Anonymous Broadcast Encryption for Small Universe and Applications in Trace and Revoke Systems

Abstract. Broadcast Encryption is a fundamental cryptographic primitive which allows a sender to send a message to any chosen target set of users. While many efficient constructions for broadcast encryption have been proposed, it is still an open question to efficiently construct an Anonymous Broadcast Encryption (AnoBE) which can hide the target set. The current state of the art AnoBE schemes from Barth, Boneh and Waters at FC '06 and from Libert, Paterson and Quaglia at PKC '12 are constructed from public key encryption and present a linear loss: the ciphertext rate between their proposed AnoBE schemes for N users and the underlying public-key encryption is N. In a restricted case for 2-user only, Phan, Pointcheval and Strefler gave a 2-user AnoBE scheme which reduced the rate from 2 to 1.5.

In this paper, we improve the state of the art for AnoBE, constructing an AnoBE scheme for k users from LWE assumption. This scheme achieves the optimal rate as it is as efficient as the underlying LWE public-key encryption. The construction comes from a simple trick, which switches the tracing algorithm in Ling, Phan, Stehlé and Steinfeld's paper at CRYPTO '14 into a broadcast encryption. The security of our scheme relies on the hardness of the $k-$LWE problem and can only support k users. We named it thus *Anonymous Broadcast Encryption for Small Universe* – AnoBES.

Efficient systems for a small universe play an important role in combinatorial traitor tracing: the code-based traitor tracing schemes, put forth by Kiayias and Yung at EUROCRYPT '02, are constructed from the combination of systems for small universe in integrating with traceability codes, namely collusion-secure code or identifying parent property code (IPP Code). By integrating our AnoBES system with IPP code in the same manner, one can immediately get an traitor tracing scheme. We can achieve a much more relevant objective which achieves both the functionality of traceability and revocation: a Trace and Revoke scheme. Our technique relies on the recent notion of Robust IPP code, introduced by Barg and Kabatiansky at IEEE IT '13. However, Barg and Kabatiansky only proved the existance of the robust IPP codes but did not provide explicit construction. We finally give two concrete and explicit constructions of the robust IPP codes which are suitable for our generic scheme, considering as parameters both the number of traitors and the number of revoked users.

Keywords. Anonymous broadcast encryption, Robust IPP Code, Trace and Revoke system.

1 Introduction

Broadcast Encryption. Broadcast encryption is a cryptographic primitive designed to efficiently distribute an encrypted content via a public channel to a designated set of users so that only privileged users can decrypt and the others users can learn nothing about the content. The first constructions of broadcast encryption were proposed by Berkovits [6] and Fiat-Naor [14]. Since then, many interesting schemes are proposed, in particular Boneh, Gentry and Waters [8] introduced a scheme with a constant size ciphertext.

Anonymous Broadcast Encryption. Privacy is important in real-life application but unfortunately, it is very difficult to hide the target set in broadcast encryption. Even though very efficient broadcast encryption were designed, no efficient anonymous broadcast encryption is constructed.

Anonymity in broadcast encryption system was considered in many works such as [5], [18], [13], [23]. The state of the art constructions from Libert 's one [5, 18] start from a public-key encryption (PKE) and result in a scheme with the ciphertext size is N times the ciphertext size of the underlying PKE scheme. Kiayias and Samari [16] proved the lower bound: ciphertext size of any anonymous broadcast encryption is $\Omega(s \cdot n)$, where s is the cardinality of the set of enabled users and n is security parameter, and $\Omega(N + n)$ for any set of revoked users $\mathcal{R}$. Hence a complexity bound sub-linear in the number of users is impossible to be achieved if the number of users N is large.

In case $N = 2$, Phan *et al.* [23] provided a construction for 2-user anonymous broadcast encryption scheme in which the ciphertext length is only approximate 1.5 times the ciphertext size of underlying ElGamal encryption scheme. We will consider in this paper the case where N is small (*i.g,* as small as the security parameter) that we call anonymous broadcast encryption for small universe (AnoBES).

Anonymous broadcast encryption for small universe and traitor tracing. Traitor tracing (TT) [12] is a cryptographic primitive used to broadcast content only to a set of authorized users, with an additional tracing property: if registered users share their secrets to allow a pirate to access the content, one can trace back at least one of these traitors. Unlike broadcast encryption, the set of legitimate users is fixed.

As we know from [8] that, any anonymous broadcast encryption (in fact, they proved this for a more restricted case of anonymity, called private linear broadcast encryption) supports tracing traitor. Therefore,

any solution for AnoBES directly implies an traitor tracing scheme for small universe.

Combinatorial methods of designing a traitor tracing consist in two steps: first, construct a small scheme, then combine these schemes to achieve a general one. This methis is proposed in the very first traitor tracing paper of Chor-Fiat-Naor [12]. In this work, we concentrate on code-based traitor tracing. Kiayias and Yung [17] integrated a 2-user traitor tracing scheme with a collusion-secure code [10] into the TT scheme. This method can be summarized as follows. Firstly, a 2-user traitor tracing scheme can be trivially obtained from applying a public-key encryption (PKE) twice, each for one user. Now, a message or a session key is divided into ℓ sub-keys. The sender essentially then encrypts each sub-key twice with PKE and gets sub-ciphertexts. Each recipient, provided sub-keys associated to a codeword of a collusion-secure code, can decrypt one of the two sub-ciphertexts for each sub-key and thus recover the whole message or session key which will be used to encrypt data. The tracing procedure consists in using the traceability in each 2-user scheme to extract a word associated with the pirate decoder. Thanks to the tracing capability of the collusion-secure code, one can then trace back one of the traitors.

This method is then generalized for $q-$ary IPP code by intergrating it with a q-user traitor tracing scheme. The q-user traitor tracing scheme can also be obtained from applying q times PKE [22]. Now, if we have AnoBES for q-user which is as efficient as the underlying PKE, we can save a factor q in the efficiency. Therefore, the design of an efficient AnoBES has an impact on the IPP code-based TT.

Trace and Revoke System with Public Traceability. As shown in the paper of Boneh and Waters (BW) at [11], traceability and revocation are very difficult to be combined. There exists only a few trace-and-revoke systems with public traceability, where the tracing procedure can be done from public tracing key. Algebraic schemes have only been achieved by Boneh and Waters, and quite recently by [27] (embeds a collusion secure code into a broadcast system), Nishimaki, Wichs and Zhandry (NWZ) [21] and by Agrawal et al. [1]. The BW and NZW schemes are quite powerful in that it supports malicious collusions of unbounded size but its ciphertexts are very large (in BW, their size grows proportionally to $\sqrt{N}$, where N is the total number of users and in NWZ, they use general functional encryption schemes). For bounded schemes where the number of traitors is small, the Agrawal et al. 's scheme is quite efficient but they only support

a weak level of tracing: black-box confirmation which the assumption that the tracer gets a suspect set that contains all the traitors.

Combinatorial schemes are considered in [20] and in [2]. The paper [20] only considered a weak form of black-box tracing while Ak et al. [2] gave a generic transformation which maps a broadcast encryption to a trace and revoke scheme and thus suffers the factor $\sqrt{N}$ in the ciphertext size. Code-based schemes are also bounded schemes but enjoy a nice property of supporting black-box tracing [17], [9], [20], [7]. Although we know that the binary collusion secure code is well suitable for traitor tracing, its shortcoming is the incapacity of supporting the revocation. In fact, to revoke a group of users, the authority has to disable the ability to decrypt of sub-keys in each position of the revoked group. In using the binary collusion secure code scenario, there are only two possibilities for sub-key of each position. Whenever the authority executes the revocation procedure, a large number of the legitimate users will be affected and cannot decrypt anymore. A non-trivial remendy is that the designer of system can choose a code with big alphabet for example $q-$ary IPP code instead of a binary collusion secure code with alphabet size two. The coincidence of sub-keys in each position then will decrease slightly. Certainly, in this case, the possibility that legitimate users will be excluded outside the system with revoked users must also be taken into account. The secret sharing scheme is a mechanism that allows us to think about an solution: a legitimate user only needs to have a certain fraction of the sub-keys to be able to recover the original message. But this makes also an advantage for the pirates: they become more stronger as they do not need to put the whole sub-keys in the pirate decoder. This means that they are permitted to delete sub-keys. Thanks to the introduction of the robust IPP of Barg *et al.* [4] which allows to identify parents even if some positions are intentionally erased, we propose a new generic method to design a trace and revoke system from robust IPP codes and AnoBES. As in the previous code-based method, the ciphertext size of the trace and revoke system is proportional to the length of the code and the ciphertext size of the AnoBES.

Contributions. We present three main results:

1. It has been an open question to generalize a PKE to an anonymous BE scheme for small universe (AnoBES for short) of ($N > 2$) users with a ciphertext rate strictly less than N. We show that we can transform LWE PKE into an AnoBES with an optimal rate. The security of our proposed schemes for k users relies on the $k-$ LWE problem [19].

2. A new method to design a trace and revoke system from an AnoBES, a secret sharing scheme and a robust IPP code. Previous methods of integrating a q-user traitor tracing with a $q-$ary IPP code result in a traitor tracing scheme. It is worth remarking that the robust IPP code, introduced by Barg et al. [4], is an interesting generalization of IPP code but to the best of our knowledge, it has not found yet any application in cryptography.
3. The use of the robust IPP code in our scheme is non black-box as we deal not only with the number of traitors but also with the number of revoked users. It is also important to remark that, in [4], the proof of existence of robust IPP codes was given but there was no explicit construction, neither the analysis of the complexity length of the code. We propose two concrete and explite construction of robust IPP codes which are suitable to be integrated with our AnoBES in the above trace and revoke system.

Our techniques. Ling *et al.* [19] introduced the first lattice-based traitor tracing scheme. There scheme is based on the $k-$LWE assumption. As they showed a polynomial-time reduction from $k-$LWE to LWE, their scheme is as efficient as the LWE encryption. A natural question is why one cannot directly rely on their scheme to design an anonymous revoke or broadcast encryption scheme. Revoking users is a very difficult task and the following simple question is still open: for a constant number of revoked users, can we design a revoke scheme that is comparably efficient as the underlying encryption. Based on $k-$LWE, it seems very hard because for revocation, essentially one need to find a vector that is "orthogonal" to all the secret vectors of the non-revoked users (so that they get the same message) and this is impossible for a large universe system. Now, concerning broadcast encryption, whenever relying on $k-$LWE, one cannot allow the adversary to corrupt more than k-users, where $k \leq m$ is bounded by the underlying lattice dimension. Therefore, at best, one can target for an anonymous broadcast encryption for a small universe. But as we discussed above, achieveng an efficient AnoBES scheme is challenging task.

Surprisingly, our construction of a AnoBES scheme comes from a simple trick: switching tracing procedure in [19] to be functional as a broadcast encryption. We first recall that, in the LPSS traitor tracing scheme of Ling et al. [19], the linear tracing technique [12] was applied: to detect a traitor in a group of suspect users, they first create a ciphertext so that every user in this group can decrypt successfully the ciphertext. In the subsequent steps, the tracer will disable one by one an user in the group from decrypting the ciphertext. We observe that if we switch the suspected

users in LPSS scheme to the legitimate users and the removed users in the suspected set to the revoked users, then we will get a broadcast encryption. Because the LPSS traitor tracing can deal with a small number (less than the dimension of the underlying lattice) of traitors, we also get a broadcast encryption for a small number of users, that we call broadcast encryption for small universe.

The main technical difficulty is to prove the anonymous property of this broadcast encryption. The anonymity requires that an adversary cannot distinguish between encryptions for two targets $\mathcal{S}_0, \mathcal{S}_1$ of its choice. If we consider an outsider adversary, defined in [13], who only corrupts users outside both $\mathcal{S}_0, \mathcal{S}_1$, then the proof is quite direct because from the $k - \mathsf{LWE}$ assumption, the encryption for $\mathcal{S}_0$ or for $\mathcal{S}_1$ looks like a random ciphertext for the adversary. It is more challenging to consider a general adversary who can also corrupt the key in the intersection of $\mathcal{S}_0$ and $\mathcal{S}_1$. Fortunately, we can exploit an intermediate theorem in [19] which informally states that the encryptions for a set $\mathcal{S}$ and for a set $\mathcal{S} \cup \{i\}$ are indistinguishable if the adversary does not corrupt the user i, even if the adversary corrupts users in $\mathcal{S}$. Thanks to this result, we can apply an hybrid argument to move an encryption for the set $\mathcal{S}_0$ (or $\mathcal{S}_1$) to an encryption for the set $\mathcal{S}_0 \cup \mathcal{S}_1$ by adding one by one user in $\mathcal{S}_1 \setminus \mathcal{S}_0$ (or in $\mathcal{S}_1 \setminus \mathcal{S}_0$, respectively).

