MSc Thesis
Graduate Program in Logic, Algorithms and Computation
($\mu\Pi\lambda\forall$)

Encryption mechanisms for multiuser environments

Aikaterini Samari

Supervisor
Aggelos Kiayias

Athens 2012

Διπλωματική εργασία
Μεταπτυχιακό πρόγραμμα Λογικής και Θεωρίας
Αλγορίθμων και Υπολογισμού
($\mu\Pi\lambda\forall$)

Μηχανισμοί Κρυπτογράφησης για περιβάλλοντα πολλών χρηστών

Αικατερίνη Σάμαρη

Επιβλέπων
Αγγελος Κιαγιάς

Αθήνα 2012

Η παρούσα διπλωματική εργασία εκπονήθηκε στα πλαίσια των σπουδών για την απόκτη ση του Διαπανεπιστημιακού Μεταπτυχιακού προγράμματος στη Λογική και Θεωρία Αλγορίθμων και Υπολογισμού που απονέμει το Τμήμα Μαθηματικών της Σχολής Θετικών Επιστημών, του Εθνικού και Καποδιστριακού Πανεπιστημίου Αθηνών. Συγγραφέας είναι η φοιτήτρια Σάμαρη Αικατερίνη του Νικολάου με αριθμό μητρώου 200911. Επιβλέπων Καθηγητής είναι Άγγελος Κιαγιάς, Επίκουρος Καθηγητής του τμήματος Πληροφορικής και τηλεπικοινωνιών του ΕΚΠΑ. Τα υπόλοιπα δύο μέλη που απαρτίζουν την τριμελή επιτροπή είναι οι Βασίλειος Ζησιμόπουλος, Καθηγητής του τμήματος Πληροφορικής και τηλεπικοινωνικών του ΕΚΠΑ, και 'Αρης Παγουρτζής, Επίκουρος Καθηγητής της σχολής Ηλεκτρολόγων Μηχανικών και Μηχανικών Ηλεκτρονικών Υπολογιστών του ΕΜΠ. Η τριμελής επιτροπή ενέκρινε την παρούσα διπλωματική εργασία την 4η Απριλίου 2012.

This text is a thesis for the Inter-University, Inter-Departmental Program of Post-Graduate Studies in Logic and the Theory of Algorithms and Computation (MPLA). Samari Aikaterini with registry number 200911, is the author. The supervisor of this thesis is Aggelos Kiayias, Assistant Professor at the Department of Informatics and Telecommunications of the National and Kapodistrian University of Athens (UOA). Aggelos Kiayias is a member of the three-member committee. The other two members are Vassilis Zissimopoulos, Professor at the Department of Informatics and Telecommunications of the University of Athens, and Aris Pagourtzis, Assistant Professor at the School of Electrical and Computer Engineering of the National and Technical University of Athens (NTUA). The committee approved this thesis on the 4th of April 2012.

Acknowledgements

Concerning the contribution of Prof. Kiagias, I would like to point out first that during the preparation of this thesis he oriented me methodically inside the vast area of cryptography. Under his guidance I managed to get a more solid knowledge of the field, my interest in the field became deeper and permanent. Secondly, the present thesis became my initiation in research, as witnessed by the publication of parts of this work. This fortunate outcome, that strengthened my confidence and determination to continue working in the field, would not have occurred without his decisive interference in any difficulty that I faced. His support was beyond a conventional guidance of a Master's thesis, hence my thanks to him are essential and not just a formality required in this occasion.

I want to acknowledge collectively the creative environment of MPLA where this work has been done. Thanks to all staff members especially the faculty members and the graduate students of MPLA.

Special acknowledgements I would like to address to my friends, Vassilis Galanis, collaborator from the Department of informatics and Telecommunications and to Giorgos Stathopoulos, Iosif Salem, Nikolaos Karvelas and Yiannis Tselekounis, graduate students of MPLA for the support they have given me with thoughts we exchanged and ongoing discussions we had.

To Professor Kirousis Eleftherios my undergraduate thesis supervisor at the Dept. Computer Engineering and Informatics at the University of Patras. Although we did not have any collaboration for the present work the subject of which is not related to my undergraduate thesis, his instructions for mathematical rigor and orderly study of each subject were particularly useful to me.

To the members of the examining committee for their seriousness and academic integrity on the consideration of my work and myself personally.

Contents

Chapter 1

Introduction

Technological development in digital media led most kinds of information to be increasingly produced, converted and distributed in digital form. Thus, the protection of digital information and the need for secure communication and privacy brought cryptographic techniques to the center of the process of creation and distribution of the digital information. This thesis focuses on several aspects related to the use of encryption in the problem of transmitting messages to multiuser environments. Specifically, it will concentrate on issues related to a class of encryption schemes called Broadcast Encryption.

The need of broadcasting digital information to a group of entities simultaneously through a unique channel arises in many aspects of daily life such as watching TV, sending email with multiple recipients, social networks or registering a file that is supposed to be accessed by many users. But is it always desirable that every party who listens to a channel to be able to access the entire broadcast information? Would it be appropriate if everybody were able to access a registered file with sensitive information? What happens in case a sender wishes to exclude some entities of a certain population from receiving a message?

Evidently, a basic issue that arises in the area of distributing digital information via a broadcast channel with many listening parties is the need of controlling the list of potential recipients, since once a message is sent via a broadcast channel, every listening party can obtain it. As indicated above, this may be an undesirable effect for the sender as he may wish to hide a message from some recipients for some specific reasons or deny the access of another non-authorized party, called an eavesdropper, that could possibly listen to the broadcast information. Consider, for example, a file system which contains files with sensitive information that must not be viewed by the entire population of users that have access to the file system. These files must be encrypted in a way that can be accessed only by the authorized users. Another example could be satellite TV. The company intends each time only the customers who have paid their subscription to be able to access the signal. This implies that a mechanism that could revoke any time the non-paying customers is necessary.

To be more precise, the general problem that arises is how encryption could be used in order for a sender to be capable of choosing arbitrarily any subset of users to be excluded from a transmission and prepare a ciphertext which is suitable for decryption by the non-revoked ones. An obvious solution could be to deliver to each user a unique key and then encrypt the message separately under the key of each user. As a result, in each transmission, the number of broadcast ciphertexts will be equal to the number of enabled users which could be quite inefficient in the cases where this set is very large. Another trivial solution could be as follows: before each transmission the center generates a key which will be delivered only to the enabled users and then encrypt the message under this key. The drawback of this solution is that it requires private channels to the recipients. A center could also generate a large amount of keys, one for each possible subset of users of a certain population, and then depending on which subgroup is selected as the enabled set, use the corresponding key for encryption. Despite the fact that a unique ciphertext will be sent each time, the number of keys required to be stored per user increases dramatically as each user has to store as many keys as the possible subsets he belongs to.

Therefore, the challenge in designing broadcast encryption schemes is to go beyond trivial solutions and deals with discovering efficient methods that can be employed to solve this problem. But how efficiency can be measured in this setting? Efficiency in this setting is measured in terms of a variety of parameters such as the length of the broadcast ciphertext, the information that has to be stored per user in order for the decryption to be achieved, the computational time that has to be spent per user in order for the broadcast ciphertext to be decrypted. Another parameter that is taken into consideration is the computational time that a sender has to spend for producing a ciphertext available for decryption by a certain subset of users. As the above parameters are correlated to one another, the performance of a broadcast encryption method is measured in terms of the interaction between the aformentioned factors. A variety of proposed constructions for broadcast encryption with different performance trade offs will be described in this thesis.

Before getting into further details, due to the fact that the objective of the use of cryptography in any setting is to guarantee some security properties, it is necessary to investigate the conditions a broadcast encryption scheme must satisfy in order to be considered secure. In general, the definition of security in any cryptographic protocol constitutes a fundamental aspect since it cannot always be globally defined. The security of a protocol may be defined in different ways which could be evaluated as weaker or stronger or can even be incomparable. This obviously depends on the requirements each definition puts forth.

Primarily, the first thing that must be clear in order for the security properties of a cryptographic protocol to be defined is the objective of a potential attacker against this protocol, or, in other words, in which cases we will say that an attacker breaks the security of the protocol. To give an example, in the setting of broadcast encryption, a malicious entity could be a revoked user who wants to learn the message. Does it suffice to consider a scheme secure if no revoked user in a transmission is able to recover the message? What happens if a malicious user compromises another one or if a number of users collaborate in order to obtain a message that are not supposed to? Should a definition according to which no coalition of revoked users can recover the message be considered as stronger? Obviously, an ideal security definition is one that concurrently captures all the possible attacks that could take place by a malicious entity called an *adversary*. Consequently, in any cryptographic construction it has to be clearly defined the adversarial model that is considered.

Besides the security model that is considered, it is important to be clear which are the resources of the attackers. It is crucial to mention that a part of modern cryptography considers security with regard to "efficient" but "reasonable" adversaries. In this approach, attackers are viewed as polynomially time bounded algorithms. We will refer extensively to this approach later as it characterizes the whole thesis.

Overview in Broadcast Encryption

Fiat and Naor in [12] were the first who studied formally the context of broadcast encryption. They provided several schemes resilient to any coalition of users but the storage requirements per user and the storage requirements of the center are parametrized by the number of users they collude. Since then, a lot of work is spent on this area.

Naor et al. in [22] were the first who proposed efficient constructions where the values of performance measures do not depend on the number of users they collude. They introduced revocation schemes for the case of *stateless receivers*. In this case, receivers do not have to update their state during the lifetime of the system, i.e. all private information can be stored exclusively during the initialization process and consequently receivers cannot adapt their state according to the history of transmissions. These schemes are very important for applications such as CD's, DVD's, Blu-ray discs where devices are not online and it is still necessary for the sender to be able to revoke devices that are possibly compromised or used illegally. The authors introduce a definitional framework for broadcast encryption schemes called *Subset Cover Framework*. In this framework, the sender generates a number of subsets of the population of users and assigns a different key to each one of them. Thus, according to the choice of the revoked set the sender has to solve a set cover problem, i.e. to find a set of subsets that contain exclusively the non-revoked users. The authors propose two different constructions with different performance trade offs that apply to this framework, the Complete Subtree method and the Subset Difference method. Many broadcast encryption schemes have been proposed since then that lie on this framework([26], [3], [15], [16] etc.). Following the terminology of [19] we will call these schemes *combinatorial* broadcast encryption schemes.

It is worth mentioning that a variant of the Subset Difference method has been adopted in the AACS standard [1]. The Advanced Access Content System (AACS) is a standard for content distribution and digital rights management that has been adopted as the access restriction mechanism for HD-DVD and Blu-ray Disc.

Another class of schemes are those that are based on algebraic structures or, in other words, the category of *structured* broadcast encryption schemes. The characteristic of this category is that the key-space has a special structure that allows construction of ciphertexts that can be decrypted only by the enabled set. Such examples are schemes based on polynomial interpolation [23, 10], where the keys of users are points of a polynomial. This class also includes schemes that rely on bilinear maps such as [6] ,[8]. The use of such algebraic structures led to schemes that behave in a totally different way in comparison to combinatorial schemes. Constructions in [6] ,[8] managed to achieve ciphertexts and private keys of constant size. As far as we know, Delerablée [8] proposed the most efficient scheme for cases of small enabled sets in the setting of public key broadcast encryption, which is proven secure in the random oracle model.

Privacy in the context of Broadcast Encryption

It can be observed that a common characteristic in all the aforementioned broadcast encryption schemes, is that they reveal the enabled set either as a part of the broadcast ciphertext or as input of the decryption algorithm. This implies that anybody that has access to the broadcast ciphertext, even if not being able to recover the plaintext, can learn exactly the members of the enabled set. Protecting the privacy of the users in the enabled set can be an equally and sometimes perhaps even more important goal than the privacy of the message. Indeed, hiding the information that one is a recipient of a message from other users and even from other recipients of the same message, is a critical security feature in any setting where the fact of receiving a message at a certain time or frequency reveal sensitive personal characteristics of the recipient. For example, in a file system, an encrypted system file under a certain account may reveal that the said account has a certain level of system privileges and this fact can assist an attacker in a more complex attack vector.

The first that put forth the notion of privacy in the setting of broadcast encryption are Barth, Boneh and Waters in [4]. Their objective is to consider another class of attacks for broadcast encryption where the goal of the attacker is to discover information about the set of enabled users rather than decrypting a ciphertext for which it is not enabled. They propose the first schemes that preserve privacy for the setting of public key broadcast encryption. Their characteristic that each user holds a pair of public-secret key and one ciphertext is prepared for each enabled user.

Further work that deals with the feature of privacy(or anonymity) in the setting of broadcast encryption is included in ([21],[11]).

Our results

Motivated by the importance of privacy as a general feature, as part of the preparation of the present thesis we made further research related to this property for broadcast encryption schemes which highlights that privacy would impose a performance penalty. The results of this work constitute the main part of this thesis. We proceed to a classification of privacy notions, i.e. full privacy, "single-target" privacy and privacy among equal sets. We show the relation between these notions. Next, we provide lower bounds on the ciphertext length for private broadcast encryption schemes with respect to the stated privacy definitions. Our main result is an impossibility result that highlights the cost of privacy for *atomic broadcast encryption schemes* (which include the class of combinatorial schemes).

We prove that any atomic scheme that has sublinear ciphertext size to the number of enabled users is susceptible to an attack against privacy. More precisely, we show that the ciphertext length of any atomic scheme that satisfies privacy among equal sets has to be at least $s{\cdot}k$, where s is the cardinality of the set of enabled users and k the security parameter. Extending this result to the full privacy notion, we prove that the ciphertext length of any atomic scheme that satisfies full privacy has to be at least $n{\cdot}k$ where n is the cardinality of the set of all users.

Finally, concerning non-atomic schemes, we proceed showing a lower bound of $\Omega(n + k)$ for all the possible enabled sets. This result concerns broadcast encryption schemes in general and is based on an *information-theoretic* argument.

The above results were submitted and accepted in the conference Information hiding, 2012 [20].

Organization of the thesis

The next chapter equips the reader with the necessary background of this work. A few definitions of mathematical terms are included in the section 2.1, while section 2.2 explains in more detail computational approach of security definitions. This logic is the basis of almost all the results that will be presented along the thesis. Chapter 3 deals with a variety of broadcast encryption schemes. First of all, a generalized formal definition for Broadcast Encryption is provided. Section 3.2 contains a formal definition for combinatorial broadcast encryption schemes together with the security requirements that they must satisfy. Two representative schemes of this category are also presented, the Complete Subtree and Subset Difference scheme. Section 3.4 elaborates on two constructions ([6] ,[8]) of the class of structured broadcast encryption schemes. According to our study of literature, these are the most efficient schemes of this category. Moreover, in section 3.3 we introduce the notion of *atomic* broadcast encryption schemes which could be viewed as a generalization of combinatorial broadcast encryption schemes.

The main part of this thesis is included in chapters 4 and 5. In these chapters we concentrate on privacy in the context of broadcast encryption. In section 4.1, we refer in detail to the work of [4] which is the first work that deals with this issue. In sections 4.2.1 and 4.2.2, we say a few words for [21],[11] respectively. Next, in section 4.3 we start presenting our results, providing at first a classification of privacy notions. Finally, in chapter 5, we proceed showing some lower bounds. Section 5.1 deals with lower bounds for atomic broadcast encryption schemes while in section 5.2 a lower bound for general broadcast encryption schemes is provided.

Chapter 2

Preliminaries

2.1 Mathematical Background

Definition 2.1. *We say that a function f is negligible if for all $c \in \mathbb{R}$ there exists $n_0 \in \mathbb{N}$ such that $f(n) \leq 1/n^c$, for all $n > n_0$.*

Definition 2.2 (Statistical Distance). *Let X, Y be two random variables distributed according to D_1, D_2 respectively, and let $V = X([\mathsf{D}_1]) \cup Y([\mathsf{D}_2])$. We define the statistical distance Δ by*

$$\Delta[X,Y] = \frac{1}{2}\sum_{u \in V}\left|\underset{X \leftarrow \mathsf{D}_1}{\mathsf{Prob}}[X = u] - \underset{Y \leftarrow \mathsf{D}_2}{\mathsf{Prob}}[Y = u]\right|$$

Definition 2.3. *Let X, Y be two random variables over ensembles D_1 and D_2 respectively. We say D_1 and D_2 are statistically indistinguishable if $\Delta[X,Y]$ is a negligible function of n.*

Definition 2.4 (statistical test). *A statistical test $\mathcal{A}$ for an ensemble $\mathsf{D} = \{\mathsf{D}_n\}_{n \in \mathbb{N}}$ is an algorithm that takes input elements from D_n and outputs values 0 or 1 for each $n \in \mathbb{N}$.*

Definition 2.5. *Consider the statistical test $\mathcal{A}$ as a function of n and let X and Y be two random variables following the ensembles $\mathsf{D}_1, \mathsf{D}_2$ respectively. We define statistical distance with respect to $\mathcal{A}$ as*

$$\Delta_{\mathcal{A}}[X,Y] = \frac{1}{2}\sum_{u \in V}\left|\underset{X \leftarrow \mathsf{D}_1}{\mathsf{Prob}}[\mathcal{A}(X) = 1] - \underset{Y \leftarrow \mathsf{D}_2}{\mathsf{Prob}}[\mathcal{A}(Y) = 1]\right|.$$

Theorem 2.1. *For all statistical tests $\mathcal{A}$, it holds that $\Delta[X,Y] \geq \Delta_{\mathcal{A}}[X,Y]$.*

Definition 2.6. *Let $\Phi \subseteq 2^n$ be a set system with n participants. We say that Φ is fully exclusive over $[n]$, if for all subsets $\mathsf{R} \subseteq 2^n$, the set $[n] \setminus \mathsf{R}$ can be partitioned into a number of disjoint subsets that belong to Φ.*

Definition 2.7. *Let $\mathbb{G}$ and $\mathbb{G}_1$ be two multiplicative cyclic groups of prime order p and g is a generator of $\mathbb{G}$. A function $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_1$ is an admissible **bilinear map** if it satisfies the following properties*

- ***Bilinearity:****For all $u, v \in \mathbb{G}$ and $a, b \in \mathbb{Z}_p^*$, it holds that $e(u^a, v^b) = e(u,v)^{ab}$.*
- ***Non-Degeneracy:*** *For all non-zero $v \in \mathbb{G}$, it holds that $e(v,v) \neq 1$.*
- ***Computability:*** *There is an efficient algorithm that computes e.*

2.2 Computational approach in Modern Cryptography

Following the book of Katz and Lindell [18], we will provide a basic idea on the computational approach for Security in Modern Cryptography. This approach combines in a very interesting way complexity theory with cryptographic techniques. Its characteristic is the fact that it does not require encryption schemes to be resilient to attacks of adversaries with unlimited computational time. However, this approach considers as secure encryption schemes that can resist attacks of efficient but realistic adversaries. In addition, it still leaves room for success from the part of the adversary, but with so small probability that success is unlikely to take place.

More precisely, at the stage of initialization of an encryption scheme, where a key generation algorithm takes place, the security parameter of the scheme is selected. The security parameter is an integer λ that represents the key-length and constitutes a determinant for the security of a cryptographic construction. The value of the security parameter is public and as a result the adversary knows it. The computational time of the adversary as well as the probability of success or failure is measured in terms of this parameter. Based on that, we proceed explaining more formally how adversaries are modeled in this approach.

- Any adversary is considered as an algorithm which runs in *probabilistic polynomial time* with respect to the value λ.
- The success probability of an adversary in order for a scheme to be secure must be a negligible function of λ (definition 2.1).

Therefore, a high-level definition of the security of any cryptographic construction is the following.

A scheme is secure if all *probabilistic polynomial time*(PPT) adversaries have advantage in breaking the scheme with probability at most negligible in λ.

For any cryptographic construction, any security definition has to make clear the type of attacks that aims to address. Furthermore, depending on the scheme and the security definition, the adversary may also have additional power. This power may arise from access to some resources, depending on the instantiation. For example, in Chosen Plaintexts Attacks, the adversary may obtain ciphertexts that correspond to messages of his choice, while in the case of Chosen Ciphertexts Attackers, an adversary might additionally to observe the behaviour of the decryption function on chosen ciphertexts.

Having settled with some the general principles that characterize security definitions, let us turn our attention to the conditions under which cryptographic constructions are considered secure. The security of cryptographic schemes relies upon computationally hard problems, i.e. problems that are not believed to be solvable in polynomial time. Two of the most common assumptions that have become the guarantee of security for plenty of schemes are the Discrete-Log assumption and the RSA assumption. As a hard problem constitutes a necessary condition for preserving security of a particular scheme, we will say that

if a hard problem A holds, then the scheme is secure according to a definition B.

However, in more complex constructions where various cryptographic techniques incorporate, this statement becomes slightly different. The security of the construction relies on the security of underlying schemes which in turn are secure under hard problems.

As the aforementioned statement implies, proofs by contradiction can be employed in order for security proofs to be provided. This argument will proceed as follows:

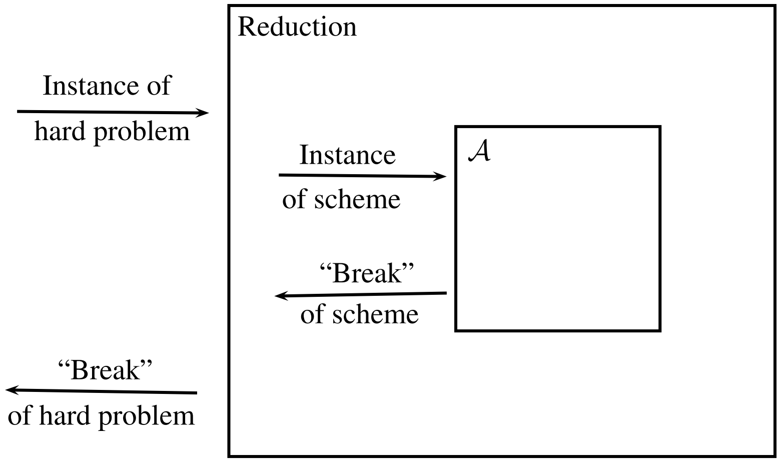

Figure 2.1: Reductions

> if there exists a PPT adversary $\mathcal{A}$ that breaks the security of the scheme, then there exists an algorithm that solves the hard problem is polynomial time.

These proofs in cryptography are called *proofs by reduction*. The goal of such a proof is to find a way to use an algorithm that breaks the security of a scheme in order to construct an algorithm that solves a particular hard problem in polynomial time. The algorithm that aims to solve the underlying hard problem is called a *reduction*. The figure 2.1 describes abstractly how a reduction operates.

