

An attacker might utilize a functional STT together with a license in one region for the purpose to intercept the Control Words on the slot interface and use the intercepted Control Words to run a STT in another region where the license is not valid. If this type of attack is feasible (the Control Words need to be transmitted real-time to another region if the transmission there is to be descrambled in real-time as well) then it could be applied already today for regional Pay-TV systems where the data broadcast is covering a super-region (*e. g.,* satellite Pay-TV, cable networks). A possible countermeasure for this type of attack is to enforce a mutual authentication of the CI module and the STT. As the underlying Pay-TV standardized technology is the vulnerability in this case our proposed system is not more secure than the content scrambling standard adopted by it.

5.5.3.2 Failures of the 2nd network

We have to take into account that the GSM networks or other 2nd radio networks are used as trusted parties in our proposal. A manipulation of a network that makes it possible to re-route a GSM cell broadcast to a different region (in a different country) would threaten the system security in the same way as the *Relocating Control Words* attack (section 5.5.3.1).

This kind of attack on a network level is unlikely performed by a single Pay-TV pirate user and the relocating attack appears to be less expensive so that this attack scenario is of minor importance.

5.6 Enforcing Regional DRM

In this section we compare the proposed architecture for regional DRM enforcement on top of existing CAS standards (as laid out in section 5.5) to other possible technical and organizational solutions. The DRM enforcement could be based on organizational measures only – or be combined with technical measures described in the next sections.

5.6.1 Organizational Measure: Regional Smartcards

If the distribution of smartcards (as part of the Conditional Access system) could be linked to the regions defined by the DRM policy (and each card stays in a region), it is not a difficult task to enforce the license restrictions. The sender would send the management messages containing the descrambling information to the set of smartcards distributed to a region. Only these cards are then able to descramble the content according to the regional licensing model. This regional licensing approach is not new and was in particular set into practice for DVD media in the nineties (by applying *regional codes*).

The attacker's task in such a scenario would then be to move the smartcard from one region to another (*e. g.,* from an *inexpensive* region to an *expensive one*) or to alter the card distribution process. The sender (in the role of the smartcard issuer) could prohibit that cards are moved across borders but this would probably not stop all potential attackers. Moreover, such a regulation might not even be enforceable by law as a violation of the subscription contract is not necessarily a breach of law in every region [74]. In this case the sender is unable to prosecute traced pirates. Such a situation renders regulations useless.

5.6.2 Technical Measures: Tamper Resistance / Trusted Computing

In order to technically prevent unauthorized movement of the user's STT (including the CI module and the smartcard), either the network or at least one component of the user's equipment has to validate the location when no other technical measures are in place. If the STT is forced to initiate a communication to the sender before a transmission there are several proposed ways to locate the equipment by letting the STT initiate a communication to the broadcast sender [37], but as we aim to operate on top of the established DVB and Conditional Access standards, we cannot expect the STT to have such a convenient call-out feature. The communication is one-way only, hence the sender does not know where the receiver is located.

As the STT itself is standard off-the-shelf hardware being sold across regions with no localization features, either the CI module or the smartcard may be augmented with extra-functionality to validate the user location.

5.6.2.1 Positioning Systems

The CI module delivered by the Pay-TV provider could incorporate a positioning module like a GPS [25, 102] or Galileo [29, 86] signal receiver. If the positioning unit operated in a secure (*i. e.,* tamper resistant) environment it would be able to securely validate the location and check the license against geographical co-ordinates. If a confidential channel to the smartcard is established, the descrambling of content will only be initiated if the correct location is determined. Note that the positioning signals are not protected, thus fake signals are still a threat (detailed in section 5.6.4).

The positioning module would unlikely be incorporated in a smartcard as the restriction on size and computational power are much higher compared to the CI module.

However, this option should be taken into consideration for completeness. Since our findings regarding the CI module augmented with a localization module are also valid for potential smartcards with localization features, so we do not elaborate on an augmented smartcard approach further.

If a PC that is acting as a set top terminal (STT) can be assumed to be in a trusted state, the positioning computations could be performed by an augmented player software running on the PC. This can in general be achieved with TCG technology [98]. This way, a standard CI module could be used while the localization device is connected to the PC (incorporating the trusted platform) itself. Implementation details on this option are detailed in section 5.6.2.

The tamper-resistant or trusted device in this scenarios needs to validate the license ticket by checking the co-ordinates of the STT against the Region ID in the ticket. If the user (her equipment) is located in the specified region, then the descrambling process is initiated. This process is specified by the generic Algorithm 3.

Algorithm 3 Ticket processing and localization

```
repeat
   read ticket
until ticket on User ID is received
get STT co-ordinates
if Region ID matches co-ordinates then
   return keys from ticket restriction field
else
   return license violation error
end if
```

5.6.3 Technical Measures: 2nd Radio Network

For this scenario we re-use the idea laid out in section 5.5.2 where an additional radio network for Pay-TV localization purposes is used. This 2nd radio network with low data throughput performance can send small amounts of key information to a *radio cell*, which is a rather tiny area (compared to the rather big regions), so that this information would be missing in other regions where only the broadcast signal is available. In order to use established technology, the individual information could be transmitted via the GSM [54] mobile phone network using the service *cell broadcast* [51]. This radio interface does require the CI module device to incorporate a basic non-voice GSM terminal card so that these cell broadcast messages can be received[6]. Regions without GSM coverage can also participate if another local radio network with cell addressability is available (*e. g.,* pager networks, analog mobile phone networks or DVB-T network entities). This enhanced CI module (the *DRM device* from section 5.5.2) replaces the ordinary CI module.

The localization is implicitly performed via the 2nd radio network and there is no need to use clients of positioning systems. In order to apply the second radio network scheme to our ticket based licenses approach each ticket is split up into two parts. The restrictions field containing the cryptographic keys are removed from the ticket that is sent via the broadcast channel. This part, which is represented by a rather small amount of data, is sent via the 2nd radio network to the user location's radio cell if the cell co-ordinates match the Region ID. Note, that Algorithm 4 does not contain any conditional statements.

[6] An alternative approach to use the GSM network is to let the network transmit the cell ID of a terminal card that has booked into the network to the broadcast center. The center can then revoke users who are not located in a specified region. Since the mobile terminal's localization by cell IDs has severe implications on user privacy (the broadcaster knows all the active subscribers' locations in such a scenario) which might violate national legislation in some regions, we do not examine this concept further.

Algorithm 4 Ticket assembling using 2nd radio network

```
repeat
    read ticket.part(User ID, Broadcast ID, Region ID, TicketNo.) (broadcast channel)
until matching User ID is received
read ticket.part(AudioConf, TicketNo.) (2nd radio network)
assemble ticket from both parts with same TicketNo.
return keys from ticket restriction field AudioConf
```

5.6.4 Security Attributes of Measures

Organizational measures: There is no obvious way to compare the strength of the organizational measures named in the preceding section to technical measures, but history shows that selling devices in a certain region and banning export to other regions is not a method to stop users from doing so. The Digital Versatile Disc (DVD) region codes (see section 3.7.1) are an example for this measure. The mechanism is rather conceived as an inconvenience than a measure to hinder individuals to seek access to foreign content [18].

Small-size devices like smartcards can easily be transported or shipped by mail and it does not require any expertise to remove the smartcard from the card reader and send it to somebody else in another region. Moreover, different national laws might not legally support the system supplier's export regulation and render it useless.

Localization with Tamper Resistant / TC Devices: While the tamper resistance property of the CI module (or a part of it) can be a significant line of defense for an attacker, the positioning radio signals are received outside the environment before the secure computation is initiated. A straightforward attack scenario would be to remove the antenna and record the signals at another location in order to replay it. The same result could be reached by generating fake signals and feed them to the positioning unit directly. Regarding the current de-facto standard positioning system (GPS), the latter task is feasible as the positioning satellites' signals are not cryptographically protected at all and fake signals can easily be generated.

A direct attack on the tamper resistant hardware by a time-consuming and laborious disaggregation the hardware in order to read secret keys is generally regarded to be too costly for an average attacker. But an attack on the hardware still needs to be considered here because the single pivotal machine command executed by the secure hardware is the **if**-statement of Algorithm 3. The attacker only needs to provoke a faulty system state at this computation step in order to successfully circumvent the trusted hardware's security feature; upon success, he does not need to read any secret keys in the device. This kind of indirect attack on tamper-resistant hardware could be rather inexpensive [5], and special protection concepts against these attacks have to be considered [66].

Using Trusted Computing for a PC acting as STT provides mechanisms for a remote attestation of a secure state during the license ticket purchasing process and enables the possibility for a sealing of the content in a way that only after successful localization of the PC the content can be consumed (implementation details on this solution are provided in section 6.1). A trusted platfotm (in particular, the player software running on it) can then process positioning signals and determine the current position and decide whether a certain content may be consumed or not (using Algorithm 3 described in section 5.6.2). However, the positioning signals are still received outside the PC and the same security weaknesses regarding fake signals do apply.

Apart from the security considerations another facet on positioning should be regarded: the standard positioning system GPS has limited indoor reception capabilities while the consumption of multimedia content often take place inside homes. Thus, the

STT (or PC) needs to be connected with an antenna outside the building walls. This requirement presents a major inconvenience to the end-users who could be reluctant to use such a technology only on account of being restricted.

2nd radio network: Our first observation is that the second radio network (*e. g.,* GSM network) is used as a trusted party in this scenario. A direct attack on the network infrastructure is considered infeasible (or too expensive compared to other attacks with similar results, see section 5.5.3.1).

As the key information needed to descramble the content is not available in other regions (by the trusted 2nd radio network), an attacker cannot gain anything from manipulating the STT, smartcard or DRM device hardware. If the information is not available in the region the attacker is located in, it could not be derived from other information stored by the user equipment at all.

An attacker might utilize a functional STT together with a license in one region for the purpose of intercepting the Control Words on the slot interface and use the intercepted data to run a STT in another region where the license is not valid. These Control Words are the secret information continuously issued by the DRM device via the Common Interface that is needed by the STT to descramble the content during a broadcast session.

A counter-measure for this type of attack is to enforce a mutual authentication of the CI module and the STT. As the underlying standardized Pay-TV technology is the vulnerability in this case the proposed system is at most as secure as the content scrambling standard adopted by it. If the scrambling algorithm is broken, then new STTs have to be rolled out anyway and the system could operate on top of this new standard technology again.

Comparison of technical measures: In order to compare the measures we start by identifying the differences first.

The tamper-resistance / TC solution of the problem does require a localization hardware item that needs to be issued together with the STT to the enduser. Such a device would increase the cost for the enduser equipment considerably. Moreover it is limited in its suitability for solving the problem as current available positioning systems can be circumvented by feeding a fake signal to the antenna input. The trusted device would still store secret information that could be used by an attacker to descramble the content if a successful attack on the secure hardware could be launched.

The 2nd radio network solution does not need additional tamper resistant hardware (a smartcard would still be used, though) and it would not store any secret information that could be used to descramble the content since the key information is unavailable outside a target region. The major difference to the positioning solution is that another (trusted) network is needed and the usage of the network services would also add cost to the global content distribution (and some extra hardware for is needed as well). The security limitation here is the amount of trust towards the second radio network management.

If the cost generated by both solutions is assumed to be comparable or negligible regarding the security considerations, the remaining differentiator is the security limitation of each solution. As the generation of fake positioning signals (or usage of copied signals) is a feasible task for an attacker, while the manipulation of a radio network is considered infeasible, and as the trusted hardware can also be subject to successful attacks, the latter solution is favored under these reasonable conditions. Note, that this decision is based on theoretical analysis only and might not withstand real-life conditions regarding cost and availability of hardware and radio networks as well as user acceptance.

5.7 Analysis of Compliance to the Requirements

We will now show that informal requirements laid out in section 5.4 are fulfilled by our proposed 2nd radio network solution while the other solutions have several shortcomings.

We denote by RegCard the regional smartcard distribution described in section 5.6.1, by 2ndNetwork the 2nd radio network solution (section 5.5.2) and by LocTC the TC based localization solution (section 5.6.2). Implementation details of 2ndNetwork are given in section 6.3, details of LocTC in section 6.1 and for a combination of both in section 6.2.

R1: **The system shall provide a global offering to users who are able to receive high-bandwidth broadcast data.**

This requirement is fulfilled by all three of the considered solutions, RegCard, 2ndNetwork and LocTC, because all three alternatives may connect to the (global) broadcast signal. The RegCard solution does not offer user mobility, though, as the cards must not be moved outside the region.

R2: **There shall be no theoretical possibility to construct a system pirate device displaying all content of all regions without a license. Moreover, a pirate device shall never display content dedicated for a region it is not located in.**

This requirement is not fulfilled by RegCard because the key content of all available regional card types could be put on *one* card, thus providing a super-pirate card (*e. g.,* in case the tamper resistance of the original cards is insufficient). Likewise, LocTC can be tricked by fake positioning signals (see section 5.6.4). Only 2ndNetwork offers the restrictions imposed by the second radio network sending key information only to a dedicated region. Outside this region the broadcast signal cannot be descrambled as long as the scrambling algorithm is assumed not to be broken.

R3: **The multimedia coding technology and the STT architecture shall adopt existing standards. There shall be no need to roll out a new broadcasting and set-top technology -** ***e. g.,*** **latest-generation STTs already shipped to the user base shall be ready to participate in a new system.**

All three of the considered solutions, RegCard, 2ndNetwork and LocTC, operate on top of the Common Interface standard, so the latest-generation STTs are supported as long as this standard belongs to state-of-the-art technology.

R4: **The STTs may be exported but shall always enforce DRM policies in the region they are located in.**

This requirement can naturally not be accomplished by RegCard since the cards are supposed only to be useful in one region. 2ndNetwork and LocTC provide either implicit or explicit localization and offer regional DRM enforcement at every geographic position so they satisfy R4.