We now explain how to construct a Trace and Revoke system (TR) from an AnoBES. As explained above, any AnoBES for q-user can be integrated with a $q-$ary IPP code to produce an traitor tracing scheme. More precisely, a message or a session key is divided into ℓ sub-keys and each sub-key is encrypted with the AnoBES scheme. A robust code allows one to trace a traitor from a pirate word even if some among ℓ positions are missing. Such a robust code gives us a possibility to remove users: the encryption is designed such that legitimate users can decrypt more than a fraction $\rho\ell$ of sub-keys and the revoked users, even colluded together, can only decrypt less than $\rho\ell$ sub-keys. By using then an $(\rho\ell, \ell)$-secret sharing scheme, we can get a revoke scheme. The tracing relies on the anonymity of AnoBES and robust IPP code: one can do tracing from each AnoBES scheme and then get a pirate word which contains significant non-empty positions. This is sufficient to perform tracing in the robust IPP code.

2 Preliminaries

2.1 Secret sharing scheme

A secret sharing scheme ($\mathcal{SSS}$) is to distribute a secret amongst a group of users, each of whom keeps a share of the secret. The $\mathcal{SSS}$ contains 2 algorithms: Share and Combine. It is defined formally as follows:

Definition 1 ($(m,n)-$Secret Sharing Scheme).
Share(k,m,n)*: Takes as input* 3 *positive integers* k,m,n*. It outputs a secret bit string* K *of length* k*, as well as* n *shares* $s_1,\ldots,s_n$*, so that any* m *of them will allow to recover* K*.*
Combine$(\{(i,s_i)\})$*: Takes as input* m *pairs* $\{(i,s_i)\}$*. It outputs the bit string* K*.*

The correctness requirement is that from any m-subset of $\{(i,s_i)\}$ generated by Share(k,m,n), the Combine algorithm outputs the bit string K generated by Share. Furthermore, the bit string K must be perfectly uniformly distributed.

2.2 $k-$LWE problem

For two matrices A,B of compatible dimensions, let $(A|B)$ and $(A\|B)$ (or sometimes $\left(\frac{A}{B}\right)$) respectively denote the horizontal and vertical concatenations of A and B. For $A\in\mathbb{Z}_q^{m\times n}$, define $\mathrm{Im}(A)=\{A\boldsymbol{s}\mid \boldsymbol{s}\in\mathbb{Z}_q^n\}\subseteq\mathbb{Z}_q^m$. For $X\subseteq\mathbb{Z}_q^m$, let $\mathrm{Span}(X)$ denote the set of all linear combinations of elements of X and define $X^\perp$ to be $\{\boldsymbol{b}\in\mathbb{Z}_q^m\mid\forall\boldsymbol{c}\in X,\langle\boldsymbol{b},\boldsymbol{c}\rangle=0\}$.

If D_1 and D_2 are distributions over a countable set X, their statistical distance $\frac{1}{2}\sum_{x\in X}|D_1(x)-D_2(x)|$ will be denoted by $\Delta(D_1,D_2)$. We also let $U(X)$ denote the uniform distribution over X.

Let $\mathbf{B}=\{\boldsymbol{b}_1,\boldsymbol{b}_2,\ldots,\boldsymbol{b}_n\}\subset\mathbb{R}^n$ consist of n linearly independent vectors. The n-dimensional lattice Λ generated by the basis $\mathbf{B}$ is

$$\Lambda=L(\mathbf{B})=\{\mathbf{B}\boldsymbol{c}=\sum_{i\in[n]}c_i\cdot\boldsymbol{b}_i\mid\boldsymbol{c}\in\mathbb{Z}^n\}.$$

For a lattice $L\subseteq\mathbb{R}^m$ and an invertible matrix $S\in\mathbb{R}^{m\times m}$, we define the Gaussian distribution of parameters L and S by $D_{L,S}(\boldsymbol{b})\sim\rho_S(\boldsymbol{b})=\exp(-\pi\|S^{-1}\boldsymbol{b}\|^2)$ for all $\boldsymbol{b}\in L$.

The q-ary lattice associated to a matrix $A\in\mathbb{Z}_q^{m\times n}$ is defined as $\Lambda^\perp(A)=\{\boldsymbol{x}\in\mathbb{Z}^m:\boldsymbol{x}^t\cdot A=\mathbf{0}\bmod q\}$. It has dimension m, and a basis can be computed in polynomial-time from A. For $\boldsymbol{u}\in\mathbb{Z}_q^m$, we define $\Lambda_{\boldsymbol{u}}^\perp(A)$ as the coset $\{\boldsymbol{x}\in\mathbb{Z}^m:\boldsymbol{x}^t\cdot A=\boldsymbol{u}^t\bmod q\}$ of $\Lambda^\perp(A)$.

Lemma 2 (Theorem 3.1, [3]). *There is a probabilistic polynomial-time algorithm that, on input positive integers $n, m, q \geq 2$, outputs two matrices $A \in \mathbb{Z}_q^{m \times n}$ and $T \in \mathbb{Z}^{m \times m}$ such that the distribution of A is within statistical distance $2^{-\Omega(n)}$ from $U(\mathbb{Z}_q^{m \times n})$; the rows of T form a basis of $\Lambda^{\perp}(A)$; each row of T has norm $\leq 3mq^{n/m}$.*

Lemma 3 (GPV algorithm, [15]). *There exists a probabilistic polynomial-time algorithm that given a basis $\mathbf{B}$ of an n-dimensional lattice $\Lambda = L(\mathbf{B})$, a parameter $s \geq \|\tilde{\mathbf{B}}\| \cdot \omega\sqrt{\log n}$ [1], output a sample from a distribution that is statistically close to $D_{L,s}$.*

Definition 4. *Let $S \in \mathbb{R}^{m \times m}$ be an invertible matrix and denote $\mathbb{T}^{m+1} = (\mathbb{R}/\mathbb{Z})^{m+1}$. The $(k, S)-\mathsf{LWE}$ problem is: given $A \hookleftarrow U(\mathbb{Z}_q^{m \times n})$, $\boldsymbol{u} \hookleftarrow U(\mathbb{Z}_q^n)$ and $\boldsymbol{x}_i \hookleftarrow D_{\Lambda^{\perp}_{-\boldsymbol{u}}(A),S}$ for $i \leq k \leq m$, the goal is to distinguish between the distributions (over $\mathbb{T}^{m+1}$)*

$$\frac{1}{q} \cdot U\Big(\mathrm{Im}\Big(\frac{\boldsymbol{u}^t}{A}\Big)\Big) + \nu_\alpha^{m+1} \quad \textit{and} \quad \frac{1}{q} \cdot U\Big(\mathrm{Span}_{i \leq k}(\mathbf{1} \| \boldsymbol{x}_i)^{\perp}\Big) + \nu_\alpha^{m+1},$$

where ν_α denotes the one-dimensional Gaussian distribution with standard deviation $\alpha > 0$.

In [19], it was shown that this problem can be reduced to LWE problem for a specific class of diagonal matrices S. In our work, we only need any such S that $(k, S) - \mathsf{LWE}$ is hard, and thus the use of S is implicit. For simplicity, we will use $k - \mathsf{LWE}$ and $(k, S) - \mathsf{LWE}$ interchangeably in this paper.

2.3 Anonymous Broadcast Encryption

Let $\mathcal{PT}$ and $\mathcal{CT}$ denote the plaintext and ciphertext space, respectively. We briefly recall the definition of broadcast encryption in [18]. A broadcast encryption scheme consists of four algorithms:

$\mathsf{Setup}(\lambda, N)$**:** Takes as input the security parameter λ, it generates the global parameters param of the system, including N the maximal number of users (receivers are implicitly represented by integers in $\mathcal{U} = \{1, \ldots, N\}$), and outputs a public key ek and a master secret key MSK.

[1] $\tilde{\mathbf{B}}$ is Gram-Schmidt orthogonalization of $\mathbf{B}$.

$\mathsf{Extract}(\mathsf{ek}, \mathsf{MSK}, i)$**:** Take as input the public key ek, the master secret key MSK and a user index $i \in \mathcal{U}$, the algorithm extracts the decryption key dk_i which is sent to the user i.

$\mathsf{Encrypt}(\mathsf{ek}, m, S)$**:** Take as input the public key ek, a message $M \in \mathcal{PT}$ and a set of previleged users $S \subseteq \mathcal{U}$, outputs a ciphertext $c \in \mathcal{CT}$, which is broadcasted to every member of S.

$\mathsf{Decrypt}(\mathsf{ek}, \mathsf{dk}_i, c)$**:** Take as input the public key ek, the decryption key dk_i of user i and a ciphertext $c \in \mathcal{CT}$. If $i \in S$, the algorithm outputs a message $M \in \mathcal{PT}$.

For correctness, we require that for all $S \subseteq \mathcal{U}$ and all $i \in \mathcal{U}$, if $c = \mathsf{Encrypt}(\mathsf{ek}, M, S)$ and dk_i is the decryption key for user $i \in S$, one then should get $M = \mathsf{Decrypt}(\mathsf{ek}, \mathsf{dk}_i, c)$ with overwhelming probability.

Semantic security against adaptive corruptions. We give the definition of semantic security by considering a security game between an adversary $\mathcal{A}$ and a challenger $\mathcal{B}$ as follows:

The challenger $\mathcal{B}$ runs algorithm $\mathsf{Setup}(\lambda, N)$ to obtain a public key ek and master secret key MSK and sends ek to adversary $\mathcal{A}$.

Phase 1. The adversary $\mathcal{A}$ adaptively issues decryption key extraction queries for any index $i \in \mathcal{U}$. The challenger runs $\mathsf{Extract}$ algorithm on index i and returns $\mathcal{A}$ a decryption key $\mathsf{dk}_i = \mathsf{Extract}(\mathsf{ek}, \mathsf{MSK}, i)$.

Challenger. The adversary chooses two messages $M_0, M_1 \in \mathcal{PT}$ and a set $\mathcal{S} \subset \mathcal{U}$ of users. We require that every index that were queried before does not include in $\mathcal{S}$. The adversary $\mathcal{A}$ passes M_0, M_1 and $\mathcal{S}$ to the challenger $\mathcal{B}$. The challenger $\mathcal{B}$ randomly chooses a bit $b \in \{0, 1\}$ and computes $c = \mathsf{Encrypt}(\mathsf{ek}, M_b, \mathcal{S})$ and gives c to the adversary $\mathcal{A}$.

Phase 2. $\mathcal{A}$ continues to adaptively issue decryption keys extraction queries on the others indices outside $\mathcal{S}$.

Guess. The adversary outputs a guess $b' \in \{0, 1\}$ and wins the game if $b' = b$.

We say that a broadcast scheme is semantically secure ($\mathsf{IND} - \mathsf{CPA}$) if all polynomial time adaptive adversaries $\mathcal{A}$ have at most negligible advantage in the above game, where $\mathcal{A}$'s advantage is defined as $\mathsf{Succ}^{\mathsf{IND-CPA}}(\mathcal{A}) = \Pr[b' = b]$, the probability that $\mathcal{A}$ wins the game

$$\mathsf{Adv}^{\mathsf{IND-CPA}}(\mathcal{A}) = |\mathsf{Succ}^{\mathsf{IND-CPA}}(\mathcal{A}) - \frac{1}{2}| = |\Pr[b' = b] - \frac{1}{2}|.$$

In our scheme will be constructed in the next section, the messages M_0, M_1 are only one bit.

Anonymity. A broadcast system called have anonymity property (AnoBE system) if it allows to address a message to a subset of the users, without revealing this previleged set even to users who successfully decrypted the message. When the number of users in our system is small, we have the notation *anonymous broadcast encryption for Small Universe* – AnoBES. We follow the definition in [18]. We define the ANOCPA security games for an anonymous broadcast encyption scheme as follows

The challenger $\mathcal{B}$ runs algorithm $\mathsf{Setup}(\lambda, N)$ to obtain a public key ek and master secret key MSK and sends ek to adversary $\mathcal{A}$.

Phase 1. The adversary $\mathcal{A}$ adaptively issues decryption key extraction queries for any index $i \in \mathcal{U}$. The challenger runs $\mathsf{Extract}$ algorithm on index i and returns $\mathcal{A}$ a decryption key $\mathsf{dk}_i = \mathsf{Extract}(\mathsf{ek}, \mathsf{MSK}, i)$.