In fact, the approach that described above is called *asymptotic* approach according to [18]. This is the approach that we will adopt in our results. Apart from this approach, there exists another part of computational approach the authors call *concrete* approach. This approach places bounds on the security parameters of a scheme and the running time of the adversary. Namely, security according to this approach is defined, in general, as follows:

> We say that a scheme is (t, ε)-secure if all t-time adversaries succeeds in breaking the scheme with probability at most ε.

It can be easily observed that if ε is a negligible function of λ and t is a polynomial function in λ, then both approaches coincide.

Security definitions according to both approaches will be met in this thesis.

Security definitions formalized as experiments

A widely applicable model of security definitions which captures most of the security definitions included in this thesis, is an interactive procedure with two parties denoted as the *adversary* and the *challenger*. The challenger can be viewed as an honest party that has full access to the secret parameters of the cryptographic scheme and thus can answer the queries of the adversary according to the restrictions placed by the game. A high level overview of such a definition could be the following:

Experiment:

1. **Initialization Stage:** Initialization of the public-secret parameters of the system. The public parameters of the system are given to the adversary

2. **Training stage:** The adversary can issue a number of queries to the oracles he may be provided.

3. **Challenge Phase:** The adversary outputs a challenge (e.g. a message or a set) and intends to receive an answer for the challenger. The answer of the adversary depends on the result of a toss of a fair coin. According to this result, the challenger prepares the answer.

4. Step 2 may be repeated.

5. **Guess:** The adversary has to guess the result of the coin toss.

6. **Answer:** If the output of the adversary coincides with the result of the coin, then the experiment outputs 1, else it outputs 0.

The security definition follows.

A scheme is secure if all PPT adversaries can guess the correct bit with probability higher than 1/2 only by a negligible function in the security parameter.

Thus, for this type of definitions, in the security proof the reduction plays the role of the challenger. This means that the reduction has to simulate perfectly the experiment in a beneficial way so as to break a hard problem with non-negligible advantage.

Chapter 3

Broadcast Encryption

3.1 Definition of Broadcast Encryption

We provide a formal definition for broadcast encryption.

Definition 3.1. *Let* K *be a key space,* M *a plaintext space and* C *ciphertext space. A broadcast encryption scheme with* n *receivers and security parameter* λ*, is defined as a tuple of algorithms* $\langle\mathsf{KeyGen}, \mathsf{Encrypt}, \mathsf{Decrypt}\rangle$*:*

- KeyGen: On input 1^n and 1^λ, it generates the set of keys $(ek, sk_1, ..., sk_n)$, where ek is the encryption key and sk_i is the decryption key assigned to a user i. It produces a language $\mathcal{L}$ which encodes the set of all possible revocation instructions to be provided as input to the encryption function.

- $\mathsf{Encrypt}$: On input a message $m \in \mathsf{M}$, the encryption key ek and a revocation instruction $\psi \in \mathcal{L}$, it outputs a ciphertext $c \in \mathsf{C}$ such that $c \leftarrow \mathsf{Encrypt}(ek, m, \psi)$.

- $\mathsf{Decrypt}$: On input a ciphertext c, such that $c \leftarrow \mathsf{Encrypt}(ek, m, \psi)$, and a decryption key sk_i assigned to the user i according to the algorithm KeyGen it outputs either m, the symbol $\perp$ which represents the failure or another string depending on the instantiation.

The above definition is general, as it does not aim to refer to a particular scheme or a class of schemes. Furthermore, due to the presence of no restrictions in the language $\mathcal{L}$, it does not only capture broadcast encryption schemes that are fully exclusive (definition 2.6). Namely, given a particular broadcast encryption scheme, the revocation capability of a sender depends on which subsets $\mathsf{R} \subseteq N$ are encoded in the language $\mathcal{L}$. In the case of fully exclusive schemes, the language $\mathcal{L}$ contains the encodings of all the subsets $\mathsf{R} \subseteq N$.

A necessary property that must be satisfied by a broadcast encryption scheme is correctness. This property guarantees that each enabled receiver is capable of recovering the plaintext in possession of the broadcast ciphertext and his private key, while at the same time no excluded user has such capability. This requirement is compacted in the definition below:

Definition 3.2 (Correctness). *We say that a broadcast encryption scheme is correct if for any* $\psi \in \mathcal{L}$ *that encodes a subset* $\mathsf{R} \subseteq N$ *and for all* $m \in \mathsf{M}$*, it holds that for any* $u \in [n] \setminus \mathsf{R}$

$$\mathsf{Prob}[\mathsf{Decrypt}(\mathsf{Encrypt}(ek, m, \psi), sk_u) = m] = 1,$$

where $(ek, sk_1, ..., sk_n)$ *is distributed according to* $\mathsf{KeyGen}(1^n)$.

3.2 Combinatorial Broadcast Encryption Schemes

A formal definition of a combinatorial broadcast encryption scheme Φ with $[n] = \{1, 2, ..., n\}$ receivers as a triple of algorithms $\langle \mathsf{KeyGen}, \mathsf{Encrypt}, \mathsf{Decrypt} \rangle$ is provided below.

- KeyGen : On input $1^n, 1^\lambda$:
  - Choose a fully exclusive broadcast encryption scheme $\Phi = \{S_j\}_{j \in \mathcal{J}}$, where $\mathcal{J}$ is the set of all the indices of the elements of Φ.
  - Generate a collection of keys $\{k_j\}_{j \in \mathcal{J}} \subseteq \mathsf{K}$ suitable for a symmetric encryption scheme $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$.
  - For any user $u \in [n]$, define the sets $\mathcal{J}_u = \{j | i \in S_j\}$ and $\mathsf{SK}_u = \{k_j | j \in \mathcal{J}_u\}$. Set $ek = \langle \Phi, \{k_j\}_{j \in \mathcal{J}} \rangle$.
- Encrypt: On input a revoked set $\mathsf{R} \subseteq [n]$, encoded in the language $\mathcal{L}$, and a plaintext $m \in \mathsf{M}$:
  - Run a subset cover algorithm which produces a collection of disjoint subsets $S = \{S_{j_1}, S_{j_2}, ..., S_{j_s}\}$ such that $\bigcup_{j=1}^{k} S_{j_\ell} = [n] \setminus \mathsf{R}$.
  - Select the set of keys $\{k_j | j \in \mathcal{J}$ and $S_j \in S\}$.
  - Employing the underlying encryption scheme $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$, compute the ciphertext $c = \langle j_1, ..., j_s, \mathsf{Enc}_{k_{j_1}}(m), ..., \mathsf{Enc}_{k_{j_s}}(m) \rangle$.
- Decrypt: On input a ciphertext c and $\mathcal{J}_u$, K_u:
  - Search if there exists in the ciphertext c an encoding j_i such that $j_i \in \mathcal{J}_u$. If there exist such an encoding, j_i, compute $\mathsf{D}_{k_{j_i}}(c_i)$ otherwise return $\perp$.

In fact, combinatorial broadcast encryption schemes are those that rely on the Subset Cover Framework introduced in [22]. We note that the message m used in the definition above represents a symmetric cryptographic key. This key constitutes a session key which is then used to encrypt the appropriate information. As a result, the users that are supposed to obtain this key are also able to decrypt this information.

In this section, we present two specific constructions of combinatorial broadcast encryption schemes investigated in [22]. The interesting thing is that both schemes rely on the complete binary tree structure in a sense that each receiver can be viewed as a leaf of a complete binary tree. On the other hand, they differ at the collection of the subsets which is the characteristic that makes these methods completely different. These constructions will be presented in detail in the subsections 3.2.2, 3.2.3.

3.2.1 Security requirements

The objective of a broadcast encryption scheme is to be able to carry securely a cryptographic key which will be then used as a session key to encrypt messages. As indicated in the definition of combinatorial broadcast encryption schemes, a basic ingredient of their structure is the underlying symmetric encryption scheme. Thus, it is necessary for the underlying encryption scheme to be able to encrypt the session key securely . An encryption scheme that forms a *Key Encapsulation Mechanism*(KEM) achieves this goal. The notion of KEM is introduced by Shoup in [25] for the context of public key encryption. In the setting of symmetric key encryption it can be formalized as a security game between a challenger and an adversary as follows (cf. [19]).

The challenger chooses randomly a key from a key-space K. The adversary, having no knowledge about the key, can obtain encrypted messages of its choice under this key asking an Encryption Oracle. He can also observe how ciphertexts are decrypted under this key making queries to a Decryption Oracle. The challenger chooses two messages and encrypts one of them under this key. Then, flipping a coin he chooses which message will return to the adversary together with the produced ciphertext. The scheme will be considered KEM-secure if the adversary has negligible advantage in distinguishing whether the plaintext-ciphertext pair is valid or not. After the receipt of the challenge, the adversary can issue only encryption queries which means that CCA-1 security is considered.

```
Experiment $\mathsf{Exp}^{kem}_{\mathcal{A}}(1^\lambda)$
  Select $k$ at random.
  $aux \leftarrow \mathcal{A}^{\mathsf{Enc}_k(\cdot),\mathsf{Dec}_k(\cdot)}$
  $m_0, m_1 \overset{r}{\leftarrow} \mathsf{M}$;
  $b \overset{r}{\leftarrow} \{0,1\}; c \leftarrow \mathsf{Enc}_k(m_1)$
  $b^* \leftarrow \mathcal{A}^{\mathsf{Enc}_k(\cdot)}(m_b, c)$
  if $b = b^*$ then return $1$ else $0$;
```

Definition 3.3. *Suppose that* $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *is a symmetric encryption scheme. We say that* $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *is* ε*-insecure if for any PPT adversary* $\mathcal{A}$ *ih holds that*

$$\left|\mathsf{Prob}[\mathsf{Exp}^{kem}_{\mathcal{A}}(1^\lambda) = 1] - \frac{1}{2}\right| \leq \varepsilon.$$

An important additional requirement for the security of a combinatorial broadcast encryption scheme is that the keys should be generated in a way that any user in possession of his keys not be able to gain any information about another key, i.e. a key that corresponds to a subset he is not a member of. The key-indististinguishability property puts forth an even stronger requirement. No coalition of users should be able to learn anything about a key that corresponds to a set they do not belong to. More precisely, the essence of this property in the following definition is that any coalition of users cannot distinguish a key associated with a subset they do not belong to from a randomly chosen key.

The key-indistinguishability property is defined with the aid of a security game between a challenger and an adversary. The adversary outputs the index of a set j_0. The challenger flipping a coin chooses whether the j_0-th key will be chosen independently at random or not. The adversary is given the keys of all users that are not members of S_{j_0} and furthermore he is able to issue encryption and decryption queries for messages of his own choice under the key k_{j_0}. We will say that a broadcast encryption scheme satisfies key-indinstinguishability if for every j_0, all polynomially bounded adversaries have negligible probability in winning the following game.

$\mathsf{EncryptionOracle}(m, j)$	$\mathsf{DecryptionOracle}(c, j)$
$retrieve\ k_j, j_0$	$retrieve\ k_j, j_0$
$return\ c \leftarrow \mathsf{Enc}_{k_j}(m)$	$return\ D_{k_j}(c)$

```
Experiment $\mathsf{Exp}_{\mathcal{A}}^{key-ind}(1^\lambda, 1^n)$
  $j_0 \leftarrow \mathcal{A}(\Phi)$
  if $b = 0$ then $(\Phi, \{k_j\}_{j\in\mathcal{J}}) \leftarrow \mathsf{KeyGen}(1^n)$
  else $(\Phi, \{k_j\}_{j\in\mathcal{J}}) \leftarrow \mathsf{KeyGen}^{j_0}(1^n)$
  $b^* \leftarrow \mathcal{A}^{\mathsf{EncryptionOracle}(\cdot),\mathsf{DecryptionOracle}(\cdot)}((\langle \mathcal{J}_u, \mathsf{K}_u\rangle_{u\notin S_{j_0}})$
  if $b = b^*$ then return 1 else 0;
```

Definition 3.4 (key-indinstinguishability). *We say that a broadcast encryption scheme satisfies the key-indistinguishability property with distinguishing probability ε, if there exists a family of key generation procedures $\{\mathsf{KeyGen}^j\}_{j\in\mathcal{J}}$ with the property that for all j, and for all PPT adversaries $\mathcal{A}$ it holds that*

$$\left|\mathsf{Prob}[\mathsf{Exp}_{\mathcal{A}}^{key-ind}(1^\lambda, 1^n) = 1] - \frac{1}{2}\right| \leq \varepsilon.$$

Now, it can be shown (theorem 3.1) that if a combinatorial broadcast encryption scheme satisfies the key indistinguishability property and the underlying encryption scheme is KEM-secure then the broadcast encryption scheme is secure in the KEM-sense. The KEM security for broadcast encryption schemes is formalized below.

$\mathsf{EncryptionOracle}(m, \mathsf{R})$	$\mathsf{CorruptOracle}(u)$	$\mathsf{DecryptionOracle}(u, c)$
$retrieve\ ek$	$\mathsf{T} \leftarrow \mathsf{T} \cup \{u\}$	$\mathsf{D} \leftarrow \mathsf{D} \cup \{(u, c)\}$
$c \leftarrow \mathsf{Encrypt}(ek, m, \mathsf{R})$	$return\ \mathsf{K}_u$	$return\ x \in \{0, 1\}$
$return\ c$		

```
Experiment $\mathsf{Exp}_{\mathcal{A}}^{BE-kem}(1^\lambda, 1^n)$
  $(ek, (\mathcal{J}_1, \mathsf{K}_1), ..., (\mathcal{J}_n, \mathsf{K}_n)) \leftarrow \mathsf{KeyGen}(1^n)$
  $\mathsf{T} \leftarrow \emptyset$
  $(state, \mathsf{R}) \leftarrow \mathcal{A}^{\mathsf{CorruptOracle}(\cdot),\mathsf{EncryptionOracle}(\cdot),\mathsf{DecryptionOracle}(\cdot)}(1^n)$
  $m_0, m_1 \overset{r}{\leftarrow} \mathsf{M}; b \overset{r}{\leftarrow} \{0, 1\}$
  $c^* \leftarrow \mathsf{Encrypt}(ek, m_1, \mathsf{R})$
  $b^* \leftarrow \mathcal{A}^{\mathsf{EncryptionOracle}(\cdot)}(guess, (c^*, m_b), state)$
  if $\exists i \in \mathsf{T}$ such that $i \notin \mathsf{R}$ then output a random bit else if $b = b^*$ then return 1 else 0;
```

Definition 3.5. *Let Φ be a combinatorial broadcast encryption scheme with n receivers. We say that Φ is ε-insecure if for all PPT adversaries $\mathcal{A}$*

$$\left|\mathsf{Prob}[\mathsf{Exp}_{\mathcal{A}}^{BE-kem}(1^\lambda, 1^n) = 1] - \frac{1}{2}\right| \leq \varepsilon.$$

Theorem 3.1. *Let Φ be a combinatorial broadcast encryption scheme with n receivers that satisfies key-indistinguishability property with distinguishing probability ε_1 and the underlying encryption scheme* $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *is ε_2-insecure in the sense of definition 3.3. Then, Φ is ε-insecure in the sense of definition 3.5, with $\varepsilon \leq 2n \cdot |\Phi| \cdot (2\varepsilon_1 + \varepsilon_2)$.*

The proof of the theorem is provided in the chapter 2 of the book [19].

3.2.2 Complete Subtree

We describe the first combinatorial construction proposed in [22], the Complete Subtree method for broadcast encryption. We denote this scheme as Φ^{CS}. Intuitively, imagine a complete binary tree with n leaves and suppose that the objective is to construct a broadcast encryption scheme with n receivers. A Complete Subtree scheme with n receivers can be viewed as a set system constructed recursively as follows. Primarily, a single user is assigned to a single leaf and the singletons that contain all the leaves are added to Φ^{CS}. Then, for each internal node of the tree, a set that includes all the leaves that of the subtree rooted from this node is added to Φ^{CS}. Without loss of generality, we assume that the number of users is a power of 2. A complete subtree set system with n receivers contains $2n-1$ subsets. The next paragraphs provide an investigation of this scheme by first describing the algorithms $\langle \mathsf{KeyGen}, \mathsf{Encrypt}, \mathsf{Decrypt} \rangle$ specified for this method and subsequently examining its performance in terms of the measures of efficiency put forward.

- KeyGen: On input 1^n, 1^λ:
  - The set of encodings $\mathcal{J}^{CS}$ is the set of all possible binary strings of length at most $\log n$. Each encoding $j \in \mathcal{J}^{CS}$ corresponds to an index of a node of the complete binary tree. The indices of the nodes are produced recursively in a top-down manner: the index of the root is the empty string ϵ, an index of a left child is constructed by appending '0' to its parent index, while an index of a right child is constructed by appending '1' to its parent index. As a result, the indices of the leaves are of length $\log n$. For simplicity, we will refer to the index of a leaf corresponds to a user or a user itself as to be the same thing.
  - A collection of keys $\{k_j\}_{j \in \mathcal{J}^{CS}} \subseteq \mathsf{K}$ suitable for a symmetric encryption scheme $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ is generated.
  - Each user u belongs to all the subsets that correspond to the nodes that are located on the path from the leaf corresponds to u to the root. Considering both the above notice and algorithm that constructs the indices, it holds that for any user u, $\mathcal{J}_u = \{[u]_\ell | \forall \ell \in \{0, 1, .., \log n\}$ and $\mathsf{SK}_u = \{k_j | j \in \mathcal{J}_u\}$. With $[u]_\ell$ we denote the substring that consists of the ℓ most significant bits of the string u.
- $\mathsf{Encrypt}$: On input a set $\mathsf{R} \subseteq [n]$ and a message $m \in \mathsf{M}$:
  - Run the Algorithm 1 which produces a subset of indices of the set $\mathcal{J}$, e.g. $\{j_1, ..., j_s\}$. Employing the underlying encryption scheme $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$, for all $j_i \in \{j_1, ..., j_s\}$ compute $\mathsf{Enc}_{k_{j_i}}(m)$.
  - Output $c = \langle j_1, ..., j_s, \mathsf{Enc}_{k_{j_1}}, ..., \mathsf{Enc}_{k_{j_s}} \rangle$.
- $\mathsf{Decrypt}$: On input a ciphertext tuple $c = \langle j_1, ..., j_s, c_1, ..., c_s \rangle$ and $(\mathcal{J}_u, \mathsf{SK}_u)$:
  - Check whether there exists a j_i in $\{j_1, ..., j_s\}$ that is a prefix of u, considering that u is represented as a bitstring of length $\log n$. If there exists such j_i, compute $\mathsf{Dec}_{k_{j_i}}(c_{j_i})$ otherwise return $\perp$.

Subset Cover algorithm for the Complete subtree set system

The subset cover algorithm for this set system is based on the idea of computing the steiner tree of the leaves that correspond to users in the revoked set R. Recall that the steiner tree of a set of leaves R, Steiner(R), is defined as the minimal subtree that connects all the leaves in the set R. Also, we say that a node of the complete binary tree is "hanging" from a steiner tree, Steiner(R), if its sibling-node is located in the steiner tree whereas it is not. Hence, given a set of revoked users R, an algorithm that discovers the subset cover of the set $[n] \setminus \mathsf{R}$ proceeds as follows.

Algorithm 1 subsetcover(n,R)

```
Construct a complete binary tree with $n$ leaves.
Find the steiner tree of the leaves of R, STEINER(R).
Output the indices of the nodes that are hanging form R.
```

It is necessary to prove that the above algorithm is *correct* in the sense that it produces a set of disjoint subsets that cover the enabled set of users, $[n] \setminus \mathsf{R}$. Consequently no revoked user can recover the transmitted message. Obviously, the correctness of the subsetcover algorithm guarantees the correctness of the scheme (definition 3.2). The next theorem captures the requirements that must be satisfied by a subset cover algorithm in order to be correct.

Theorem 3.2 (Correctness). *The subset cover algorithm is correct if for all* $\mathsf{R} \subseteq 2^{[n]}$

1. *Each leaf* $u \notin \mathsf{R}$ *exists in exactly one subset in the result of the algorithm* subsetcover.

2. *Each leaf* $u \in \mathsf{R}$ *does not exist to any subset of the resulted partition.*

Proof. The correctness property derives explicitly from the subset cover algorithm. More precisely, if $u \notin \mathsf{R}$ then there is a node on the path from the leaf u to the root that hangs from the steiner tree STEINER(R). On the other side, if $u \in \mathsf{R}$ there does not exist such a node because the path from u to the root is part of the steiner tree. ■

Theorem 3.3. *If we assume that* r *is the size of the revoked set* R, *where* $0 < r < n$, *then the cover size resulted from the subset cover algorithm has size at most* $r \log(n/r)$.

3.2.3 Subset Difference

Similarly to the previous scheme, the subset difference scheme, Φ^{SD}, defined over a collection of sets, relies on the complete binary tree structure in which each leaf corresponds to one user. In this case, each set is encoded as pair of nodes (v_i, v_j) with v_i being an ancestor of v_j which means that the set contains all the leaves-users that are children of v_i excluding those that are children of the node v_j. Thus, the notation used for each set is $S_{i,j}$ given that this is represented with the pair (v_i, v_j).

Now, we will describe the algorithm that finds the subset cover of a set $[n] \setminus \mathsf{R}$, given that R is the revoked set.

subsetcover: On input n, R:

1. Set $T = \mathsf{STEINER(R)}$.

2. Find a node v in the tree T that has two children v_ℓ and v_r with each one being an ancestor of a single leaf. Denote the leaf that is a descendant of v_ℓ as v_i and the leaf that is a descendant of

v_r as v_j . If no such node exists, i.e., there is only one leaf left in the tree, then set $v_i = v_j$ to the leaf, set v to be the root and $v_\ell = v_r = v$.

3. If $v_\ell = v_i$, then add the subset S_{j_1} with encoding $j_1 = (v_\ell, v_i)$ to the broadcast pattern. Likewise, if $v_r = v_j$, then add the subset S_{j_2} with encoding $j_2 = (v_r, v_j)$ to the broadcast pattern.
4. Remove from T all the descendants of v and make v a leaf. Set T to be the tree resulted from this procedure and repeat the step 1. The algorithm will end when the tree T consists of a single node, the root of the complete binary tree.

Theorem 3.4. *For any set* $\mathsf{R} \subseteq [n]$, *the above algorithm partitions the set* $[n] \setminus \mathsf{R}$ *into* $2r - 1$ *subsets, assuming that* r *is the cardinality of the set* R.