R5: **The STT technology shall adopt the same tamper-proof and system**

security requirements as existing Pay-TV systems. Established conditional access solutions shall be used.

Since all three of the considered solutions, RegCard, 2ndNetwork and LocTC adopt the established CAS by design, the existing tamper-proof and system security properties of the existing standards are in place.

R6: **An existing state-of-the-art Pay-TV provider shall be able to extend the regional coverage to a global coverage while the legacy user base can still be served with the same technology.**

If the legacy user base is equipped with standard STTs all three architectures will be able to serve them, but 2ndNetwork and LocTC still need to roll-out a piece of extra hardware (DRM Device or Positioning Receiver) to permit the legacy users to become mobile and use their equipment anywhere.

R7: **If the system security is compromised in one region the system security in other regions shall not be affected necessarily.**

Although the inclusion of this requirement is understandable from a broadcaster's point of view, it lacks some logical coherence since it is not clear what a geographical confined failure of the security mechanism might involve. However, the schemes RegCard and 2ndNetwork do provide mechanisms to map certain keys being needed to compute a broadcast session key to certain regions in a way that outside a region the compromised keys are useless. For 2ndNetwork we can point out that a failure of a local radio network does not jeopardize system security in other regions while for LocTC a local failure itself is not a sound notion at all.

6 Implementation Issues

In this part we examine the practical issues relating to implementation of our proposed schemes and mechanisms into current and upcoming consumer electronics and personal computers regarding the consumption of digital multimedia content. We focus on technology related to the DVB standards family, a suite of internationally accepted, open standards for digital television broadcasts maintained by the *DVB Project*, an industry consortium with about 300 members. See section 3.9.1 for details on the adoption of DVB as digital television technology standard.

6.1 Trusted Computing

6.1.1 Concepts

In this section we detail the architecture on location based license acquisition introduced in sections 5.6.2 and 5.6.4.

If a PC that is acting as a set top terminal (STT) is to be trusted it is required to be booted up in a secure manner. This as achieved with TCG technology [98] by starting from an implicitly trusted component. This component validates the next piece of machine code to be executed by the CPU and passes control over to it after approval. This process continues until the operating system is loaded into memory and holds the trust by its own architecture. This continuous sequence of operations of passing control is denoted as *chain of trust*.

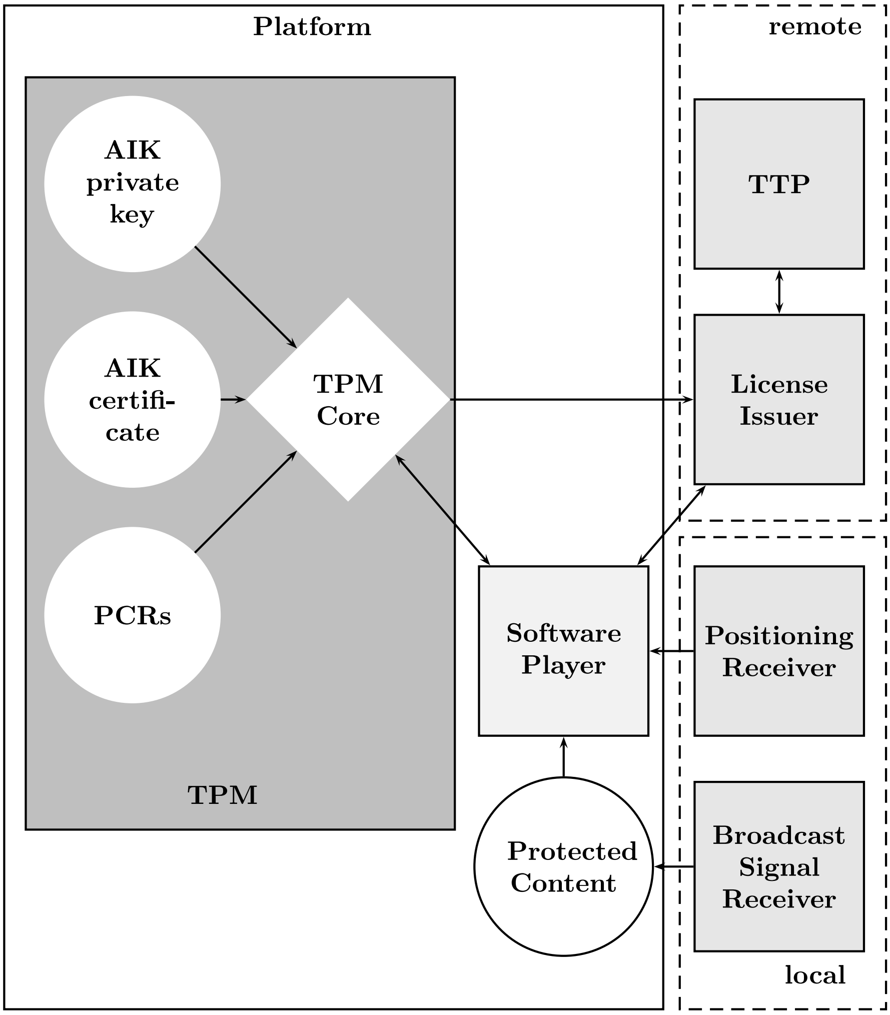

Figure 6.1: Implementation using TPM

During the manufacturing of the TPM instance the Endorsement Key (EK) pair is created and stored as the permanent identity of the platform for later usage of acquiring attestation identity keys (AIKs). The platform also holds a single storage root key (SRK). Details on the different roles of the keys are described in section 3.8.

See Fig. 6.1 for details on a suggested architecture regarding location based license acquisition and rights enforcement. The acquisition and consumption is performed in by executing the following steps.

1. User requests license ticket for the consumption of a certain transmission.

2. Remote license issuer requests attestation of the platform

3. The TPM collects PCRs to attest the trusted state of the platform including the (special) software player that is connected to a compliant hardware receiving positioning signals

4. PCRs with corresponding logs are signed with a AIK private key and sent to the remote party

5. License Issuer validates AIK certificate (if necessary with the assistance of a trusted third party) and verifies signature on PCRs. Protocol is aborted in case of a failure.

6. License ticket is issued and sent to the platform's player software.

7. Protected content is received via broadcast channel.

8. Mutual authentication between software player and positioning signal receiver. Evaluate location of the platform.

9. The software player extracts the content key from the license and allows real-time consumption if the location matches the license restrictions and / or

10. The content is stored sealed on the hard disk so it can be consumed at a later time when the correct system state is in place.

The implementation allows the enforcement of the location dependent license restrictions. Consumption of the content is possible after the location is determined either real-time during the broadcast transmission of later when the sealed storage of the content is accessed (see section 3.8.3 for the TPM sealing function).

Note, that the (trusted) software player receives the positioning information from authenticated hardware but it cannot necessarily ensure that the signals themselves are authentic[1] or that the positioning receivers is local to the platform.

6.2 CPCM Implementation

Regarding the management of digital rights, the standard DVB-CPCM (*DVB Content Protection and Copy Management*, short: CPCM) has been party released by the end of 2005. CPCM is a system for protecting commercial digital content delivered to consumer products and home networks. The specification [27], also called *CPCM Blue Book*, is made up of several parts, some of which specify signaling details, and other parts which explain the rationale behind the specification, including implementation issues. A *Reference Model* provides the framework for the CPCM system. It is the foundation upon which the remaining specification elements are built. Further parts of the standards are expected to be released at midyear 2006.

6.2.1 Propagation of CPCM Cryptographic Keys

The content according to CPCM is delivered together with a license data file, (the *CPCM Content License*). The license includes the following items:

[1] GPS positioning information signals for example can be generated or recorded and replayed by a pirate owning special equipment unless the *GPS Precise Positioning Service* is used, which is unfortunately limited to users specifically authorized by the U.S. military forces [25], and special countermeasures are applied [102].

- Content License Version Information,
- Content Instance Identifier (CIID),
- Content License Creator (CLC),
- Usage State Information (USI),
- Authorized Domain Identifier (ADID),
- Content Descrambling Keys and / or
- Auxiliary Data fields.

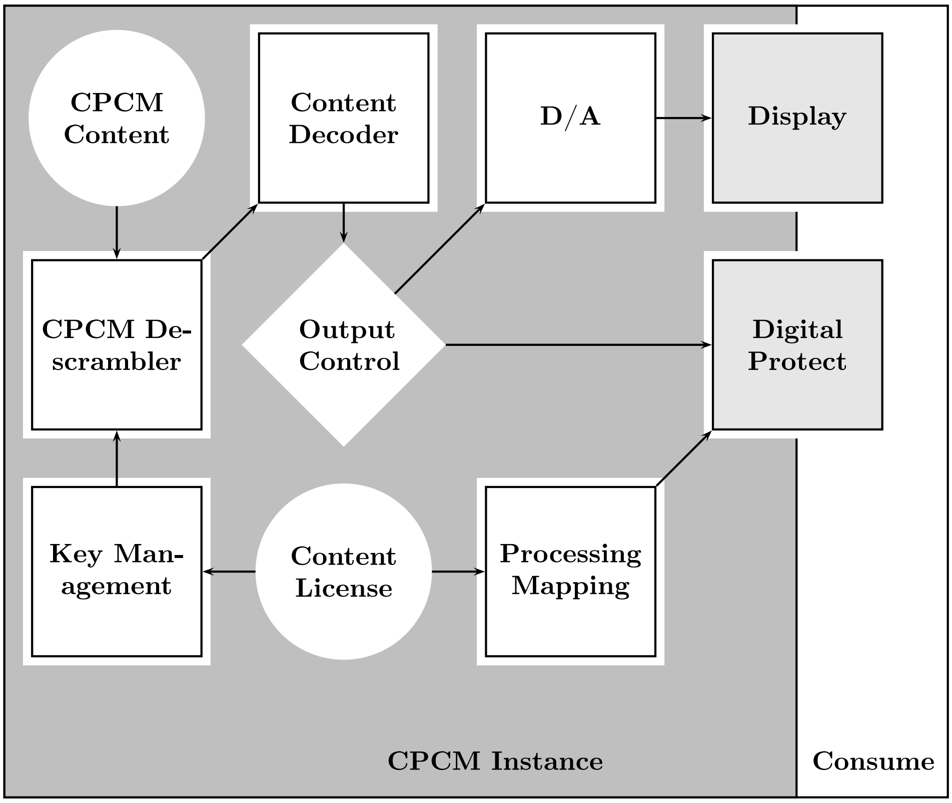

Figure 6.2: CPCM Consumption Point (simplified)

The license information is adequately processed by any CPCM Consumption Point (see Fig. 6.2). The *Key Management* extracts the cryptographic keys needed by the descrambler while the *Output Control* unit enforces the restrictions regarding analog or protected digital consumption; digital output requires the usage rights (after processing and mapping to the technical requirements of the digital consumption entity) to be added to the digital content

The Usage State Information is the metadata that describes the authorized usage for each content in an adequate rights expression language (not further specified by [27]) and the Authorized Domain (identified by the ADID) is a set of CPCM compliant devices of a single household, while the cryptographic keys (if existing in a license) are stored as Descrambling Keys that are needed by a *CPCM Descrambler* being a part of any compliant device that can be used for consumption of CPCM content[2].

The Content License Creator identifies the original creator of the (initial) Content License upon acquisition into the CPCM system. The digital broadcast of a television program is one possible kind of acquisition but there is also support of reading storage media input into the system, among other acquisition processes being out of scope for our purposes, *e. g.,* on-demand type content delivery which does not require a broadcast encryption approach. The Auxiliary Data is defined as "any other data pertaining to the Content Item to which the Content License refers" and can be used for transmitting parameters or identifiers for our implementation purposes.

The CPCM standard [27] does not specify a certain cryptographic algorithm or key size; there will be a future release *CPCM cryptographic tools* as an upcoming part of the standard (announced in [27]) that will suggest a set of ciphers with the appropriate

[2] Devices that provide only storage functionality do not require to incorporate a scrambler or descrambler in order to be compliant.

mode of operation and also a random number generator but the suggested set of ciphers and algorithms always remains subject to future changes.

6.2.2 CPCM Key Blocks for the Proposed Schemes

We will now construct the key-block that is generated during Phase 1 of the improved Improved Biased Sub-set Scheme (see section 4.3.3 for reference). The formal mathematical description requires to send the following structure

$$\big((\pi^{-1}(1), E_{k_{N_1}}(s_1)), (\pi^{-1}(2), E_{k_{N_2}}(s_2)), \ldots, (\pi^{-1}(s), E_{k_{N_s}}(s_s))\big)$$

that represents a concatenation of s encrypted shares.

We follow the CPCM collection of available data formats [27] and use

- uns. int: unsigned integer, most significant bit first
- bitstring: bit string, left bit first

for encoding the data fields of the key-block and implement a structure as defined in Table 6.1.