Challenger. The adversary chooses a message $M \in \mathcal{PT}$ and two distinct subsets $S_0, S_1 \subset \mathcal{U}$ of users with the same size. We require that $\mathcal{A}$ has not issued key queries for any index $i \in S_0 \triangle S_1 = (S_0 \setminus S_1) \cup (S_1 \setminus S_0)$. The adversary $\mathcal{A}$ passes M and S_0, S_1 to the challenger $\mathcal{B}$. The challenger $\mathcal{B}$ randomly chooses a bit $b \in \{0, 1\}$, computes $c = \mathsf{Encrypt}(\mathsf{ek}, M, S_b)$ and sends c to $\mathcal{A}$.

Phase 2. $\mathcal{A}$ adaptively issues decryption key extraction queries on indices $i \notin S_0 \triangle S_1$ and obtains a decryption key dk_i.

Guess. The adversary outputs a guess $b' \in \{0, 1\}$ and wins the game if $b' = b$.

We denote by $\mathsf{Succ}^{\mathsf{ANOCPA}}(\mathcal{A}) = \Pr[b' = b]$ the probability that $\mathcal{A}$ wins the game, and its advantage is

$$\mathsf{Adv}^{\mathsf{ANOCPA}}(\mathcal{A}) = |\mathsf{Succ}^{\mathsf{ANOCPA}}(\mathcal{A}) - \frac{1}{2}| = |\Pr[b' = b] - \frac{1}{2}|.$$

We say that the Π scheme is *anonymous secure against chosen plaintext attacks* – ANOCPA if all polynomial-time adversaries $\mathcal{A}$ have at most negligible advantage in the above game.

3 $k-$LWE Based Broadcast Encryption for Small Universe

The present section will be first devoted to construct a broadcast encryption for small universe scheme (AnoBES) from $k-$LWE problem. Let N be the maximal number of users (receivers are implicitly represented by integers in $\mathcal{U} = \{1, \ldots, N\}$) and $\mathcal{S} \subset \mathcal{U}$. Given a security parameter n, we suppose that parameters q, m, α, S are chosen so that the $(k, S) -$ LWE

problem is hard to solve as presented in [19]. We require that $N \leq k$, thus the systems works for small universe only. Our build will enable any privilege users inside $\mathcal{S}$ might decrypt successfully a ciphertext which is distributed from broadcaster.

$\mathsf{Setup}(n, N)$**:** Take as input the security parameter n and maximal number of users N. Broadcaster uses the algorithm as in Lemma 2 to generate 2 matrices $(A, T) \in \mathbb{Z}_q^{m \times n} \times \mathbb{Z}^{m \times m}$ and picks $\boldsymbol{u}$ uniformly in $\mathbb{Z}_q^n$. We set a master secret key $\mathsf{MSK} = (A, T)$ and a public key $\mathsf{ek} = A^+$, with $A^+ = (\boldsymbol{u}^t \| A)$.

$\mathsf{Extract}(\mathsf{ek}, \mathsf{MSK}, j)$**:** Take as input the public key ek, the master secret key MSK and a user index $j \in \mathcal{U}$, the algorithm calls GPV algorithm as in Lemma 3 using the basis $\Lambda^{\perp}(A)$ consisting of the rows of T and the standard deviation matrix S. The broadcaster obtains a sample $\boldsymbol{x}_j$ from $D_{\Lambda^{\perp}_{-\boldsymbol{u}}(A), S}$. The algorithm outputs decryption key $\mathsf{dk}_j = \boldsymbol{x}_j^+ := (1 \| \boldsymbol{x}_j) \in \mathbb{Z}^{m+1}$ for user j.

$\mathsf{Encrypt}(\mathsf{ek}, M, \mathcal{S})$**:** Take as input the public key ek, a message $M \in \mathcal{PT} = \{0, 1\}$ and a set of users $\mathcal{S} \subseteq \mathcal{U}$. To encrypt M, choose a vector $\boldsymbol{y} \in \mathbb{Z}_q^{m+1}$ from the distribution $U\left(\mathrm{Span}_{i \in \mathcal{S}}(\boldsymbol{x}_i^+)^{\perp}\right)$, $\boldsymbol{e} \hookleftarrow \lfloor \nu_{\alpha q} \rceil^{m+1}$ and outputs $\boldsymbol{c} \in \mathcal{CT}$, which is broadcasted to every member of $\mathcal{S}$ as follows:

$$\boldsymbol{c} = \boldsymbol{y} + \boldsymbol{e} + \left(\frac{M \lfloor q/2 \rfloor}{\boldsymbol{0}}\right),$$

whereas $\lfloor x \rfloor$ denotes the greatest integer less than or equal to x.

$\mathsf{Decrypt}(\mathsf{ek}, \mathsf{dk}_j, \boldsymbol{c})$**:** Take as input the public key ek, a decryption key $\mathsf{dk}_j = \boldsymbol{x}_j^+$ of user j and a ciphertext $\boldsymbol{c} \in \mathcal{CT}$. The function $\mathsf{Decrypt}$ will return 0 if $\langle \boldsymbol{x}_j^+, \boldsymbol{c} \rangle$ is closer 0 than to $\lfloor q/2 \rfloor$ modulo q, otherwise output 1.

Correctness. We require that for given subset $\mathcal{S} \subseteq \mathcal{U}$ and all $j \in \mathcal{S}$, if $\boldsymbol{c} = \mathsf{Encrypt}(\mathsf{ek}, m, \mathcal{S})$ and dk_j is the decryption key for user $j \in \mathcal{S}$, we then recover $M = \mathsf{Decrypt}(\mathsf{ek}, \mathsf{dk}_j, \boldsymbol{c})$ with overwhelming probability. Indeed, for each user $j \in \mathcal{S}$ and $\boldsymbol{y} \hookleftarrow U\left(\mathrm{Span}_{i \in \mathcal{S}}(\boldsymbol{x}_i^+)^{\perp}\right)$, we have $\langle \boldsymbol{x}_j^+, \boldsymbol{y} \rangle = 0$. Therefore,

$$\begin{aligned} \langle \boldsymbol{x}_j^+, \boldsymbol{c} \rangle &= \langle \boldsymbol{x}_j^+, \boldsymbol{y} \rangle + \langle \boldsymbol{x}_j^+, \boldsymbol{e} \rangle + \langle \boldsymbol{x}_j^+, \left(\frac{M \lfloor q/2 \rfloor}{\boldsymbol{0}}\right) \rangle \mod q \\ &= \langle \boldsymbol{x}_j^+, \boldsymbol{e} \rangle + M \lfloor q/2 \rfloor \mod q, \end{aligned}$$

where $\boldsymbol{e} \hookleftarrow \lfloor \nu_{\alpha q} \rceil^{m+1}$. According to arguments in [19], the quantity $\langle \boldsymbol{x}_j^+, \boldsymbol{e} \rangle$ is relative small modulo q with overwhelming probability. The user will apply the function Decrypt and obtain the original message with overwhelming probability. Therefore, for every such user in $\mathcal{S}$ can decrypt sucessfully that ciphertext.

We now consider the security of the scheme: an adversary which is allowed to corrupt any user outside $\mathcal{S}$, cannot break the semantic security of the scheme.

Theorem 5. *Under the assumption that the* $k - \mathsf{LWE}$ *problem is hard, for any* $N \leq k$*, the* AnoBES *scheme* Π *constructed as above is* $\mathsf{IND} - \mathsf{CPA}$ *secure.*

Proof. We leave the proof in the appendix A.1.

We next consider anonymous property of the AnoBES scheme which is stated and proved in following Theorem.

Theorem 6. *Under the assumption that the* $k - \mathsf{LWE}$ *problem is hard, for any* $N \leq k$*, our scheme is* ANOCPA*-secure.*

Proof. We will prove by considering a sequence of games as follows:

Game $\mathbf{G}_0$: This is the real game security defined as in the security model. We repeat the interaction between the challenger $\mathcal{B}$ and the adversary $\mathcal{A}$ as following

Setup. The challenger uses the algorithm as in Lemma 2 to generate a matrix $A \hookleftarrow U(\mathbb{Z}_q^{m \times n})$ and picks $\boldsymbol{u}$ uniformly in $\mathbb{Z}_q^n$. Then the public key is set $\mathsf{ek} = A^+$, with $A^+ = (\boldsymbol{u}^t \| A)$ and given to $\mathcal{A}$.

Phase 1. When $\mathcal{A}$ asks decryption key for user i, $\mathcal{B}$ replies $\boldsymbol{x}_i^+ = (1 || \boldsymbol{x}_i)$, whereas $\boldsymbol{x}_i \hookleftarrow D_{\Lambda_{-\boldsymbol{u}}^{\perp}(A),S}$.

Challenger phase. $\mathcal{A}$ chooses 2 subsets $\mathcal{S}_0, \mathcal{S}_1$, a message M and sends to $\mathcal{B}$. The challenger picks randomly $b \in \{0, 1\}$ and gives $\mathcal{A}$ a ciphertext $\boldsymbol{c}$ taken from one of 2 distributions (distribution $\mathcal{D}_b$, over $\mathbb{T}^{m+1}$)

$$\mathcal{D}_0 = U\Big(\mathrm{Span}_{i \in \mathcal{S}_0}(\boldsymbol{x}_i^+)^{\perp}\Big) + \lfloor \nu_{\alpha q} \rceil^{m+1} + \begin{pmatrix} M\lfloor q/2 \rfloor \\ 0 \end{pmatrix},$$

$$\mathcal{D}_1 = U\Big(\mathrm{Span}_{i \in \mathcal{S}_1}(\boldsymbol{x}_i^+)^{\perp}\Big) + \lfloor \nu_{\alpha q} \rceil^{m+1} + \begin{pmatrix} M\lfloor q/2 \rfloor \\ 0 \end{pmatrix}.$$

Phase 2. In this step, $\mathcal{A}$ continues querying to get decryption keys with the limitations as mentioned before (query indices $i \in \Big(\mathcal{U} - (\mathcal{S}_0 \triangle \mathcal{S}_1)\Big)$). $\mathcal{B}$ gets $\boldsymbol{x}_i^+$ from $D_{\Lambda_{-\boldsymbol{u}}^{\perp}(A),S}$ and answers $\mathcal{A}$.

Guess. $\mathcal{A}$ gives a guess b' for b.

Game $\mathbf{G}_1$: In this game, the inputs and the settings of this game are identical to the ones of **Game $\mathbf{G}_0$**. In challenger phase, the adversary $\mathcal{A}$ received a ciphertext from one of two following distributions:

$$\mathcal{D}_0 = U\Big(\mathrm{Span}_{i\in\mathcal{S}_0}(\boldsymbol{x}_i^+)^\perp\Big) + \lfloor\nu_{\alpha q}\rceil^{m+1} + \Big(\frac{M\lfloor q/2\rfloor}{0}\Big),$$

$$\mathcal{D}_1^1 = U\Big(\mathrm{Span}_{i\in(\mathcal{S}_1\cup\{j_1\})}(\boldsymbol{x}_i^+)^\perp\Big) + \lfloor\nu_{\alpha q}\rceil^{m+1} + \Big(\frac{M\lfloor q/2\rfloor}{0}\Big),$$

where $\boldsymbol{x}_{j_1}^+ \hookleftarrow D_{\Lambda^\perp_{-\boldsymbol{u}}(A),S}$, $j_1 \in \mathcal{S}_0 - \mathcal{S}_1$.

Here we notice that the adversary $\mathcal{A}$ does not know the key $\boldsymbol{x}_{j_1}^+$ because $\mathcal{A}$ only can choose the keys with index in $\mathcal{U} - (\mathcal{S}_0 \triangle \mathcal{S}_1)$. Since $k - \mathsf{LWE}$ is hard, by applying Theorem 25 [19], two distributions

$$\mathcal{D}_1 = U\Big(\mathrm{Span}_{i\in\mathcal{S}_1}(\boldsymbol{x}_i^+)^\perp\Big) + \lfloor\nu_{\alpha q}\rceil^{m+1} + \Big(\frac{M\lfloor q/2\rfloor}{0}\Big),$$

$$\mathcal{D}_1^1 = U\Big(\mathrm{Span}_{i\in(\mathcal{S}_1\cup\{j_1\})}(\boldsymbol{x}_i^+)^\perp\Big) + \lfloor\nu_{\alpha q}\rceil^{m+1} + \Big(\frac{M\lfloor q/2\rfloor}{0}\Big),$$

are indistinguishable. This means that the difference between the advantage of $\mathcal{A}$ in **Game $\mathbf{G}_1$** and **Game $\mathbf{G}_0$** is negligible.