3.2.4 Remarks

The following table represents the precise performance of both schemes.

	Key-storage/user	Ciphertext length	Computation time	Decryption op.
Complete Subtree	$\log n + 1$	$r \log(n/r)$	$O(\log \log n)$	1
Subset Difference	$1/2 \cdot \log^2 n$	$2r - 1$	$O(\log n)$	1

Considering the performance trade-offs of both schemes, we observe that they can be very efficient in cases where a sender wishes to broadcast a message to a large set of users. Specifically, the Subset Difference method is more efficient in terms of the ciphertext length as it gets rids of the $\log N$ factor that appears in the ciphertext size in the complete subtree method. Based on the structrure of this method, Halevy and Shamir in [16] introduce a method called "Layered Subset Difference"(LSD) which also achieves ciphertext length $O(r)$ improving at the same time the key storage required per user to $O(\log^{1+\epsilon} n)$, for $\epsilon > 0$ being any fixed constant. Goodrich et al. in [15], based on the same scheme but using different key-handling techniques, reduce the key storage to $O(\log n)$. These schemes constitute the most efficient schemes in the class of combinatorial broadcast encryption schemes.

Moreover, Dodis and Fazio [9] extended these methods(Complete Subtree, Subset Difference, Layered Subset Difference) to the setting of public key broadcast encryption. They take advantage of the priviledges of Hierarchical Identity Based Encryption (HIBE) [14, 17] in order to achieve public key size and storage of the Center $O(1)$. In the CS method the ciphertext consists of $r \log(N/r)$ identity based encryptions, each user stores $O(\log N)$ keys and needs to perform a single identity based decryption. In the SD method, the ciphertext consists of $(2r-1)$ hierarchical identity based encryptions, each user stores $O(\log^2 N)$ keys and needs to perform a single hierarchical identity based decryption.

3.3 Atomic Broadcast Encryption Schemes

We introduce a new class of schemes called atomic broadcast encryption schemes. Atomic schemes have the characteristic that the ciphertext can be broken into a number of discrete components and each recipient when decrypting it applies a decryption function to one or more of those components. A formal definition is provided above.

Definition 3.6. *An atomic broadcast encryption scheme with n receivers is defined as a tuple of algorithms* $(\mathsf{KeyGen}, \mathsf{Encrypt}, \mathsf{Decrypt})$:

- KeyGen: On input $1^n, 1^\lambda$, it generates the set of keys $(ek, \mathsf{SK}_1, ..., \mathsf{SK}_n)$, where ek is the encryption key and SK_i is the decryption key assigned to a user i. Each decryption key SK_i is a set which consists of elements $\{sk_{ij}\}_{j=1}^{\ell}$ (we call those atomic keys) for some value ℓ which is not necessarily the same for each user. It also produces the description of a language $\mathcal{L}$ which encodes all the possible subsets of users that may be provided as input to the encryption function.

- $\mathsf{Encrypt}$: On input a message m, the encryption key ek and a revocation instruction $\mathsf{R} \in \mathcal{L}$, it outputs a ciphertext C such that $C \leftarrow \mathsf{Encrypt}(ek, m, \mathsf{R})$ which, among possibly other values, contains a number of components $c_1, ..., c_\rho$ (we call those the atomic ciphertexts of C).

- $\mathsf{Decrypt}$: On input a ciphertext C, such that $C \leftarrow \mathsf{Encrypt}(ek, m, \mathsf{R})$ and a decryption key SK_i: It outputs m if $i \notin \mathsf{R}$ and some value $x \neq m$ if $i \in \mathsf{R}$. Depending on the instantiation, x could be the symbol $\perp$, or some plaintext sampled independently of m.

For atomic broadcast encryption schemes we further assume the existence of a deterministic algorithm called $\mathsf{Decryptmatching}$ which matches the atomic ciphertexts of a ciphertext tuple C with the atomic keys under which they are decrypted. In all cases we are aware of, this algorithm is part of the Decryption algorithm.

Proposition 1. *The broadcast encryption schemes that rely on the Subset Cover Framework [22] are atomic.*

Proof. A scheme that relies on the Subset Cover Framework can be viewed as an instantiation of an atomic scheme in the following way:

- Each atomic ciphertext is a ciphertext encrypted under the key related to a subset produced by the subset cover algorithm and each user decrypts a single ciphertext of the ciphertext tuple.

- The encodings of the subsets produced by the subset cover algorithm are placed at the beggining of each ciphertext as additional information.

- The decryption algorithm returns $\perp$ if the user that tries to decrypt is revoked.

■

3.4 Broadcast Encryption schemes based on algebraic structures

In section 3.2 we stated that all the broadcast encryption schemes that rely on the subset cover framework are not efficient in cases where the revoked set is large. Boneh, Gentry and Waters in [6] and Delerablée in [8], overcome this obstacle using pairings over elliptic curves. They provide constructions that achieve constant size ciphertexts and private keys for revoked sets of any size. These constructions apply to the setting of public key broadcast encryption and more precisely the second one [8] relies on Identity Based Broadcast Encryption. It is important to mention that the efficiency of these schemes takes place at the expense of the public key size which can be very large as we will see below in detail.

3.4.1 Collusion Resistant Broadcast Encryption With Short Ciphertexts and Private Keys

Boneh, Gentry and Waters [6] propose two constructions for the setting of public key broadcast encryption. The characteristic of the first construction is that the ciphertext and the private keys are of constant size while on the other side the public key has size linear to the entire population of receivers. The second construction is a generalized construction which achieves a performance trade-off between public key size and ciphertext size, while at the same time the size of the secret key remains constant. Primarily, let us define exactly the model of public key broadcast encryption considered in [6].

- Setup: On input 1^n, where n is the population of users λ the security parameter, it outputs a public key PK and n private keys $d_1, ..., d_n$.
- $\mathsf{Encrypt}$: On input an enabled set $S \subseteq [n]$ and PK, it outputs a header Hdr which is a ciphertext that encrypts a symmetric key K. Let M be the message the set S is supposed to decrypt. This message is encrypted under the symmetric key K using an encryption function F_K. The derived ciphertext C_M is called broadcast body.
- $\mathsf{Decrypt}$: On input $S, i, d_i, Hdr, \mathsf{PK}$, if $i \in S$, then the algorithm outputs the message encryption key K. The key K can then be used to decrypt the broadcast body and obtain the message body M .

Next, we present the first construction in [6].

- Setup: On input n, let $\mathbb{G}$ be a bilinear group of prime order p.
  - Pick a random generator g of the group $\mathbb{G}$ and select α, γ uniformly at random from the group $\mathbb{Z}_p$.
  - Compute $g_i = g^{(\alpha^i)} \in \mathbb{G}$, $\forall i \in \{1, 2, .., n, n+2, .., 2n\}$ and $v = g^\gamma$.
  - For $i = 1$ to n, compute $d_i = g_i^\gamma$.
  - Set $\mathsf{PK} = \{g, g_1, ..., g_n, g_{n+2}, ..., g_{2n}, v\}$ and d_i be the secret key corresponds to each user i.
- $\mathsf{Encrypt}$: On input a set $S \subseteq [n]$ and PK:
  - Select a random $t \in \mathbb{Z}_p$. Set the message to be broadcast $K = e(g, g_{n+1})^t = e(g, g)^{t\alpha^{n+1}}$.
  - Compute $C_0 = g^t$ and $C_1 = (v \cdot \prod_{j \in S} g_{n+1-j})^t$
  - Output (C_0, C_1). This pair of ciphertexts constitutes the Header.
- $\mathsf{Decrypt}$: On input $S, i, d_i, \mathsf{PK}, (C_0, C_1)$:
  - Compute $K = e(g_i, C_1)/e(d_i \cdot \prod_{\substack{i \in S \\ i \neq j}} g_{n+1-j+i}, C_0)$.

It is necessary to prove that the above scheme is correct in the sense that user-member of the enabled set S can recover the key K while at the same time this is impossible for every user $i \notin S$.

Theorem 3.5 (Correctness). *For each user $u \in S$, it holds that*

$$\mathsf{Prob}[\mathsf{Decrypt}(S, i, d_i, \mathsf{PK}, (C_0, C_1)) = K] = 1,$$

where (C_0, C_1) is the output of the algorithm $\mathsf{Encrypt}$ *executed with input* S, PK.

Proof. Suppose that $u \in S$. We have that,

$$e(g_u, C_1) = e(g^{\alpha^u}, g^{\gamma} \cdot \prod_{j \in S} g_{n+1-j})^t = e(g^{\alpha^u}, g^{\gamma + \sum_{j \in S} \alpha^{n+1-j}})^t = e(g, g)^{t\alpha^u\gamma + t\sum_{j \in S} \alpha^{n+1-j+u}}. \quad (3.1)$$

Also, it holds that

$$e(d_u \cdot \prod_{\substack{j \in S \\ j \neq u}} g_{n+1-j+u}, C_0) = e(g^{\alpha^u\gamma + \sum_{j \in S, j \neq u} \alpha^{n+1-j+u}}, g^t) = e(g, g)^{t\alpha^u\gamma + t\sum_{\substack{j \in S \\ j \neq u}} \alpha^{n+1-j+u}}. \quad (3.2)$$

From the relations (3.1),(3.2)

$$\begin{aligned} e(g_u, C_1)/e(d_u \cdot \prod_{\substack{j \in S \\ j \neq u}} g_{n+1-j+u}, C_0) &= e(g, g)^{t\sum_{j \in S} \alpha^{n+1-j+u} - t\sum_{\substack{j \in S \\ j \neq u}} \alpha^{n+1-j+u}} \\ &= e(g, g)^{t\alpha^{n+1}} = K. \quad (3.3) \end{aligned}$$

■

The bottom line of the second construction is that each user can be viewed as an element of a matrix with dimensions $A \times B$ such that $n = AB$. This construction demands the execution of A parallel instances of the first scheme each one of which can broadcast the selected message to B users at most. This is the reason why this scheme is called B- Broadcast encryption. All the users share the same public key parameters $(g_1, ..., g_B, g_{B+2}, ..., g_{2B})$ and the decryption procedure is completely related to the position of the user into the matrix $A \times B$.

- Setup : On input n, let $\mathbb{G}$ be a bilinear group of prime order p.
  - Pick a random generator g of the group $\mathbb{G}$ and select $\alpha, \gamma_1, .., \gamma_A$ uniformly at random from the group $\mathbb{Z}_p$.
  - Compute $g_i = g^{(\alpha^i)} \in \mathbb{G}, \forall i \in \{1, 2, .., B, B{+}2, .., 2B\}$ and $v_j = g_i^{\gamma}, \forall j \in \{1, 2, ..., A\}$.
  - For each user $i \in [n]$, compute $a = \lceil \frac{i}{A} \rceil$ and $b = i \mod B$ and $d_i = g_b^{\gamma_a}$.
  - Set $\mathsf{PK} = \{g, g_1, ..., g_B, g_{B+2}, ..., g_{2B}, v_1, .., v_A\}$ and d_i be the secret key corresponds to the user i.
- $\mathsf{Encrypt}$: On input S, PK:

  - Select a random $t \in \mathbb{Z}_p$. Set the message to be broadcast $K = e(g, g_{B+1})^t = e(g, g)^{t\alpha^{B+1}}$.
  - For $\ell = 1$ to A, define the sets $\hat{S}_\ell$, S_ℓ as follows:

$$\hat{S}_\ell = S \cap \{(\ell - 1)B + 1, ..., \ell B\} \text{ and } S_\ell = \{x - \ell B + B | x \in \hat{S}_\ell\}.$$

  - Compute $C_0 = g^t$, $C_1 = (v_1 \cdot \prod_{j \in S_1} g_{B+1-j})^t$,...,$C_A = (v_A \cdot \prod_{j \in S_A} g_{B+1-j})^t$
  - Output $Hdr = (C_0, C_1, ..., C_A)$.

- Decrypt : On input $S, i, sk_i, \mathsf{PK}, (C_0, C_1, ..., C_A)$:
  - Compute $a = \lceil \frac{i}{A} \rceil$, $b = i \mod B$. Find S_a.
  - Compute $K = e(g_b, C_a)/e(sk_i \cdot \prod_{\substack{i \in S_a \\ j \neq i}} g_{B+1-j+b}, C_0)$.

We note that $\hat{S}_\ell$ is a set which consists of all the enabled users that fall in ℓ-th row and the set S_ℓ contains the indices that are assigned to the users of $\hat{S}_\ell$ with respect to ℓ-th row. The property of correctness is proved similarly to the first construction. It is easy to observe that in case $A = 1$, this scheme coincides with the previous one. If $A = B = \sqrt{n}$, the resulted scheme has ciphertext length and public key size $O(\sqrt{n})$ while the size of the private key is preserved constant.

Security proof

Now, we will turn our attention to the security proof of the above constructions. As a security model, the authors consider CCA-security against *static* adversaries. At the first step of the security experiment, the adversary outputs the enabled set S^* that aims to attack, i.e. to extract information about the key this enabled set decrypts. Then, the challenger generates the parameters of the system and computes the public key and the private keys. It gives to the adversary the public key together with the private keys of all the users that do not belong to the set published by the adversary during the first step. The adversary can issue a number of decryption queries for ciphertexts that are prepared for enabled sets S that are subsets of S^*. The challenger computes a key K in the way the encryption algorithm suggests and executes the encryption algorithm for the set S^*. In the sequel, he picks randomly a key from the key-space K and flips a random coin in order to select in order to decide in which position each of the two keys will be placed. Then, he sends to the adversary a tuple with the prepared ciphertext with the pair of keys. In order for a scheme to be secure, the adversary must not be able to understand which is the key that the ciphertext encrypts. The formal definition follows.

Experiment $\mathsf{Exp}^{cca-sec}$:

- **Init :** The adversary $\mathcal{A}$ outputs a set $S^* \subseteq [n]$.

- **Setup :** The challenger runs Setup on input n and obtains a public key PK and n private keys $d_1, ..., d_n$. He sends PK and d_i, for all $i \notin S^*$.

- **Phase 1 :** $\mathcal{A}$ issues addaptively $q_1, ..., q_m$ decryption queries on input (u, S, Hdr), with $S \subseteq S^*$ and $u \in S$. The challenger answers preparing $\mathsf{Decrypt}(u, d_u, S, Hdr, \mathsf{PK})$.

- **Challenge :** The challenger runs $\mathsf{Encrypt}(\mathsf{PK}, S^*)$ and obtains (K, Hdr^*). He chooses randomly a value $b \in \{0,1\}$ and sets $K_b = K$. K_{1-b} is a key that he selects at random form the key space K. The challenger returns to the adversary the (Hdr^*, K_0, K_1).

- **Phase 2:** $\mathcal{A}$ operates as in Phase 1 submitting $q_{m+1}, ..., q_D$ with the restriction $Hdr \neq Hdr^*$.

- **Guess:** $\mathcal{A}$ outputs $b' \in \{0,1\}$.

The experiment returns 1 if and only if $b = b'$.

Definition 3.7. *Let $\mathsf{WIN}_{\mathcal{A}}$ be the event that $\mathcal{A}$ wins the previous experiment. We say that a broadcast encryption scheme is (t, ε, n, q_D)-CCA secure is for all t-time algorithms $\mathcal{A}$ that issue q_D decryption queries $\left|\mathsf{Prob}[\mathsf{WIN}_{\mathcal{A}}] - \frac{1}{2}\right| \leq \varepsilon$.*

The authors show that their constructions are semantically secure under the decision (t, ℓ, ε)-BDHE assumption. The decision ℓ-BDHE problem in a group $\mathbb{G}$ is defined as follows:

Definition 3.8. *Let $\vec{y}_{g,\alpha,\ell} = (g_1, ..., g_\ell, g_{\ell+1}, ..., g_{2\ell})$, $\mathcal{P}_{BDHE}$ be a distribution $(g, h, \vec{y}_{g,\alpha,\ell}, e(g_{\ell+1}, h))$ and $\mathcal{R}_{BDHE}$ be the distribution $(g, h, \vec{y}_{g,\alpha,\ell}, T)$, where $g, h \xleftarrow{r} \mathbb{G}$, $\alpha \xleftarrow{r} \mathbb{Z}_p$ and $T \xleftarrow{r} \mathbb{G}_1$. A statistical test has advantage ε in solving decision ℓ-BDHE in $\mathbb{G}$ if*

$$\left|\mathsf{Prob}[\mathcal{B}(g, h, \vec{y}_{g,\alpha,\ell}, e(g_{\ell+1}, h)) = 0] - \mathsf{Prob}[\mathcal{B}(g, h, \vec{y}_{g,\alpha,\ell}, T) = 0]\right| \geq \varepsilon. \tag{3.4}$$

Definition 3.9. *We say that the decision ℓ-BDHE holds in $\mathbb{G}$ if no t-time adversary has advantage at least ε in breaking the ℓ-BDHE decisional problem.*

Having provided the necessaary definitions, we proceed to the main theorem that shows that the generlaized scheme is under the decision (t, ℓ, ε)-BDHE assumption.

Theorem 3.6. *Let $\mathbb{G}$ be a bilinear group of prime order p. For any positive integers B, n such that $n > B$, a B-broadcast encryption scheme is secure assuming that the (t, ε, B)-BDHE assumption holds in $\mathbb{G}$.*

Proof. Let $\mathcal{A}$ be a t-time adversary such that $\left|\mathsf{Prob}[\mathsf{WIN}_{\mathcal{A}}] - \frac{1}{2}\right| \geq \varepsilon$. We will construct a t-time statistical test $\mathcal{B}$ that breaks definition 3.9. The algorithm $\mathcal{B}$ takes inputs either of the form $(g, h, \vec{y}_{g,\alpha,\ell}, e(g_{\ell+1}, h))$ or $(g, h, \vec{y}_{g,\alpha,\ell}, T)$ and proceed as follows:

1. $\mathcal{B}$ runs $\mathcal{A}$.

2. $\mathcal{A}$ outputs the challenge $S^* \subseteq [n]$.

3. $\mathcal{B}$ computes the sets $\hat{S}_i$ and S_i for $i = 1, ..., A$. Then $\mathcal{B}$ selects g, α unoformly at random and sets $\mathsf{PK} = (g, \vec{y}_{g,\alpha,\ell}, v_1, ..., v_A)$, where each v_i is computed as follows:

   - $\mathcal{B}$ chooses randomly a value $u_i \in \mathbb{Z}_p$. Using the values from $g, \vec{y}_{g,\alpha,\ell}$, $\mathcal{B}$ computes $v_i = g^{u_i}\Big(\prod_{j \in S_a} g_{B+1-j}\Big)^{-1}$. Due to the fact that g, α, u_i are selected uniformly at random, v_i from the value computed in the construction and thus PK is indinstinguishable from that of the construction.

4. $\mathcal{B}$ computes the private keys as follows: For each $i \notin S$,

$$d_i = g_b^{u_i} \cdot \prod_{j \in S_a} (g_{B+1-j+b})^{-1}.$$

5. $\mathcal{B}$ computes $Hdr = (h, h^{u_1}, ..., h^{u_A})$. This is a valid encryption for a key $K = e(g_{B+1,B})^t$, where t is a randomly chosen value. Precisely, $h = g^t$, for some $t \in Z_p$(as in C_0) and for all $i = 1, ..., A$

$$h^{u_i} = \Big(g^{u_i}\big(\prod_{j \in S_i} g_{B+1-j}\big)^{-1} \prod_{j \in S_i} g_{B+1-j}\Big)^t = \big(v_i \cdot \prod_{j \in S_i} g_{B+1-j}\big)^t$$

6. $\mathcal{B}$ flips a coin and chooses a value $b \in \{0, 1\}$. He sets $K_b =, e(g_{\ell+1}, h)$ or $K_b = T$ depending on the input. Next, $\mathcal{B}$ selects a random key from the key space as K_{1-b}. $\mathcal{B}$ sends to $\mathcal{A}$ the tuple (Hdr, K_0, K_1).

7. $\mathcal{B}$ outputs 0 of the result of the experiment is 1 and vice versa.

We observe that $\mathcal{B}$'s responses to $\mathcal{A}$ are consistent to the protocol and thus he simulates the protocol perfectly. This arises from the way the public key is computed, from the ciphertexts computed at the step 4 of the algorithm and the fact that

$$d_i = g_b^{u_i} \cdot \prod_{j \in S_a} (g_{B+1-j+b})^{-1} = \Big(g^{u_i} \cdot \prod_{j \in S_a} (g_{B+1-j})^{-1}\Big)^{\alpha^b} = v_i^{\alpha^b}.$$

According to the above algorithm, we observe that if the input of $\mathcal{B}$ is a tuple $(g, h, \vec{y}_{g,\alpha,\ell}, e(g_{\ell+1}, h))$ which means that $K_b = e(g_{\ell+1}, h)$, the adversary has advantage ε in guessing the correct answer and break the semantic security of the scheme. Namely, we have that

$$\mathsf{Prob}[\mathcal{B}(g, h, \vec{y}_{g,\alpha,\ell}, e(g_{\ell+1}, h)) = 0] = \mathsf{Prob}[\mathsf{WIN}_{\mathcal{A}}].$$

From the initial assumption, we have that

$$\Big|\mathsf{Prob}[\mathsf{WIN}_{\mathcal{A}}] - \frac{1}{2}\Big| \geq \varepsilon. \tag{3.5}$$

On the other hand, in case the input of $\mathcal{B}$ is $(g, h, \vec{y}_{g,\alpha,\ell}, T)$, both keys that are provided to $\mathcal{A}$ are selected randomly and as a result $\mathcal{A}$ has no advantage in distinguishing which is the one encrypted in Hdr. Consequently,

$$\mathsf{Prob}[\mathcal{B}(g, h, \vec{y}_{g,\alpha,\ell}, T) = 0] = \mathsf{Prob}[\mathsf{WIN}_{\mathcal{A}}] = \frac{1}{2}.$$

Consequently, ti holds that

$$\Big|\mathsf{Prob}[\mathcal{B}(g, h, \vec{y}_{g,\alpha,\ell}, e(g_{\ell+1}, h)) = 0] - \mathsf{Prob}[\mathcal{B}(g, h, \vec{y}_{g,\alpha,\ell}, T) = 0]\Big| = \Big|\mathsf{Prob}[\mathsf{WIN}_{\mathcal{A}}] - \frac{1}{2}\Big| \geq \varepsilon.$$

The proof is completed.