Structure	**No. of bits**	**Format**
BS_KeyBlock1() {		
$Count\_s$	16	uns. int
for ($i = 1$; $i \leq Count\_s$; $i++$) {		
sub-set ID	32	uns. int
encrypted share i	128	bitstring
}		
}		

Table 6.1: Key-block 1, CPCM implementation

Regarding phase 2 the structure to be encoded is given by

$$\Big((u_1, E_{k_{u_1}^{indv}}(k_S)), (u_2, E_{k_{u_2}^{indv}}(k_S)), \ldots, (u_g, E_{k_{u_g}^{indv}}(k_S))\Big)$$

and can be implemented as defined in Table 6.2 for further parsing by the g users being a member of the set $\mathcal{T} - \mathcal{T}_s'$ in the second phase of the improved Biased Sub-set Scheme (as depicted in section 4.3.3) and for the purpose of establishing the combined scheme (following Definition 4.5.2), we also fix a key-block definition for the SD revocation scheme (Table 6.3) to encrypt content key K using a key encryption algorithm E (see

Structure	**No. of bits**	**Format**
BS_KeyBlock2() {		
$Count\_g$	16	uns. int
for ($i = 1$; $i \le Count\_g$; $i++$) {		
user ID	32	uns. int
encrypted content key	128	bitstring
}		
}		

Table 6.2: Key-block 2, CPCM implementation

Structure	**No. of bits**	**Format**
SD_KeyBlock() {		
$Count\_m$	16	uns. int
for ($k = 1$; $k \le Count\_m$; $k++$) {		
node ID i_k	32	uns. int
node ID j_k	32	uns. int
}		
for ($k = 1$; $k \le Count\_m$; $k++$) {		
encrypted content key	128	bitstring
}		
}		

Table 6.3: SD Key-block, CPCM implementation

Definition 3.5.8) with keys $L_{i_1,j_1}, L_{i_2,j_2}, \ldots, L_{i_m,j_m}$ respectively and generate the tuple

$$((i_1, j_1), (i_2, j_2), \ldots, (i_m, j_m), E_{L_{i_1,j_1}}(K), E_{L_{i_2,j_2}}(K), \ldots E_{L_{i_m,j_m}}(K))$$

while the parameters that can vary after the set-up phase, *i. e.,* the number of shares d needed to reconstruct the secret and the choice of a certain scheme (*e. g.,* Biased Sub-set or Combined Scheme) can be fixed in the Auxiliary Data fields (Table 6.4).

Structure	**No. of bits**	**Format**
BS_AuxData() {		
d	16	uns. int
scheme_select	8	uns. int
}		

Semantics: *scheme_select*	Description of *scheme_select*
0	Improved Biased Sub-set Scheme
1	Revocation Scheme
2	Combined Scheme
3 . . . 255	Reserved

Table 6.4: Auxiliary Data, CPCM implementation

Note, that the auxiliary data fields are intended by the CPCM standard authors to provide “a signaling path for both present and future systems” thus a possible incorporation of our proposed schemes into the CPCM Blue Book is not dependent on the inclusion of the schemes into the first published set of official CPCM cryptographic tools.

6.2.3 Location Based Usage Rights

CPCM compliant devices need to be mapped to an Authorized Domain (AD) in order to enable consumption or propagation of certain content.

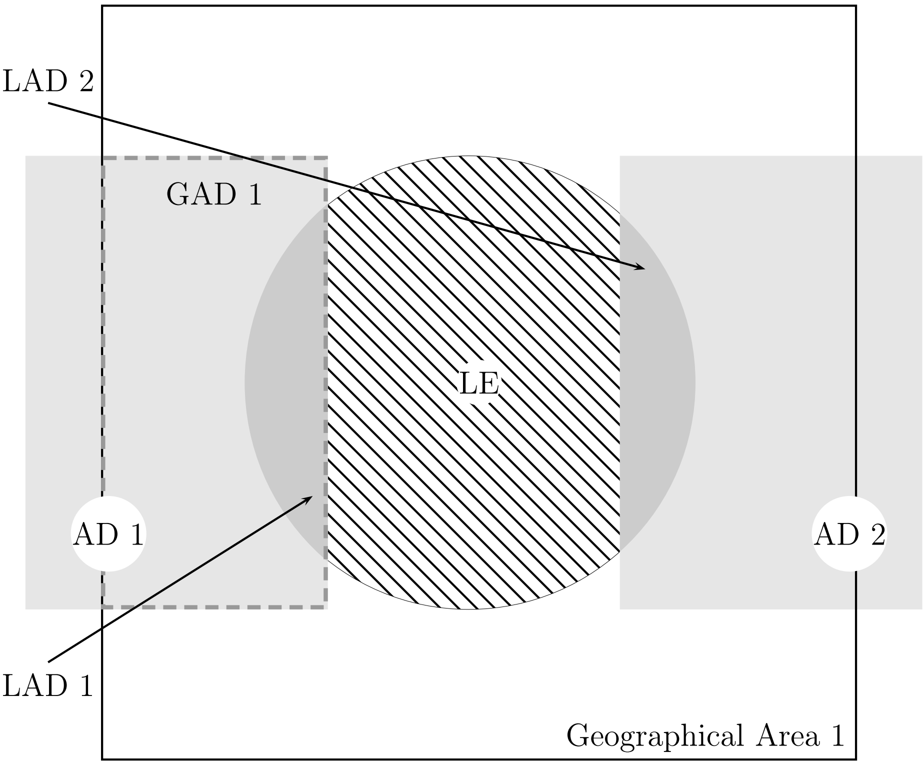

Figure 6.3: CPCM Area Intersections

The AD is a set of CPCM compliant devices of a single household but it is not required that this household is limited to a geographic area because also mobile devices or several locations of one household are regarded[3] in this context. However, restrictions or special usage rights considering the geographic location are explicitly stated and a Local Environment (LE) is defined as the "the immediate vicinity around a CPCM Device". It approximates to the physical dimension of a house. The basic intention

[3] The standard specifically names a possible *Main Home*, *Second Home* (*e. g.,* summer house), *In-Car Device* among other possible elements of an Authorized Domain

with the LE is to allow the Propagation of CPCM Content over the home (local area) network, without any AD binding, but to be able to prevent its propagation over Wide Area Networks [27]. This enables the household members to use one device for content purchase or content reception (*e. g.,* a recording device) and another device in the same LE for consumption (*e. g.,* a television set in another room). The compliant devices could access each other's stored content and distribute it within the household as long as they belong to the same AD and LE (and the spreading is allowed by the usage rights). The logical intersection of the AD and the LE is described by the Localized Authorized Domain (LAD) being the set of all devices of one household located in one home while the intersection of a geographic area and the AD is referred to as the Geographically-constrained Authorized Domain (GAD).

Example 6.2.1. Possible intersections are illustrated in Fig. 6.3. A GAD and a LAD are always a proper sub set of an AD (as depicted in the figure), a LAD is also a proper sub-set of a LE; a LE, GAD and LAD are always a proper sub set of an GA while an LE and LAD may intersect only to some extent as it is the case with an AD and a GA that may share some area without being sub-sets of each other (also shown in the example figure); in other words: an Authorized Domain may not be covered by one Geographical Area.

The propagation of content is allowed, in accordance with other usage rights, if the CPCM device can determine using at least one Geo_Data structure (see Table 6.5) that it is within the specified area. The first two Geo_Coding_Methods (URL[4] and

[4] The URL coding method is not a fully specified option in the current CPCM standard documents [27] but provides a mechanism to point to a web page from which the Geographic_Area structure can be read. By appending to this URL a slash symbol followed by a Geo_Coding_Method number, the device can get the Geo_Data expressed with that particular geo-coding method instead of the whole Geographic_Area structure.

Structure	**No. of bits**	**Format**
Geographic_Area() {		
Loop_Count	8	uns. int
for ($i = 1$; $i \leq Loop\_Count$; $i++$) {		
Geo_Coding_Method	8	uns. int
Length	16	uns. int
Geo_Data()	$8 \cdot Length$	bitstring
}		
}		

Semantics:

Geo_Coding_Method	Description of *Geo_Coding_Method*
0	Indirect method (URL)
1	ISO 3166-1-alpha-2 Country Codes
2	Second Radio Network cell IDs
3 . . . 255	Reserved

Table 6.5: CPCM Geographic Area Definition

country codes[5]) are specified by the CPCM standard while the third coding method (second radio network cell IDs) are our proposal for the implementation of the second radio network approach (as laid out in section 5.5.2) into CPCM.

The cell IDs used for the third coding method can be described with 32 bits[6] which is expected to cater for adequate determination of a country and a region inside this country for digital rights management purposes. If a more granular determination of a geographic region is required (*e. g.,* exact co-ordinates) more bytes (as determined by the *length*-parameter can be used to shrink the cell to a smaller-sized area.

[5] A comma-delimited list of included country codes (2 ASCII characters each) is optionally followed by a comma-delimited list of excluded country codes. Moreover, it is defined that the *AA* country code represents the whole world.

[6] In the case that a GSM network is used: the GSM cell identifier has a length of 16 bits and additionally the GSM *Location Area* (*e. g.,* the operator's country) is coded using another 16 Bits. This, this information could be directly read into the data fields without further processing.

6.2.4 Proximity Control

The CPCM proximity test is the mechanism to determine whether two compliant devices are local with respect to each other at the time the test is performed. According to [27] the test may also be run prior to content exchange or on a periodic (or irregular) basis. The proximity test tools may involve secure CPCM communications between the two devices performing the test and may also be used by AD management when determining whether a new device can join an AD.

Algorithm 5 CPCM Proximity Test by 2nd Radio Network

```
repeat
    pause
until secure CPCM channel between devices is established
get remote $Geo\_Data()$
get local $Geo\_Data()$
if remote $Geo\_Data()$ is adjacent to local $Geo\_Data()$ then
    return TRUE
else
    return FALSE
end if
```

For devices that offer a CPCM Geographic Area determination according to our proposal in the preceding section 6.2.3, we can establish a proximity test that does not rely on a signal round trip time measurement between the devices. We can take advantage of the fact that our proposed Second Radio Network cell IDs (see Table 6.5) can describe a small-sized area and let the proximity test confine itself to check whether both devices are located in the same (or adjacent) cell (formalized by Algorithm 5).

The proximity test is not an implementable CPCM specification by itself, yet. Subsequent specifications are expected to define detailed formats and functions and a compliance and interoperability framework under which compliant products are to be constructed will be part of a future release (this was announced with the publication of the current available specifications). The proximity test method presented in this section could be one of these functions in a subsequent specification.

6.3 CAS: ECM / EMM Implementation

6.3.1 Concepts

A conditional access system (CA system or CAS) is a service that allows broadcasters to limit certain programming products to certain users [20, 76, 81]. This is performed by encrypting (also called *scrambling* in this context) the broadcaster's programs.

CAS offers capabilities such as pay-per-view (PPV) and time-bounded subscriptions, the ability to restrict access to certain material (*e. g.,* adult movies) and the ability to direct messages to specific set-top terminals. The broadcaster applying this Pay-TV technology sends *Entitlement Control Messages (ECMs) and Entitlement Management Messages (EMMs)* to the STT in order to carry out the usage rights and usage restrictions within the user equipment.

At the receiving end, it is the task of the set-top terminal (STT) to descramble the CAS encryption and decode the MPEG data for consumption.

The tuner portion of the STT receives the broadcast, demodulates it and sends the resulting data to the transport stream generator which is the part of the STT that reconstitutes the transport stream. This stream consists of packets and each packet header carries a program identifier (PID). All packets with PID value 1 are unencrypted and are used by the demux processor to set up the conditional access table (CAT) and the program map table (PMT). The CAT identifies all the PID values of the transport packets containing the EMMs while private data associated with the program is included in PMT – *e. g.,* the PID value of the packet containing the ECM.

The data contained in these two messages (the EMM and the ECM) are necessary to descramble the encrypted transmission.

The EMM is an encrypted message that contains conditional access information about the user rights to the consumption of such services as cable, terrestrial or satellite television programs. The ECM contains access criteria and a scrambled short-lived session key called a *control word* (CW). The EMM is related to the authorization of services (not single transmissions) and allows a particular user or a defined geographic region to access these services. It contains the encrypted service key (SK). Typically a SK is changed every few months to discourage pirates [76]. The descrambled SK is used as the key to descramble the ECM containing the CW or containing meta-data to generate CWs. The CW is the key to the transport-stream descrambler during a certain time-frame.

Currently, most Pay-TV subscribers own an STT equipped with a *Common Interface* [20], which is connected with a *CI module* incorporating a smartcard reader where the user will put in a smartcard issued by his Pay-TV provider (see section 3.9.2 for details). EMMs and ECMs are sent via Common Interface to the CI modules which, assisted by a smartcard storing secret key information, generates the CWs being returned via the interface to the STT.

6.3.2 Implementation

The CAS architecture laid out in the preceding section generally allows an implementation of a broadcast encryption system because the ECMs and EMMs can be used to transmit the header information to the CI module where the the scheme's algorithm can be implemented without the need of specifying additional STT functionality. The smartcard allows for secure storage of the user's private key information and also acts as a container that can be sent (*e. g.,* via postal service) to the user in case of a subscription change or re-keying process. However, this architecture does support stateful schemes since the smartcard can be used for re-keying operations through EMMs as well as activation and de-activation of services provided by the card. Thus, broadcast encryption schemes tailored for stateless receivers do not take full advantage of the available mechanisms and can be inefficient in such a way that the header length of a certain transmission is dependent on the set of privileged users of that transmission only but not on any preceding transmissions with prior key agreements that could be re-used. In contrast to this statefulness issue, any implementation that is restricted to stateless mechanisms could be advantageous in a situation where the receiving STT was disconnected from the broadcast communication for an extensive period of time and missed re-keying operations carried out within this period. Using stateless protocols in the CAS architecture on special occasions (*e. g.,* during or shortly after re-keying operations) can be the fall back position to realize maximum quality of service in the sense that every subscriber is able to consume the content she is entitled to.

Structure	**No. of bits**	**Format**
CA_message_section() {		
table_id	8	uns. int
section_syntax_indicator	1	bitstring
DVB_reserved	1	bitstring
ISO_reserved	2	bitstring
CA_section_length	12	uns. int
for $(i = 1; i \leq CA\_section\_length; i++)$ {		
CA_data_byte	8	bitstring
}		
}		

Table 6.6: CA Message Definition

The transport of Conditional Access information, such as ECMs and EMMs is described by Table 6.6 and the data structure as a whole is limited to a maximum length of 256 bytes [26], thus we can deploy a payload of 253 bytes per ECM (or EMM) message for our purposes. Note, that the DVB standards define the interface to the conditional access system and the format of the ECM / EMM messages but not the cryptographic key information encoded in the messages. The Conditional Access implementation itself could be built on a proprietary basis without the need (or the benefit) to follow a standard and all major European Pay-TV stations use proprietary CA systems [74].