Game $\mathbf{G}_\tau$: We assume that $\kappa = |\mathcal{S}_0 - \mathcal{S}_1|$ and $\mathcal{S}_0 - \mathcal{S}_1 = \{j_1, j_2, \ldots, j_\kappa\}$. For each $2 \le \tau \le \kappa$, we consider a sequence of $\kappa - 1$ games. We set $\mathcal{T}_1 = \mathcal{S}_1 \cup \{j_1\}$ and $\mathcal{T}_\tau = \mathcal{T}_{\tau-1} \cup \{j_\tau\}$. It implies $\mathcal{T}_\kappa = \mathcal{S}_0 \cup \mathcal{S}_1$. In each game in this sequence, the inputs and the settings of this game are identical to the ones of previous games. In challenger phase, the adversary $\mathcal{A}$ received a ciphertext from one of two following distributions:

$$\mathcal{D}_0 = U\Big(\mathrm{Span}_{i\in\mathcal{S}_0}(\boldsymbol{x}_i^+)^\perp\Big) + \lfloor\nu_{\alpha q}\rceil^{m+1} + \Big(\frac{M\lfloor q/2\rfloor}{0}\Big)$$

$$\mathcal{D}_1^\tau = U\Big(\mathrm{Span}_{i\in\mathcal{T}_\tau}(\boldsymbol{x}_i^+)^\perp\Big) + \lfloor\nu_{\alpha q}\rceil^{m+1} + \Big(\frac{M\lfloor q/2\rfloor}{0}\Big),$$

Since the adversary $\mathcal{A}$ does not know the any key in the set $\boldsymbol{x}_{j_\tau}^+$ and the problem $k - \mathsf{LWE}$ is hard, we apply Theorem 25 [19], two distributions

$$\mathcal{D}_1^{\tau-1} = U\Big(\mathrm{Span}_{i\in\mathcal{T}_{\tau-1}}(\boldsymbol{x}_i^+)^\perp\Big) + \lfloor\nu_{\alpha q}\rceil^{m+1} + \Big(\frac{M\lfloor q/2\rfloor}{0}\Big)$$

$$\mathcal{D}_1^\tau = U\Big(\mathrm{Span}_{i\in\mathcal{T}_\tau}(\boldsymbol{x}_i^+)^\perp\Big) + \lfloor\nu_{\alpha q}\rceil^{m+1} + \Big(\frac{M\lfloor q/2\rfloor}{0}\Big),$$

are indistinguishable for each τ. This means that the difference between the advantage of $\mathcal{A}$ in the sequence of games **Game** $\mathbf{G}_\tau$, $1 \leq \tau \leq \kappa$ is negligible.

Game $\mathbf{G}_{\kappa+\eta}$**:** We assume that $\iota = |\mathcal{S}_1 - \mathcal{S}_0|$ and $\mathcal{S}_1 - \mathcal{S}_0 = \{j_1, j_2, \ldots, j_\iota\}$. For each $1 \leq \eta \leq \iota$, we consider a sequence of ι games. We set $\mathcal{T}'_1 = \mathcal{S}_0 \cup \{j_1\}$ and $\mathcal{T}'_\eta = \mathcal{T}'_{\eta-1} \cup \{j_\eta\}$. It implies $\mathcal{T}'_\iota = \mathcal{S}_0 \cup \mathcal{S}_1$. In each game in this sequence, the inputs and the settings of this game are identical to the ones of previous games. In challenger phase, the adversary $\mathcal{A}$ received a ciphertext from one of two following distributions:

$$\mathcal{D}_0^\eta = U\Big(\mathrm{Span}_{i\in\mathcal{T}'_\eta}(\boldsymbol{x}_i^+)^\perp\Big) + \lfloor \nu_{\alpha q} \rceil^{m+1} + \Big(\frac{M\lfloor q/2 \rfloor}{0}\Big),$$

$$\mathcal{D}_1^\kappa = U\Big(\mathrm{Span}_{i\in(\mathcal{S}_0\cup\mathcal{S}_1)}(\boldsymbol{x}_i^+)^\perp\Big) + \lfloor \nu_{\alpha q} \rceil^{m+1} + \Big(\frac{M\lfloor q/2 \rfloor}{0}\Big).$$

It means that we keep fix the distribution $\mathcal{D}_1^\kappa$ and replace the distribution $\mathcal{D}_0$ by

$$\mathcal{D}_0^\eta = U\Big(\mathrm{Span}_{i\in\mathcal{T}'_\eta}(\boldsymbol{x}_i^+)^\perp\Big) + \lfloor \nu_{\alpha q} \rceil^{m+1} + \Big(\frac{M\lfloor q/2 \rfloor}{0}\Big),$$

we set $\mathcal{D}_0^\eta = \mathcal{D}_0$. By the same argument as in previous games, in the view of the adversary $\mathcal{A}$, two distributions $\mathcal{D}_0^{\eta-1}$ and $\mathcal{D}_0^\eta$ are indistinguishable under the hardness of $k - \mathsf{LWE}$, this means that two following distributions

$$\mathcal{D}_0^{\eta-1} = U\Big(\mathrm{Span}_{i\in\mathcal{T}'_{\eta-1}}(\boldsymbol{x}_i^+)^\perp\Big) + \lfloor \nu_{\alpha q} \rceil^{m+1} + \Big(\frac{M\lfloor q/2 \rfloor}{0}\Big)$$

$$\mathcal{D}_0^\eta = U\Big(\mathrm{Span}_{i\in\mathcal{T}'_\eta}(\boldsymbol{x}_i^+)^\perp\Big) + \lfloor \nu_{\alpha q} \rceil^{m+1} + \Big(\frac{M\lfloor q/2 \rfloor}{0}\Big),$$

are indistinguishable for each $1 \leq \eta \leq \iota$. Therefore the difference between the advantage of $\mathcal{A}$ in the sequence of games **Game** $\mathbf{G}_{\eta+\kappa}$, $1 \leq \eta \leq \iota$ is negligible. We recall that in the last game ($\eta = \iota$), $\mathcal{A}$ will receive a challenger ciphertext taken from

$$U\Big(\mathrm{Span}_{i\in(\mathcal{S}_0\cup\mathcal{S}_1)}(\boldsymbol{x}_i^+)^\perp\Big) + \lfloor \nu_{\alpha q} \rceil^{m+1} + \Big(\frac{M\lfloor q/2 \rfloor}{0}\Big),$$

$$U\Big(\mathrm{Span}_{i\in(\mathcal{S}_0\cup\mathcal{S}_1)}(\boldsymbol{x}_i^+)^\perp\Big) + \lfloor \nu_{\alpha q} \rceil^{m+1} + \Big(\frac{M\lfloor q/2 \rfloor}{0}\Big).$$

Obviously, the advantage of adversary $\mathcal{A}$ of the game is equal to zero since these distributions are identical.

We conclude that our scheme AnoBES is ANOCPA-secure under the hardness of $k-$LWE problem.

4 Trace and Revoke System from AnoBES and Robust IPP Codes

We firstly dedicate this section to recall the concept of Robust IPP code, proposed by Barg et al. [4] and then present a construction of the TR system from Robust IPP and AnoBES. Due to the property of implemented Robust IPP, we emphasize that our TR system supports to trace efficiently at least one traitor who contributes to build a smart pirate device (allowed to erasure several bits from a word that embedded in that device).

4.1 Robust IPP code

Before recalling the formal definition of Robust IPP code, we revisit the notation of IPP code. Let Σ be an alphabet set containing q symbols.

If $\mathcal{C} = \{w_1, \ldots, w_N\} \subset \Sigma^\ell$ then $\mathcal{C}$ is called a $q-$ary code of size N and length ℓ. Each $w_i \in \mathcal{C}$ is called a codeword and we write $w_i = (w_{i,1}, w_{i,2}, \ldots, w_{i,\ell})$ where $w_{i,j} \in \Sigma$ is called the $j-$th component of the codeword w_i. We denote by (ℓ, N, q) a code with size N, length ℓ over an alphabet size q. Let Δ denote the minimum Hamming distance of the code $\mathcal{C}$.

Given a positive integer t, a subset of codewords $X = \{w_1, w_2, \ldots, w_t\} \subset \mathcal{C}$ is called a coalition of size t. The set of descendants of X, denoted $\mathsf{desc}(X)$, is defined by

$$\mathsf{desc}(X) = \Big\{x \in \Sigma^\ell \mid x_j \in \{w_{i,j} \mid w_i \in X\}, 1 \leq j \leq \ell\Big\},$$

codewords in the coalition X is call parents. Define a $t-$descendant ($t-$envelope) of the code $\mathcal{C}$, denoted $\mathsf{desc}_t(\mathcal{C})$, as follows:

$$\mathsf{desc}_t(\mathcal{C}) = \bigcup_{X \subset \mathcal{C}, |X| \leq t} \mathsf{desc}(X).$$

The $\mathsf{desc}_t(\mathcal{C})$ consists of all $\ell-$tuples that could be generated by some coalition of size at most t. Codes with identifiable parent property (IPP codes) are defined below

Definition 7. *Given a code $\mathcal{C} = (\ell, N, q)$, let $t \geq 2$ be an integer. $\mathcal{C}$ is called a $t-$IPP code if for all $x \in \mathsf{desc}_t(\mathcal{C})$, there exists $w \in \mathcal{C}$ such that for any $X \subset \mathcal{C}$, if $|X| \leq t$ and $x \in \mathsf{desc}(X)$ then $w \in X$.*

In a $t-\mathsf{IPP}$ code, given a descendant $x \in \mathsf{desc}_t(\mathcal{C})$, we can always identify at least one of its parent codewords with error-free.

Let $X_i = \{w_{1,i}, w_{2,i}, \ldots, w_{t,i}\}$ be the set of the $i-$th coordinates of the coalition X. If the cardinality of X_i is equal to 1, say $|X_i| = 1$, the coordinate i is called undetectable, else it is called detectable. The set of detectable coordinates for the coalition X is denoted by $D(X)$.

In [10], Boneh and Shaw considered a more general coalition, called wide-sense envelope of the coalition X as follows:

$$\Big\{(x_1, \ldots, x_\ell) \in \Sigma^\ell \cup \{*\} \mid x_i = y_i, i \notin D(X)\Big\}.$$

This means that any symbol of Σ or erased symbols $*$ are allowed in the detectable coordinates. Only detectable coordinates of descendant are allowed to modify the values by following the notation Marking assumption. The notation Robust IPP code is an intermediate concept between the IPP and fingerprinting codes in the sense that robust IPP codes allow a limited number of coordinates do not follow their parents. These coordinates allowed to deviate by breaking the marking assumption.

Let $X \subset \Sigma^\ell, |X| \leq t$ be a coalition. For $i = 1, \ldots, \ell$, let X_i be the set of the $i-$th coordinates of the elements of a coalition X. Assume that the coalition X forms x following the marking assumption rule except εn coordinates that can deviate from this rule. Call a coordinate i of $x \in \mathsf{desc}(X)$ is a mutation if $x_i \notin X_i$ and consider mutations of two types: erasures, where x_i is replaced by an erasure symbol $*$ and arbitrary symbol $y_i \in \Sigma - X_i$.

Denote by $\mathsf{desc}(X)_\varepsilon$ the set of all vectors x formed from the vectors in the coalition X so that $x_i \in X_i$ for $\ell(1-\varepsilon)$ coordinates i and x_i is a mutation in at most $\varepsilon\ell$ coordinates. Codes with robust identifiable parent property (Robust IPP codes) are defined below

Definition 8. *Code $\mathcal{C} \subset \Sigma^\ell$ is a $(t, \varepsilon) - \mathsf{IPP}$ code (robust $t - \mathsf{IPP}$ code) if*

$$\bigcap_{X \subset \mathcal{C}, |X| \leq t, x \in \mathsf{desc}(X)_\varepsilon} X \neq \emptyset.$$

In words: the code $\mathcal{C}$ guarantees exact identification of at least one member of the coalition X of size at most t for any collusion with at most $\varepsilon\ell$ mutations. In the case $\varepsilon = 0$, the robust IPP becomes IPP codes.

In our purpose, we aim to build a broadcast encryption supporting a traitor tracing algorithm in the case that: even some malicious users (traitors) in the system intend to collude on their private keys to build a powerful pirate decoder device, we can identify exactly (error-free) at least

one traitor who leaks the secret information to contribute to the pirate. The powerful decoder we mentioned is that it can erase some information (delete some bits) of the embedded word in the device to avoid a tracing procedure. We thus consider the case: any coordinate of $x \in \mathsf{desc}(X)_\varepsilon$ can mutate, and the mutant coordinates always become erasures.