■

3.4.2 Identity-Based Broadcast Encryption with constant size ciphertexts and private keys

Delerablée in [8] proposes a broadcast encryption scheme still for the setting of public key broadcast encryption which does also achieve ciphertexts and private keys of constant size.The advantage of this scheme compared to the first scheme of [6] is that the size of the public key in [8] is linear to the maximum number of enabled users and not linear to the number of the whole population of users. This difference implies that this scheme can be very efficient especially in the cases where the sender wishes to broadcast a message to a small number of users.

More precisely, the construction of Delerablée relies on the setting of Identity-based broadcast encryption. Identity-based encryption introduced by Shamir in [24]. The characteristic of this primitive is that every user is able to generate on its own the string-identity that corresponds to his public key and then obtain the appropriate private key. This is achieved by interacting with a trusted authority called a Private Key Generator (PKG). The Private Key generator holds a master public key(MPK) and a master secret key (MSK). The master secret key is used for the generation of the private keys of the users.

The Identity-based model extended to the context of broadcast encryption is similar in the following way. The author formalizes it as a tuple of four algorithms $(\mathsf{Setup}, \mathsf{Extract}, \mathsf{Encrypt}, \mathsf{Decrypt})$. The Setup algorithm generates the parameters of the system, a master secret key, MSK, and a public key PK. MSK is provided to the to the PKG in order to be able to generate private keys for new members through the $\mathsf{Extract}$ algorithm. In order to encrypt a message for a set of identities, a sender uses the identitites of the appropriate users and PK. Each user uses his private key for decryption.

The IBBE scheme of [8] is defined over a tuple $(\mathsf{Setup}, \mathsf{Extract}, \mathsf{Encrypt}, \mathsf{Decrypt})$ as follows:

- Setup: On input a security parameter λ and an integer m:
    - Generate a bilinear map group system $\mathcal{B} = (\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e(\cdot,\cdot))$, such that $\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T$ are cyclic groups of prime order p, with $|p| = \lambda$.
    - Choose randomly two group generators $g \in \mathbb{G}_1$ and $h \in \mathbb{G}_2$ and a value $\gamma \in \mathbb{Z}_p^*$.
    - Choose a hash function $\mathcal{H}$, such that $\mathcal{H} : \{0,1\}^* \to \mathbb{Z}_p^*$.
    - Set $\mathsf{PK} = \{w, v, h, h^\gamma, ..., h^{\gamma^m}\}$, with $w = g^\gamma$, $v = e(g, h)$ and $\mathsf{MSK} = (g, \gamma)$.
- $\mathsf{Extract}$: On input MSK, ID compute:
$$sk_{\mathsf{ID}} = g^{\frac{1}{\gamma + \mathcal{H}(\mathsf{ID})}}$$
- $\mathsf{Encrypt}$: On input $S = \{\mathsf{ID}_i\}_{i=1}^s$, PK:
    - Pick k randomly from $\mathbb{Z}_p^*$. Compute the key to be broadcast as $K = e(g, h)^k$.
    - Output (C_1, C_2) such that:
$$C_1 = w^{-k} \text{ and } C_2 = h^{k \prod_{i=1}^s (\gamma + \mathcal{H}(\mathsf{ID}_i))}.$$
- $\mathsf{Decrypt}$: On input $(S, \mathsf{ID}, sk_{\mathsf{ID}}, (C_1, C_2), \mathsf{PK})$: Compute
$$K = \Big(e(C_1, h^{p_{i,S}(\gamma)}) \cdot e(sk_{\mathsf{ID}}, C_2)\Big)^{\frac{1}{\prod_{j=1, j\neq i}^s \mathcal{H}(\mathsf{ID}_j)}}, \text{ where} \tag{3.6}$$

$$p_{i,S}(\gamma) = \frac{1}{\gamma}\Big(\prod_{j=1, j\neq i}^{s} (\gamma + \mathcal{H}(\mathsf{ID}_j)) - \prod_{j=1, j\neq i}^{s} \mathcal{H}(\mathsf{ID}_j)\Big)$$

Theorem 3.7 (Correctness). *For each user $u \in S$, it holds that*

$$\mathsf{Prob}[\mathsf{Decrypt}((S, \mathsf{ID}_u, sk_{\mathsf{ID}}, (C_1, C_2), \mathsf{PK})) = K] = 1.$$

Proof. Let u be an enabled user, i.e. $u \in S$:

$$e(C_1, h^{p_{u,S}(\gamma)}) = e(g,h)^{-k\Big(\prod_{j=1,j\neq u}^{s}(\gamma+\mathcal{H}(\mathsf{ID}_j)) - \prod_{j=1,j\neq u}^{s}\mathcal{H}(\mathsf{ID}_j)\Big)}$$

$$e(sk_{\mathsf{ID}_u}, C_2) = e(g^{\frac{1}{\gamma+H(\mathsf{ID}_u)}}, h^{k\prod_{i=1}^{s}(\gamma+\mathcal{H}(\mathsf{ID}_i))}) = e(g,h)^{k\prod_{j=1,j\neq u}^{s}(\gamma+\mathcal{H}(\mathsf{ID}_j))}$$

$$\begin{aligned} e(C_1, h^{p_{u,S}(\gamma)}) \cdot e(sk_{\mathsf{ID}_u}, C_2) &= e(g,h)^{-k\cdot\prod_{j=1,j\neq u}^{s}(\gamma+\mathcal{H}(\mathsf{ID}_j)) + k\cdot\prod_{j=1,j\neq u}^{s}\mathcal{H}(\mathsf{ID}_j) + k\prod_{j=1,j\neq u}^{s}(\gamma+\mathcal{H}(\mathsf{ID}_j))} \\ &= e(g,h)^{k\cdot\prod_{j=1,j\neq u}^{s}\mathcal{H}(\mathsf{ID}_j)} \qquad (3.7) \end{aligned}$$

As a result, we have that

$$\Big(e(g,h)^{k\cdot\prod_{j=1,j\neq u}^{s}\mathcal{H}(\mathsf{ID}_j)}\Big)^{\frac{1}{\prod_{j=1,j\neq u}^{s}\mathcal{H}(\mathsf{ID}_j)}} = K.$$

■

Security

The security model adopted is IND-CCA-sID security as this is defined in the section 2.2 of the paper [8]. This scheme is proven IND-CCA-sID-secure in the random oracle model under the $(f, g, F) - GDDHE$ assumption.

Chapter 4

Privacy in the context of Broadcast Encryption

As already mentioned during the introduction, the first that put forth the notion of privacy in the setting of broadcast encryption are Barth et al. in [4]. They consider a class of attacks for broadcast encryption where the objective of the attacker is to obtain information about the set of enabled users. To address this important problem, Barth et al. [4] introduced a security model for private broadcast encryption and provided a first solution. The scheme of [4] applies to the public key setting and has the characteristic of being linear in the number of users, i.e. $\Theta(s \cdot k)$ where s is the number of enabled users and k is the security parameter. They define security against chosen-ciphertext attacks due to the importance of a scheme to be able to resist to active attacks. Then, they provide two constructions with different performance which are secure in terms of this definition. We will elaborate on their work in detail.

Motivated by the above, we provide various results suggesting the latter state of affairs by proving tight lower bounds for the ciphertext length of private broadcast encryption schemes. We outline our results below.

First, we study the formalization of the notion of privacy in the context of private broadcast encryption. We introduce three security formulations. The first notion we consider is inspired by that in [4] : it allows the adversary to interact with the broadcast encryption system by obtaining encryption and decryption queries as well as corrupting recipients. Upon completion of a first stage the adversary provides two target sets of users to be revoked $\mathsf{R}_0, \mathsf{R}_1$. Then, provided that $|\mathsf{R}_0| = |\mathsf{R}_1|$, the adversary receives as a challenge an encryption of a randomly chosen message M with the set of users R_b revoked where b is a random bit. The challenge also contains the message M. The adversary has to guess the bit b under the constraint that it does not submit the challenge ciphertext to a decryption oracle and does not control any user in the symmetric difference $\mathsf{R}_0 \triangle \mathsf{R}_1$. We call this level of privacy $\mathsf{priv\text{-}eq}$.

We observe $\mathsf{priv\text{-}eq}$ is quite insufficient for many reasonable attack settings. Specifically, for a certain ciphertext the adversary may be absolutely certain that the set of users R is revoked and only wishes to test whether an additional target user i is also revoked or not. Clearly the objective of this attack is not captured by the above definition since in this case it holds that $\mathsf{R}_0 = \mathsf{R}$ and $\mathsf{R}_1 = \mathsf{R} \cup \{i\}$, two sets of different cardinality. We formalize this notion of privacy as $\mathsf{priv\text{-}st}$. It is very easy to see that there exist schemes that satisfy $\mathsf{priv\text{-}eq}$ and fail $\mathsf{priv\text{-}st}$; in particular, any scheme that leaks the cardinality of the set of revoked users is such a candidate and in fact the scheme of [4] is one such scheme.

Taking this one step further we introduce *full privacy* to be the property where the adversary cannot distinguish any two sets $\mathsf{R}_0, \mathsf{R}_1$; we term this notion as priv-full. We then prove that in fact priv-st and priv-full are equivalent.

Armed with this definitional basis we proceed to our lower bounds. We first consider the case of *atomic* broadcast encryption schemes. The private schemes of [4] satisfy this condition and it is also quite common in the wide class of combinatorial broadcast encryption schemes; all the (non-private) schemes in ([22],[16],[15], [26], [3]) are atomic.

For such atomic schemes, we prove that any scheme that satisfies the priv-eq condition is susceptible to an attack against privacy in the case when the ciphertext drops below $s \cdot k$ where s is the cardinality of the set of enabled users. This means that a lower bound of $\Omega(s \cdot k)$ is in place. We then present an atomic private broadcast encryption scheme with this complexity hence showing the lower bound is tight. The scheme itself is a standard linear length construction; the scheme applies equally to the symmetric and public-key setting and abstracts the necessary properties needed for privacy to the existence of secure key-private encryption mechanism in the KEM sense [25]. We present a similar set of results for the priv-full level of privacy; in this case KEM security is sufficient and the corresponding tight bound is $\Theta(n \cdot k)$.

Having settled the case of atomic broadcast encryption, we switch our focus to the setting of general private broadcast encryption schemes (that are not necessarily atomic). We first show using an information theoretic argument that any broadcast encryption scheme should exhibit some ciphertexts of length $\Omega(n+k)$. Using this as a stepping stone we then prove that if a broadcast encryption scheme is assumed to be private in the sense of priv-st, priv-full, it will have to provide a ciphertext of length $\Omega(n+k)$ for any set of revoked users R hence a complexity bound sublinear in the number of users is impossible to be achieved if full privacy is desired.

Before presenting our results, we will refer extensively to the work of Barth, Boneh and Waters [4] as they are the first who adress the problem of privacy in the context of broadcast encryption. Furthermore, we will briefly describe some related work that has been conducted independently of ours.

4.1 Private Broadcast Encryption

A private broadcast encryption system is defined by Barth et al. in [4] as a tuple of algorithms $\langle \mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Encrypt}, \mathsf{Decrypt} \rangle$.

- $\mathsf{Setup}(\lambda)$: Generates the global parameters of the system I.
- $\mathsf{KeyGen}(I)$: Generate public-secret key pairs (pk, sk).
- $\mathsf{Encrypt}(S, m)$: Given that $S = \{pk_1, ..., pk_s\}$ and a message m, generate a ciphertext C to be decrypted by the users of the set S.
- $\mathsf{Decrypt}(sk_i, C)$: If $pk_i \in S$ it returns m. If $pk_i \notin S$ or C is malformed the algorithm can return $\perp$.

In this model, each user holds a different public-secret key pair and the sender encrypts the message separately using the public key of each enabled user. The authors adopt a security model for private broadcast encryption schemes, called recipient privacy, which is defined using a game between a challenger and an adversary. As soon as the challenger outputs the global parameters of the system, the adversary outputs two sets of users S_0, S_1 of equal size. Then, the adversary is given the public keys of all users in the set $S_0 \cap S_1$ and the secret keys of all users in the set $S_0 \cup S_1$ and he is also provided access to a Decryption Oracle. The scheme will be considered secure if the adversary cannot distinguish between a ciphertext prepared for decryption by the users of the set S_0 and a ciphertext prepared for decryption by the users of the set S_1. Furthermore, this model provides security against *static* adversaries due to the fact that once the global parameters of the system are published, the users have to output the challenged sets not being able to gain any additional information about the behavior of the system. A formal definition is provided below:

Security game:

- **Init**: Run $\mathsf{Setup}(\lambda)$ and output the global parameters I of the system. Give I to the adversary. The adversary outputs $S_0, S_1 \subseteq [n]$ with $|S_0| = |S_1|$.
- **Setup**: For each $i \in S_0 \cup S_1$ run $\mathsf{KeyGen}(I) \rightarrow (pk_i, sk_i)$. Give the adversary all the pk_i and each sk_i for each $i \in S_0 \cap S_1$.
- **Phase 1**: The adversary submits a number of queries of the form (u, C), which are answered by the challenger by returning $\mathsf{Decrypt}(sk_i, C)$.
- **Challenge**: The adversary challenges a message m. The challenger chooses randomly a value $b \in \{0, 1\}$ and returns $C^* \leftarrow \mathsf{Encrypt}(S_b, m)$ to the adversary.
- **Phase 2**: The adversary is allowed to issue decryption queries similarly to phase 1 with the restriction that $C \neq C^*$.
- **Guess**: The adversary outputs $b^* \in \{0, 1\}$.

The adversary wins the game if and only if $b^* = b$.

Definition 4.1 (Security). *A private broadcast encryption system is (t, q, n, ϵ)-CCA-Recipient-Private if, for all t-time adversaries $\mathcal{A}$ the probability that $\mathcal{A}$ wins the above game using recipients of size at most n, making q decryption queries, is at most $1/2 + \epsilon$.*

Since each user is assigned a different public-secret key pair and the security model preserves privacy only for sets of equal cardinality, a construction that simply omits the enabled set in the decryption should be considered sufficient. Nevertheless, the aim of the authors is to consider a class of attacks, called active attacks, which cannot be barred such a simple construction. The authors provide an example of an active attack in an encrypted file system in order to show the insecurity of such a scheme with respect to this attacks.

Suppose that there is a malicious user A who is authorized to access a file F that is encrypted under a symmetric key K. This user, which is in this case the attacker, intends to extract information about which other users are eligible to access the file F. Suppose that he wishes to distinguish which of the users B and D are authorized to access F as well. Let $C^* = \{K\}_{K_A}||\{K\}_{K_D}||\{F\}_K$ ciphertext the adversary receives, where K_A is the adversary's public key. As a result, $\mathcal{A}$ obtains K and consequently the file F as well. He can then prepare a new ciphertext $C = \{K\}_{K_D}||\{F'\}_K$. My making a decryption query (u, C) with u be the index of the user D, $\mathcal{A}$ checks whether D can decrypt F' or not. If he is able obtain F', this subsequently implies that he is able to obtain F.

Their idea in order to prevent active attacks is to modify this simple model employing an one-time digital signature scheme. The sender encrypts the session key K together with the verification key of the signature under the key of each enabled user. Then, he signs the entire ciphertext with the signing key. As a result, if someone intends to extract a component of this ciphertext and place it to another one, he has to sign the new ciphertext under the same verification scheme. This is considered difficult if the one-time digital signature scheme is one-time strongly existentially unforgeable [1]. The first construction of [4] is described below:

- $\mathsf{Setup}(1^\lambda)$: Run $\mathsf{Init}(1^\lambda)$ and output the global parameters I.
- $\mathsf{KeyGen}(I)$: For each user $i \in [n]$, run $\mathsf{Gen}(1^\lambda)$ in order to output a pair $(\mathsf{pk}_i, \mathsf{sk}_i)$.
- $\mathsf{Encrypt}(S, m)$:
  - Run $\mathsf{SigGen}(1^\lambda)$ and output $(\mathsf{VK}, \mathsf{SK})$.
  - Choose a random symmetric key K and for each $i \in S$ compute $c_{pk_i} = \mathsf{Enc}_{\mathsf{PK}_i}(\mathsf{VK}||K)$.
  - Concatenate all the resulted ciphertexts c_{pk_i} in random order. The result is a ciphertext C_1
  - $C_2 = E_K(m)$.
  - Run $\mathsf{Sign}_{\mathsf{SK}}(C_1||C_2)$ and output σ.
  - Return $C = \sigma||C_1||C_2$.
- $\mathsf{Decrypt}(C, sk)$: If $C_1 = c_1||c_2||...||c_n$, For each $i \in [n]$:
  - Run $\mathsf{Dec}(c_i, sk)$. If the output is $\perp$ continue to the next i otherwise the result is $\mathsf{VK}||K$.
  - Then, if $\mathsf{Verify}(\sigma, C_1||C_2) = 1$ return $D_K(m)$, otherwise return $\perp$.

The above construction requires an underlying public key encryption scheme $(\mathsf{Init}, \mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ that is *strongly correct*. This property guarantees that is almost impossible a decryption operation under a key applied to a ciphertext encrypted under another public key to return a non-$\perp$ symbol. This property is necessary for the decryption algorithm as each user, in possession of his private key, makes decryption operations until the result being a non-$\perp$ symbol. Furthermore, it is required that the scheme $(\mathsf{Init}, \mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ to be key-indistinguishable under CCA-attacks[5]. This notion guarantees that ciphertexts produced using this scheme do not leak information about the key under which it is encrypted. As a result, an adversary that participates in the security experiment gains no knowledge on the ciphertext components that are encrypted under the keys of the users that are outside of the set $S_0 \cap S_1$.

Theorem 4.1. *If* $(\mathsf{Init}, \mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *is both* ε_1*-strongly-correct and* (t, q, ε_2)*-CCA-key-private and* $(\mathsf{SigGen}, \mathsf{Sig}, \mathsf{Ver})$ *is* $(t, 1, \varepsilon_3)$*-strongly-existentially-unforgeable, the above construction is* $(t, q, n, n(\varepsilon_1 + \varepsilon_2 + \varepsilon_3))$*-CCA-recipient-private.*

As already mentioned, the ciphertext length of this scheme is $\Theta(|S| \cdot \lambda)$. The number of decryption operations is at most $|S|$, as the user has to decrypt each component until a non-$\perp$ result is returned or none of the components can be recovered. The authors modify the construction above in a way that the number of decryption operations is reduced to one. Namely, each ciphertext component contains a

[1] An adversary cannot create a new signature even for a message that is already signed

tag which is a hashed value, that provides information for the user that is supposed to decrypt it. This information is provided in way that can be decoded only by the authorized user. An additional value that is placed as part of the secret key for this purpose. Consequently, a user just parses the ciphertext until the corresponding component is found. This scheme is secure in the random oracle model.

4.2 Related work

4.2.1 Anonymous Broadcast Encryption: Adaptive Security and Efficient Constructions in the Standard Model

Libert, Paterson and Quaglia [21] have studied the problem of "anonymous broadcast encryption". They provide constructions for the setting of public key broadcast encryption with ciphertext length of either $\Theta(s \cdot k)$ or $\Theta(n \cdot k)$. These constructions are proven secure against *adaptive* adversaries. More precisely, their first construction is an "Anonymous broadcast Encryption scheme"(ANOBE) with ciphertext $\Theta(n \cdot k)$ which is based on an underlying IND-CCA secure Public key encryption scheme. Their second construction is an ANOBE scheme with ciphertext size of $\Theta(s \cdot k)$ which built from an underlying key-private and weakly robust public key encryption scheme. A variant of this construction that is based on an Identity-based encryption scheme is proposed. Then, their main focus is to enable efficient decryption in the standard model in the setting where the ciphertext is of length $\Theta(s \cdot k)$. In this case, the known schemes that satisfy privacy, require from the users to test sequentially until they find the proper element they can decrypt. In the public-key setting this can be an arduous task if the number of enabled users is large; by using some randomized tagging mechanism it is possible to improve the decryption time complexity. Our modeling is consistent with that of [21] and our lower bounds readily apply to their setting as well.

4.2.2 Outsider-anonymous Broadcast Encryption with Sublinear Ciphertexts

Fazio and Perera in [11], introduce an intermediate notion of anonymity called *outsider-anonymity*. In an *Outsider-anonymous broadcast encryption scheme*, if the adversary is a user that belongs to the revoked set, he can gain no information about the enabled set. On the other side, in case the adverary is member of the enabled set, she may extract information about some other users that belong to it. Taking advantage of the benefits of this relaxation, the authors employ the public key variant of Complete Subtree method [9], in order to provide constructions that achieve sublinear ciphertext size. The public key variant is used together with use of an Anonymous Identity Based encryption scheme [2] as an underlying encryption scheme together. A type of padding to the ciphertext produced by the Complete Subtree method as well as a random permutation on the derived components is necessary in order to prevent leakage of information related to ciphertext length and the position of ciphertexts.