Structure	**No. of bits**	**Format**
CA_message_section() {		
table_id	8	uns. int
section_syntax_indicator	1	bitstring
DVB_reserved	1	bitstring
ISO_reserved	2	bitstring
CA_section_length := 192	12	uns. int
% BS Scheme Payload:		
BS_ECM_1() {		
broadcast ID	32	uns. int
sub-set ID	32	uns. int
encrypted share i	128	bitstring
}		
}		

Table 6.7: Biased Sub-set phase 1, CAS ECM implementation

Since the available 253 bytes for an ECM message do most likely not cater for the biased sub-set key block (see for analogy the CPCM implementation data structure detailed in Table 6.1) we are required to split the key block into several messages each containing a triple

$$\Big(broadcast\_ID, \pi^{-1}(1), E_{k_{N_i}}(s_i)\Big)$$

that represents one of the s encrypted shares ($1 \leq i \leq s$). The *broadcast_ID* is used here so that the CI module (together with the smartcard) can filter and re-concatenate the relevant encrypted shares to a key block that might be sent during a protocol run of another broadcast key agreement with a different ID.

Structure	**No. of bits**	**Format**
CA_message_section() {		
(header information)	8+1+1+2+12	see Tab. 6.7
% BS Scheme Payload:		
BS_ECM_2() {		
broadcast ID	32	uns. int
user ID	32	uns. int
encrypted content key	128	bitstring
}		
}		

Table 6.8: Biased Sub-set phase 2, CAS ECM implementation

The second phase can then analogously be implemented as defined in Table 6.8 for content key distribution to the users of the set $\mathcal{T} - \mathcal{T}_s'$ as defined for the improved Biased Sub-set Scheme (formalized in section 4.3.3) and for implementation of the combined scheme (as in Definition 4.5.2), we also fix an ECM definition for the SD revocation scheme (Table 6.9).

Each ECM of Table 6.9 format represents a triple

$$\Big(broadcast\_ID, (i_k, j_k), E_{L_{i_k, j_k}}(K)\Big)$$

of a total of m such triples so that the key block

$$\big((i_1, j_1), (i_2, j_2), \ldots, (i_m, j_m), E_{L_{i_1, j_1}}(K), E_{L_{i_2, j_2}}(K), \ldots E_{L_{i_m, j_m}}(K)\big)$$

Structure	**No. of bits**	**Format**
CA_message_section() {		
(header information)	8+1+1+2+12	see Tab. 6.7
% BS Scheme Payload:		
SD_ECM() {		
broadcast ID	32	uns. int
node ID i_k	32	uns. int
node ID j_k	32	uns. int
encrypted content key	128	bitstring
}		
}		

Table 6.9: SD Key Agreement, CAS ECM implementation

can be reconstructed by the CI Module together with the smartcard to decrypt the content key K using a key decryption algorithm E^{-1} (see Definition 3.5.8).

7 Conclusions

In this dissertation we proposed schemes for efficient broadcast key establishment that offer a tradeoff between the ratio of free-riders and other parameters (overall key size or message header size). The schemes do not require stateful receivers and the second one is unconditionally secure (disregarding the existence of free-riders). Free-riders will also be prevented if revocation schemes are used together with our proposed schemes so a combined scheme consisting of a well established revocation scheme and our proposed new scheme is also constituted.

Moreover, we presented a concept of transforming existing regional conditional access technology to a global digital rights management system enforcing regional license requirements. We identified some obstacles regarding DRM enforcement that possibly prevent a global technological offering although such a system could reduce the transmission costs significantly. We sketched a new global conditional access system for broadcasting audiovisual content that could be implemented on top of existing technology and that would use standard hardware architectures already rolled out in the user space (set-top terminals). We also presented evidence that the proposed new scheme will not be less secure (in a relaxed notion) than existing state-of-the-art Pay-TV systems.

Useful answers to the problem of enforcing a DRM system that needs to consider location-dependent licensing policies can be based on very different technical or organizational prerequisites. We considered some of them in this thesis and provided a line of argument that might help a broadcaster or consumer electronics manufacturer to reach a decision in favor of one certain approach.

Regarding the broadcast encryption schemes that form a building block of a digital rights management system for multimedia transmissions we also prepared information on parameter dependencies of the proposed schemes that facilitate comparisons of these schemes to other published schemes in this area. Furthermore, simulation data was compiled for realistic parameter values suggesting that the schemes are essentially appropriate for implementation.

Finally, we outlined possible implementations of our proposed cryptographic schemes and technical mechanisms within current and upcoming industrial standard frameworks to show that the proposed building blocks are indeed suitable for generally agreed conditional access and DRM standards.

A Sample Data

A.1 Comparison to other Broadcast Encryption Schemes

In this section we evaluate our probabilistic broadcast encryption scheme by comparing its performance with that of existing broadcast encryption schemes. Comparison will be mainly in terms of communication overhead (*i. e.,* broadcast message length), storage (number of keys and public storage) per user, as well as computational complexity per user. The latter will be measured in terms of *major* operations, *i. e.,* we will count the number of PRNG executions and decryption operations, and also the long-integer multiplications and exponentiations but emphasize the long-integer operations with an asterisk symbol.

Furthermore, we focus our comparison on *1-resilient schemes* and rather *small numbers of users* n (in the order of 10000). These restrictions make comparisons between the different schemes possible: 1-resilient schemes are the basic building blocks for constructing k-resilient schemes using Fiat and Naor's construction [36] described in section 3.4.4 from several independent instancesn of a 1-resilient scheme. Note, that this comparison is *unfair* to schemes being infinity-resilient; these are the trivial schemes and the revocation schemes (CS, SD).

Parameter results are derived by the corollaries 3.4.7 (on page 41), 3.5.13 (on page 61), 4.2.6 (on page 108) and 4.5.5 (on page 134).

Table A.1 shows several sample data values for the proposed share-wise key distribution scheme. In this case half of the user base is in the privileged set $\mathcal{T}$ (*i. e.,* $|\mathcal{T}| = |\mathcal{N}|/2$), while the other half is not (generic case), and resilience is fixed as $k = 1$. Table A.2 shows results for a smaller set of users $\mathcal{N}$ where more than half of the users (600 from 1000) are in the privileged set.

BE scheme	user keys	total keys	shares	header bits	user ops	FR_{rat}
Fiat&Naor [36]	10,000	10,001	NA	10,000	5,000	0
Fiat&Naor [36] (OWF)	14	20,001	NA	10,000	5,000 *	0
Fiat&Naor [36] (Root)	1 (+ 10,000 PKs)	10,003	NA	10,000	5,001 **	0
Biased Sub-set Scheme	500,000	1,000,000	1000	144,754	1,000	0.16
Combined Scheme	500,588	1,009,999	1000	239,170	1,001	0
Trivial 1	2^{9999}	2^{10000}	NA	10,000	0	0
Trivial 2	1	10,000	NA	330,000	1	0
CS Revocation [82]	14	19,999	NA	320,000	1	0
SD Revocation [82]	588	9,999	NA	575,000	1	0

Table A.1: Performance results: our 1-resilient scheme compared to the 1-resilient schemes of Fiat and Naor [36] and revocation schemes of Naor, Naor and Lotspiech [82]. $|\mathcal{N}| = 10000$, $|\mathcal{T}| = 5000$, size of keys and shares is 64 bits. (*) plus initialization phase. (**) long-integer operations plus one exponentiation.

BE scheme	user keys	total keys	shares	header bits	user ops	FR_{rat}
Fiat&Naor [36]	1,000	1,001	NA	1,000	600	0
Fiat&Naor [36] (OWF)	10	2,001	NA	1,000	600 *	0
Fiat&Naor [36] (Root)	1 (+ 1,000 PKs)	1,003	NA	1,000	601 **	0
Biased Sub-set Scheme	500,000	1,000,000	100	13,874	100	0.07
Combined Scheme	500,588	1,009,999	100	24,374	101	0
Trivial 1	2^{999}	2^{1000}	NA	1,000	0	0
Trivial 2	1	1,000	NA	39,400	1	0
CS Revocation [82]	10	999	NA	39,146	1	0
SD Revocation [82]	251	999	NA	42,000	1	0

Table A.2: Performance results: our 1-resilient scheme compared to the 1-resilient schemes of Fiat and Naor [36] and revocation schemes of Naor, Naor and Lotspiech [82]. $|\mathcal{N}| = 1000$, $|\mathcal{T}| = 600$, size of keys and shares is 64 bits. (*) plus initialization phase. (**) long-integer operations plus one exponentiation.

A.2 Simulation Results

A.2.1 Methodology

In order to gather numerical data regarding parameters and efficiency of our probabilistic broadcast encryption schemes, several methods could be considered:

1. The approximation formulae taken from Theorem 4.2.4 and Table 4.2 can be used.
2. The scheme can be partially simulated while some values are approximated by their average value.
3. The scheme can be fully simulated.

While the first alternative can be used to quickly settle a set of parameters for a realistic use-case, there is a risk of a large magnitude of error since each partial error of an sub-approximation is carried over and accumulated to the resulting values.

The second alternative provides pretty accurate numerical values that were used to generate the visualization figures A.3 – A.6. It proved to be a good method to generate data that does not dither too heavily making it easier to visualize the outcome of probabilistic computations. It also allows to collect data for large user key spaces (*i. e.,* large number of pre-distributed sub-set keys) as full simulations of these cases are very costly. The accuracy drops for boundary values (*e. g.,* cases where almost none or all users are in the target set) and this was considered in the parameter choices.

Instead of generating M sub-sets each containing half of the users, M integers are chosen at random each having the same hypergeometric distribution as the number of privileged users in s random sub-set (containing half of the users). The s highest

```
be1plot:=proc(N,m,s,tt,tt2,step) localp1,X,A,B,Y,Z,L,L2,L3,L4,L5,targets,
freeriders,messages,messNNL,T,t,PFR,d,XTS,TS,XNT,FRRAT,FR,XT,SUCRAT,SUC;
description\"BS scheme approx.\"; L:=[]; L2:=[]; L3:=[]; L4:=[]; L5:=[];
for T from tt by step to tt2 do
  d:=1/2*s;
  XTS:=(Statistics:-RandomVariable)
  (Normal(1/2*T,sqrt(1/4*(T*(1-T/N)*N)/(N-1))));
  TS:=evalf((Statistics:-Quantile)(XTS,1 - s/m)); PFR:=(1/2*N-TS)/(N-T);
  XNT:=(Statistics:-RandomVariable)(Normal(PFR*s,sqrt(s*PFR*(1-PFR))));
  FRRAT:=evalf(1 - (Statistics:-CDF)(XNT,d));
  FR:=FRRAT*(N - T);
  XT:=(Statistics:-RandomVariable)
  (Normal((TS*s)/T,sqrt((TS*s*(1-TS/T))/T)));
  SUCRAT:=evalf(1 - (Statistics:-CDF)(XT,d)); SUC:=SUCRAT*T;
  messages:=s+T - SUC; messNNL:=messages+round(FR*1.25);
  L:=[op(L),[T,SUC]]; L2:=[op(L2),[T,FR]]; L3:=[op(L3),[T,messages]];
  L4:=[op(L4),[T,messNNL]]; L5:=[op(L5),[T,(N-T)*1.25]];
end do;
[L,L2,L3,L4,L5]
end proc
```

Figure A.1: BS Approximation Maple Script

values (out of M) are then determined. These maximum biased values are then used to approximate a probability (taken from the median of the s highest values) for the events that (i.) a privileged and (ii.) a non-privileged users are in a sorted sub-set (and can thus decrypt a transmitted share).

The binomial distribution is subsequently employed to determine the number of privileged users receiving enough shares as well as the number of non-privileged users becoming free-riders. From these numbers the messages for phase 2 can be calculated as well as the number of messages necessary to revoke the free-riders.

This partial simulation approach was implemented by the author with the *Maple 10* computer algebra system [23]. See Fig. A.2 for an exemplary script that generates

```
be2plot:=proc(N,T,m,ss,ss2,step)
local p1,X,A,B,Y,Z,s,L,L2,L3,L4,targets,freeriders,messages,messNNL;
description\"BS scheme\";
L:=[]; L2:=[]; L3:=[]; L4:=[];
if T<=1/2*N then
  X:=(Statistics:-RandomVariable)(Hypergeometric(N,T,1/2*N))
else
  X:=(Statistics:-RandomVariable)(Hypergeometric(N,N - T,1/2*N))
end if;
A:=(Statistics:-Sample)(X,m); B:=(Statistics:-Sort)(A);
for s from ss by step to ss2 do
  if T<=1/2*N then
    p1:=evalf(B[round(m - 1/2*s)])/T
  else p1:=evalf(1/2*N - B[round(1/2*s)])/T
  end if;
  Y:=(Statistics:-RandomVariable)(Binomial(s,p1));
  Z:=(Statistics:-RandomVariable)(Binomial(s,(1/2*N-T*p1)/(N-T)));
  targets:=round(T*(Statistics:-Probability)(1/2*s<=Y));
  freeriders:=round((N - T)*(Statistics:-Probability)(1/2*s<=Z));
  messages:=s+T - targets; messNNL:=messages+round(freeriders*1.25);
  L:=[op(L),[s,targets]]; L2:=[op(L2),[s,freeriders]];
  L3:=[op(L3),[s,messages]]; L4:=[op(L4),[s,messNNL]]
end do;
[L,L2,L3,L4]
end proc
```

Figure A.2: BS Partial Simulation Maple Script

simulation data comparing different numbers of shares while all other parameters are fixed while the script from Fig. A.1 does the approximation according to Theorem 4.2.4.