4.2 Trace and Revoke scheme from AnoBES and robust IPP code

Our aim addresses to construct a Trace and Revoke (TR) scheme from the efficient tool AnoBES. In our approach, we are going to combine the $t-$IPP robust code $\mathcal{C} = (\ell, N, q)$, proposed by Barg et al. [4] with AnoBES scheme. It is progressed as follows:

Given a $q-$ary robust $t-$IPP code $\mathcal{C} = (\ell, N, q)$ with length ℓ, size N, mimimum Hamming distance Δ over alphabet $\Sigma = \{1, \ldots, q\}$, we denote by $\mathcal{C} = \{w_1, \ldots, w_N\}$ the codewords of code $\mathcal{C}$ and $w_i = (w_{i,1}, \ldots, w_{i,\ell})$. Since $\mathcal{C}$ is a robust IPP code, it means that any combination of t users can make a word, having at most $\varepsilon\ell$ erasure positions. We put symbols $*$ in these erasure positions. The word then puts in a pirate device.

Let $(\rho\ell, \ell)$-secret sharing scheme, where $\rho \in (0, 1)$. We assume that $\rho = 1 - \varepsilon$ and we will explain later the reason why our assumption is justified. Let r be maximum number of revoked users. We require that the parameter r, with the purpose of revocation, chosen so that

$$\Delta > \ell\Big(1 - \frac{1-\rho}{r}\Big). \tag{1}$$

We denote by $[N] = \{1, \ldots, N\}$ the set of N users. We define a *mixture* $S = (S_1, \ldots, S_\ell)$ over Σ^ℓ is a sequence of ℓ subsets of Σ, i.e. $S_i \subseteq \Sigma$. Given a vector $\boldsymbol{\omega} = (\omega_1, \ldots, \omega_\ell) \in \Sigma^\ell$, the *agreement* between $\boldsymbol{\omega}$ and a mixture S is defined to be the number of positions $i \in [\ell]$ for which $\omega_i \in S_i$:

$$\mathrm{AGR}(\boldsymbol{\omega}, S) = \sum_{i=1}^{\ell} \mathbf{1}_{\omega_i \in S_i}.$$

We will construct a broadcast system Γ for the set $[N]$ as follows: We identify each user $i \in [N]$ with the codeword $w_i = (w_{i,1}, \ldots, w_{i,\ell})$ in $\mathcal{C}$, whereas $w_{i,j}$ is the j-th of the codeword $w_i \in \mathcal{C}$. By assigning each user in

Γ to a set with ℓ sub-keys, we have

$$\begin{aligned}
\mathsf{dk}_1 &= (\mathsf{dk}_{1,w_{1,1}}, \ldots, \mathsf{dk}_{j,w_{1,j}}, \ldots, \mathsf{dk}_{\ell,w_{1,\ell}}) \\
\mathsf{dk}_2 &= (\mathsf{dk}_{1,w_{2,1}}, \ldots, \mathsf{dk}_{j,w_{2,j}}, \ldots, \mathsf{dk}_{\ell,w_{2,\ell}}) \\
&\vdots \\
\mathsf{dk}_N &= (\mathsf{dk}_{1,w_{N,1}}, \ldots, \mathsf{dk}_{j,w_{N,j}}, \ldots, \mathsf{dk}_{\ell,w_{N,\ell}}).
\end{aligned}$$

Therefore, the number of decryption keys that can be generated for the broadcast system with N users is equal to $q\ell$. In our system, at any coordinate component of the decryption key, we have at most q sub-keys. We consider a one-to-one correspondence between the set of q sub-keys and the set of decryption keys of q users in AnoBES system. Consequently, to broadcast a message K for the set of N users, it is necessary to decompose K to ℓ components and we encrypt j^{th}-component with a subset of q sub-keys $j \in [\ell]$. This action is equivalent to apply ℓ times the algorithm Encrypt of the scheme AnoBES.

In a nutshell, in formally, to build a broadcast system for N users, we concatenate ℓ instantiations of the scheme AnoBES (for q users) according to an $q-$ary code $\mathcal{C}$. In particular, we will combine the code $\mathcal{C}$ with robust IPP code $\mathcal{C}$. Our construction consists of 4 algorithms: Setup, KeyDer, Enc and Dec. It progresses as follows:

$\mathsf{Setup}(n, N)$**:** Take as input the security parameter n and the size of the code $\mathcal{C}$. By calling the procedure $\mathsf{AnoBES.Setup}(n, q)$. We obtain a public key ek and a master secret key MSK.

$\mathsf{KeyDer}(\mathsf{ek}, \mathsf{MSK}, i)$**:** Take as index $i \in [N]$ for each user, using MSK to extract ℓ decryption keys for user i,

$$\mathsf{dk}_i = (\mathsf{sk}_{1,w_{i,1}}, \ldots, \mathsf{sk}_{j,w_{i,j}}, \ldots, \mathsf{sk}_{\ell,w_{i,\ell}}),$$

where $w_{i,j}$ be the value at position j of codeword w_i. Here,

$$\mathsf{sk}_{j,w_{i,j}} = \mathsf{AnoBES.Extract}(\mathsf{ek}, \mathsf{MSK}, w_{i,j}) \in \mathbb{Z}^{m+1}, j \in [\ell].$$

$\mathsf{Enc}(\mathsf{ek}, K, \mathcal{R})$**:** Take as input a set of revoked users $\mathcal{R} \subset \mathcal{C}$, where the cardinality of $\mathcal{R}$ is at most r. The message K will be broadcasted to the target set $\mathcal{C} - \mathcal{R}$. We call the procedure $\mathsf{Share}(\ell, \rho\ell, \ell)$ of $(\rho\ell, \ell)$-secret sharing scheme. The $\mathsf{Share}(\ell, \rho\ell, \ell)$ algorithm outputs a secret $K \in \mathcal{PT}^\ell$ and ℓ shares $K_1, \ldots, K_\ell$. At least $\rho\ell$ of the shares are needed to recover the message K. We broadcast using the following mixture

$$\mathcal{M} = (\mathcal{M}_1, \ldots, \mathcal{M}_\ell) = (\Sigma - \mathcal{R}[1], \ldots, \Sigma - \mathcal{R}[\ell]),$$

where $\mathcal{R}[j] = \cup_{i \in \mathcal{R}} w_{i,j}$. The ciphertext has form

$$\begin{aligned}\boldsymbol{c} &= (c_1, \ldots, c_\ell) \in \mathcal{CT}^\ell, \\ &= \Big(\mathsf{AnoBES.Encrypt}(\mathsf{ek}, K_1, \mathcal{M}_1), \ldots, \mathsf{AnoBES.Encrypt}(\mathsf{ek}, K_\ell, \mathcal{M}_\ell)\Big),\end{aligned}$$

where each c_j is the output of the encryption algorithm $\mathsf{AnoBES.Encrypt}$ with input K_j and $\mathcal{M}_j$. It means that each part of message, K_j is encrypted with keys $\{\mathsf{sk}_{j,w_{i,j}}\}_{i \notin \mathcal{R}}$.

To recover K, we first observe that the mixture needs to have at least $\rho\ell$ non-empty sets . The mixture can be seen as a set of descendants (words having at most $\varepsilon\ell$ mutant coordinates) created by the codewords in $\mathcal{C} - \mathcal{R}$. The empty sets are identical with symbols $*$. Because the code $\mathcal{C}$ only allows the words have at most $\varepsilon\ell$ erasure coordinates, there are at least $\ell - \varepsilon\ell$ non-empty sets. It implies that we can choose $\rho = 1 - \varepsilon$.

$\mathsf{Dec}(\mathsf{ek}, \mathsf{dk}_i, \boldsymbol{c})$**:** Take as input ciphertext $\boldsymbol{c} \in \mathcal{CT}^\ell$ and decryption key dk_i of user i. For each $j \in [\ell]$, the user call the decryption function $\mathsf{AnoBES.Decrypt}(\mathsf{ek}, \mathsf{sk}_{j,w_{i,j}}, c_j)$ of AnoBES scheme on sub-keys $\mathsf{sk}_{j,w_{i,j}}$ to obtain shared values K_j. By recalling the function $\mathsf{Combine}(\{j, K_j\})$ of the secret sharing scheme, the user recovers successfully the original message K.

$\mathsf{Tracing}$**:** We consider the mixture $\mathcal{M}$ as in Enc procedure. Let $\mathcal{T}$ be the subset of $\mathcal{U} - \mathcal{R}$ with at most t elements (traitors). We assume the pirate produces a ϵ-useful pirate decoder $\mathcal{D}$, it can decrypt any normal ciphertext in form

$$\begin{aligned}\boldsymbol{c} &= (c_1, \ldots, c_j, \ldots, c_\ell) \\ &= \Big(\mathsf{AnoBES.Encrypt}(\mathsf{ek}, K_1, \mathcal{M}_1), \ldots, \mathsf{AnoBES.Encrypt}(\mathsf{ek}, K_\ell, \mathcal{M}_\ell)\Big),\end{aligned}$$

with non-negligible probability. We denote here $\mathcal{M}_j = \{j_\iota\}_{\iota=1}^{\leq q}$ or $\mathcal{M}_j = \emptyset$ for all $j = 1, \ldots, \ell$. We consider the tracing procedure as below:

For $j = 1$ to ℓ, do the following:

1. While $\mathcal{M}_j \neq \emptyset$, do the following:
   (a) Let $\mathsf{cnt} \leftarrow 0$.
   (b) Repeat the following steps $W \leftarrow 8n(q/\epsilon)^2$ times:
      i. $c_j = \mathsf{AnoBES.Encrypt}(\mathsf{ek}, K_j, \mathcal{M}_j)$.
      ii. Call oracle $\mathcal{O}^{\mathcal{D}}$ on input $\boldsymbol{c} = (c_1, \ldots, c_j, \ldots, c_\ell)$. If $\mathcal{O}^{\mathcal{D}}(\boldsymbol{c}, K) = 1$ then $\mathsf{cnt} \leftarrow \mathsf{cnt} + 1$.
   (c) Let $\tilde{p}_{j,j_\iota}$ be the fraction of times that $\mathcal{D}$ decrypted the ciphertexts correctly. We have $\tilde{p}_{j,j_\iota} = \mathsf{cnt}/W$.

(d) $\mathcal{M}_j = \mathcal{M}_j - \{j_\iota\}$.

2. If there exists an index $j_\iota \in \mathcal{M}_j$ for which $\tilde{p}_{j,j_\iota} - \tilde{p}_{j,j_{\iota'}} \geq \epsilon/4q\ell$ for all $j_{\iota'} \in \mathcal{M}_j$ then
   (a) the key j_ι is accused and $\omega_j = j_\iota$,
   (b) $c_j = \mathsf{AnoBES.Encrypt}(\mathsf{ek}, K_j, \mathcal{M}_j)$
   else $c_j = \mathsf{random}$ and $\omega_j = *$.

End for.

From the pirate word $\boldsymbol{\omega} = (\omega_1, \ldots, \omega_\ell)$ found after the Loop finished, call tracing procedure in robust IPP code on input $\boldsymbol{\omega}$. The Tracing returns a traitor.

The above Tracing algorithm, we note that the decryption probabilities of the pirate device does not change significantly in every iteration step because even if the tracer detects a non-negligible decryption probability of pirate decoder, it will reset the modified component to normal component. After step 2, the tracer will find out a letter of pirate word at position j. The value of position is either a symbol in the alphabet or erasure symbol.

We can proof easily that the tracing algorithm returns at least $\rho\ell$ keys. Indeed, if the output of algorithm provides $t < \rho\ell$ keys then the ciphertext in the final iteration step ℓ will appear t normal components and the pirate device can still certainly decrypt the ciphertext. This is a contradition because in the setting of our system it is impossible for any decoder device (included the pirate device) to decrypt the ciphertext successfully. Therefore, our tracing algorithm will output at least $\rho\ell$ pirate keys. We put these keys together with $*$ symbols in a word. Since the scheme Γ employs robust IPP code $\mathcal{C}$, the tracer uses the property of robust IPP for the pirate word which found from the black-box tracing to identify at least one user who contributes to build the pirate device.

Correctness. We consider the correctness of our TR system: for all user $i \in [N]$ and all messages K. Whenever the user i is given a ciphertext $\boldsymbol{c}$, he can decrypt successful if the user $i \notin \mathcal{R}$. Indeed, since $\mathcal{C}$ is the code having the minimum Hamming distance that satisfies the inequality (1), any user i is in $\mathcal{C} - \mathcal{R}$, we have $\text{AGR}(w_i, \mathcal{M}) \geq \rho\ell$. Indeed,

$$\text{AGR}(w_i, \mathcal{M}) \geq \ell - r(\ell - \Delta) \geq \rho\ell.$$

It implies the user i has at least $\rho\ell$ sub-keys that agree with the mixture $\mathcal{M}$ and recovers at least $\rho\ell$ sub-message K_i. By calling the function Dec, he will receive the original message K.