4.3 Privacy notions for Broadcast Encryption

In this section, we provide some privacy definitions for broadcast encryption and show the relation between them. We define privacy in broadcast encryption using an experiment between a challenger and an adversary. The adversary is given access to an Encryption Oracle which means that he is capable of obtaining ciphertext-message pairs that can be decrypted by an enabled set of users of his choice. Also, he is able to derive the secret keys of a selected set of users, by submitting a number of queries to a Corruption Oracle. We will distinguish three levels of privacy in our formalization. In the most general type (full privacy), priv-full, the adversary should be unable to distinguish between any two sets of revoked users as long as the corrupted users do not cover the symmetric difference of the two sets. In the case of "single target" privacy, priv-st, the adversary wishes to understand whether a single (target) user is included in an (otherwise) known revoked set. Finally, in privacy among equal sets, priv-eq, is identical to the case of priv-full with the additional restriction that the adversary should challenge on two sets with equal cardinality. Formally, we have the following:

```
$\mathsf{EncryptionOracle}(\mathsf{R})$
    $retrieve\ ek$
    $m \xleftarrow{r} \mathsf{M}$
    $c \leftarrow \mathsf{Encrypt}(ek, m, \mathsf{R})$
    $return\ (c, m)$
```

```
$\mathsf{CorruptOracle}(u)$
    $\mathsf{T} \leftarrow \mathsf{T} \cup \{u\}$
    $return\ \mathsf{K}_u$
```

```
$\mathsf{DecryptionOracle}(u, c)$
    $\mathsf{D} \leftarrow \mathsf{D} \cup \{(u, c)\}$
    $retrieve\ \mathsf{K}_u$
    $return\ \mathsf{Decrypt}(\mathsf{K}_u, c)$
```

```
Experiment $\mathsf{Exp}_{\mathcal{A}}^{\mathsf{priv\text{-}x}}(1^n, 1^\lambda)$
  $(ek, \mathsf{K}_1, \ldots, \mathsf{K}_n) \leftarrow \mathsf{KeyGen}(1^n, 1^\lambda)$
  $\mathsf{T} \leftarrow \emptyset$
  $(state, \mathsf{R}_0, \mathsf{R}_1) \leftarrow \mathcal{A}^{\mathsf{CorruptOracle}(\cdot),\mathsf{EncryptionOracle}(\cdot),\mathsf{DecryptionOracle}(\cdot)}(1^n)$
  $b \xleftarrow{r} \{0, 1\}$
  $m \xleftarrow{r} \mathsf{M}$
  $c^* \leftarrow \mathsf{Encrypt}(ek, m, \mathsf{R}_b)$
  $b^* \leftarrow \mathcal{A}^{\mathsf{CorruptOracle}(\cdot),\mathsf{EncryptionOracle}(\cdot),\mathsf{DecryptionOracle}(\cdot)}(guess, (c^*, m), state)$
  if $\big(\exists i \in \mathsf{T}$ such that $i \in (\mathsf{R}_0 \triangle \mathsf{R}_1)\big) \vee$
  $\big(\exists (i, c) \in \mathsf{D}$ such that $i \in (\mathsf{R}_0 \triangle \mathsf{R}_1)$ and $c = c^*\big)$
  then output a random bit else if $b = b^*$ then return $1$ else $0$;
```

Definition 4.2 (Privacy). *Let Φ be a fully exclusive broadcast encryption scheme with n receivers. We say that Φ is private* priv-x*, if for all PPT adversaries $\mathcal{A}$,*

$$\mathsf{Prob}[\mathsf{Exp}_{\mathcal{A}}^{\mathsf{priv\text{-}x}}(1^\lambda, 1^n) = 1] \leq \frac{1}{2} + \varepsilon,$$

where ε is a negligible function of λ and λ is the security parameter.

The sets of users the adversary is able to corrupt with respect to the above definition are presented in the figure 4.1.

Based on the definition above, we provide three different definitions for privacy whose differences concern the form of the challenge $(\mathsf{R}_0, \mathsf{R}_1)$.

- We call $\mathsf{Exp}^{\mathsf{priv\text{-}full}}$ the experiment in which $\mathsf{R}_0, \mathsf{R}_1$ can be any set which is subset of $[n]$.
- With $\mathsf{Exp}^{\mathsf{priv\text{-}st}}$, we define the experiment where $\mathsf{R}_0, \mathsf{R}_1$ have to be of the form R and $\mathsf{R} \cup \{i\}$, accordingly.
- With $\mathsf{Exp}^{\mathsf{priv\text{-}eq}}$, we define the experiment where $\mathsf{R}_0, \mathsf{R}_1$ have to be of equal size. Consequently, it is necessary to add one more or-factor, $(|\mathsf{R}_0| \neq |\mathsf{R}_1|)$, in the condition of the last line of the experiment, in order for the experiment to output a random bit if the adversary challenges sets of unequal size.

We then proceed to show relations between the three notions of privacy.

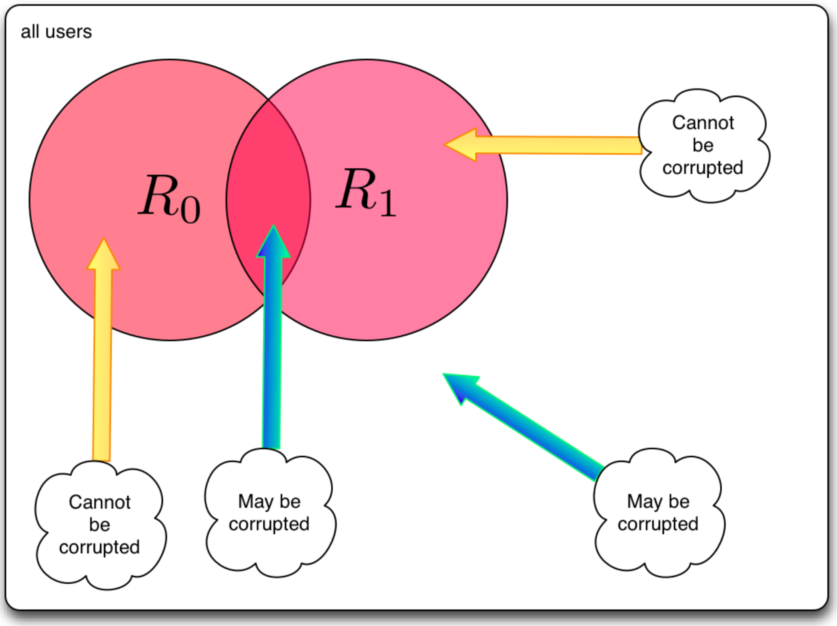

Figure 4.1: Restrictions on corrupted users

Theorem 4.2. 1. *Privacy definitions* priv-st *and* priv-full *are equivalent.*

2. *Privacy definition* priv-full *implies the privacy definition* priv-eq.

3. *Privacy definition* priv-eq *does not imply privacy definition* priv-st.

Proof. 1. We need to prove two directions in order to show that these definitions are equivalent. The easy direction is the one which says that privacy definition priv-full implies privacy definition priv-st. If we assume that there exists a PPT adversary $\mathcal{A}$ that breaks privacy definition priv-st challenging a pair $(\mathsf{R}, \mathsf{R} \cup \{i\})$ with non-negligible advantage α, this adversary also breaks privacy definition priv-full considering that $\mathsf{R}_0 = \mathsf{R}$ and $\mathsf{R}_1 = \mathsf{R} \cup \{i\}$. The opposite direction will be derived from the lemma 1.

2. Assuming that there exists a PPT adversary that breaks privacy definition priv-eq having advantage α, then the same adversary does also break privacy definition priv-full with non-negligible advantage α.

3. It suffices to provide a broadcast encryption scheme which satisfies the definition priv-eq but not private according to the definition priv-full. Let Φ be a broadcast encryption scheme which is priv-eq secure. Now consider Φ' to be exactly like Φ but with the added feature that the encryption algorithm appends at the end of all ciphertexts the cardinality of the revoked set. It is obvious that this scheme is inherently incapable of satisfying privacy definition priv-full (while it remains priv-eq). Such schemes exist under standard cryptographic assumptions as we will see in section 5.1.1.

■

Lemma 1. *Let Φ be a broadcast encryption scheme with n receivers. If there exists a PPT adversary that has at least advantage α in breaking privacy definition* priv-full*, with α non-negligible, then there exists a PPT adversary that breaks privacy definition* priv-st *with probability at least $1/2 + \alpha/n$.*

Sketch of proof: Let $\mathcal{A}$ be a PPT adversary that breaks priv-full definition having at least advantage α, where α is non-negligible. Conditioning on the fact that $\mathcal{A}$ breaks privacy for a pair of sets $(\mathsf{R}_0, \mathsf{R}_1)$, we consider a sequence of sets $P_0, ..., P_{k-1}$, where $k = |\mathsf{R}_0 \triangle \mathsf{R}_1| + 1$, $P_0 = \mathsf{R}_0$ and $P_{k-1} = \mathsf{R}_1$. We set $m = |\mathsf{R}_0 \setminus \mathsf{R}_1|$ and we define P_i as follows: if $i \in \{0, \ldots, m\}$ $P_i = P_{i-1} \setminus \{j\}$, for some user $j \in \mathsf{R}_0 \setminus \mathsf{R}_1$, otherwise $P_i = P_{i-1} \cup \{j'\}$ for some user $j' \in \mathsf{R}_1 \setminus \mathsf{R}_0$. Namely, all the members of this sequence are supersets of $\mathsf{R}_0 \cap \mathsf{R}_1$ and every pair of consecutive sets are of the form $(\mathsf{R}, \mathsf{R} \cup \{i\})$ for some R. This sequence of sets is represented schematically in figure 4.2. We denote as $\mathcal{A}_1$ the part of the algorithm $\mathcal{A}$ that corresponds to the training stage of the experiment, i.e. before the output of challenge, while with $\mathcal{A}_2$ we denote $\mathcal{A}$'s steps after the receipt of the response. Together with the challenge pair $(\mathsf{R}_0, \mathsf{R}_1)$, $\mathcal{A}_1$ outputs a random variable $state$.

We construct a PPT adversary $\mathcal{B}$ that breaks definition priv-st as follows: $\mathcal{B}$ runs $\mathcal{A}_1$ until he outputs the challenge pair $(\mathsf{R}_0, \mathsf{R}_1)$ together with $state$. Then $\mathcal{B}$ makes a guess $j \in \{0, \ldots, k-2\}$ and challenges the corresponding pair. Due to the structure of the sequence, if $j \in \{0, \ldots m-1\}$ $\mathcal{B}$ challenges (P_{j+1}, P_j), otherwise challenges (P_j, P_{j+1}). The received response is provided together with $state$ to $\mathcal{A}_2$. Then, if $j \in \{0, \ldots, m-1\}$ $\mathcal{B}$ outputs the complement of $\mathcal{A}_2$'s output, otherwise outputs $\mathcal{A}_2$'s output. We conclude that $\mathcal{B}$ breaks definition priv-st with advantage at least α/n. ■

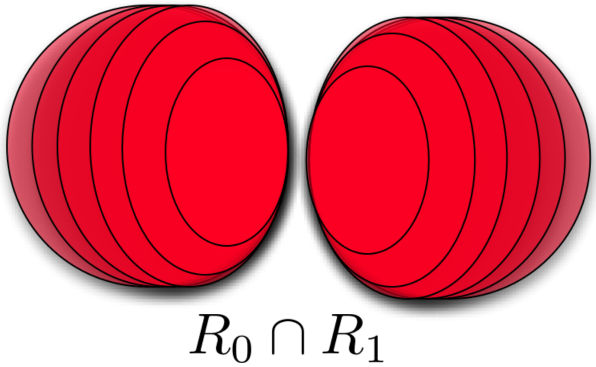

Figure 4.2: Sequence of hybrid sets

Proof. According to the assumption of the theorem we have that there exists a PPT adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ such that

$$\mathsf{Prob}[\mathsf{Exp}_{\mathcal{A}}^{\mathsf{priv\text{-}full}}(1^n, 1^\lambda) = 1] \geq \frac{1}{2} + \alpha, \tag{4.1}$$

where α is a non-negligible function of λ. We will construct a PPT adversary $\mathcal{A}' = (\mathcal{A}'_1, \mathcal{A}'_2)$ that breaks priv-st privacy definition with probability at least $1/2 + \alpha/n$. We consider the sequence of sets $P_1, ..., P_{k-1}$ as this is defined in the above sketch of proof. Note that with P'_b we denote either P_j or P_{j+1}, depending on the choice of the challenger.

More precisely, the algorithm $\mathcal{A}' = (\mathcal{A}'_1, \mathcal{A}'_2)$ proceeds as follows:

$\mathcal{A}'_1$:

1. Run the algorithm $\mathcal{A}_1$ until he challenges $\mathsf{R}_0, \mathsf{R}_1$ and outputs $state$.
2. Guess $j \in \{0, 1, ..., k-2\}$.
3. If $j \in \{0, 1, ..., m-1\}$ challenge (P_{j+1},P_j) otherwise challenge (P_j,P_{j+1}).

$\mathcal{A}'_2$: On input $(\langle m, \mathsf{Encrypt}(ek, m, P'_b)\rangle, state)$

1. Execute $\mathcal{A}_2(\langle m, \mathsf{Encrypt}(ek, m, P'_b)\rangle, state)$.
2. If $j \in \{0, 1, \ldots, m-1\}$ output $\overline{b^*}$, where b^* is the output of $\mathcal{A}_2$. Otherwise output b^*.

Now, we fix two sets $\mathsf{R}_0, \mathsf{R}_1$ and we consider the following events:

$$X_1 = \text{"}\mathsf{Exp}_{\mathcal{A}}^{\mathsf{priv\text{-}full}}(1^n, 1^\lambda) = 1 \text{ given that } \mathcal{A} \text{ challenges } (\mathsf{R}_0, \mathsf{R}_1)\text{"}$$
$$X_0 = \text{"}\mathsf{Exp}_{\mathcal{A}}^{\mathsf{priv\text{-}full}}(1^n, 1^\lambda) = 0 \text{ given that } \mathcal{A} \text{ challenges } (\mathsf{R}_0, \mathsf{R}_1)\text{"}.$$

We set $\alpha_{\mathsf{R}_0,\mathsf{R}_1} = \mathsf{Prob}[\mathsf{Exp}_{\mathcal{A}}^{\mathsf{priv\text{-}full}}(1^n, 1^\lambda) = 1 | \mathcal{A} \text{ challenges } (\mathsf{R}_0, \mathsf{R}_1)] - \dfrac{1}{2}$.
From the relation (4.1), we have that

$$\mathsf{Prob}[\mathsf{Exp}_{\mathcal{A}}^{\mathsf{priv\text{-}full}}(1^n, 1^\lambda) = 1] = \sum_{\mathsf{R}_0,\mathsf{R}_1} \mathsf{Prob}[\mathsf{Exp}_{\mathcal{A}}^{\mathsf{priv\text{-}full}}(1^n) = 1 | \mathcal{A} \text{ challenges } \mathsf{R}_0, \mathsf{R}_1]\mathsf{Prob}[\mathcal{A} \text{ challenges } \mathsf{R}_0, \mathsf{R}_1] \tag{4.2}$$

which implies that

$$\sum_{\mathsf{R}_0,\mathsf{R}_1} \alpha_{\mathsf{R}_0,\mathsf{R}_1} \mathsf{Prob}[\mathcal{A} \text{ challenges } \mathsf{R}_0, \mathsf{R}_1] \geq \alpha.$$

For simplicity, we denote as $p_j = \mathsf{Prob}[\mathcal{A}_2(\langle m, \mathsf{Encrypt}(ek, m, P_j), state\rangle) = 1]$. Suppose that the challenger in the experiment $\mathsf{Exp}^{\mathsf{priv\text{-}full}}$ encrypts a message m for the enabled set $[n] \setminus \mathsf{R}_0$ and the algorithm $\mathcal{A}_2$ returns 1. This means that the adversary made a wrong guess and consequently the result of the experiment is 0. If the challenger encrypts a message for the set $[n] \setminus \mathsf{R}_1$ and $\mathcal{A}_2$ returns 1, the experiment succeeds returning 1. Based on these remarks, it holds that

$$\begin{aligned} p_0 &= \mathsf{Prob}[\mathcal{A}_2(\langle m, \mathsf{Encrypt}(ek, m, \mathsf{R}_0)\rangle, state) = 1] = \mathsf{Prob}[X_0 | b = 0], \\ p_{k-1} &= \mathsf{Prob}[\mathcal{A}_2(\langle m, \mathsf{Encrypt}(ek, m, \mathsf{R}_1)\rangle, state) = 1] = \mathsf{Prob}[X_1 | b = 1]. \end{aligned} \tag{4.3}$$

Making some simple calculations we have that

$$p_{k-1} - p_0 = 2\alpha_{\mathsf{R}_0,\mathsf{R}_1}. \tag{4.4}$$

Next, we define the event $Z_{\mathsf{R}_0,\mathsf{R}_1} = \text{"}\mathsf{Exp}_{\mathcal{A}'}^{\mathsf{priv\text{-}st}}(1^n, 1^\lambda) = 1 \text{ given that } \mathcal{A} \text{ challenges } \mathsf{R}_0, \mathsf{R}_1\text{"}$.

$$\begin{aligned} \mathsf{Prob}[\mathsf{Exp}_{\mathcal{A}'}^{\mathsf{priv\text{-}st}}(1^n, 1^\lambda) = 1] &= \sum_{\mathsf{R}_0,\mathsf{R}_1} \mathsf{Prob}[\mathsf{Exp}_{\mathcal{A}'}^{\mathsf{priv\text{-}st}}(1^n) = 1 | \mathcal{A} \text{ challenges } \mathsf{R}_0, \mathsf{R}_1]\mathsf{Prob}[\mathcal{A} \text{ challenges } \mathsf{R}_0, \mathsf{R}_1] \\ &= \sum_{\mathsf{R}_0,\mathsf{R}_1} \mathsf{Prob}[Z_{\mathsf{R}_0,\mathsf{R}_1}]\mathsf{Prob}[\mathcal{A} \text{ challenges } \mathsf{R}_0, \mathsf{R}_1]. \end{aligned} \tag{4.5}$$

Moreover, it holds that

$$\begin{aligned}\mathsf{Prob}[Z_{\mathsf{R}_0,\mathsf{R}_1}] &= \mathsf{Prob}[Z_{\mathsf{R}_0,\mathsf{R}_1}|b=0]\mathsf{Prob}[b=0] + \mathsf{Prob}[Z_{\mathsf{R}_0,\mathsf{R}_1}|b=1]\mathsf{Prob}[b=1] \\ &= \frac{1}{2}\big(\mathsf{Prob}[Z_{\mathsf{R}_0,\mathsf{R}_1}|b=0] + \mathsf{Prob}[Z_{\mathsf{R}_0,\mathsf{R}_1}|b=1]\big). \qquad (4.6)\end{aligned}$$

In order to compute $\mathsf{Prob}[Z_{\mathsf{R}_0,\mathsf{R}_1}|b=0]$, $\mathsf{Prob}[Z_{\mathsf{R}_0,\mathsf{R}_1}|b=1]$ we make the following observations: In case the challenger chooses $b=0$, the ciphertext provided as input to $\mathcal{A}_2'$ corresponds to the revoked set P_{j+1} if $j \in \{0,...,m-1\}$ or to the revoked set P_j in case $j \in \{m,\ldots,k-2\}$. Due to the fact the the result of the experiment is 1, $\mathcal{A}_2'$ must return 0. Similarly, if the challenger chooses $b=1$, the given input corresponds to the revoked set P_j if $j \in \{0,...,m-1\}$ or to the revoked set P_{j+1} in case $j \in \{m,\ldots,k-2\}$ and as a result $\mathcal{A}_2'$ must return 1.

$$\begin{aligned}\mathsf{Prob}[Z_{\mathsf{R}_0,\mathsf{R}_1}|b=0] &= \frac{1}{k-1}\Big(\sum_{j=0}^{m-1}\mathsf{Prob}[\mathcal{A}_2'(\langle m, \mathsf{Encrypt}(ek,m,P_{j+1})\rangle, state)=0] \\ &+ \sum_{j=m}^{k-2}\mathsf{Prob}[\mathcal{A}_2'(\langle m, \mathsf{Encrypt}(ek,m,P_j)\rangle, state)=0]\Big) \\ &= \frac{1}{k-1}\Big(\sum_{j=0}^{m-1}\mathsf{Prob}[\mathcal{A}_2(\langle m, \mathsf{Encrypt}(ek,m,P_{j+1})\rangle, state)=1] \\ &+ \sum_{j=m}^{k-2}\big(1-\mathsf{Prob}[\mathcal{A}_2(\langle m, \mathsf{Encrypt}(ek,m,P_j)\rangle, state)=1]\big)\Big) \\ &= \frac{1}{k-1}\Big(k-1-m+\sum_{j=0}^{m-1}p_{j+1}-\sum_{j=m}^{k-2}p_j\Big). \qquad (4.7)\end{aligned}$$

$$\begin{aligned}\mathsf{Prob}[Z_{\mathsf{R}_0,\mathsf{R}_1}|b=1] &= \frac{1}{k-1}\Big(\sum_{j=0}^{m-1}\mathsf{Prob}[\mathcal{A}_2'(\langle m, \mathsf{Encrypt}(ek,m,P_j)\rangle, state)=1] \\ &+ \sum_{j=m}^{k-2}\mathsf{Prob}[\mathcal{A}_2'(\langle m, \mathsf{Encrypt}(ek,m,P_{j+1})\rangle, state)=1]\Big) \\ &= \frac{1}{k-1}\Big(\sum_{j=0}^{m-1}\mathsf{Prob}[\mathcal{A}_2(\langle m, \mathsf{Encrypt}(ek,m,P_j)\rangle, state)=0] \\ &+ \sum_{j=m}^{k-2}\mathsf{Prob}[\mathcal{A}_2(\langle m, \mathsf{Encrypt}(ek,m,P_{j+1})\rangle, state)=1]\Big) \\ &= \frac{1}{k-1}\Big(\sum_{j=0}^{m-1}\big(1-\mathsf{Prob}[\mathcal{A}_2(\langle m, \mathsf{Encrypt}(ek,m,P_j)\rangle, state)=1]\big) \\ &+ \sum_{j=m}^{k-2}\mathsf{Prob}[\mathcal{A}_2(\langle m, \mathsf{Encrypt}(ek,m,P_{j+1})\rangle, state)=1]\Big) \\ &= \frac{1}{k-1}\Big(m+\sum_{j=m}^{k-2}p_{j+1}-\sum_{j=0}^{m-1}p_j\Big). \end{aligned} \tag{4.8}$$

From the relations (4.6), (4.7), (4.8), we have

$$\begin{aligned}\mathsf{Prob}[Z_{\mathsf{R}_0,\mathsf{R}_1}] &= \frac{1}{2(k-1)}\Big(k-1+\sum_{j=0}^{m-1}p_{j+1}-\sum_{j=m}^{k-2}p_j+\sum_{j=m}^{k-2}p_{j+1}-\sum_{j=0}^{m-1}p_j\Big) \\ &= \frac{1}{2(k-1)}\Big(k-1+\sum_{j=0}^{k-2}p_{j+1}-\sum_{j=0}^{k-2}p_j\Big) \\ &= \frac{1}{2}+\frac{1}{2(k-1)}\cdot\sum_{j=0}^{k-2}(p_{j+1}-p_j) \\ &= \frac{1}{2}+\frac{1}{2(k-1)}\cdot(p_{k-1}-p_0). \end{aligned} \tag{4.9}$$

Thus, from the relation (4.4) it holds that

$$\mathsf{Prob}[Z_{\mathsf{R}_0,\mathsf{R}_1}] = \frac{1}{2}+\frac{1}{2(k-1)}\cdot 2\alpha_{\mathsf{R}_0,\mathsf{R}_1} \tag{4.10}$$