The third alternative provides exact numbers for all parameters and is useful as a *reality check* for the other options. Moreover, it can be used for constructing visualizations that depict the ranges of values that result from probabilistic effects. It was used for the figures A.7 – A.9. The downside of the full simulation approach is that it requires a rather large computation time, especially for excessive parameter choices (*e. g.,* more than 10 million sub-sets), where it can easily take several hours for an average personal computer to compute one data item. The full simulation was implemented during a practical course at the Chair of Network and Data Security, Ruhr-University Bochum by Andreas Krügersen [68] under the author's supervision.

A.2.2 Partial Simulations Results

A.2.2.1 Combined Scheme vs. Sub-set Difference

In Fig. A.3 the focus is on the Combined Scheme (Def. 4.5.2) and its intersection with Naor *et al.*'s Sub-set Difference Method (Def. 3.5.8); so both schemes do not include any free-riders. Note, that there is an average case scenario (as in the other simulations) so the SD scheme is assumed to generate $1.25(|\mathcal{N}| - |\mathcal{T}|)$ messages (and not the $2(|\mathcal{N}| - |\mathcal{T}|)$ for the worst case). The number of sub-sets (*i. e.,* the number of pre-distributed sub-set keys) is chosen very low and a user batch size of 1000 users is assumed here. The number of shares parameter is 100 in the upper sub-figure and 50 in the lower; half of the shares are needed to decrypt the session key. Apparently, 50 shares is a better choice, at least for most of the simulated values $|\mathcal{T}|$ on the x-axis. It also can be estimated that both variants of the Combined Scheme are more efficient (in the number of messages) than the SD method when more than 300 users are revoked from the target set.

A.2.2.2 Batch Sizes

In the next figure (Fig. A.4) a larger batch size $(10,000)$ is fixed and more sub-set keys are pre-distributed (a million), the number of shares is also increased accordingly.

Again the focus in on the intersection point and the ratio of revoked users at this point is about the same value as in Fig. A.3 , but slightly higher. The lower sub-figure shall demonstrate that the strategy to split a large user base into batches of 1000 or $10,000$ users is indeed legitimate since the Combined Scheme's performance for large user sizes (here: a million users) is low.

A.2.2.3 Accepting Free-Riders

In Fig. A.5 the Biased Sub-Set scheme without free-rider elimination is simulated with reasonable parameters (same as in Fig. A.3 but with more sub-set keys) and different numbers of shares (half of them are used to compute the session key) on the x-axis. Here, the effect that free-riders are reduced by sending more phase 1 messages, becomes visible. The total number of messages (shares plus phase 2 messages) is slightly increasing depicting that the reduced number of phase 2 messages is more than compensated by the messages transmitting the shares. In the lower sub-picture also the number of messages for the Combined Scheme (Def. 4.5.2) come into place and we can perceive a minimization problem: A very large number of shares increases the total number of messages by the share messages while a very low number of shares provokes a high number of free-riders so the Combined Scheme's performance suffers from the revocation messages. In between these choices an optimal value $s \approx 120$ for this case $(|\mathcal{T}|) = 400)$ can be estimated.

A.2.2.4 Sub-Set Keys

Finally, Fig. A.6 shows in contrast to Fig. A.5 that for a large batch size other values for s (about a thousand shares) are to be considered. Moreover, the lower sub-figure indicates the enhancement of the scheme's message efficiency for a larger number of pre-distributed sub-set keys (here: ten million). The factor-10 increase of key-size yields a reduction of total number of messages of about 20 per-cent.

A.2.3 Full Simulations Results

A.2.3.1 Free-Riders vs. Target Users

In Fig. A.7 the number of free-riders is fully simulated for different numbers of users. For each value of $|\mathcal{T}|$ five simulations are run showing the *dithering* of free-rider numbers due to the probabilistic nature of the Biased Sub-set scheme. The number of sub-sets (*i. e.,* the number of pre-distributed sub-set keys) is chosen very low and a user batch size of 1000 users and 100 shares are assumed here, identical to Fig. A.3. The absolute number of free-riders is then transformed to the ratios (*i. e.,* divided by the number of non-privileged users) for the lower sub-figure and the values are rounded to nearest per-cents to emphasize the probabilistic spread. The dithering is clearly visible but there is also strong indication that the BS scheme does not produce outlier values.

A.2.3.2 Phase 1 Users, Shares

Fig. A.8 contrasts to Fig. A.7 by a higher number of pre-distributed sub-set keys and transmitted shares, it also includes the BS phase 1 users, *i. e.,* the number of users receiving enough shares in phase 1 of the improved Biased Sub-set scheme to be able to decrypt the session key. It becomes visible that the BS phase 1 user ratio (being the success ratio in the terminology of Table 4.1) drops for the events where only few users are revoked while the free-riders vanish at the same time.

A.2.3.3 Reconstruction Threshold Parameter

Yet another parameter to be considered is focused on in Fig. A.9. The minimum number of shares needed to compute the session key is varied from half (100) of the transmitted 200 shares to 95 and 105 shares. Apparently it is useful to increase the threshold value in case of a small number of privileged users as the number of free-riders is reduced heavily while the success in phase 1 is only slightly reduced. The same argument applies to the case where many users are in the target set: here the threshold can be decreased to get a better success rate while the number of free-riders is only slightly increased.

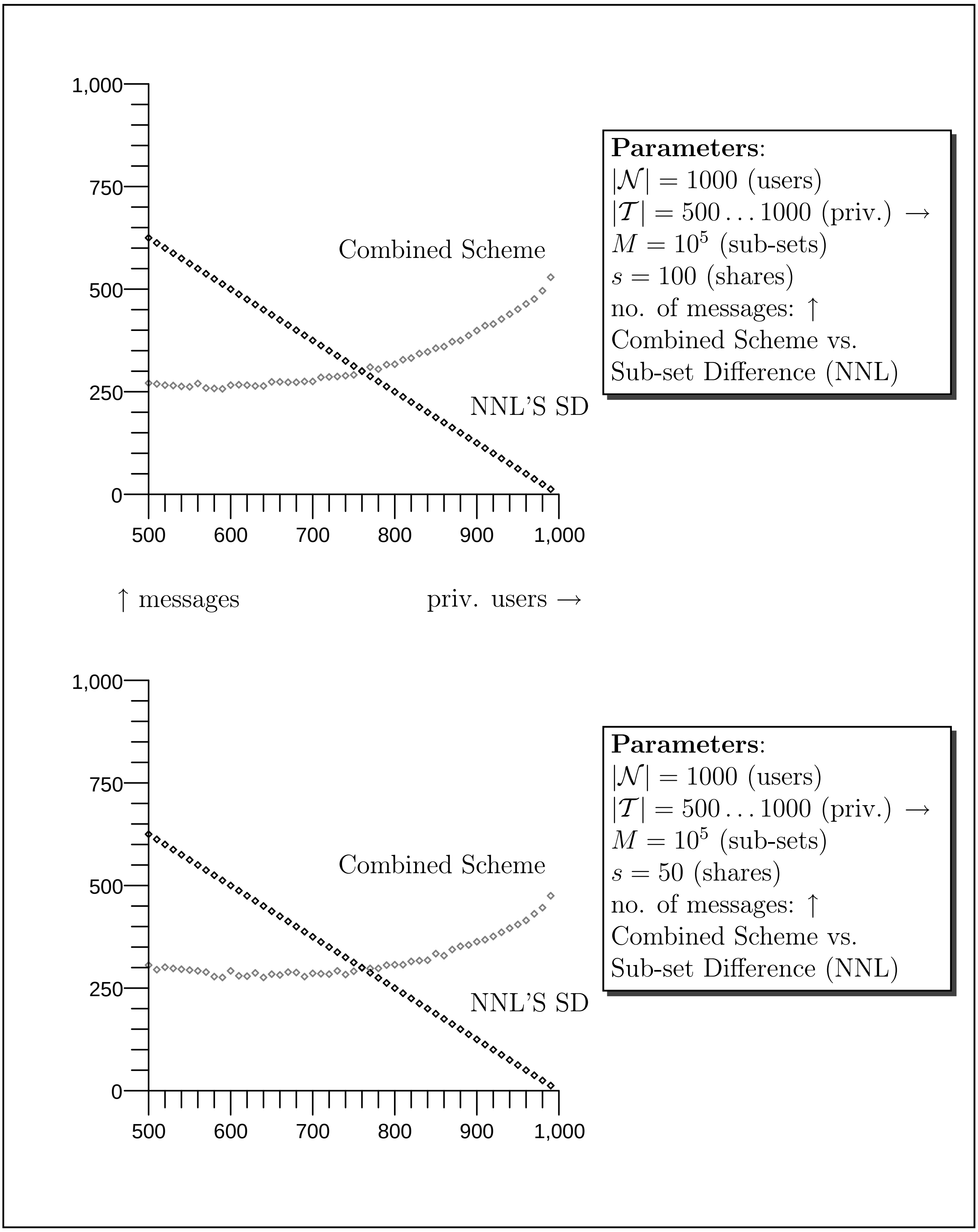

Figure A.3: No Free-riders (Compare 50 to 100 Shares)

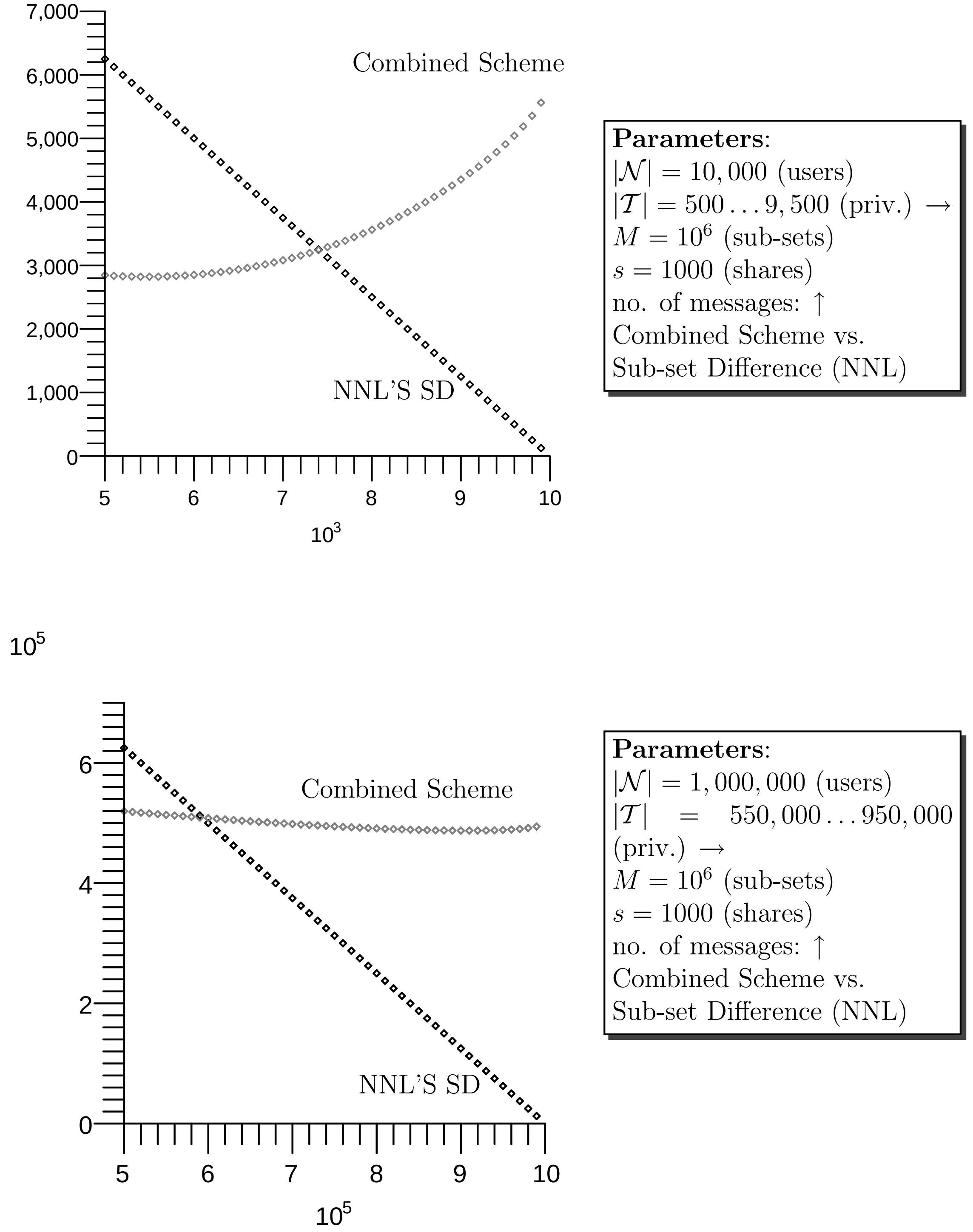

Figure A.4: No Free-riders (Batches vs. Full Set)