We will see as below efficiency of the trace and revoke scheme based on a robust IPP code in sense that we will consider parameters of scheme in the number of decryption keys per each user and the length of ciphertext.

After black-box tracing procedure, we get a pirate word. With a given pirate word, to ensure that the identify algorithm can return at least a traitor from a $t-$collusion, we need to have some following relations between parameters. See Proposition 3.1 ([4])

$$\Delta/\ell > 1 - \Big(\frac{1-\varepsilon}{t^2} - \frac{\varepsilon}{t}\Big),$$

whereas $0 < \varepsilon < (t+1)^{-1}$. Therefore, for the system to be able to trace and revoke, we need a q-ary codes in which

$$\begin{aligned}\Delta &> \ell \cdot \max\left\{1 - \frac{1-\rho}{r}, 1 - \left(\frac{1-\varepsilon}{t^2} - \frac{\varepsilon}{t}\right)\right\} \\ &= \ell \cdot \left(1 - \min\left\{\frac{1-\rho}{r}, \frac{1-\varepsilon}{t^2} - \frac{\varepsilon}{t}\right\}\right).\end{aligned}$$

The number of keys per user is ℓ and the ciphertext size is at most ℓ times the ciphertext size of AnoBES. We just proved the following theorem.

Theorem 9. *Given*

- $\mathcal{C} = (\ell, N, q)$ *be a robust* $t -$ IPP *of Hamming distance* $\mathcal{C}$*, with at most* $\varepsilon\ell$ *erasure positions veryfying* $0 < \varepsilon < (t+1)^{-1}$*;*
- *a* $(\rho\ell, \ell)$ *secret sharing scheme, for* $\rho \in (0, 1)$*;*
- *an anonymous broadcast encryption for* q *users* AnoBES*;*

satisfying the following condition

$$\Delta/\ell > 1 - \min\left\{\frac{1-\rho}{r}, \frac{1-\varepsilon}{t^2} - \frac{\varepsilon}{t}\right\}. \tag{2}$$

Then Γ*, constructed as above, is a* TR *scheme for* N *users in which we can revoke up to* r *users and trace succesfully at least one traitor from any coalition up to* t *traitors.*

Ciphertext Size. We now consider the ciphertext size of the scheme Γ. This turns out to evaluate the length of the Robust IPP code. We can not directly use the analysis in the paper [4] because of the two raisons:

- in [4], the proof of existence of robust IPP codes was given but there was no explicit construction, neither the analysis of the complexity length of the code.
- robust IPP codes only deals with the number of traitor. In our scheme, we need moreover to take into account of the number of the revoked users that satisfying the condition (2).

Explicit construction of codes. We set $\delta := \Delta/\ell$ the rate of the code $\mathcal{C}$. **Construction 1:** We will consider a code with the rate δ satisfying the Gilbert-Varshamov bound. To do this, let us pick

$$1 - \min\left\{\frac{1-\rho}{r}, \frac{1-\varepsilon}{t^2} - \frac{\varepsilon}{t}\right\} < \delta \leq 1 - \frac{1}{q}.$$

According to Gilbert-Varshamov theorem (Theorem 4.10, [25]), there exists a $q-$ary code $\mathcal{C}$ with rate $R(\mathcal{C}) = \frac{1}{\ell}\log_q N$ satisfying

$$R(\mathcal{C}) \geq 1 - H_q(\delta) - o(1),$$

where $H_q(\delta)$ is the q-ary entropy function $H_q : [0,1] \to \mathbb{R}$ defined by

$$H_q(\delta) = \delta \log_q \frac{q-1}{\delta} + (1-\delta)\log_q \frac{1}{1-\delta}.$$

We choose

$$d = \max\left\{\frac{r}{1-\rho}, \frac{t^2}{(1-\varepsilon)-\varepsilon t}\right\}.$$

Therefore

$$1 - 1/d < \delta \leq 1 - \frac{1}{q}.$$

To ensure the obtained code is not a random code, we apply a derandomization procedure Porat-Rothschild [24]. This means that we give an explicit construction for the code $\mathcal{C}$. It progresses as follows:

We choose $\delta = 1 - \frac{1}{d+1}$. Obviously, we do not want large δ because that only can reduce the size of the code. To satisfy $\delta \leq 1 - \frac{1}{q}$ we need $q \geq d+1$. Since $q \geq d+1$, we choose $q = \Theta(d)$. Next, we need to estimate the value of $1 - H_q(\delta)$. We will use the fact that $\log(1+x) \approx x$ for small x extensively.

$$\begin{aligned}
1 - H_q(\delta) &= 1 - \delta\log_q(q-1) + \delta\log_q\delta + (1-\delta)\log_q(1-\delta) \\
&= 1 - \log_q(q-1) + (1-\delta)\log_q[(q-1)(1-\delta)] + \delta\log_q\delta \\
&= \frac{\log\left(\frac{q}{q-1}\right)}{\log q} + \frac{\log[(q-1)/(d+1)]}{(d+1)\log q} - \frac{d}{d+1}\frac{\log(1+1/d)}{\log q} \\
&\approx \frac{1}{(q-1)\log q} + \frac{\log[\Theta(1)]}{(d+1)\log q} - \frac{1}{(d+1)\log q} \\
&= \Theta\left(\frac{1}{d\log q}\right).
\end{aligned}$$

Since $R(\mathcal{C}) \geq 1 - H_q(\delta) - o(1)$, we omit small terms and obtain $R(\mathcal{C}) = 1 - H_q(\delta)$. Moreover, $R(\mathcal{C}) = \frac{1}{\ell} \log_q N$, it implies the length of the code is

$$\ell = \frac{\log_q N}{R(\mathcal{C})} = \frac{\log_q N}{1 - H_q(\delta)} = O\left(d \log q \log_q N\right) = O(d \log N).$$

In brief, we obtain

$$q = \Theta(d) \text{ and } \ell = O(d \log N).$$

and finally gets

$$\ell = O\left(\max\left\{\frac{r}{1-\rho}, \frac{t^2}{(1-\epsilon) - \epsilon t}\right\} \log N\right).$$

Construction 2: In another circumstance, we consider our code $\mathcal{C}$ in Reed-Solomon setting: we also pick

$$d = \max\left\{\frac{r}{1-\rho}, \frac{t^2}{(1-\epsilon) - \epsilon t}\right\}.$$

The Reed-Solomon code has $\delta = \frac{\ell - k + 1}{\ell} = 1 - \frac{k}{\ell} + \frac{1}{\ell}$, whereas k is dimension of code $\mathcal{C}$. In this case, if we choose $\ell = kd$ then $\delta > 1 - 1/d$. Hence, to use Reed-Solomon code we need to pick $q \geq \ell = kd$ such that $q^k \geq N$ or, equivalently, $\ell \log q \geq d \log N$.

For example, we can pick $q = \ell \approx \frac{2d \log N}{\log(d \log N)}$ and $k \approx \frac{\log N}{\log q}$. In this case, the length of the code is

$$\ell = O\left(\frac{2d \log N}{\log(d \log N)}\right).$$

Semantic Security. We now consider the security of the scheme Γ.

Theorem 10. *Assume that the scheme* AnoBES *is* IND − CPA*-secure, then the scheme* Γ *is also* IND − CPA*-secure scheme.*

Proof. We leave the proof in Appendix A.2.

Remark.

- The length of the above robust IPP codes is approximately the length of the best collusion secure code which essentially $O(t^2 \log N)$ [26]. Suppose that one can construct an AnoBES which is as efficient as the underlying PKE, then our proposed robust IPP code based trace and revoke schemes has the same ciphertext size as the state of the art collusion secure code based traitor tracing schemes. Note that one cannot revoke users in the collusion secure code based traitor tracing schemes.

- We provided a construction of AnoBES which is as efficient as the underlying LWE PKE. We raise an open question of constructing AnoBES schemes from the other standard encryptions such as ElGamal, RSA, Paillier encryptions without a significant loss in the ciphertext size.
- Boneh-Naor [9] and Billet-Phan [7] provided a solution to trace traitors from imperfect pirate device, with short ciphertext size. Their schemes built from a robust collusion secure code and a PKE. We can completely follow these methods to obtain a traitor tracing scheme from a robust IPP code and an AnoBES with short ciphertext size, namely a constant factor of the ciphertext size of the AnoBES.

References

1. Shweta Agrawal, Sanjay Bhattacherjee, Duong Hieu Phan, Damien Stehlé, and Shota Yamada. Efficient public trace and revoke from standard assumptions: Extended abstract. In *CCS*, pages 2277–2293. ACM, 2017. (Page 3.)
2. Murat Ak, Aggelos Kiayias, Serdar Pehlivanoglu, and Ali Aydin Selcuk. Generic construction of trace and revoke schemes. Cryptology ePrint Archive, Report 2012/531, 2012. http://eprint.iacr.org/2012/531. (Pages 4 and 31.)
3. J. Alwen and C. Peikert. Generating shorter bases for hard random lattices. *Theor. Comput. Science*, 48(3):535–553, 2011. (Page 8.)
4. Alexander Barg and Grigory Kabatiansky. Robust parent-identifying codes and combinatorial arrays. *IEEE Trans. Information Theory*, 59(2):994–1003, 2013. (Pages 4, 5, 15, 17, and 21.)
5. Adam Barth, Dan Boneh, and Brent Waters. Privacy in encrypted content distribution using private broadcast encryption. In Giovanni Di Crescenzo and Avi Rubin, editors, *FC 2006*, volume 4107 of *LNCS*, pages 52–64. Springer, Heidelberg, February / March 2006. (Page 2.)
6. Shimshon Berkovits. How to broadcast a secret (rump session). In Donald W. Davies, editor, *EUROCRYPT'91*, volume 547 of *LNCS*, pages 535–541. Springer, Heidelberg, April 1991. (Page 2.)
7. Olivier Billet and Duong Hieu Phan. Efficient traitor tracing from collusion secure codes. In Reihaneh Safavi-Naini, editor, *ICITS 08*, volume 5155 of *LNCS*, pages 171–182. Springer, Heidelberg, August 2008. (Pages 4 and 24.)
8. Dan Boneh, Craig Gentry, and Brent Waters. Collusion resistant broadcast encryption with short ciphertexts and private keys. In Victor Shoup, editor, *CRYPTO 2005*, volume 3621 of *LNCS*, pages 258–275. Springer, Heidelberg, August 2005. (Page 2.)
9. Dan Boneh and Moni Naor. Traitor tracing with constant size ciphertext. In Peng Ning, Paul F. Syverson, and Somesh Jha, editors, *ACM CCS 08*, pages 501–510. ACM Press, October 2008. (Pages 4 and 24.)
10. Dan Boneh and James Shaw. Collusion-secure fingerprinting for digital data (extended abstract). In Don Coppersmith, editor, *CRYPTO'95*, volume 963 of *LNCS*, pages 452–465. Springer, Heidelberg, August 1995. (Pages 3 and 16.)