Finally, from the relations (4.5), (4.10) we conclude that

$$\begin{aligned}\mathsf{Prob}[\mathsf{Exp}^{\mathsf{priv\text{-}st}}_{\mathcal{A}'}(1^n, 1^\lambda) = 1] &= \sum_{\mathsf{R}_0,\mathsf{R}_1} \Big(\frac{1}{2} + \frac{1}{2(k-1)}(2\alpha_{\mathsf{R}_0,\mathsf{R}_1})\Big)\mathsf{Prob}[\mathcal{A} \text{ challenges } (\mathsf{R}_0, \mathsf{R}_1)]. \\ &= \frac{1}{2} + \frac{1}{2(k-1)} \sum_{\mathsf{R}_0,\mathsf{R}_1} (2\alpha_{\mathsf{R}_0,\mathsf{R}_1})\mathsf{Prob}[\mathcal{A} \text{ challenges } (\mathsf{R}_0, \mathsf{R}_1)] \\ &\geq \frac{1}{2} + \frac{\alpha}{k-1} \\ &\geq \frac{1}{2} + \frac{\alpha}{n}. \end{aligned} \tag{4.11}$$

■

Chapter 5

Lower Bounds for Private Broadcast Encryption

5.1 Lower bounds for Atomic Broadcast Encryption schemes

In this section we will show lower bounds for atomic broadcast encryption schemes as these are defined in section 3.3. Given that we will provide lower bounds, we provide a weaker definition of privacy which departs from definition priv-eq in the existence of the CorruptOracle and DecryptionOracle in the security game. More precisely, the adversary is not given access to a Decryption Oracle and instead of being provided access to a Corruption Oracle, he is given access to an Atomic Decryption Oracle which operates as follows:

$$\mathsf{AtDecOr}(j,t,C) = \begin{cases} 0 & \text{if no atomic ciphertext in } C \text{ is supposed to be decrypted} \\ & \text{under the key } sk_{jt} \\ \perp & \text{if the number of keys in the set } \mathsf{SK}_j \text{ are less than } t \\ 1 & \text{if there exists an atomic ciphertext that can be decrypted} \\ & \text{under the key } sk_{jt} \end{cases}$$

```
$\mathsf{EncryptionOracle}(\mathsf{R})$            | $\mathsf{AtDecOr}(j,t,C)$
  $retrieve\ ek$                       |   $\mathsf{E} \leftarrow \mathsf{E} \cup \{(j,t)\}$
  $m \overset{r}{\leftarrow} \mathsf{M}$                            |   $return\ x \in \{0,1,\perp\}$
  $c \leftarrow \mathsf{Encrypt}(ek, m, \mathsf{R})$       |
  $return\ (c, m)$                     |
Experiment $\mathsf{Exp}_{\mathcal{A}}^{\mathsf{priv\text{-}eq\text{-}at}}(1^n)$
  $(ek, \mathsf{K}_1, ..., \mathsf{K}_n) \leftarrow \mathsf{KeyGen}(1^n)$
  $\mathsf{T} \leftarrow \emptyset$
  $(state, \mathsf{R}_0, \mathsf{R}_1) \leftarrow \mathcal{A}^{\mathsf{AtDecOr}(\cdot),\mathsf{EncryptionOracle}(\cdot)}(1^n)$
  $b \overset{r}{\leftarrow} \{0,1\}$
  $m \overset{r}{\leftarrow} \mathsf{M}$
  $c^* \leftarrow \mathsf{Encrypt}(ek, m, \mathsf{R}_b)$
```

```
$b^* \leftarrow \mathcal{A}^{\mathsf{AtDecOr}(\cdot),\mathsf{EncryptionOracle}(\cdot)}(guess, (c^*, m), state)$
if $\big(\exists(i, \cdot) \in \mathsf{E}$ such that $i \in (\mathsf{R}_0 \triangle \mathsf{R}_1)\big) \vee \big(|\mathsf{R}_0| \neq |\mathsf{R}_1|\big)$
then output a random else if $b = b^*$ then return 1 else 0;
```

The experiment $\mathsf{Exp}_{\mathcal{A}}^{\mathsf{priv\text{-}eq\text{-}at}}$ is defined identically to $\mathsf{Exp}_{\mathcal{A}}^{\mathsf{priv\text{-}eq}}$ with the oracle AtDecOr substituting the corruption and decryption oracles.

Definition 5.1. *Let Φ be a broadcast encryption scheme with n receivers. We say that Φ is private* priv-eq-at*, if for all PPT adversaries $\mathcal{A}$,*

$$\mathsf{Prob}[\mathsf{Exp}_{\mathcal{A}}^{\mathsf{priv\text{-}eq\text{-}at}}(1^\lambda, 1^n) = 1] \leq \frac{1}{2} + \varepsilon,$$

where ε is a negligible function of λ and λ be the security parameter.

The following proposition is easy:

Proposition 2. *Any broadcast encryption scheme Φ that satisfies privacy definition* priv-eq*, does also satisfy privacy definition* priv-eq-at.

Proof. It is easy to see that assuming the existence of a PPT adversary $\mathcal{A}$ that has non-negligible advantage in breaking privacy definition priv-eq-at, there is a PPT adversary $\mathcal{B}$ that breaks privacy definition priv-eq with the same advantage as $\mathcal{A}$ executing $\mathcal{A}$ inside him. The proof relies on the fact that $\mathcal{B}$ can perfectly answer the queries submitted by $\mathcal{A}$ to the Atomic Decryption Oracle because of his access to a Corrupt Oracle.

■

Theorem 5.1. *(Lower bound for atomic schemes) Let Φ be an atomic broadcast encryption scheme and suppose that there exists an enabled set $S \subseteq [n]$ such that the number of atomic ciphertexts included in the prepared ciphertext C_S are less that $|S|$. Then, the scheme is* not *private according to definition* priv-eq-at.

Proof. We will assume that for every R the atomic ciphertexts produced by the algorithm Encrypt are always decrypted under the same set of atomic keys (in the other case, if the algorithm Encrypt flips a number of coins in order to decide the atomic keys that will be used, then the same argument we present below can take place with the only difference that in this case the adversary will have to run a number of times the algorithm Encrypt for the set R_0 to approximate the distribution). Let us assume that there exists such a set S_0 and let C_{S_0} be a ciphertext produced by the algorithm Encrypt on input ek, m, R_0 with $\mathsf{R}_0 = [n] \setminus S_0$. Then, according to the pigeonhole principle, there exists at least one atomic ciphertext c_k in the ciphertext C_{S_0} that can be decrypted by at least two users $i, j \in [n]$. As a result, the ciphertext c_k can be decrypted under an atomic key sk_m which is a member of both sets SK_i, SK_j, where SK_i, SK_j are the atomic decryption keys of i and j accordingly. Given this an adversary $\mathcal{A}$ that breaks privacy can be constructed following the logic presented below:

1. If $i, j \in [n]$ are two users which decrypt the same atomic ciphertext in a ciphertext tuple C_{S_0}, where $C_{S_0} \leftarrow \mathsf{Encrypt}(ek, m, \mathsf{R}_0)$, select a set R_1 such that $|\mathsf{R}_1| = |\mathsf{R}_0|$, $i \in \mathsf{R}_1$ and $j \notin \mathsf{R}_1$. Choose arbitrarily the other $|\mathsf{R}_1| - 1$ members of $|\mathsf{R}_1|$ and challenge $\mathsf{R}_0, \mathsf{R}_1$.

2. When the response C^* is received, issue a query R_0 to the Encryption Oracle which is replied with a ciphertext C.

3. Submit a number of queries of the form (j, t, C) to the Atomic Decryption Oracle, for all the possible values of t, starting form $t = 1$, until $\mathsf{AtDecOr}$ returns $\perp$. If we ignore the symbol $\perp$, the output of this procedure is a bitstring x_1 of length s, where s is the number of atomic keys included in the decryption key of SK_j.

4. Repeat the same procedure submitting queries on inputs of the form (j, t, C^*) and obtain a bitstring x_2 of length k (note that this is allowed since j is enabled in both challenge ciphertexts). If it holds that $x_1 \neq x_2$, then answer 1 else 0.

■

Corollary 5.1. *Any atomic broadcast encryption scheme with n receivers with ciphertext length less that n cannot be private according to definition* priv-full.

Proof. If $\mathsf{R} = \emptyset$ and the atomic ciphertexts are less that n, the assumption of the Theorem 5.1 takes place for $S = [n]$. It is easily observed that the fact that the challenged sets $\mathsf{R}_0, \mathsf{R}_1$ were of equal length played no crucial role in the proof of Theorem 5.1. Thus, we can apply exactly the same arguments with $\mathsf{R} = \emptyset$ being the one set in the challenge. ■

Corollary 5.2. *For any atomic broadcast encryption scheme Φ with $[n]$ receivers which private according to* priv-eq *definition, it holds that for any enabled set $S \subseteq [n]$, the ciphertext length is $\Omega(k \cdot |S|)$ bits, where k is the maximum size of an atomic ciphertext. For any broadcast encryption scheme which is private according to* priv-full *definition, the ciphertext length is $\Omega(k \cdot n)$ for all the enabled sets $S \subseteq [n]$.*

5.1.1 Constructions of Atomic Private Broadcast Encryption schemes

In this section, we present matching schemes for the lower bounds of the previous section. We focus on CCA-1 security for simplicity but our results can be easily extended to CCA-2 security. We consider security in the sense of key encapsulation mechanisms (KEM). The definitions of these section are slightly different compared to the definitions of section 3.2.1.

```
Experiment $\mathsf{Exp}_{\mathcal{A}}^{KEM}(1^\lambda)$
  Select $k$ at random.
  $aux \leftarrow \mathcal{A}^{\mathsf{Enc}_k(\cdot),\mathsf{Dec}_k(\cdot)}$
  $m_0, m_1 \xleftarrow{r} \mathsf{M}$;
  $b \xleftarrow{r} \{0,1\}; c \leftarrow \mathsf{Enc}_k(m_b)$
  $b^* \leftarrow \mathcal{A}^{\mathsf{Enc}_k(\cdot)}(m_1, c)$
  if $b = b^*$ then return 1 else 0;
```

Definition 5.2. *We say that the symmetric encryption scheme* (Gen, Enc, Dec) *is KEM-secure if for any probabilistic polynomial time adversary $\mathcal{A}$ it holds that*

$$\mathsf{Prob}[\mathsf{Exp}_{\mathcal{A}}^{KEM}(1^\lambda) = 1] \leq \frac{1}{2} + \varepsilon,$$

where ε is a negligible function of λ.

```
$\mathsf{EncryptionOracle}(\mathsf{R})$
  $m \overset{r}{\leftarrow} \mathsf{M}$
  $retrieve\ ek$
  $c \leftarrow \mathsf{Encrypt}(ek, m, \mathsf{R})$
  $return\ (m, c)$

$\mathsf{CorruptOracle}(u)$
  $\mathsf{T} \leftarrow \mathsf{T} \cup \{u\}$
  $return\ \mathsf{K}_u$

$\mathsf{DecryptionOracle}(u, c)$
  $\mathsf{D} \leftarrow \mathsf{D} \cup \{(u, c)\}$
  $retreive\ \mathsf{K}_u$
  $return\ \mathsf{Decrypt}(\mathsf{K}_u, c)$
```

```
Experiment $\mathsf{Exp}_{\mathcal{A}}^{BE-KEM}(1^n, 1^\lambda)$
  $(ek, \mathsf{K}_1, \ldots, \mathsf{K}_n) \leftarrow \mathsf{KeyGen}(1^n, 1^\lambda)$
  $\mathsf{T} \leftarrow \emptyset$
  $\mathsf{R} \leftarrow \mathcal{A}^{\mathsf{CorruptOracle}(\cdot),\mathsf{EncryptionOracle}(\cdot),\mathsf{DecryptionOracle}(\cdot)}(\cdot)$
  $b \overset{r}{\leftarrow} \{0, 1\}$
  $m_0, m_1 \overset{r}{\leftarrow} \mathsf{M}$
  $c^* \leftarrow \mathsf{Encrypt}(ek, m_b, \mathsf{R})$
  $b^* \leftarrow \mathcal{A}^{\mathsf{EncryptionOracle}(\cdot)}(c^*, m_1)$
  If $\mathsf{T} \not\subseteq \mathsf{R}$ then output a random bit
  else if $b = b^*$ then return $1$ else $0$;
```

Definition 5.3. *Let Φ be a broadcast encryption scheme with n receivers. We say that the broadcast encryption scheme Φ is KEM-secure if for any PPT adversary $\mathcal{A}$ it holds that*

$$\mathsf{Prob}[\mathsf{Exp}_{\mathcal{A}}^{BE-KEM}(1^n, 1^\lambda) = 1] \leq \frac{1}{2} + \varepsilon,$$

for ε being a negligible function of λ.

Now, we introduce the notion of key-privacy. This notion captures the fact that an entity in possession of none of two keys and given a ciphertext-plaintext pair, cannot distinguish under which key the plaintext is encrypted. The formal definition follows.

```
Experiment $\mathsf{Exp}_{\mathcal{A}}^{\mathsf{key\text{-}priv}}(1^\lambda)$
  Select $k_0 \leftarrow \mathsf{Gen}(1^\lambda)$; $k_1 \leftarrow \mathsf{Gen}(1^\lambda)$
  $aux \leftarrow \mathcal{A}^{\mathsf{Enc}_{k_0}(\cdot),\mathsf{Enc}_{k_1}(\cdot),\mathsf{Dec}_{k_0}(\cdot),\mathsf{Dec}_{k_1}(\cdot)}$
  $m \overset{r}{\leftarrow} \mathsf{M}$
  $b \overset{r}{\leftarrow} \{0, 1\}$;$c \leftarrow \mathsf{Enc}_{k_b}(m)$
  $b^* \leftarrow \mathcal{A}^{\mathsf{Enc}_{k_0}(\cdot),\mathsf{Enc}_{k_1}(\cdot)}(m, c)$
  if $b = b^*$ then return $1$ else $0$;
```

Definition 5.4. *We say that the symmetric encryption scheme* $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *is key-private if for any PPT adversary $\mathcal{A}$ it holds that*

$$\mathsf{Prob}[\mathsf{Exp}_{\mathcal{A}}^{\mathsf{key\text{-}priv}}(1^\lambda) = 1] \leq \frac{1}{2} + \varepsilon,$$

where ε is a negligible function of λ.

Having provided the necessary definitions, let us continue with presenting two instantiations of the class of atomic broadcast encryption schemes.

Scheme 1. This scheme is defined as a tuple of algorithms $(\mathsf{KeyGen}, \mathsf{Encrypt}, \mathsf{Decrypt})$ which are described below. A basic component of the scheme is the underlying symmetric encryption scheme $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$.

- KeyGen : On input $1^n, 1^\lambda$:
  - For any user $i \in [n]$ run the algorithm $\mathsf{Gen}(1^\lambda)$ which generates a key k_i. The encryption key is $ek = \{k_j\}_{j\in[n]}$.
- $\mathsf{Encrypt}$: On input a message m and a revoked set R:
  - By employing the scheme $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ compute a ciphertext tuple c as follows: For each $i \in [n] \setminus \mathsf{R}$ compute $\mathsf{Enc}_{k_i}(m)$. Perform a random permutation f to the ciphertext components which results to a ciphertext tuple of length s, where s is the cardinality of the set $[n] \setminus \mathsf{R}$.
- $\mathsf{Decrypt}$: On input a ciphertext $c = \langle c_1, ..., c_s\rangle$ and a key k_u:
  - Starting from c_1, try to decrypt each ciphertext component under the key k_u. If there exists c_j that is supposed[1] to be decrypted by u, return $\mathsf{Dec}_{k_u}(c_j)$.

Scheme 2. This scheme is defined as a tuple of algorithms $(\mathsf{KeyGen}, \mathsf{Encrypt}, \mathsf{Decrypt})$ which we describe below. A basic component of the scheme is the underlying symmetric encryption scheme $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$.

- KeyGen : On input $1^n, 1^\lambda$:
  - For any user $i \in [n]$ run the algorithm $\mathsf{Gen}(1^\lambda)$ which generates a key k_i. The encryption key is $ek = \{k_j\}_{j\in[n]}$.
- $\mathsf{Encrypt}$: On input a message m and a revoked set R:
  - By employing a scheme $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ compute a ciphertext tuple c of length n as follows: For any user $i \in [n]$, if $i \in \mathsf{R}$ choose randomly a message $m' \in \mathsf{M}$, compute $E_{k_i}(m')$ and place $E_{k_i}(m')$ at the i-th position. If $i \notin \mathsf{R}$, compute $\mathsf{Enc}_{k_i}(m)$ and place it to the i-th position.
- $\mathsf{Decrypt}$: On input a ciphertext $c = \langle c_1, ..., c_n\rangle$ and a key k_u of a user u:
  - Compute $\mathsf{Dec}_{k_u}(c_u)$.

It can be easily observed that Scheme 1 achieves ciphertext length $\Theta(s \cdot \lambda)$ and Scheme 2 achieves ciphertext length $\Theta(n \cdot \lambda)$. The following theorems show that these schemes are priv-eq and priv-full accordingly and therefore meet the lower bounds in the corollaries 5.2, 5.1. As the logic of all the proofs of this section is similar we make some general statements in the appendix that will help our proofs to be more elegant.

[1] In order to determine this *strong* correctness is required; this notion means that applying a wrong key to a ciphertext results to a special fail message to be returned. This can be achieved e.g., by appending a value $H(M)$ to all plaintexts M (here H is a hash function); we omit further details.

Theorem 5.2. *If Scheme 1 satisfies that the underlying scheme* $(\mathsf{Gen}, \mathsf{Dec}, \mathsf{Enc})$ *is key-private then Scheme 1 is private according to the definition* $\mathsf{priv\text{-}eq}$.

Proof. Considering that there exists a PPT adversary $\mathcal{A}$ that breaks $\mathsf{priv\text{-}eq}$ with non-negligible advantage α, we will construct a PPT adversary $\mathcal{B}$ that breaks key privacy. Similarly to the proof of lemma 1, given $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$, we can construct a PPT adversary $\mathcal{A}' = (\mathcal{A}'_1, \mathcal{A}'_2)$ that breaks $\mathsf{priv\text{-}eq}$ definition only for challenged pairs of the form $(P_\ell, P_{\ell+1})$ with advantage at least α/n. To further elaborate, assuming that $(\mathsf{R}_0, \mathsf{R}_1)$ is the challenge of $\mathcal{A}$, we define the sequence $P_0, ..., P_{k-1}$ where $P_0 = \mathsf{R}_0$, $P_{k-1} = \mathsf{R}_1$ and P_ℓ is R_0 with the first ℓ elements of $\mathsf{R}_0 \setminus \mathsf{R}_1$ being replaced with the first ℓ elements of $\mathsf{R}_1 \setminus \mathsf{R}_0$. Thus, the sets $P_\ell, P_{\ell+1}$ differ only at the $(\ell + 1)$-th element. The difference between the corresponding enabled sets $[n] \setminus P_\ell$ and $[n] \setminus P_{\ell+1}$ is that in $[n] \setminus P_\ell$ there is an enabled user in $\mathsf{R}_1 \setminus \mathsf{R}_0$ which is replaced with a user of $\mathsf{R}_0 \setminus \mathsf{R}_1$ in $[n] \setminus P_{\ell+1}$. $\mathcal{A}'_1$ configures his challenge running $\mathcal{A}_1$. Namely, when $\mathcal{A}_1$ outputs $(\mathsf{R}_0, \mathsf{R}_1)$, $\mathcal{A}'_1$ guesses ℓ and challenges $(P_\ell, P_{\ell+1})$. Note that $\mathcal{A}'_2$ and $\mathcal{A}_2$ are the same algorithms. Moreover, when a challenger interacts with $\mathcal{A}'$ in the experiment $\mathsf{priv\text{-}eq}$ the case $b' = 0$ is the case in which the challenger replies with a ciphertext that is supposed to be decrypted by the users in $[n] \setminus P_\ell$. Now, based on $\mathcal{A}'$ we describe the way $\mathcal{B}$ proceeds.

1. $\mathcal{B}$ guesses $i, j \in [n]$ and runs $n-2$ times the algorithm $\mathsf{Gen}(1^\lambda)$ in order to generate the private keys for the other users in $[n]$. We assume that user i owns the key k_1 while user j owns the key k_0.

2. $\mathcal{B}$ runs $\mathcal{A}'_1$. When $\mathcal{A}'_1$ issues a query $u \neq i, j$ to the Corruption Oracle, $\mathcal{B}$ replies returning the corresponding key k_u. If $u = i$ or $u = j$, $\mathcal{B}$ returns 0. If $\mathcal{A}'_1$ issues a query R to the Encryption Oracle with the users i or j not being in the enabled set, $\mathcal{B}$ prepares a ciphertext for a randomly chosen message $m \in \mathsf{M}$ using the generated keys of the first step. If i or j or both are enabled, $\mathcal{B}$ chooses randomly a message $m \in \mathsf{M}$ and issues the query m to the corresponding oracle $\mathsf{Enc}_{k_0}(\cdot)$ or $\mathsf{Enc}_{k_1}(\cdot)$. Prepare the other components using the keys generated in the step 1 and then permute them choosing a random permutation f. $\mathcal{B}$ responds to the Decryption Oracle queries in a similar way, this time by invoking the oracles $\mathsf{Dec}_{k_0}(\cdot), \mathsf{Dec}_{k_1}(\cdot)$ if necessary.

3. $\mathcal{A}'_1$ challenges $P_\ell, P_{\ell+1}$ and $\mathcal{B}$ outputs aux. If $P_\ell, P_{\ell+1}$ are not the sets which differ at the existence of users i, j, with $i \in P_\ell$ and $j \in P_{\ell+1}$, $\mathcal{B}$ outputs 0.

4. Otherwise $\mathcal{B}$ prepares a ciphertext tuple of length $s = |[n] \setminus \mathsf{R}_0|$, encrypting the message m in the received challenge $(m, \mathsf{Enc}_{k_b}(m))$ for the common users of the sets $[n] \setminus P_\ell$ and $[n] \setminus P_{\ell+1}$ and then placing in the appropriate position (according to a random permutation f) the component $\mathsf{Enc}_{k_b}(m)$ of the received challenge . The result is provided to $\mathcal{A}'_2$.

5. $\mathcal{B}$ outputs $\mathcal{A}_2$'s result.

We define as $\mathsf{FAIL} = \{\mathcal{B}$ guesses wrongly the pair i, j at the step $1\}$. It holds that

$$\mathsf{Prob}[\mathsf{FAIL}] = 1 - 1/n^2.$$

1. $\mathcal{B}$ guesses $i, j \in [n]$ and runs $n-2$ times the algorithm $\mathsf{Gen}(1^\lambda)$ in order to generate the private keys for the other users in $[n]$. We assume that a user i owns the key k_1 while user j owns the key k_0.