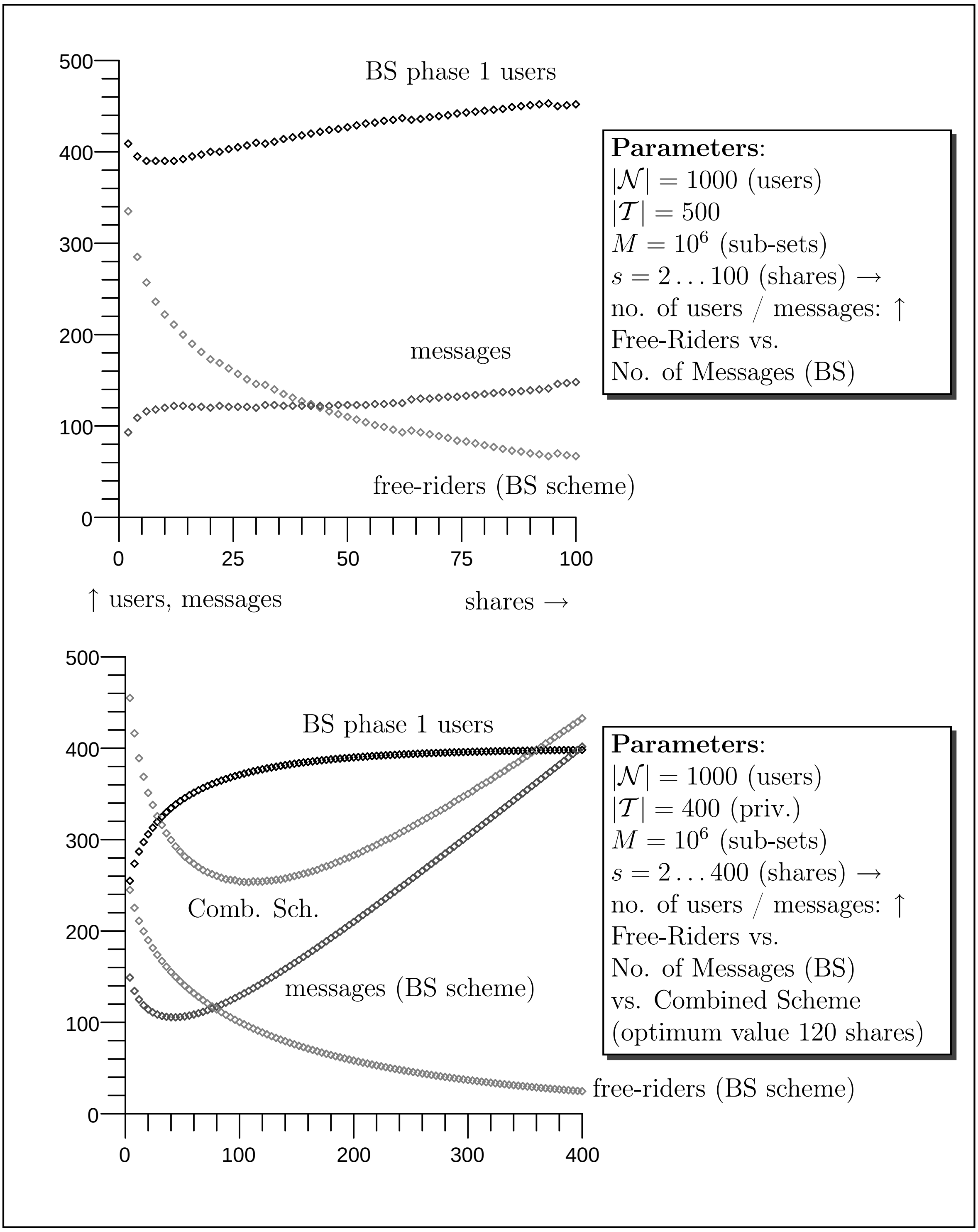

Figure A.5: Choosing Number of Shares Parameter

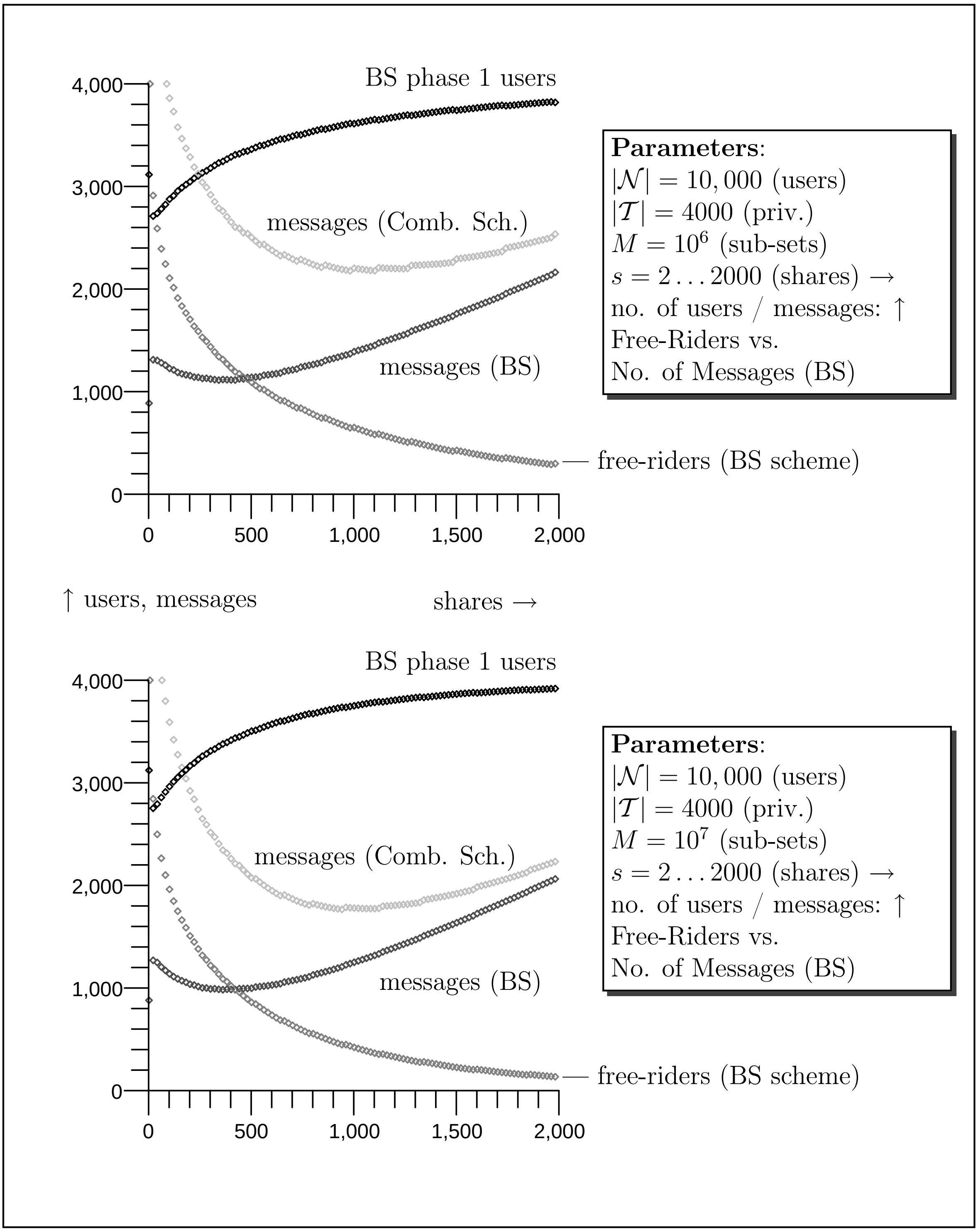

Figure A.6: 10^6 Sub-sets vs. 10^7 Sub-sets

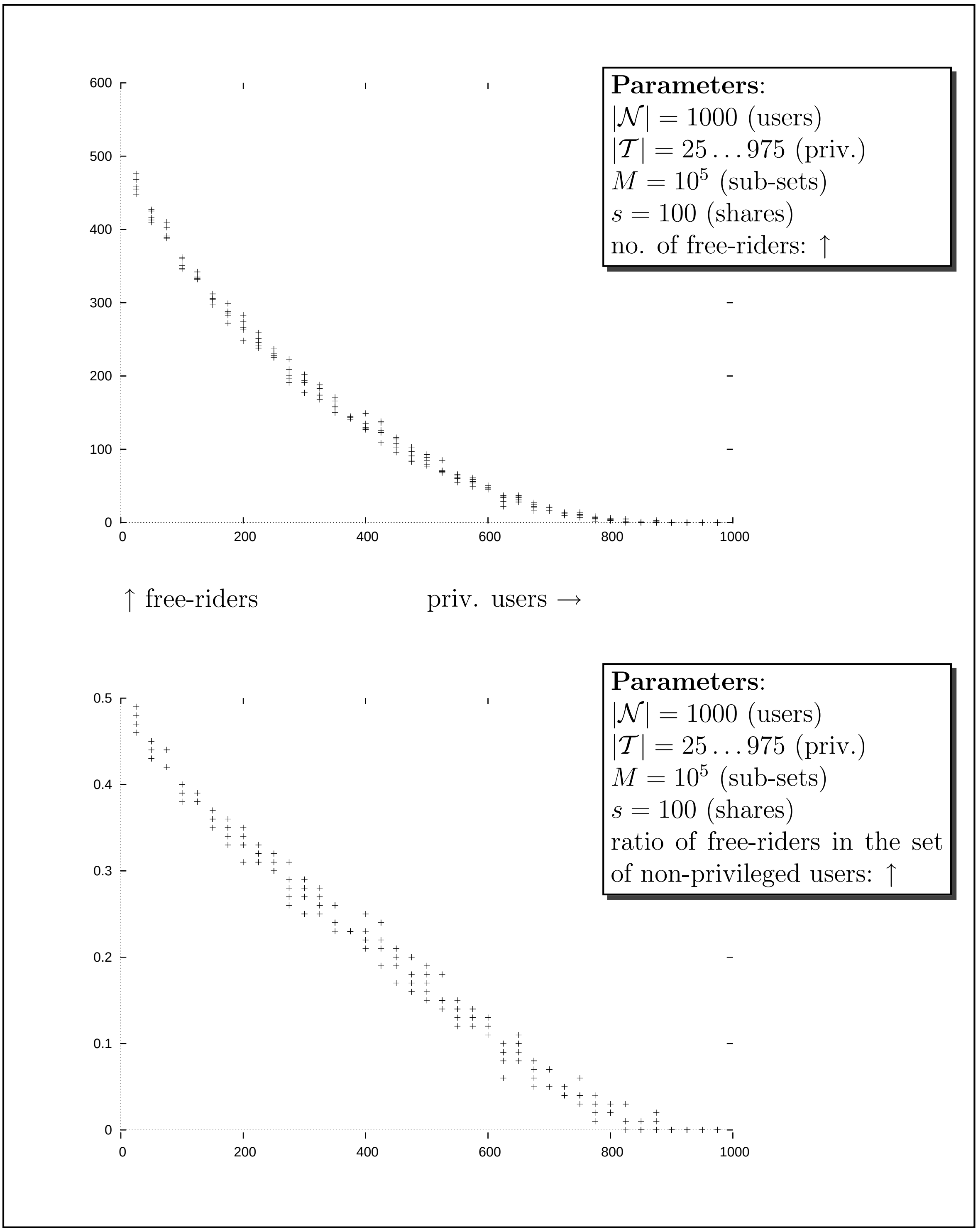

Figure A.7: Full Simulation: Determine No. of Free-Riders / Ratio

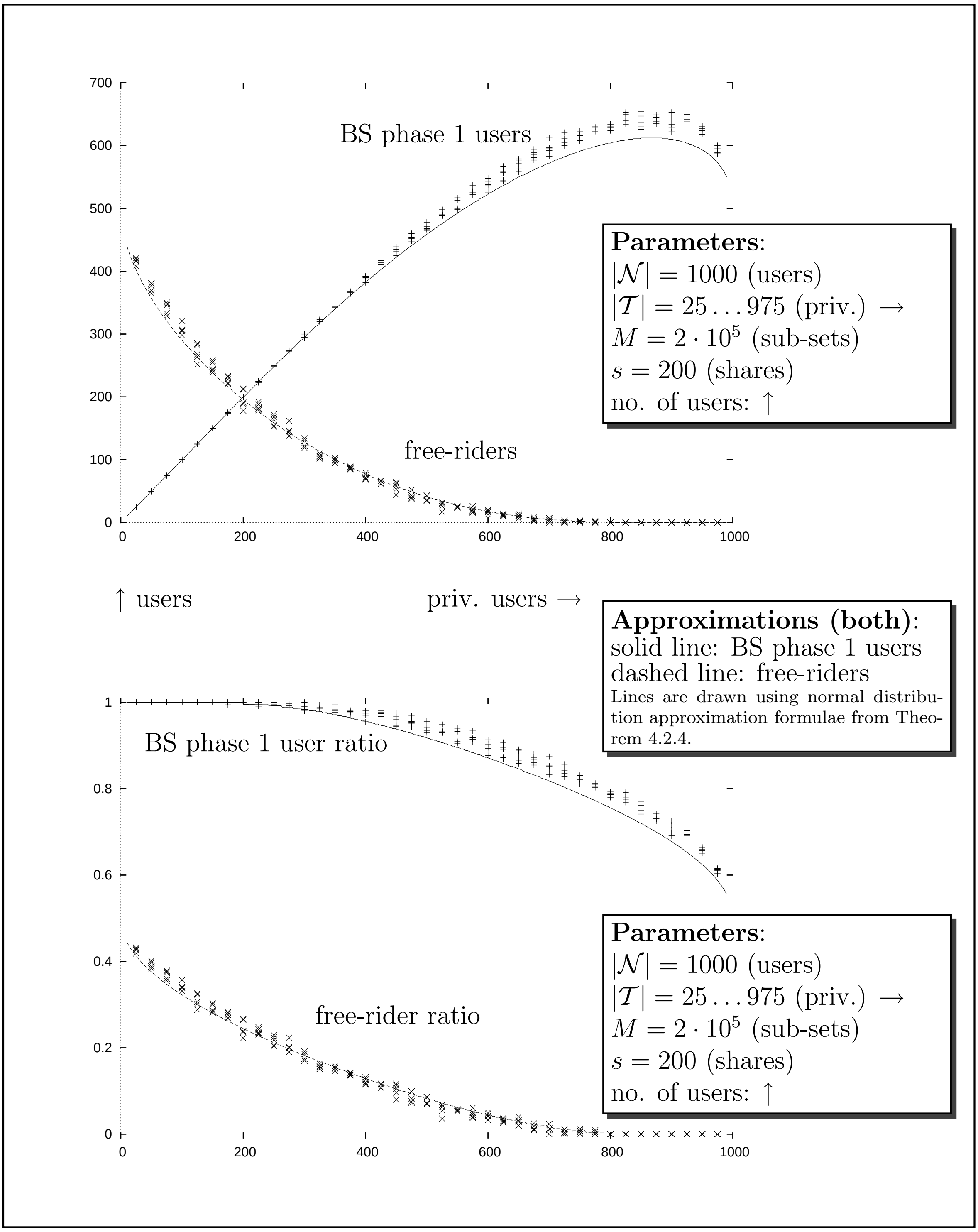

Figure A.8: Full Simulation: Free-Riders vs. BS Phase 1 Users; Ratios

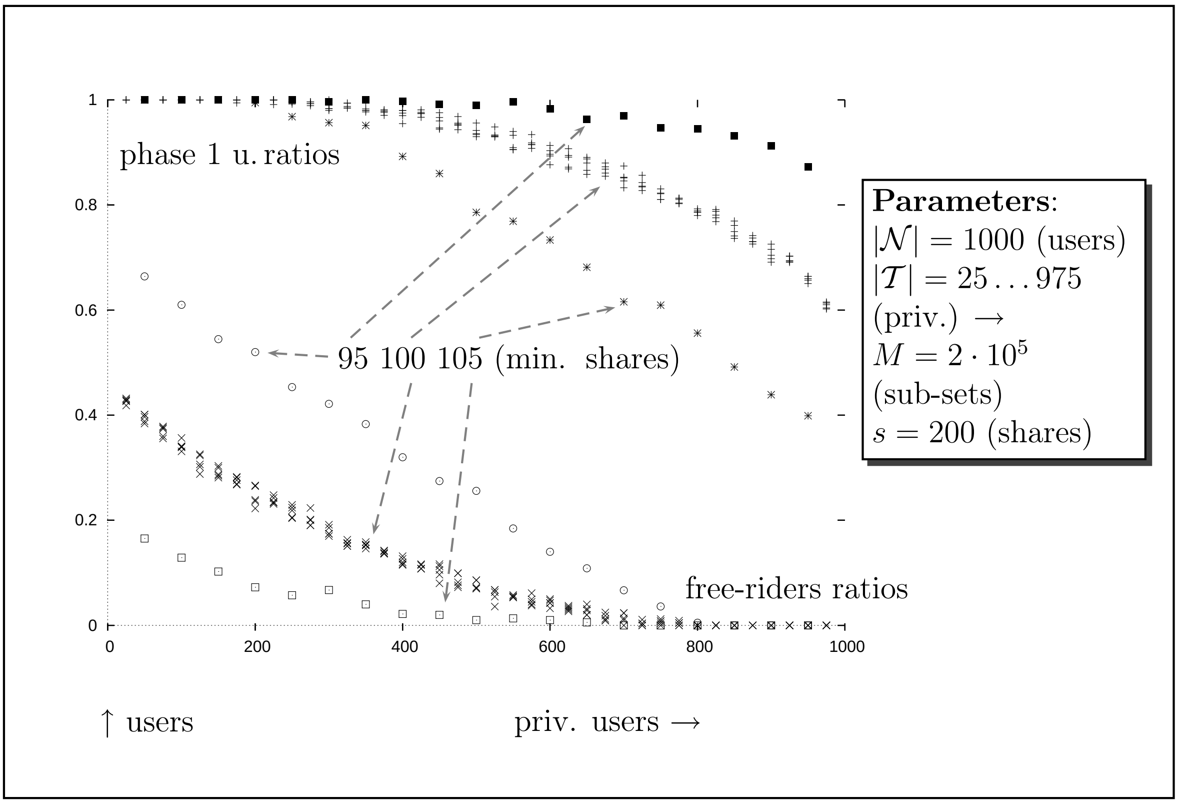

Figure A.9: Full Simulation: Minimum Shares

B Symbols and Frequent Notations

Notation / Symbol	**Meaning**
b	number of batches
$\mathcal{B}$	set of broadcast messages
$B(n, p)$	binomial distribution
Broadcast	broadcast algorithm of a broadcast encryption system
BS scheme	Biased Sub-set Scheme (section 4.2)
d	number of bits needed for user's exhaustive key search
	(Bit-wise Biased Sub-set Scheme)
	or number of shares needed to reconstruct secret key
	(Improved Biased Sub-set Scheme)
d'	number of bits needed for pirate's exhaustive key search
	(Bit-wise Biased Sub-set Scheme)
Decrypt	decryption algorithm of a broadcast encryption system
FR_{rat}	free-rider ratio after phase 1 of BS scheme
$\Phi(\cdot)$	distribution function
$\Phi^{-1}(\cdot)$	quantile function
$i = 1 \ldots M$	index of a sub-set

k	resiliency parameter
K	session key, broadcast key, media key
k_S	session key (BS scheme)
$\|k_S\|$	length of session key in bits (BS scheme)
k_{N_1}	sub-set key of sub-set N_1
k_u^{indv}	individual key of user u
$\mathcal{K}$	set of (all available) keys
LSD scheme	Halevy and Shamir's Layered Sub-set Difference scheme [49]
M	number of sub-sets N_i'
$N(\mu, \sigma^2)$	Gaussian distribution
$\mathcal{N}$	set of users of a broadcast encryption system
$\|\mathcal{N}\|$	number of users
n	number of users (sometimes)
N_i'	sub-set containing half of the users
N_i	sorted sub-set with index i
$N_i \geq_{\mathcal{T}} N_j$	sub-set N_i contains equal or more privileged users than sub-set N_j
π	permutation (for the purpose to sort sub-sets)
$p = \frac{\|\mathcal{T}\|}{\|\mathcal{N}\|}$	probability that user is privileged, context: Bernoulli (p, n) distribution
P_C	collusion success probability
P_u	private key information of user u
p_t	probability: t users are privileged
s	number of transmitted bits / shares

SD scheme	Sub-set Difference Scheme by Naor *et al.* [82]
Setup	set-up algorithm of a broadcast encryption system
SUC_{rat}	success ratio after phase 1 of BS scheme
$\|\mathcal{T}\|$	number of privileged users
t_i	priv. users in unsorted sub-set i
t_i'	priv. users in sorted sub-set i
$\overline{t_s}$	average number of privileged users in sorted sub-set
$\mathcal{T}_0 \subseteq \mathcal{N}$	augmented target set (target plus free-riders)
$\mathcal{T}_i'$	target users that can compute the session key after i bits / shares are sent (BS scheme)
u	user of a broadcast encryption scheme
$\mathcal{U}_i$	user batch with index i

Bibliography

[1] Michel Abdalla, Yuval Shavitt, and Avishai Wool. Key management for restricted multicast using broadcast encryption. *IEEE/ACM Trans. Netw.*, 8(4):443–454, 2000.

[2] Jérôme Adda and Marco Ottaviani. The transition to digital television. *Economic Policy*, 20(41):160–209, 2005.

[3] André Adelsbach and Ulrich Greveler. A broadcast encryption scheme with free-riders but unconditional security. *Safavi-Naini, Reihaneh; Yung, Moti (Eds.): Proceedings of the First International Conference on Digital Rights Management, Sydney 2005, Lecture Notes in Computer Science Vol. 3919*, 2006.

[4] André Adelsbach and Jörg Schwenk. Key-assignment strategies for CPPM. In *Proceedings of the 2004 ACM Multimedia and Security Workshop*, pages 107–115, 2004.

[5] R. Anderson and M. Kuhn. Low cost attacks on tamper resistant devices. *IWSP: 5th International Workshop on Security Protocols, LNCS 1361, Springer-Verlag*, pages 125–136, 1997.

[6] Mark Armstrong. Competition in the Pay-TV market. *Journal of the Japanese and International Economies*, 13(4):257–280, 1999.

[7] N. Attrapadung, K. Kobara, and H. Imai. Key derivation patterns for broadcast encryption and key predistribution schemes. In *ASIACRYPT '03*, volume 2894, pages 374–391, 2003.

[8] H. Bauer. *Wahrscheinlichkeitstheorie.* de Gruyter, fifth edition, 1996.

[9] S. Berkovits. How to broadcast a secret. In *CRYPTO '91: Proceedings of the 11th Annual International Cryptology Conference on Advances in Cryptology*, pages 535–541. Springer-Verlag, 1991.

[10] Stephen Mooney Bill Rosenblatt, Bill Trippe. *Digital Rights Management: Business and Technology.* Wiley, first edition, November 2001.

[11] Rolf Blom. An optimal class of symmetric key generation systems. In *Advances in Cryptology - Eurocrypt '84*, pages 335–338. Springer-Verlag, 1984.