11. Dan Boneh and Brent Waters. A fully collusion resistant broadcast, trace, and revoke system. In Ari Juels, Rebecca N. Wright, and Sabrina De Capitani di Vimercati, editors, *ACM CCS 06*, pages 211–220. ACM Press, October / November 2006. (Page 3.)
12. Benny Chor, Amos Fiat, and Moni Naor. Tracing traitors. In Yvo Desmedt, editor, *CRYPTO'94*, volume 839 of *LNCS*, pages 257–270. Springer, Heidelberg, August 1994. (Pages 2, 3, and 5.)
13. Nelly Fazio and Irippuge Milinda Perera. Outsider-anonymous broadcast encryption with sublinear ciphertexts. In Marc Fischlin, Johannes Buchmann, and Mark Manulis, editors, *PKC 2012*, volume 7293 of *LNCS*, pages 225–242. Springer, Heidelberg, May 2012. (Pages 2 and 6.)
14. Amos Fiat and Moni Naor. Broadcast encryption. In Douglas R. Stinson, editor, *CRYPTO'93*, volume 773 of *LNCS*, pages 480–491. Springer, Heidelberg, August 1994. (Page 2.)
15. C. Gentry, C. Peikert, and V. Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In *Proc. of STOC*, pages 197–206. ACM, 2008. Full version available at http://eprint.iacr.org/2007/432.pdf. (Page 8.)
16. Aggelos Kiayias and Katerina Samari. Lower bounds for private broadcast encryption. In *Information Hiding*, volume 7692 of *Lecture Notes in Computer Science*, pages 176–190. Springer, 2012. (Page 2.)
17. Aggelos Kiayias and Moti Yung. Traitor tracing with constant transmission rate. In Lars R. Knudsen, editor, *EUROCRYPT 2002*, volume 2332 of *LNCS*, pages 450–465. Springer, Heidelberg, April / May 2002. (Pages 3 and 4.)
18. Benoît Libert, Kenneth G. Paterson, and Elizabeth A. Quaglia. Anonymous broadcast encryption: Adaptive security and efficient constructions in the standard model. In Marc Fischlin, Johannes Buchmann, and Mark Manulis, editors, *PKC 2012*, volume 7293 of *LNCS*, pages 206–224. Springer, Heidelberg, May 2012. (Pages 2, 8, and 10.)
19. San Ling, Duong Hieu Phan, Damien Stehlé, and Ron Steinfeld. Hardness of k-LWE and applications in traitor tracing. In Juan A. Garay and Rosario Gennaro, editors, *CRYPTO 2014, Part I*, volume 8616 of *LNCS*, pages 315–334. Springer, Heidelberg, August 2014. (Pages 4, 5, 6, 8, 11, 12, 13, 27, and 33.)
20. Hung Q. Ngo, Duong Hieu Phan, and David Pointcheval. Black-box trace and revoke codes. 67(3):418–448, November 2013. (Page 4.)
21. Ryo Nishimaki, Daniel Wichs, and Mark Zhandry. Anonymous traitor tracing: How to embed arbitrary information in a key. In Marc Fischlin and Jean-Sébastien Coron, editors, *EUROCRYPT 2016, Part II*, volume 9666 of *LNCS*, pages 388–419. Springer, Heidelberg, May 2016. (Page 3.)
22. Duong Phan, Reihaneh Safavi-Naini, and Dongvu Tonien. Generic construction of hybrid public key traitor tracing with full-public-traceability. In Michele Bugliesi, Bart Preneel, Vladimiro Sassone, and Ingo Wegener, editors, *ICALP 2006, Part II*, volume 4052 of *LNCS*, pages 264–275. Springer, Heidelberg, July 2006. (Page 3.)
23. Duong Hieu Phan, David Pointcheval, and Mario Strefler. Message-based traitor tracing with optimal ciphertext rate. In Alejandro Hevia and Gregory Neven, editors, *LATINCRYPT 2012*, volume 7533 of *LNCS*, pages 56–77. Springer, Heidelberg, October 2012. (Page 2.)
24. Ely Porat and Amir Rothschild. Explicit non-adaptive combinatorial group testing schemes. In Luca Aceto, Ivan Damgård, Leslie Ann Goldberg, Magnús M. Halldórsson, Anna Ingólfsdóttir, and Igor Walukiewicz, editors, *ICALP 2008, Part I*, volume 5125 of *LNCS*, pages 748–759. Springer, Heidelberg, July 2008. (Page 22.)

25. Ron Roth. *Introduction to Coding Theory*. Cambridge University Press, New York, NY, USA, 2006. (Page 22.)
26. Gábor Tardos. Optimal probabilistic fingerprint codes. In *35th ACM STOC*, pages 116–125. ACM Press, June 2003. (Page 23.)
27. Xingwen Zhao and Hui Li. Codes based tracing and revoking scheme with constant ciphertext. In Tsuyoshi Takagi, Guilin Wang, Zhiguang Qin, Shaoquan Jiang, and Yong Yu, editors, *ProvSec 2012*, volume 7496 of *LNCS*, pages 318–335. Springer, Heidelberg, September 2012. (Page 3.)

A Missing Proofs

A.1 Proof of Theorem 5

Proof. We consider the sequence of following games between a challenger $\mathcal{B}$ and an attacker $\mathcal{A}$. We start considering the real game defined as in the security model.

Game $\mathbf{G}_0$: Setup. The challenger generates matrix $A \hookleftarrow U(\mathbb{Z}_q^{m\times n})$ and $\boldsymbol{u} \hookleftarrow U(\mathbb{Z}_q^n)$. The challenger sends public key $\mathsf{ek} = A^+$, with $A^+ = (\boldsymbol{u}^t \| A)$ to $\mathcal{A}$.

Phase 1. $\mathcal{A}$ queries descryption keys for several users $i \in \{1, \ldots, N\}$. $\mathcal{B}$ samples $\boldsymbol{x}_i \hookleftarrow D_{\Lambda^\perp_{-\boldsymbol{u}}(A),S}$ and gives $\boldsymbol{x}_i^+$ to $\mathcal{A}$, where $\boldsymbol{x}_i^+ := (1\|\boldsymbol{x}_i) \in \mathbb{Z}^{m+1}$.

Challenger phase. The adversary selects two messages $M_0, M_1 \leftarrow \mathcal{PT} = \{0,1\}$, a subset of users $\mathcal{S} \subset \mathcal{U}$ so that queried indices must outside $\mathcal{S}$. $\mathcal{A}$ then sends M_0, M_1 and $\mathcal{S}$ to $\mathcal{B}$. The challenger picks randomly $b \hookleftarrow U(\{0,1\})$, outputs a challenge ciphertext (of the message M_b) sampled from one of two following distributions:

$$\mathcal{D}_0 = U\Big(\mathrm{Span}_{i\in\mathcal{S}}(\boldsymbol{x}_i^+)^\perp\Big) + \lfloor \nu_{\alpha q} \rceil^{m+1} + \Big(\frac{M_0\lfloor q/2 \rfloor}{0}\Big),$$

$$\mathcal{D}_1 = U\Big(\mathrm{Span}_{i\in\mathcal{S}}(\boldsymbol{x}_i^+)^\perp\Big) + \lfloor \nu_{\alpha q} \rceil^{m+1} + \Big(\frac{M_1\lfloor q/2 \rfloor}{0}\Big).$$

Phase 2. The adversary continues querying to decryption keys with the limited condition that $\mathcal{A}$ only queries to indices outside $\mathcal{S}$.

Guess. $\mathcal{A}$ gives a guess b' for b.

In each next games, the adversary will be received challenge ciphertexts in that the challenger gradually removes one key $\boldsymbol{x}_i^+$ from the distributions of ciphertext.

Game $\mathbf{G}_1$: The challenger now makes one small change to the previous game. It means that every steps in this game are coincide to

previous one but the challenge ciphertext sampled from one of two distributions $\mathcal{D}_0^1$ and $\mathcal{D}_1^1$.

$$\mathcal{D}_0^1 = U\Big(\mathrm{Span}_{i\in\mathcal{S}}(\{\boldsymbol{x}_i^+\} - \{\boldsymbol{y}\})^\perp\Big) + \lfloor \nu_{\alpha q} \rceil^{m+1} + \Big(\frac{M_0\lfloor q/2 \rfloor}{0}\Big),$$

$$\mathcal{D}_1^1 = U\Big(\mathrm{Span}_{i\in\mathcal{S}}(\{\boldsymbol{x}_i^+\} - \{\boldsymbol{y}\})^\perp\Big) + \lfloor \nu_{\alpha q} \rceil^{m+1} + \Big(\frac{M_1\lfloor q/2 \rfloor}{0}\Big),$$

whereas $\boldsymbol{y} = \boldsymbol{x}_i^+, i \in \mathcal{S}$. Applying Theorem 25 in [19], in the view of $\mathcal{A}$, two pairs of distributions

$$\mathcal{D}_0 = U\Big(\mathrm{Span}_{i\in\mathcal{S}}(\boldsymbol{x}_i^+)^\perp\Big) + \lfloor \nu_{\alpha q} \rceil^{m+1} + \Big(\frac{M_0\lfloor q/2 \rfloor}{0}\Big),$$

$$\mathcal{D}_0^1 = U\Big(\mathrm{Span}_{i\in\mathcal{S}}(\{\boldsymbol{x}_i^+\} - \{\boldsymbol{y}\})^\perp\Big) + \lfloor \nu_{\alpha q} \rceil^{m+1} + \Big(\frac{M_0\lfloor q/2 \rfloor}{0}\Big)$$

and

$$\mathcal{D}_1 = U\Big(\mathrm{Span}_{i\in\mathcal{S}}(\boldsymbol{x}_i^+)^\perp\Big) + \lfloor \nu_{\alpha q} \rceil^{m+1} + \Big(\frac{M_1\lfloor q/2 \rfloor}{0}\Big),$$

$$\mathcal{D}_1^1 = U\Big(\mathrm{Span}_{i\in\mathcal{S}}(\{\boldsymbol{x}_i^+\} - \{\boldsymbol{y}\})^\perp\Big) + \lfloor \nu_{\alpha q} \rceil^{m+1} + \Big(\frac{M_1\lfloor q/2 \rfloor}{0}\Big).$$

are indistingushable under the assumption that $k - \mathsf{LWE}$ is hard to solve. Therefore, the difference of the advantage of the adversary $\mathcal{A}$ in two games is negligible.

Similarly, we consider $\ell - 1$ games more, whereas $\ell = |\mathcal{S}|$ and reach to the final game.

Game $\mathbf{G}_\ell$: Also, the challenger makes one small change to the previous games. Every steps in this game are coincide to previous one but the challenge ciphertext sampled from one of two distributions $\mathcal{D}_0^\ell$ and $\mathcal{D}_1^\ell$.

$$\mathcal{D}_0^\ell = U\Big(\mathbb{Z}_q^{m+1}\Big) + \lfloor \nu_{\alpha q} \rceil^{m+1} + \Big(\frac{M_0\lfloor q/2 \rfloor}{0}\Big),$$

$$\mathcal{D}_1^\ell = U\Big(\mathbb{Z}_q^{m+1}\Big) + \lfloor \nu_{\alpha q} \rceil^{m+1} + \Big(\frac{M_1\lfloor q/2 \rfloor}{0}\Big).$$

Obviously, the advantage of $\mathcal{A}$ in this game is equal to zero.

In brief, we have a sequence of games where the final game **Game $\mathbf{G}_\ell$** has zero-avantage and the difference of two successive games **Game $\mathbf{G}_{\ell-1}$**,**Game $\mathbf{G}_\ell$**, for all $1 \leq i \leq \ell$, is negligible. Therefore, the scheme Π is $\mathsf{IND} - \mathsf{CPA}$ secure.

A.2 Proof of Theorem 10

Proof. We consider a sequence of games starting with **Game $\mathbf{G_0}$** as follows:

Game $\mathbf{G}_0$: This is the real game as defined in the security model. The challenger generates ℓ public keys $\{\mathsf{ek}_i\}_{i=1}^{\ell}$ and chooses robust IPP code $\mathcal{C} = \{w_1, \ldots, w_N\}$ which then gives to adversary $\mathcal{A}_\Gamma$. In the **Phase 1**, $\mathcal{A}_\Gamma$ queries descryption keys for user $i \in \{1, \ldots, N\}$ and obtains dk_i, where

$$\mathsf{dk}_i = (\mathsf{sk}_{1,w_{i,1}}, \ldots, \mathsf{sk}_{j,w_{i,j}}, \ldots, \mathsf{sk}_{\ell,w_{i,\ell}}),$$

where $\mathsf{sk}_{j,w_{i,j}}$ is a decryption key extracted from the scheme Π by call the algorithm

$$\Pi.\mathsf{Extract}(\mathsf{ek}_j, \mathsf{MSK}_j, w_{i,j}).$$

In the **Challenger phase**, the adversary selects two messages $m^0, m^1 \in \mathcal{PT}^\ell$ and a subset of revokers $\mathcal{R} \subset \mathcal{C}$. The challenger picks randomly $b \hookleftarrow \{0,1\}$, calls the procedure $\mathsf{Share}(\ell, \rho\ell, \ell)$ to get ℓ shares $m_1^b, \ldots, m_\ell^b$ for the message m^b and outputs a ciphertext $\Pi.\mathsf{Encrypt}(\mathsf{ek}_j, m_j^b, \mathcal{M}_j)_{j=1}^{\ell}$, where

$$(\mathcal{M}_1, \ldots, \mathcal{M}_\ell) = (\Sigma - \mathcal{R}[1], \ldots, \Sigma - \mathcal{R}[\ell]), \mathcal{R}[j] := \bigcup_{i|w_i \in \mathcal{R}} \{w_{i,j}\}.$$