2. $\mathcal{B}$ runs $\mathcal{A}'_1$. When $\mathcal{A}'_1$ issues a query $u \neq i, j$ to the Corruption Oracle, $\mathcal{B}$ replies returning the corresponding key k_u. If $u = i$ or $u = j$, $\mathcal{B}$ returns 0. If $\mathcal{A}'_1$ issues a query R to the Encryption Oracle with the users i or j not being in the enabled set, $\mathcal{B}$ prepares a ciphertext for a randomly chosen message $m \in \mathsf{M}$ using the generated keys of the first step. If i or j or both are enabled, $\mathcal{B}$ chooses randomly a message $m \in \mathsf{M}$ and issues the query m to the corresponding oracle $\mathsf{Enc}_{k_0}(\cdot)$ or $\mathsf{Enc}_{k_1}(\cdot)$. Prepare the other components using the keys generated in the step 1 and then permute them choosing a random permutation f. $\mathcal{B}$ responds to the Decryption Oracle queries in a similar way, this time by invoking the oracles $\mathsf{Dec}_{k_0}(\cdot), \mathsf{Dec}_{k_1}(\cdot)$ if necessary.

3. $\mathcal{A}'_1$ challenges $P_\ell, P_{\ell+1}$ and $\mathcal{B}$ outputs aux. If $P_\ell, P_{\ell+1}$ are not the sets which differ at the existence of users i, j, with $i \in P_\ell$ and $j \in P_{\ell+1}$, output 0.

4. Otherwise $\mathcal{B}$ prepares a ciphertext tuple of length $s = |[n] \setminus \mathsf{R}_0|$, encrypting the message m in the received challenge $(m, \mathsf{Enc}_{k_b}(m))$ for the common users of the sets $[n] \setminus P_\ell$ and $[n] \setminus P_{\ell+1}$ and then placing in the appropriate position (according to a random permutation f) the component $\mathsf{Enc}_{k_b}(m)$ of the received challenge . The result is provided to $\mathcal{A}'_2$.

5. $\mathcal{B}$ outputs $\mathcal{A}'_2$'s result.

We define as $\mathsf{FAIL} = \{\mathcal{B} \text{ guesses wrongly the pair } i, j \text{ at the step } 1\}$. It holds that

$$\mathsf{Prob}[\mathsf{FAIL}] = 1 - 1/n^2.$$

We define as $\mathsf{FAIL} = \{\mathcal{B} \text{ guesses wrongly the pair } i, j \text{ at the step } 1\}$. It holds that

$$\mathsf{Prob}[\mathsf{FAIL}] = 1 - 1/n^2.$$

If the challenger who interacts with $\mathcal{B}$ in the experiment $\mathsf{Exp}_{\mathcal{B}}^{\mathsf{key\text{-}priv}}$ selects $b = 0$, which means that he replies with $(m, \mathsf{Enc}_{k_0}(m))$, we have that the user j is enabled in ciphertext tuple prepared for $\mathcal{A}'$. This is the case where the prepared ciphertext corresponds to the revoked set P_ℓ which implies that $b' = 0$ in the execution of the experiment $\mathsf{Exp}_{\mathcal{A}'}^{\mathsf{priv\text{-}eq}}$. As a result we have that

$$\mathsf{Prob}[\mathcal{B} \text{ outputs } 0 | b = 0, \neg\mathsf{FAIL}] = \mathsf{Prob}[\mathcal{A}' \text{ outputs } 0 | b' = 0], \tag{5.1}$$

$$\mathsf{Prob}[\mathcal{B} \text{ outputs } 1 | b = 1, \neg\mathsf{FAIL}] = \mathsf{Prob}[\mathcal{A}' \text{ outputs } 1 | b' = 1]. \tag{5.2}$$

Considering the relation (8) we have that

$$\begin{aligned} \mathsf{Prob}[\mathsf{Exp}_{\mathcal{B}}^{\mathsf{key\text{-}priv}}(1^\lambda) = 1] &= \frac{1}{2}\Big(1 - \frac{1}{n^2} + \frac{1}{n^2} \cdot \mathsf{Prob}[\mathcal{A}' \text{ outputs } 0 | b' = 0] + \frac{1}{n^2} \cdot \mathsf{Prob}[\mathcal{A}' \text{ outputs } 1 | b' = 1]\Big) \\ &\geq \frac{1}{2}(1 - \frac{1}{n^2}) + \frac{1}{n^2}(\frac{1}{2} + \frac{\alpha}{n}) = \frac{1}{2} + \frac{\alpha}{n^3}. \end{aligned} \tag{5.3}$$

■

Theorem 5.3. *If Scheme 2 is a broadcast encryption scheme in which the underlying scheme* $(\mathsf{Gen}, \mathsf{Dec}, \mathsf{Enc})$ *is* KEM*-secure, then Scheme 2 is private according to definition* $\mathsf{priv\text{-}full}$.

Proof. Let us assume that Scheme 2 does not satisfy $\mathsf{priv\text{-}full}$ definition. Recall that we have already proven that definitions $\mathsf{priv\text{-}full}$ and $\mathsf{priv\text{-}st}$ are equivalent. Thus, we have that Scheme 2 does not satisfy $\mathsf{priv\text{-}st}$ definition. As a result, there exists a PPT adversary $\mathcal{A}$ such that

$$\mathsf{Prob}[\mathsf{Exp}_{\mathcal{A}}^{\mathsf{priv\text{-}st}}(1^n, 1^\lambda) = 1] \geq \frac{1}{2} + \alpha,$$

for α non-negligible. This implies that

$$\begin{aligned} \mathsf{Prob}[\mathsf{Exp}_{\mathcal{A}}^{\mathsf{priv\text{-}st}}(1^n, 1^\lambda) = 1] &= \frac{1}{2}\big(\mathsf{Prob}[\mathcal{A} \text{ outputs } 1 | b' = 1] + \mathsf{Prob}[\mathcal{A} \text{ outputs } 0 | b' = 0]\big) \\ &\geq \frac{1}{2} + \alpha. \end{aligned} \tag{5.4}$$

Then, we will construct a PPT adversary $\mathcal{B}$ that breaks KEM-security of the underlying scheme $(\mathsf{Gen}, \mathsf{Dec}, \mathsf{Enc})$. $\mathcal{B}$ proceeds as follows:

1. $\mathcal{B}$ guesses $i \in [n]$.

2. For all $j \in [n]$ such that $i \neq j$, $\mathcal{B}$ runs $\mathsf{Gen}(1^\lambda)$ algorithm in order to generate k_j.

3. $\mathcal{B}$ runs $\mathcal{A}$.

4. When $\mathcal{A}$ issues a query $u \neq i$ to the Corruption Oracle, $\mathcal{B}$ returns k_u. If $u = i$, $\mathcal{B}$ returns 0.

5. Whenever $\mathcal{A}$ issues a query R to the Encryption Oracle, $\mathcal{B}$ chooses a message $m \in \mathsf{M}$ and checks whether $i \in \mathsf{R}$. If $i \in \mathsf{R}$, $\mathcal{B}$ chooses randomly a message m' and makes the query m' to $\mathsf{Enc}_k(\cdot)$ in order to obtain $\mathsf{Enc}_k(m')$. $\mathcal{B}$ places the response $\mathsf{Enc}_k(m')$ to the i-th position and then for the other users in R computes the encryption of m' under their private key while for the users in $[n] \setminus \mathsf{R}$ computes the encryption of the message m under their private key. If $i \notin \mathsf{R}$, $\mathcal{B}$ proceeds in the same way with the difference that $\mathcal{B}$ submits the query m to $\mathsf{Enc}_k(\cdot)$. In case $\mathcal{A}$ submits a decryption query (u, c), if $u = i$, $\mathcal{B}$ finds the i-th component c_i and submits the query (i, c_i) to $\mathsf{Dec}_k(\cdot)$ oracle. If $u \neq i$, $\mathcal{B}$ replies using the private key k_u. In case c is malformed $\mathcal{B}$ returns $\perp$.

6. $\mathcal{A}$ challenges $(\mathsf{R}, \mathsf{R} \cup \{u\})$ and $\mathcal{B}$ outputs aux.

7. if $u \neq i$, $\mathcal{B}$ returns 0. As soon as the challenger replies with $(m_1, \mathsf{Enc}_k(m_b))$, $\mathcal{B}$ chooses randomly a message $m' \in \mathsf{M}$ and for all $j \in \mathsf{R}$ prepares $\mathsf{Enc}_{k_j}(m')$. For all $j \notin \mathsf{R}$ with $j \neq i$, $\mathcal{B}$ prepares $\mathsf{Enc}_{k_j}(m_1)$ and places $\mathsf{Enc}_k(m_b)$ at the i-th position of the ciphertext tuple.

8. If $\mathcal{A}$ continues issuing queries to the Encryption Oracle or the Corruption Oracle, $\mathcal{B}$ replies in the same way as in steps 4,5.

9. $\mathcal{A}$ outputs b^*.

10. $\mathcal{B}$ outputs $\overline{b^*}$.

We define the event $\mathsf{FAIL} = \{\mathcal{B} \text{ guesses wrongly the user } i \text{ in step } 1\}$. It holds that

$$\mathsf{Prob}[\mathsf{FAIL}] = \frac{n-1}{n}.$$

If the challenger in the experiment $\mathsf{Exp}_{\mathcal{B}}^{KEM}$ selects $b = 0$, this means that the ciphertext placed in the position i is $\mathsf{Enc}_k(m_0)$, which in turn implies that user i is revoked. Consequently, this is the case $b' = 1$ in the experiment $\mathsf{Exp}_{\mathcal{A}}^{\mathsf{priv\text{-}st}}$. Based on the above arguments, we have that

$$\mathsf{Prob}[\mathcal{B} \text{ outputs } 0|b = 0, \neg\mathsf{FAIL}] = \mathsf{Prob}[\mathcal{A} \text{ outputs } 1|b' = 1], \tag{5.5}$$

$$\mathsf{Prob}[\mathcal{B} \text{ outputs } 1|b = 1, \neg\mathsf{FAIL}] = \mathsf{Prob}[\mathcal{A} \text{ outputs } 0|b' = 0]. \tag{5.6}$$

From the relations (8), (5.4), (5.5),(5.6), we have that

$$\mathsf{Prob}[\mathsf{Exp}_{\mathcal{B}}^{KEM}(1^\lambda) = 1] \geq \frac{n-1}{2n} + \frac{1}{n}\big(\frac{1}{2} + \alpha\big) = \frac{1}{2} + \frac{\alpha}{n}. \tag{5.7}$$

■

Apart from the type of privacy each scheme preserves, it have also to be secure in the KEM-sense as highlighted in the section 3.2.1. As a result, it remains to show that the broadcast encryption schemes Scheme 1 and Scheme 2 are BE-KEM-secure, i.e. they are secure under the definition 5.4. The proofs of security are similar and we prove this only for Scheme 2.

Theorem 5.4. *If the underlying encryption scheme* $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *is KEM-secure then Scheme 1 is BE-KEM secure.*

Proof. Suppose that there exists a PPT adversary $\mathcal{A}$ for which $\mathsf{Prob}[\mathsf{Exp}_{\mathcal{A}}^{BE-KEM}(1^n, 1^\lambda) = 1] \geq \frac{1}{2} + \alpha$, for α non-negligible. We consider a sequence of experiments $\mathsf{Exp}_0^{\mathcal{A}}$, ..., $\mathsf{Exp}_n^{\mathcal{A}}$ where $\mathsf{Exp}_0^{\mathcal{A}}$ is identical to the experiment $\mathsf{Exp}_{\mathcal{A}}^{BE-KEM}$. We define as $\mathsf{Exp}_v^{\mathcal{A}}$ the experiment which operates exactly as $\mathsf{Exp}_0^{\mathcal{A}}$ slightly modified in a way that the first v enabled users are given the encryption of a randomly chosen plaintext under rather that the encryption of the appropriate plaintext. If s is the size of the enabled set, for $v = s, s+1, ..., n$ the experiments are the same. When we refer to the index of a user, we mean the index he is assigned during the Key Generation step.

Now, let $\mathcal{B}$ be an adversary that participates in the experiment $\mathsf{Exp}_{\mathcal{B}}^{KEM}$. $\mathcal{B}$ proceeds as follows:

1. $\mathcal{B}$ guesses i.
2. $\mathcal{B}$ runs $\mathsf{Gen}(1^\lambda)$ $n-1$ times in order to generate the private keys for all the users in $[n]$ except for i.
3. When $\mathcal{A}$ issues a query R to the Encryption Oracle, $\mathcal{B}$ prepares the appropriate ciphertext tuple encrypting a message m using the keys generated in the previous step. If $i \notin \mathsf{R}$, $\mathcal{B}$ asks a query m to $\mathsf{Enc}_k(\cdot)$. $\mathcal{B}$ performs a random permutation f and returns a ciphertext tuple that consists of $|[n] \setminus \mathsf{R}|$ components. We note that each time a query is imposed $\mathcal{B}$ chooses a different permutation. $\mathcal{B}$ responds to the queries imposed to the decryption oracle using the keys generated at the second step and issuing queries to $\mathsf{Dec}(\cdot)$ if necessary. If $\mathcal{A}$ issues a query $u \neq i$ to the Corruption Oracle, $\mathcal{B}$ answers returning the key k_u. If $\mathcal{A}$ issues the query i, $\mathcal{B}$ returns 0.
4. $\mathcal{A}$ outputs R and $\mathcal{B}$ outputs aux.
5. If the first enabled user is not i, then $\mathcal{B}$ outputs 0. Otherwise, according to a random permutation f places the ciphertext c of the received challenge (m_1, c) at the position that corresponds to i. Then, he chooses randomly a message m' from the plaintext space and flips a perfect coin b'. Set $m'_{b'} = m_1$ and $m'_{1-b'} = m'$. Encrypt the message $m_{b'}$ for the enabled users except for i.

6. $\mathcal{A}$ outputs b^*.

7. $\mathcal{B}$ outputs the result of the experiment.

We set $p_0 = \mathsf{Prob}[\mathsf{Exp}_0^{\mathcal{A}} = 1]$ and $p_1 = \mathsf{Prob}[\mathsf{Exp}_1^{\mathcal{A}} = 1]$. Furthermore, we define the event

$$\mathsf{FAIL} = \{\mathcal{B} \text{ guesses wrongly at the first step}\}.$$

We observe that conditionally to $b = 0$ and the fact that $\mathcal{B}$ does not fail, the simulated experiment executed inside $\mathsf{Exp}_{\mathcal{B}}^{KEM}$ is identical to $\mathsf{Exp}_1^{\mathcal{A}}$. This is because in the position of the first enabled user i an encryption of a random plaintext m_0 is placed. On the other side, in case $b = 1$ in the experiment $\mathsf{Exp}_{\mathcal{B}}^{KEM}$ and $\neg\mathsf{FAIL}$, the experiment executed inside $\mathcal{B}$ is identical to $\mathsf{Exp}_0^{\mathcal{A}}$ due to the fact that $\mathsf{Enc}_k(m_1)$ is placed at i-th position. In both cases, the answer provided to $\mathcal{A}$ is $(m_1', \mathsf{Encrypt}(ek, \mathsf{R}, m_{b'}'))$ where $m_{b'}' = m_1$.

Based on the above observations we have that

$$\mathsf{Prob}[\mathsf{Exp}_{\mathcal{B}}^{KEM}(1^\lambda) = 1 | b = 0, \neg\mathsf{FAIL}] = \mathsf{Prob}[\mathsf{Exp}_1^{\mathcal{A}} = 0] = 1 - p_1, \tag{5.8}$$

$$\mathsf{Prob}[\mathsf{Exp}_{\mathcal{B}}^{KEM}(1^\lambda) = 1 | b = 1, \neg\mathsf{FAIL}] = \mathsf{Prob}[\mathsf{Exp}_0^{\mathcal{A}} = 1] = p_0. \tag{5.9}$$

Consequently,

$$\begin{aligned} \mathsf{Prob}[\mathsf{Exp}_{\mathcal{B}}^{KEM}(1^\lambda) = 1] &= \frac{1}{2} \cdot \frac{n-1}{n} + \frac{1}{2} \cdot \big(1 - p_1 + p_0\big) \cdot \frac{1}{n} \\ &= \frac{1}{2} + \frac{1}{2n} \cdot (p_0 - p_1). \end{aligned} \tag{5.10}$$

According to the assumption of the theorem we have that $\mathsf{Prob}[\mathsf{Exp}_{\mathcal{B}}^{KEM}(1^\lambda) = 1] \leq \frac{1}{2} + \varepsilon$. Considering this relation as well as (5.15), it holds that $p_0 - p_1 \leq 2n \cdot \varepsilon$. Applying exactly the same arguments, for every $i \in \{0, 1, \ldots, n-1\}$ we have that $p_i - p_{i+1} \leq 2n \cdot \varepsilon$. Summing all these relations for both sides, we have that

$$p_0 - p_n \leq 2n^2 \cdot \varepsilon. \tag{5.11}$$

Due to the fact that $\mathsf{Exp}_n^{\mathcal{A}}$ is the experiments where all the enabled users, receive an encrypted random plaintext, this implies that $p_n = \mathsf{Prob}[\mathsf{Exp}_n^{\mathcal{A}}(1^\lambda) = 1] = \frac{1}{2}$. As a result, from the relation (5.16) we have that

$$\mathsf{Prob}[\mathsf{Exp}_{\mathcal{B}}^{KEM} = 1] \leq \frac{1}{2} + 2n^2 \cdot \varepsilon. \tag{5.12}$$

This is a contradiction because of our initial assumption and the fact that the factor $2n^2 \cdot \varepsilon$ is negligible.

■

Theorem 5.5. *If the underlying encryption scheme* $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *is KEM-secure then Scheme 2 is BE-KEM secure.*

Proof. Suppose that there exists a PPT adversary $\mathcal{A}$ for which $\mathsf{Prob}[\mathsf{Exp}_{\mathcal{A}}^{BE-KEM}(1^n, 1^\lambda) = 1] \geq \frac{1}{2} + \alpha$, for α non-negligible. We consider a sequence of experiments $\mathsf{Exp}_0^{\mathcal{A}}, ..., \mathsf{Exp}_n^{\mathcal{A}}$ where $\mathsf{Exp}_0^{\mathcal{A}}$ is identical to the experiment $\mathsf{Exp}_{\mathcal{A}}^{BE-KEM}$. We define as $\mathsf{Exp}_v^{\mathcal{A}}$ the experiment which operates exactly as $\mathsf{Exp}_0^{\mathcal{A}}$ slightly modified in a way that the first v enabled user to be given the encryption of a randomly chosen plaintext under rather that the encryption of the appropriate plaintext. If s is the size of the enabled set, for $v = s, s+1, ..., n$ the experiments are defined indentically.

Now, let $\mathcal{B}$ be an adversary of the experiment $\mathsf{Exp}_{\mathcal{B}}^{KEM}$. $\mathcal{B}$ operates as follows:

1. $\mathcal{B}$ guesses i.
2. $\mathcal{B}$ runs $\mathsf{Gen}(1^\lambda)$ $n-1$ times in order to generate the private keys for all the users in $[n]$ except for i.
3. When $\mathcal{A}$ issues a query to the encryption oracle, $\mathcal{B}$ answers using the keys generated in the previous step. In case i is enabled in the query $\mathcal{B}$ asks $\mathsf{Enc}_k(\cdot)$ and places the answer to the i-th position. $\mathcal{B}$ acts similarly to the queries issued to the decryption oracle. If $\mathcal{A}$ issues a query $u \neq i$ to the Corruption Oracle, $\mathcal{B}$ answers returning the key k_u generated in the step 2. If $\mathcal{A}$ issues the query i, $\mathcal{B}$ returns 0.
4. $\mathcal{A}$ outputs R and $\mathcal{B}$ outputs aux.
5. If the first enabled user is not i, then $\mathcal{B}$ outputs 0. Otherwise he places the ciphertext c of the received challenge (m_1, c) at the i-th position. Then, he chooses randomly a message m' from the plaintext space M and flips a perfect coin b'. Set $m'_{b'} = m_1$ and $m'_{1-b'} = m'$. Encrypt the message $m_{b'}$ for the enabled users except for i, while for the revoked ones encrypt a message m'', randomly chosen form the plaintext space.
6. $\mathcal{A}$ outputs b^*.
7. $\mathcal{B}$ outputs the result of the experiment.

We set $p_0 = \mathsf{Prob}[\mathsf{Exp}_0^{\mathcal{A}} = 1]$ and $p_1 = \mathsf{Prob}[\mathsf{Exp}_1^{\mathcal{A}} = 1]$. Furthermore, we define the event

$$\mathsf{FAIL} = \{\mathcal{B} \text{ the guess at the first step is wrong }\}.$$

We observe that conditionally to $b = 0$ and the fact that $\mathcal{B}$ does not fail, the simulated experiment executed inside $\mathsf{Exp}_{\mathcal{B}}^{KEM}$ is identical to $\mathsf{Exp}_1^{\mathcal{A}}$. This is because in the position of the first enabled user i an encryption of a random plaintext m_0 is placed. On the other side, in case $b = 1$ in the experiment $\mathsf{Exp}_{\mathcal{B}}^{KEM}$ and $\neg\mathsf{FAIL}$, the experiment executed inside $\mathcal{B}$ is identical to $\mathsf{Exp}_0^{\mathcal{A}}$ due to the fact that $\mathsf{Enc}_k(m_1)$ is placed at i-th position. In both cases, the answer provided to $\mathcal{A}$ is $(m'_1, \mathsf{Encrypt}(ek, \mathsf{R}, m'_{b'}))$ where $m'_{b'} = m_1$.