[12] C. Blundo, L. F. Mattos, and D. R. Stinson. Generalized Beimel-Chor schemes for broadcast encryption and interactive key distribution. *Theoretical Computer Science*, 200(1):313–334, 1998.

[13] Carlo Blundo, Alfredo De Santis, Amir Herzberg, Shay Kutten, Ugo Vaccaro, and Moti Yung. Perfectly-secure key distribution for dynamic conferences. In *CRYPTO '92: Proceedings of the 24th Annual International Cryptology Conference on Advances in Cryptology*, pages 471–486. Springer-Verlag, 1992.

[14] Dan Boneh and Alice Silverberg. Applications of multilinear forms to cryptography. *Contemporary Mathematics*, 324:71–90, 2003.

[15] Dan Boneh and Brent Waters. Collusion resistant broadcast encryption with short ciphertexts and private keys. IACR Cryptology ePrint Archive, Report 2005/018, 2005.

[16] David Brown. Pay-TV business planning. Technical Report www.im-reports.com/PTVBP, International Marketing Reports, June 2003.

[17] Ran Canetti, Juan Garay, Gene Itkis, Daniele Micciancio, Moni Naor, and Benny Pinkas. Multicast security: A taxonomy and some efficient constructions. In *INFOCOMM'99*, 1999.

[18] David B. Carroll. Proposal for a specific exemption from prohibition on circumvention of technological measures that control access to a certain class of copyrighted works. comment on the Digital Millennium Copyright Act. Technical report, December 2002.

[19] Martin Cave and Robert W Crandall. Sports rights and the broadcast industry. *Economic Journal*, 111(469):4–26, 2001.

[20] CENELEC. Common Interface specification for conditional access and other digital video broadcasting decoder applications. Technical Report EN 50221, Technical Committee TC 206, October 1997.

[21] Artem Chlegov. All about DVD part 2: The mess with formats. Online publication http://www.dvdsoftwareguide.com/all-about-dvd-2-guide.html, 2005.

[22] CinemaNow. CinemaNow was founded in 1999, and its investors include Menlo Ventures, Microsoft, Lions Gate Entertainment, Cisco Systems and Blockbuster. online, status of aug. 2005: http://www.cinemanow.com.

[23] Robert M. Corless. *Symbolic Recipes: Scientific Computing with Maple*. Springer, first edition, February 2005.

[24] Ivan B. Damgård. Practical and provably secure release of a secret and exchange of signatures. *Journal of Cryptology*, 8(4):201–222, November 1995.

[25] Peter H. Dana. Global Positioning System (GPS) time dissemination for real-time applications. *Real-Time systems*, pages 9–40, December 1997.

[26] Digital Video Broadcasting Project. Support for use of scrambling and conditional access within digital broadcasting systems. Technical report, February 1997.

[27] Digital Video Broadcasting Project. Digital Video Broadcasting (DVB); content protection and copy management part, DVB document a094, blue book consisting of three parts: SB 1496 DVB CPCM reference model, SB 1497 DVB CPCM usage state information, SB 1498 DVB CPCM abbreviations, definitions and terms. Technical report, 2005.

[28] Yevgeniy Dodis and Nelly Fazio. Public-key broadcast encryption for stateless receivers. In Joan Feigenbaum, editor, *ACM Workshop in Digital Rights Management—DRM 2002*, volume 2696 of *Lecture Notes in Computer Science*, pages 61–80. Springer-Verlag, 2003.

[29] L. Dutton, D. Rumens, W. Forrest, and L. Ruiz. Galileo's services. In *Proceedings of ION GPS 2002, Oregon, USA, September 24-27*, 2002.

[30] DVD Forum. DVD specifications for read only disk - part 1, physical specifications, version 1.01. Technical report, 1997.

[31] DVD Forum. DVD specifications for read only disk - part 2, file system specifications, version 1.01. Technical report, 1997.

[32] DVD Forum. DVD specifications for read only disk - part 3, video specifications, version 1.01. Technical report, 1997.

[33] DVD Forum. DVD video recording for rewritable and recordable discs - part 3, video recording, version 1.0. Technical report, 1997.

[34] J. Eberspächer and H. Vogel. *GSM - Global System for Mobile Communication.* Teubner, Stuttgart, second edition, 1999.

[35] Kim et al. MPEG-4 audio v.2. Technical Report JTC1/SC29/WG11/N2670, ISO/IEC,Audio Subgroup, March 1999.

[36] Amos Fiat and Moni Naor. Broadcast encryption. In *CRYPTO '93: Proceedings of the 13th Annual International Cryptology Conference on Advances in Cryptology*, pages 480–491. Springer-Verlag, 1993.

[37] Eran Gabber and Avishai Wool. How to prove where you are. *Proceedings of the 5th ACM Conference on Computer and Communications Security*, pages 142–149, November 1998.

[38] Juan A. Garay, Jessica Staddon, and Avishai Wool. Long-lived broadcast encryption. *Lecture Notes in Computer Science*, 1880:333–343, 2000.

[39] Oded Goldreich. *Modern Cryptography, Probabilistic Proofs and Pseudorandomness.* Springer-Verlag, 1998.

[40] S. Goldwasser and S. Micali. Probabilistic encryption. *Special issue of Journal of Computer and Systems Sciences*, 28(2):270–299, 1984.

[41] Shafi Goldwasser and Mihir Bellare. Lecture notes on cryptography. Summer Course “Cryptography and Computer Security” at MIT, 1996–1999, 1999.

[42] M. T. Goodrich, J. Z. Sun, and R. Tamassia. Efficient tree-based revocation in groups of low-state devices. In *CRYPTO ’04: Proceedings of the 24th Annual International Cryptology Conference on Advances in Cryptology*, pages 511–527. Springer-Verlag, 2004.

[43] Michael T. Goodrich, Jonathan Z. Sun, and Roberto Tamassia. Efficient tree-based revocation in groups of low-state devices. In Matthew K. Franklin, editor, *CRYPTO*, volume 3152 of *Lecture Notes in Computer Science*, pages 511–527. Springer-Verlag, 2004.

[44] Ulrich Greveler. How Pay-TV becomes E-Commerce. *Proceedings of the 7th International IEEE Conference on E-Commerce Technology 2005*, 2005.

[45] Ulrich Greveler. DRM für Multimedia-Broadcasts – wie sieht das PayTV der Zukunft aus? *Patrick Horster (Ed.), D.A.CH Security 2006, Düsseldorf, March 2006*, pages 260–267, 2006.

[46] Ulrich Greveler. Enforcing Regional DRM for Multimedia Broadcasts with and without Trusted Computing. *Safavi-Naini, Reihaneh; Yung, Moti (Eds.): Proceedings of the First International Conference on Digital Rights Management, Sydney 2005, Lecture Notes in Computer Science Vol. 3919*, 2006.

[47] Ulrich Greveler. Patentierung kryptographischer Verfahren, die an Hochschulen entwickelt wurden. *Proceedings of the GI Fachtagung Sicherheit 2006, Magdeburg, Feb. 2006*, pages 329–332, 2006.

[48] S. Guha and S. Khuller. Greedy strikes back: Improved facility location algorithms. In *SODA: ACM-SIAM Symposium on Discrete Algorithms (A Conference on Theoretical and Experimental Analysis of Discrete Algorithms)*, 1998.

[49] Dani Halevy and Adi Shamir. The LSD broadcast encryption scheme. In Moti Yung, editor, *CRYPTO*, volume 2442 of *Lecture Notes in Computer Science*, pages 47–60. Springer-Verlag, 2002.

[50] David Harbord and Marco Ottaviani. Contracts and competition in the Pay-TV market. Online publication http://citeseer.ist.psu.edu/472994.html, 2001.

[51] Ian Harris. Technical realization of short message service cell broadcast (SMSCB). Technical Report 3GPP TS 03.41, 3rd Generation Partnership Project (3GPP), June 1996.

[52] Hewlett-Packard and Mitsubishi and Philips Electronics et al. DVD+RW 4.7 GBytes, basic format specifications, version 1.1. Technical report, 2001.

[53] Hewlett-Packard and Mitsubishi and Philips Electronics et al. DVD+R 4.7 gbytes, basic format specifications, version 1.0. Technical report, 2002.

[54] Friedhelm Hillebrand. *GSM and UMTS - The Creation of Global Mobile Communication*. Wiley, first edition, January 2002.

[55] R. Hinden and S. Deering. IP version 6 addressing architecture. RFC 2373, IETF, July 1998.

[56] Jung Yeon Hwang, Dong Hoon Lee, and Jongin Lim. Generic transformation for scalable broadcast encryption schemes. In *CRYPTO '05: Proceedings of the 25th Annual International Cryptology Conference on Advances in Cryptology*, pages 276–292. Springer-Verlag, 2005.

[57] D. J. Iles. Operational DVB-T SFN experience in Australia. In *IBC online paper http://www.broadcastpapers.com/tvtran/tvtran.htm*, September 2003.

[58] Intel, IBM, Matsushita, and Toshiba. Content protection for prerecorded media specification, DVD book, revision 0.93. Technical report, 2001.

[59] Intel, IBM, Matsushita, and Toshiba. C2 block cipher specification, revision 1.0. Technical report, 2003.

[60] Intel, IBM, Matsushita, and Toshiba. Content protection for prerecorded media specification, introduction and common cryptographic elements, revision 1.0. Technical report, 2003.

[61] Intel, IBM, Matsushita, and Toshiba. Content protection for prerecorded media specification, SD Memory card book / SD-Video part, revision 0.95. Technical report, 2005.

[62] Inter-American Telecommunication Commission. A global digital TV standard for Latin America and the Caribbean. Information Document 459/04, Organization of American States, July 2004.

[63] Keith Hill Jan Bormans. MPEG-21 overview v.5. Technical Report JTC1/SC29/WG11/N5231, ISO/IEC, Requirements Group, October 2002.

[64] Nam-Su Jho, Jung Yeon Hwang, Jung Hee Cheon, Myung-Hwan Kim, Dong Hoon Lee, and Eun Sun Yoo. One-way chain based broadcast encryption scheme. In *Advances in Cryptology - Eurocrypt '05*, pages 559–574. Springer-Verlag, 2005.

[65] Dieter Jungnickel. *Graphen, Netzwerke und Algorithmen.* B.I. Wiss.-Verl., third edition, 1994.

[66] O. Kommerling and M. Kuhn. Design principles for tamper resistant smartcard processors. *Proceedings of the USENIX Workshop on Smartcard Technology*, pages 9–20, 1999.

[67] Ulrich Krengel. *Einführung in die Wahrscheinlichkeitstheorie und Statistik.* Vieweg, eight edition, 2005.

[68] Andreas Krügersen. Assignment report: Implementation of the biased sub-set broadcast encryption scheme, unpublished manuscript, practical course at the Chair of Network and Data Security, Ruhr-University Bochum, February 2006.

[69] R. Kumar, S. Rajagopalan, and A. Sahai. Coding constructions for blacklisting problems without computational assumptions. In *CRYPTO '99: Proceedings of the 19th Annual International Cryptology Conference on Advances in Cryptology*, pages 609–623. Springer-Verlag, 1999.

[70] Ravi Kumar, Sridhar Rajagopalan, and Amit Sahai. Coding constructions for blacklisting problems without computational assumptions. *Lecture Notes in Computer Science*, 1666:609–623, 1999.

[71] Klaus Kursawe and Christian Stüble. Improving end-user security and trustworthiness of TCG-platforms. In *Proceedings of the 33. GI-Fachtagung, Frankfurt, September 2003.* GI, 2003.

[72] J. Levy, M. Ford-Levy, and A. Levine. Broadcast television: Survivor in a sea of competition. Working Paper Series 37, Federal Communications Commission, 2002.

[73] Claudia Loebbecke and Marcia Falkenberg. Can Internet-based TV succeed? towards a sequential framework for market entry. In *Proceedings of the 15th Bled Electronic Commerce Conference eReality: Constructing the eEconomy, Bled, Slovenia, June 17 - 19, 2002*, 2002.

[74] Claudia Loebbecke and Matthias Fischer. Business opportunities and risks from Pay-TV piracy: The case of Europe. In *Proceedings of the Eleventh Americas Conference on Information Systems, Omaha, NE, USA August 11th-14th 2005*, pages 462–471, 2005.

[75] Michael Luby and Jessica Staddon. Combinatorial bounds for broadcast encryption. In *Advances in Cryptology - Eurocrypt '98*, pages 512–526. Springer-Verlag, 1998.

[76] Mark Massel. *Digital television, DVB-T COFDM and ATSC 8-VSB.* Digitaltvbooks.Com, first edition, October 2000.

[77] Ueli Maurer. Information-theoretic cryptography. In *CRYPTO '99: Proceedings of the 19th Annual International Cryptology Conference on Advances in Cryptology*, pages 1666–1676. Springer-Verlag, 1999.

[78] Microsoft. Microsoft Next-Generation Secure Computing Base - Technical FAQ. online, status of Oct 2005, http://www.microsoft.com/technet/archive/security/news/ngscb.mspx.

[79] Miodrag J. Mihaljevic. Key management schemes for stateless receivers based on time varying heterogeneous logical key hierarchy. In *ASIACRYPT '03*, volume 2894, pages 137–154, 2003.

[80] MovieLink. Movielink service is owned and operated by Movielink, LLC, a joint venture of Metro-Goldwyn-Mayer Studios, Paramount Pictures, Sony Pictures Entertainment, Universal Studios and Warner Bros. online, status of Aug. 2005: http://www.movielink.com.

[81] Sylvain-Victor Nahum and Philippe Stransky. Conditional access data decrypting system (united states patent application). Technical Report 20050254648, US Patent and Trademark Office, 2005.

[82] Dalit Naor, Moni Naor, and Jeffrey B. Lotspiech. Revocation and tracing schemes for stateless receivers. In *CRYPTO '01: Proceedings of the 21st Annual International Cryptology Conference on Advances in Cryptology*, pages 41–62. Springer-Verlag, 2001.

[83] Dalit Naor, Moni Naor, and Jeffrey B. Lotspiech. Revocation and tracing schemes for stateless receivers. In *Electronic Colloquium on Computational Complexity. Report No. 43 (Extended version of the Crypto '01 submission)*, 2002.

[84] Moni Naor and Benny Pinkas. Efficient trace and revoke schemes. In *FC '00: Proceedings of the 4th International Conference on Financial Cryptography*, pages 1–20. Springer-Verlag, 2001.

[85] Katia Obraczka. Multicast transport protocols: A survey and taxonomy. *IEEE Communications Magazine*, pages 94–102, January 1998.

[86] Olivier Onidi. Galileo is launched. In *Proceedings of ION GPS 2002, Oregon, USA, September 24-27*, 2002.

[87] Benny Pinkas. Efficient state updates for key management. In *Digital Rights Management Workshop*, pages 40–56, 2001.

[88] Hans-Jürgen Prömel and Angelika Steger. *The Steiner Tree Problem.* Vieweg, Braunschweig, first edition, 2002.

[89] F.J. Riggins and H.-S. Rhee. Toward a unified view of electronic commerce. *Communications of the ACM*, 41(10):88–95, 1976.

[90] Uwe Schöning. *Theoretische Informatik kurz gefasst.* BI Wissenschaftsverlag, first edition, January 1992.

[91] Adi Shamir. How to share a secret. *Commun. ACM*, 22(11):612–613, 1979.

[92] Adi Shamir. On the generation of cryptographically strong pseudo-random number sequences. In *ACM Trans. Comput. Sys.*, pages 38–44, 1983.

[93] Yuval Shavitt, Peter Winkler, and Avishai Wool. On the economics of multicasting. *Netnomics*, 6(1):1–20, 2004.

[94] Alan T. Sherman and David A. McGrew. Key establishment in large dynamic groups using one-way function trees (submitted may 20, 1998). *IEEE Transactions on Software Engineering*, 29(5):444–458, 2003.

[95] Frank A. Stevenson. Cryptanalysis of contents scrambling system. Online publication http://www.dvd-copy.com/news/cryptanalysis-of-contents-scrambling-system.htm, 1999.

[96] Jim Taylor. DVD Demystified – DVD Frequently Asked Questions. online: http://www.dvddemystified.com/dvdfaq.html. revision 2005-11, November 2005.

[97] TCG. Trusted Computing Group Homepage. online, status of Oct 2005, http://www.trustedcomputinggroup.org/.

[98] Trusted Computing Group. TCG TPM specification version 1.2 revision 85, TPM main, part 1, design principles. Technical report, Trusted Computing Group, February 2005.

[99] Trusted Computing Group. TCG TPM specification version 1.2 revision 85, TPM main, part 3, design principles. Technical report, Trusted Computing Group, February 2005.

[100] Christoph Wagner and Andreas Grünwald. DTV Update: Berlin area completes switch-over by mid 2003. *International Journal of Communications Law and Policy*, 7, 2003.

[101] Wallner, Harder, and Agee. Key management for multicast: Issues and architectures. RFC 2627, IETF, June 1999.

[102] Jon S. Warner and Roger G. Johnston. GPS spoofing countermeasures. Los Alamos research paper LAUR-03-6163. Technical report, December 2003.

[103] Chung Kei Wong, Mohamed Gouda, and Simon S. Lam. Secure group communications using key graphs. In *Proceedings ACM SIGCOMM '98, Vancouver, B.C.* ACM, 1998.

Curriculum Vitae

Personal Data

Ulrich Greveler
Hattinger Straße 768 a
44 879 Bochum, Germany

Tel.: (02 34) 4 14 94 05
E-Mail: ulrich.greveler@nds.rub.de
Web: http://www.nds.rub.de/greveler/

Born on September 16th, 1972
in Gießen, Germany

Education

1979–1992	School, Grammar School, degree *Abitur* at Liebigschule Gießen
1992–1998	Mathematics and Computer Science studies at Justus Liebig University Gießen, Germany
1996	Visiting Student, University of Reading, United Kingdom
1998	University degree: *Diplom-Mathematiker* (mathematician)

Professional Experience

06/1995–03/1998	Justus Liebig University Gießen, computer service center; **student worker** (part-time)
05/1998–08/2001	TÜV Informationstechnik GmbH, Essen, Germany; **consultant, project manager**

08/2001–03/2004	Smarttrust GmbH, München, Germany / Smarttrust AB, Stockholm, Sweden; **technical sales manager**
since 04/2004	Ruhr University Bochum, Faculty of Electrical Engineering and Information Technology; **researcher**

Hiermit erkläre ich, dass die eingereichte Dissertation von mir selbständig und ohne unerlaubte Hilfe ausgeführt und verfasst wurde und dass sie nicht in dieser oder ähnlicher Form früher bei dieser oder einer anderen in- oder ausländischen Hochschule als Dissertation eingereicht worden ist.

. . . .