Based on the above observations we have that

$$\mathsf{Prob}[\mathsf{Exp}_{\mathcal{B}}^{KEM}(1^\lambda) = 1 | b = 0, \neg\mathsf{FAIL}] = \mathsf{Prob}[\mathsf{Exp}_1^{\mathcal{A}} = 0] = 1 - p_1, \tag{5.13}$$

$$\mathsf{Prob}[\mathsf{Exp}_{\mathcal{B}}^{KEM}(1^\lambda) = 1 | b = 1, \neg\mathsf{FAIL}] = \mathsf{Prob}[\mathsf{Exp}_0^{\mathcal{A}} = 1] = p_0. \tag{5.14}$$

Consequently,

$$\begin{aligned}\mathsf{Prob}[\mathsf{Exp}_{\mathcal{B}}^{KEM}(1^\lambda) = 1] &= \frac{1}{2} \cdot \frac{n-1}{n} + \frac{1}{2} \cdot \big(1 - p_1 + p_0\big) \cdot \frac{1}{n} \\ &= \frac{1}{2} + \frac{1}{2n} \cdot (p_0 - p_1). \end{aligned} \tag{5.15}$$

According to the assumption of the theorem we have that $\mathsf{Prob}[\mathsf{Exp}_{\mathcal{B}}^{KEM}(1^\lambda) = 1] \leq \frac{1}{2} + \varepsilon$. Considering this relation as well as (5.15), it holds that $p_0 - p_1 \leq 2n \cdot \varepsilon$. Applying exactly the same arguments, for every $i \in \{0, 1, \ldots, n-1\}$ we have that $p_i - p_{i+1} \leq 2n \cdot \varepsilon$. Summing all these relations for both sides, we have that

$$p_0 - p_n \leq 2n^2 \cdot \varepsilon. \tag{5.16}$$

Due to the fact that $\mathsf{Exp}_n^{\mathcal{A}}$ is the experiments where all the enabled users, receive an encrypted random plaintext, this implies that $p_n = \mathsf{Prob}[\mathsf{Exp}_n^{\mathcal{A}}(1^\lambda) = 1] = \frac{1}{2}$. As a result, from the relation (5.16) we have that

$$\mathsf{Prob}[\mathsf{Exp}_{\mathcal{B}}^{KEM} = 1] \leq \frac{1}{2} + 2n^2 \cdot \varepsilon. \tag{5.17}$$

This is a contradiction because of our initial assumption and the fact that the factor $2n^2 \cdot \varepsilon$ is negligible. ■

5.2 Lower bounds for general Broadcast Encryption schemes

We now turn our attention to the setting of general, unrestricted broadcast encryption schemes. We will prove that any scheme that is private in the sense of priv-st, priv-full has ciphertext length that with reasonably high probability is linear. First of all, the following theorem is necessary. It captures the requirement that the ciphertext length must not expose any information about the revoked set in a private broadcast encryption scheme. We denote as $|x|$, the number of bits of the value x.

Theorem 5.6. *For all the sets* $\mathsf{R} \subseteq [n]$*, we define the random variable*

$$S_{\mathsf{R}} : \mathsf{Encrypt}(ek, m, \mathsf{R}) \rightarrow |\mathsf{Encrypt}(ek, m, \mathsf{R})|,$$

where ek is an encryption key and m is a plaintext chosen from a message space M. *Suppose that Φ is a broadcast encryption scheme with n receivers, and let* R, R' *be two sets. If Φ is private according to* priv-full *definition, then for all* $\mathsf{R}, \mathsf{R}' \subseteq [n]$ *and for all the PPT statistical tests D, it holds that* $\Delta_D[S_{\mathsf{R}}, S_{\mathsf{R}'}] < \varepsilon$.

Proof. Suppose that there exists a pair of sets R, R' and a PPT statistical test D such that $\Delta_D[S_{\mathsf{R}}, S_{\mathsf{R}'}] \geq \alpha$, with α non-negligible. Then, there is a PPT adversary $\mathcal{A}$ that breaks definition priv-full with advantage at least $\alpha/2$ following the steps below.
Phase 1:

- Challenge R, R'

Phase 2: On input $\langle m, \mathsf{Encrypt}(ek, m, \mathsf{R}_b)\rangle$

- Compute $|\mathsf{Encrypt}(ek, m, \mathsf{R}_b)|$.
- Run D on input $|\mathsf{Encrypt}(ek, m, \mathsf{R}_b)|$.
- Return the output of D.

The adversary can execute the algorithm D a number of times in order to understand whether it is biased to 1 on input S_{R} or vice versa. Without loss of generality we assume that D returns 1 with greater probability in case it takes as input $|\mathsf{Encrypt}(ek, m, \mathsf{R}')|$. As a result, we have that

$$\mathsf{Prob}[D(S_{\mathsf{R}'}) = 1] - \mathsf{Prob}[D(S_{\mathsf{R}}) = 1] \geq \alpha.$$

We note that if $\mathcal{A}$ is biased to 1 on input $S_{\mathsf{R}'}$ we can consider the adversary $\overline{\mathcal{A}}$ in oder to obtain the same results.

$$\begin{aligned}\mathsf{Prob}[\mathsf{Exp}_{\mathcal{A}}^{\mathsf{priv\text{-}full}}(1^{\lambda})=1] &= \frac{1}{2}\Big(\mathsf{Prob}[\mathsf{Exp}_{\mathcal{A}}^{\mathsf{priv\text{-}full}}=1|b=0]+\mathsf{Prob}[\mathsf{Exp}_{\mathcal{A}}^{\mathsf{priv\text{-}full}}=1|b=1]\Big)\\ &= \frac{1}{2}\Big(\mathsf{Prob}[D(S_{\mathsf{R}})=0]+\mathsf{Prob}[D(S'_{\mathsf{R}})=1]\Big)\\ &= \frac{1}{2}\Big(1-\mathsf{Prob}[D(S_{\mathsf{R}})=1]+\mathsf{Prob}[D(S_{\mathsf{R}'})=1]\Big)\\ &\geq \frac{1}{2}+\frac{\alpha}{2}. \end{aligned} \tag{5.18}$$

■

Next, we will prove a lower bound on the ciphertext size that any private broadcast encryption scheme can achieve. Our proof is based on a standard information theoretic fact (cf. [7]), which is presented below:

Fact 5.7. *Suppose there is a randomized procedure $Enc : \{0,1\}^n \times \{0,1\}^r \to \{0,1\}^m$ and a decoding procedure $Dec : \{0,1\}^m \times \{0,1\}^r \to \{0,1\}^n$ such that*

$$\mathsf{Prob}_{r\in U_r}[Dec(Enc(x,r),r)=x]\geq\delta.$$

Then, $m \geq n - \log\frac{1}{\delta}$.

Let Φ be a broadcast encryption scheme with n receivers. For every $\mathsf{R}\subseteq[n]$ we define the event $E_{\mathsf{R},i}$ as

$$(\mathsf{Decrypt}(\mathsf{SK}_i,c)\neq m\wedge i\notin\mathsf{R})\vee(\mathsf{Decrypt}(\mathsf{SK}_i,c)=m\wedge i\in\mathsf{R}),$$

where $c=\mathsf{Encrypt}(ek,m,\mathsf{R})$. Observe that this event combines the correctness error and a condition that violates the security of the scheme. Thus, in any useful broadcast encryption scheme we anticipate that this event will happen with small probability.

We next prove that an upper bound on the probability of this event implies a lower bound on a certain ciphertext length distribution.

Theorem 5.8. *Let Φ be a broadcast encryption scheme with n receivers and let $\varepsilon(\lambda)$ be the upper bound of all the probabilities $\mathsf{Prob}[E_{\mathsf{R},i}]$. If for any λ there exists some β for which $\varepsilon(\lambda)<\frac{1}{2n}-\frac{\beta}{n}$, then there exists a set $\mathsf{R}\subseteq[n]$ such that $\mathsf{Prob}[S_{\mathsf{R}}\geq n]>\beta$.*

Proof. Recall the definition of S_{R}:

$$S_{\mathsf{R}}:\mathsf{Encrypt}(ek,m,\mathsf{R})\to|\mathsf{Encrypt}(ek,m,\mathsf{R})|.$$

We define a procedure f which is an encoding procedure of a set $\mathsf{R}\subseteq[n]$, while f^{-1} is a decoding procedure. The procedure f is a randomized procedure that takes two arguments $\rho\in\{0,1\}^r$ and $\mathsf{R}\subseteq[n]$. We note that ρ depends on the security parameter λ and represents all the coins needed in order for the system to setup and the encryption encryption.

$f(\rho,\mathsf{R})$:

1. Using ρ, compute a message m and the key ek which will be used by the encryption algorithm

2. Compute $\mathsf{Encrypt}(ek, m, \mathsf{R})$.

3. If $|\mathsf{Encrypt}(ek, m, \mathsf{R})| \geq n$, output 0^{n-1} else $\mathsf{Encrypt}(ek, m, \mathsf{R})$.

We denote as ψ the output of the procedure f. Regarding the above randomized encoding procedure, there exists a decoding procedure which is described below:

```
$f^{-1}(\psi, \rho)$:
1. Use $\rho$ to compute $\mathsf{SK}_1, ..., \mathsf{SK}_n$.
2. Execute the following algorithm:
   $\mathsf{R} := \emptyset$.
   For $i = 1$ to $n$
           if $\mathsf{Decrypt}(\mathsf{SK}_i, \psi) \neq m$ then $\mathsf{R} := \mathsf{R} \cup \{i\}$ else $\mathsf{R}$.
```

Considering the definition of the decoding procedure, we say that f^{-1} fails when its result is $\mathsf{R}' \neq \mathsf{R}$, given that R is the encoded set. This happens either in case an event $E_{\mathsf{R},i}$ takes place or the output of f is 0^{n-1}. With δ we denote the probability that the procedure f^{-1} succeeds.

In order to prove the theorem, we assume that for any λ for which there exists a β such that $\varepsilon(\lambda) < \dfrac{1}{2n} - \dfrac{\beta}{n}$ it holds that for all $\mathsf{R} \subseteq [n]$, $\mathsf{Prob}[S_{\mathsf{R}} \geq n] \leq \beta$. Let us define a fixed a value λ. From the assumption we have already made, it holds that $\mathsf{Prob}[f \text{ outputs } 0^{n-1}] \leq \beta$ which subsequently means that $\mathsf{Prob}[f^{-1} \text{ fails }] \leq n \cdot \varepsilon(\lambda) + \beta$. Consequently, we have that $\delta \geq 1 - n \cdot \varepsilon(\lambda) - \beta$.

Due to the fact that the length of the encoding produced by f^{-1} is always $n-1$ bits at most, using the fact 5.7, we have that

$$n - 1 \geq n - \log \frac{1}{\delta} \Rightarrow \varepsilon(\lambda) \geq \frac{1}{2n} - \frac{\beta}{n}, \tag{5.19}$$

which is a contradiction. ■

Lemma 2. *Let Φ be a private broadcast encryption scheme with n receivers and a security parameter λ for which $\beta < 1/2$ and β non-negligible as a function of λ. Then, for all $\mathsf{R} \subseteq [n]$, it holds that $\mathsf{Prob}[S_{\mathsf{R}} \geq n] \geq \alpha$, for α non-negligible.*

Proof. We assume that there exists a set R_0 such that $\mathsf{Prob}[S_{\mathsf{R}_0} \geq n] < \delta$, where δ is a negligible function of λ. We construct the following statistical test D:

D: On input S_{R}: If $S_{\mathsf{R}} \geq n$ return 1 else return 0.

According to the Theorem 5.8, we have that there exists a set R_1 for which

$$\mathsf{Prob}[S_{\mathsf{R}_1} \geq n] > \beta.$$

As a result, we have that

$$\mathsf{Prob}[D(S_{\mathsf{R}_1}) = 1] - \mathsf{Prob}[D(S_{\mathsf{R}_0}) = 1] > \beta - \delta,$$

which is non-negligible. This contradicts to Theorem 5.6. ■

Corollary 5.3 (Lower bound for general private broadcast encryption schemes). *For any broadcast encryption scheme Φ which is private in the sense of definition* priv-full, priv-st, *the ciphertext is of length $\Omega(n + k)$.*

The additive factor k stems from the fact that at least one ciphertext should be present in the encryption of a message m for any enabled set S.

Chapter 6

Conclusion

In this thesis, we studied the problem of broadcast encryption. At first, we presented the categories of broadcast encryption schemes, dividing them into combinatorial and structured schemes. We next provided several constructions that apply to both classes of broadcast encryption schemes, analyzing at the same time the security requirements that they satisfy. Then, we turned our focus to the feature of privacy in the setting of broadcast encryption which has not yet received much attention. Having described the work that has been conducted with regard to this feature, we continued with presenting some new results in this area. The detailed presentation of these results was the main part of this thesis.

The provided lower bounds highlight the high costs that privacy may incur for the case of atomic broadcast encryption schemes. The fact that privacy for atomic schemes requires a linear number of ciphertexts in the number of users, leaves little room for improvement in terms of the ciphertext size. If the objective is to attain full privacy, this result suggests that our attention should be turned to non-atomic schemes. For this case our lower bound is much weaker. It is thus an interesting open problem to design a fully private scheme with sublinear ciphertext size (or prove that such scheme is impossible).

Bibliography

[1] AACS , `http://www.aacsla.com/`.

[2] Michel Abdalla, Mihir Bellare, Dario Catalano, Eike Kiltz, Tadayoshi Kohno, Tanja Lange, John Malone-Lee, Gregory Neven, Pascal Paillier, and Haixia Shi. Searchable encryption revisited: Consistency properties, relation to anonymous ibe, and extensions. *J. Cryptology*, 21(3):350--391, 2008.

[3] Nuttapong Attrapadung and Hideki Imai. Practical broadcast encryption from graph-theoretic techniques and subset-incremental-chain structure. *IEICE Transactions*, 90-A(1):187--203, 2007.

[4] Adam Barth, Dan Boneh, and Brent Waters. Privacy in encrypted content distribution using private broadcast encryption. In Giovanni Di Crescenzo and Aviel D. Rubin, editors, *Financial Cryptography*, volume 4107 of *Lecture Notes in Computer Science*, pages 52--64. Springer, 2006.

[5] Mihir Bellare, Alexandra Boldyreva, Anand Desai, and David Pointcheval. Key-privacy in public-key encryption. In Colin Boyd, editor, *ASIACRYPT*, volume 2248 of *Lecture Notes in Computer Science*, pages 566--582. Springer, 2001.

[6] Dan Boneh, Craig Gentry, and Brent Waters. Collusion resistant broadcast encryption with short ciphertexts and private keys. In Victor Shoup, editor, *CRYPTO*, volume 3621 of *Lecture Notes in Computer Science*, pages 258--275. Springer, 2005.

[7] Anindya De, Luca Trevisan, and Madhur Tulsiani. Time space tradeoffs for attacks against one-way functions and prgs. In Tal Rabin, editor, *CRYPTO*, volume 6223 of *Lecture Notes in Computer Science*, pages 649--665. Springer, 2010.

[8] Cécile Delerablée. Identity-based broadcast encryption with constant size ciphertexts and private keys. In Kaoru Kurosawa, editor, *ASIACRYPT*, volume 4833 of *Lecture Notes in Computer Science*, pages 200--215. Springer, 2007.

[9] Yevgeniy Dodis and Nelly Fazio. Public key broadcast encryption for stateless receivers. In Joan Feigenbaum, editor, *Digital Rights Management Workshop*, volume 2696 of *Lecture Notes in Computer Science*, pages 61--80. Springer, 2002.

[10] Yevgeniy Dodis, Nelly Fazio, Aggelos Kiayias, and Moti Yung. Scalable public-key tracing and revoking. In *PODC*, pages 190--199, 2003.

[11] Nelly Fazio and Irippuge Milinda Perera. Outsider-anonymous broadcast encryption with sublinear ciphertexts. In Fischlin et al. [13], pages 225--242.

[12] Amos Fiat and Moni Naor. Broadcast encryption. In Douglas R. Stinson, editor, *CRYPTO*, volume 773 of *Lecture Notes in Computer Science*, pages 480--491. Springer, 1993.

[13] Marc Fischlin, Johannes Buchmann, and Mark Manulis, editors. *Public Key Cryptography - PKC 2012 - 15th International Conference on Practice and Theory in Public Key Cryptography, Darmstadt, Germany, May 21-23, 2012. Proceedings*, volume 7293 of *Lecture Notes in Computer Science*. Springer, 2012.

[14] Craig Gentry and Alice Silverberg. Hierarchical id-based cryptography. *IACR Cryptology ePrint Archive*, 2002:56, 2002.

[15] 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, 2004.

[16] 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, 2002.

[17] Jeremy Horwitz and Ben Lynn. Toward hierarchical identity-based encryption. In Lars R. Knudsen, editor, *EUROCRYPT*, volume 2332 of *Lecture Notes in Computer Science*, pages 466--481. Springer, 2002.

[18] Jonathan Katz and Yehuda Lindell. *Introduction to Modern Cryptography*. Chapman and Hall/CRC Press, 2007.

[19] Aggelos Kiayias and Serdar Pehlivanoglu. *Encryption for Digital Content*, volume 52 of *Advances in Information Security*. Springer, 2010.

[20] Aggelos Kiayias and Katerina Samari. Lower bounds for private broadcast encryption. In *Information Hiding*, 2012.

[21] Benoît Libert, Kenneth G. Paterson, and Elizabeth A. Quaglia. Anonymous broadcast encryption: Adaptive security and efficient constructions in the standard model. In Fischlin et al. [13], pages 206--224.

[22] Dalit Naor, Moni Naor, and Jeffery Lotspiech. Revocation and tracing schemes for stateless receivers. In Joe Kilian, editor, *CRYPTO*, volume 2139 of *Lecture Notes in Computer Science*, pages 41--62. Springer, 2001.

[23] Moni Naor and Benny Pinkas. Efficient trace and revoke schemes. *Int. J. Inf. Sec.*, 9(6):411--424, 2010.

[24] Adi Shamir. Identity-based cryptosystems and signature schemes. In G. R. Blakley and David Chaum, editors, *CRYPTO*, volume 196 of *Lecture Notes in Computer Science*, pages 47--53. Springer, 1984.

[25] Victor Shoup. A proposal for an iso standard for public key encryption. *IACR Cryptology ePrint Archive*, 2001:112, 2001.

[26] Pan Wang, Peng Ning, and Douglas S. Reeves. Storage-efficient stateless group key revocation. In Kan Zhang and Yuliang Zheng, editors, *ISC*, volume 3225 of *Lecture Notes in Computer Science*, pages 25--38. Springer, 2004.

Appendix

The following lemma that helps us to avoid repeating the same arguments in many proofs.

Lemma 3. *Let $\mathcal{B}$ be an algorithm that outputs a bit and* $\mathsf{Exp}_{\mathcal{B}}$ *an algorithm that depends on $\mathcal{B}$ and outputs a bit, too. Also, $\mathcal{B}$'s objective is to predict a bit b that is computed from the view of* $\mathsf{Exp}_{\mathcal{B}}$*.* $\mathsf{Exp}_{\mathcal{B}}$ *returns 1 if and only if $\mathcal{B}$ predicts the correct bit. Then,*

$$\mathsf{Prob}[\mathsf{Exp}_{\mathcal{B}} = 1] = \mathsf{Prob}[\mathcal{B} \textit{ outputs } 0]\mathsf{Prob}[b = 0] + \mathsf{Prob}[\mathcal{B} \textit{ outputs } 1]\mathsf{Prob}[b = 1]. \tag{1}$$

Proof. The lemma is derived directly from the definition of $\mathsf{Exp}_{\mathcal{B}}$ and the Law of total probability. ■

For the cases we consider in this work, the algorithm chooses uniformly at random the bit b. As a result, it holds that

$$\begin{aligned}\mathsf{Prob}[\mathsf{Exp}_{\mathcal{B}} = 1] &= \mathsf{Prob}[\mathcal{B} \text{ outputs } 0|b = 0]\mathsf{Prob}[b = 0] + \mathsf{Prob}[\mathcal{B} \text{ outputs } 1|b = 1]\mathsf{Prob}[b = 1]\\ &= \frac{1}{2}\Big(\mathsf{Prob}[\mathcal{B} \text{ outputs } 0|b = 0] + \mathsf{Prob}[\mathcal{B} \text{ outputs } 1|b = 1]\Big). \end{aligned} \tag{2}$$

Now, we consider an event FAIL which represents a wrong guess of $\mathcal{B}$ during the execution of the experiment in each proof. Thus, we have that

$$\begin{aligned}\mathsf{Prob}[\mathcal{B} \text{ outputs } 0|b = 0] &= \mathsf{Prob}[\mathcal{B} \text{ outputs } 0|b = 0, \mathsf{FAIL}]\mathsf{Prob}[\mathsf{FAIL}]\\ &+ \mathsf{Prob}[\mathcal{B} \text{ outputs } 0|b = 0, \neg\mathsf{FAIL}]\mathsf{Prob}[\neg\mathsf{FAIL}]. \end{aligned} \tag{3}$$

$$\begin{aligned}\mathsf{Prob}[\mathcal{B} \text{ outputs } 1|b = 1] &= \mathsf{Prob}[\mathcal{B} \text{ outputs } 1|b = 1, \mathsf{FAIL}]\mathsf{Prob}[\mathsf{FAIL}]\\ &+ \mathsf{Prob}[\mathcal{B} \text{ outputs } 1|b = 1, \neg\mathsf{FAIL}]\mathsf{Prob}[\neg\mathsf{FAIL}]. \end{aligned} \tag{4}$$

From the relations (3), (4), (2) we conclude that

$$\begin{aligned}\mathsf{Prob}[\mathsf{Exp}_{\mathcal{B}} = 1] = &\frac{\mathsf{Prob}[\mathsf{FAIL}]}{2}\Big(\mathsf{Prob}[\mathcal{B} \text{ outputs } 0|b = 0, \mathsf{FAIL}] + \mathsf{Prob}[\mathcal{B} \text{ outputs } 1|b = 1, \mathsf{FAIL}]\Big)\\ + &\frac{\mathsf{Prob}[\neg\mathsf{FAIL}]}{2}\Big(\mathsf{Prob}[\mathcal{B} \text{ outputs } 0|b = 0, \neg\mathsf{FAIL}] + \mathsf{Prob}[\mathcal{B} \text{ outputs } 1|b = 1, \neg\mathsf{FAIL}]\Big). \end{aligned} \tag{5}$$

Due to the fact that all the algorithms we will consider always output 0 when the event FAIL takes place, we have that

$$\mathsf{Prob}[\mathcal{B} \text{ outputs } 0|b=0,\mathsf{FAIL}] = 1, \tag{6}$$

$$\mathsf{Prob}[\mathcal{B} \text{ outputs } 1|b=1,\mathsf{FAIL}] = 0. \tag{7}$$

Consequently,

$$\begin{aligned}\mathsf{Prob}[\mathsf{Exp}_{\mathcal{B}}=1] = \frac{\mathsf{Prob}[\mathsf{FAIL}]}{2} + \frac{\mathsf{Prob}[\neg\mathsf{FAIL}]}{2}\Big(&\mathsf{Prob}[\mathcal{B} \text{ outputs } 0|b=0,\neg\mathsf{FAIL}]\\ &+\mathsf{Prob}[\mathcal{B} \text{ outputs } 1|b=1,\neg\mathsf{FAIL}]\Big).\end{aligned} \tag{8}$$