In the **Phase 2**, when $\mathcal{A}_\Gamma$ received the ciphertext, sampled from one of two computationally indistinguishable distributions

$$\mathcal{D}_0 = \Big(\Pi.\mathsf{Encrypt}(\mathsf{ek}_1, m_1^0, \mathcal{M}_1), \ldots, \Pi.\mathsf{Encrypt}(\mathsf{ek}_\ell, m_\ell^0, \mathcal{M}_\ell)\Big)$$
$$\mathcal{D}_1 = \Big(\Pi.\mathsf{Encrypt}(\mathsf{ek}_1, m_1^1, \mathcal{M}_1), \ldots, \Pi.\mathsf{Encrypt}(\mathsf{ek}_\ell, m_\ell^1, \mathcal{M}_\ell)\Big),$$

$\mathcal{A}_\Gamma$ outputs a guess b' for b. Let $\mathsf{Adv}_{\mathcal{A}_\Gamma}^{\mathbf{Game\ 0}}(\mathcal{D}_0, \mathcal{D}_1)$ be the advantage of $\mathcal{A}_\Gamma$ with two given distributions $\mathcal{D}_0$ and $\mathcal{D}_1$. The advantage is defined by:

$$\begin{aligned}\mathsf{Adv}_{\mathcal{A}_\Gamma}^{\mathbf{Game\ 0}}(\mathcal{D}_0, \mathcal{D}_1) &= \Big|2\Pr\left[\mathcal{A}_\Gamma(\mathcal{D}_b) = b\right] - 1\Big| \\ &= \Big|\Pr\left[\mathcal{A}_\Gamma(\mathcal{D}_0) = 1\right] - \Pr\left[\mathcal{A}_\Gamma(\mathcal{D}_1) = 1\right]\Big|.\end{aligned}$$

Game $\mathbf{G}_1$: The challenger now makes one small change to the **Game $\mathbf{G_0}$**. Namely, instead of encrypting the first share m_1^0 with all keys in the mixture $\mathcal{M}_1$, we encrypt m_1^1 with the mixture $\mathcal{M}_1$. It means that the challenger only changes the first coordinate in $\mathcal{D}_0$ and does not do

anything with $\mathcal{D}_1$. In this game, all steps are same as **Game $\mathbf{G_0}$** except as mentioned above ciphertext. Thus, $\mathcal{A}_\Gamma$ will receive a challenger ciphertext, sampled from one of two computationally indistinguishable distributions $\mathcal{D}_0^1$ and $\mathcal{D}_1$, where

$$\mathcal{D}_0^1 = \Big(\Pi.\mathsf{Encrypt}(\mathsf{ek}_1, m_1^1, \mathcal{M}_1), \quad \Pi.\mathsf{Encrypt}(\mathsf{ek}_2, m_2^0, \mathcal{M}_2) \\ , \ldots, \Pi.\mathsf{Encrypt}(\mathsf{ek}_\ell, m_\ell^0, \mathcal{M}_\ell)\Big)$$

We denote advantage of the adversary in this game by $\mathsf{Adv}_{\mathcal{A}_\Gamma}^{\mathbf{Game\ 1}}(\mathcal{D}_0^1, \mathcal{D}_1)$. And from this, we can see that

$$\Big|\Pr\left[\mathcal{A}_\Gamma(\mathcal{D}_0) = 1\right] - \Pr\left[\mathcal{A}_\Gamma(\mathcal{D}_1) = 1\right]\Big| \leq \Big|\Pr\left[\mathcal{A}_\Gamma(\mathcal{D}_0) = 1\right] - \Pr\Big[\mathcal{A}_\Gamma(\mathcal{D}_0^1) = 1\Big]\Big| \\ + \Big|\Pr\Big[\mathcal{A}_\Gamma(\mathcal{D}_0^1) = 1\Big] - \Pr\left[\mathcal{A}_\Gamma(\mathcal{D}_1) = 1\right]\Big|.$$

Therefore, we have

$$\mathsf{Adv}_{\mathcal{A}_\Gamma}^{\mathbf{Game\ 0}}(\mathcal{D}_0, \mathcal{D}_1) \leq \mathsf{Adv}_{\mathcal{A}_\Gamma}^{\mathbf{Game\ 1}}(\mathcal{D}_0^1, \mathcal{D}_1) + \varepsilon_1,$$

where ε_1 is a quantity, defined by

$$\varepsilon_1 := \Big|\Pr_{x \leftarrow \mathcal{D}_0}[\mathcal{A}_\Gamma(x) = 1] - \Pr_{x \leftarrow \mathcal{D}_0^1}[\mathcal{A}_\Gamma(x) = 1]\Big|.$$

Clame 1. ε_1 is bounded by an avantage of the attacker in AnoBES scheme, namely

$$\varepsilon_1 \leq \mathsf{Adv}_{\mathsf{AnoBES}}.$$

Indeed, assume the contrary, that there exists a polynomial time attacker $\mathcal{A}_{\mathsf{DIST}}$ is able to distinguish between 2 distributions $\mathcal{D}_0$ and $\mathcal{D}_0^1$ with a non-negligible probability. We build a simulator S to break the Π scheme as follows:
The simulator takes as input a public key ek_Π and generates $(\ell - 1)$ pairs public key and secret key $\{\mathsf{ek}_i, \mathsf{MSK}_i\}_{i=2}^{\ell}$. S passes $\mathsf{ek} = (\mathsf{ek}_\Pi, \mathsf{ek}_2, \ldots, \mathsf{ek}_\ell)$ to $\mathcal{A}_{\mathsf{DIST}}$.
S also collects some parameters such as: the shares $\{m_1^0, \ldots, m_\ell^0\}$, $\{m_1^1, \ldots, m_\ell^1\}$ and the family of mixture $\{\mathcal{M}_1, \mathcal{M}_2, \ldots, \mathcal{M}_\ell\}$. By quering to the challenger of the scheme Π with the shares m_1^0, m_1^1 and the mixture $\mathcal{M}_1$, S receives a ciphertext has form, $\mathsf{Encrypt}(\mathsf{ek}_1, m_1^b, \mathcal{M}_1)$, bit b chosen randomly by the challenger. The others ciphertexts

$\{\mathsf{Encrypt}(\mathsf{ek}_j, m_j^0, \mathcal{M}_j)_{j=2}^{\ell}\}$ generated by the simulator as well to establish a full ciphertext

$$\Big(\mathsf{Encrypt}(\mathsf{ek}_1, m_1^b, \mathcal{M}_1), \mathsf{Encrypt}(\mathsf{ek}_2, m_2^0, \mathcal{M}_2), \ldots, \mathsf{Encrypt}(\mathsf{ek}_\ell, m_\ell^0, \mathcal{M}_\ell)\Big).$$

By our assumption, $\mathcal{A}^{\mathsf{DIST}}$ can distinguish efficiently 2 distributions as above, as soon as $\mathcal{A}^{\mathsf{DIST}}$ outputs bit b, the simulator S will return the same value b. We see that if $m_1^0 = m_1^1$, two distributions $\mathcal{D}_0$ and $\mathcal{D}_0^1$ are coincide.

In brief, we already built an efficient simulator to break the scheme Π and it is a contradiction because Π is $\mathsf{IND-CPA}$ secure.

Game $\mathbf{G_2}$: This game is identical with **Game 1** with the different that the challenger only changes the second coordinate in $\mathcal{D}_0^1$ by $\Pi.\mathsf{Encrypt}(\mathsf{ek}_2, m_2^1, \mathcal{M}_2)$ and still does not do anything with $\mathcal{D}_1$. Thus, $\mathsf{Adv}_{\mathcal{A}_\Gamma}$ will receive a challenger ciphertext, sampled from one of two computationally indistinguishable distributions $\mathcal{D}_0^2$ and $\mathcal{D}_1$, where

$$\mathcal{D}_0^2 = \Big(\Pi.\mathsf{Encrypt}\ \ (\mathsf{ek}_1, m_1^1, \mathcal{M}_1),$$
$$\Pi.\mathsf{Encrypt}(\mathsf{ek}_2, m_2^1, \mathcal{M}_2), \ldots, \Pi.\mathsf{Encrypt}(\mathsf{ek}_\ell, m_\ell^0, \mathcal{M}_\ell)\Big)$$

We denote advantage of the adversary in this game by $\mathsf{Adv}_{\mathcal{A}_\Gamma}^{\mathbf{Game\ 2}}(\mathcal{D}_0^2, \mathcal{D}_1)$. And from this, we have

$$\mathsf{Adv}_{\mathcal{A}_\Gamma}^{\mathbf{Game\ 1}}(\mathcal{D}_0^1, \mathcal{D}_1) \leq \mathsf{Adv}_{\mathcal{A}_\Gamma}^{\mathbf{Game\ 2}}(\mathcal{D}_0^2, \mathcal{D}_1) + \varepsilon_2,$$

where ε_2 is a quantity, defined by

$$\varepsilon_2 := \Big|\Pr_{x \leftarrow \mathcal{D}_0^1}[\mathcal{A}_\Gamma(x) = 1] - \Pr_{x \leftarrow \mathcal{D}_0^2}[\mathcal{A}_\Gamma(x) = 1]\Big|.$$

By an argument analogous to the **Clame 1**. We get $\varepsilon_2 \leq \mathsf{Adv}_{\mathsf{AnoBES}}$.

Game $\mathbf{G_\ell}$: We substitute the ℓ^{th} coordinate of the distribution $\mathcal{D}_0^\ell$ by $\Pi.\mathsf{Encrypt}(\mathsf{ek}_\ell, m_\ell^1, \mathcal{M}_\ell)$ and still keep no change the distribution $\mathcal{D}_1$. $\mathsf{Adv}_{\mathcal{A}_\Gamma}$ will receive a challenger ciphertext, sampled from one of two computationally identical distributions $\mathcal{D}_0^\ell$ and $\mathcal{D}_1$. We denote advantage of the adversary in this game by $\mathsf{Adv}_{\mathcal{A}_\Gamma}^{\mathbf{Game\ }\ell}(\mathcal{D}_0^\ell, \mathcal{D}_1)$. And from this, we have

$$\mathsf{Adv}_{\mathcal{A}_\Gamma}^{\mathbf{Game\ }\ell-\mathbf{1}}(\mathcal{D}_0^{\ell-1}, \mathcal{D}_1) \leq \mathsf{Adv}_{\mathcal{A}_\Gamma}^{\mathbf{Game\ }\ell}(\mathcal{D}_0^\ell, \mathcal{D}_1) + \varepsilon_\ell = \varepsilon_\ell,$$

where ε_ℓ is a quantity, defined by

$$\varepsilon_\ell := \Big|\Pr_{x \leftarrow \mathcal{D}_0^{\ell-1}}[\mathcal{A}_\Gamma(x) = 1] - \Pr_{x \leftarrow \mathcal{D}_0^{\ell}}[\mathcal{A}_\Gamma(x) = 1]\Big| \leq \mathsf{Adv}_{\mathsf{AnoBES}}.$$

Putting the above arguments altogether and applying triangle inequality we have:

$$\left|\mathsf{Adv}^{\mathbf{Game\ 0}}_{\mathcal{A}_\Gamma}(\mathcal{D}_0,\mathcal{D}_1)\right| = \left|\mathsf{Adv}^{\mathbf{Game\ 0}}_{\mathcal{A}_\Gamma}(\mathcal{D}_0,\mathcal{D}_1) - \mathsf{Adv}^{\mathbf{Game}\ \ell}_{\mathcal{A}_\Gamma}(\mathcal{D}_0^\ell,\mathcal{D}_1)\right|$$
$$\leq \sum_{i=1}^{\ell} \varepsilon_\ell \leq \ell.\mathsf{Adv}_{\mathsf{AnoBES}}.$$

A.3 Public Trace and Revoke for AnoBES

Adapted from the definition of the trace and revoke system in [2], we will define a trace and revoke system in the black-box model, namely aBE system with a black-box tracing algorithm. A Trace and Revoke (TR) system consists of algorithms as below:

Setup(λ, N)**:** Takes as input the security parameter λ, maximal number of user N and a maximum malicious coalition size t. It outputs the global parameters param of the system, a public key ek, a master secret key MSK and a tracing key TK, where TK is public.

Extract$(\mathsf{ek}, \mathsf{MSK}, i)$**:** Take as input the public key ek, the master secret key MSK and a user index $i \in \mathcal{U}$, the algorithm extracts the decryption key dk_i which is sent to the user i.

Encrypt$(\mathsf{ek}, M, \mathcal{S})$**:** Take as input the public key ek, a message $M \in \mathcal{PT}$ and a set of target users $\mathcal{S} \subset \mathcal{U} = \{1, 2, \ldots, N\}$, outputs a ciphertext $c \in \mathcal{CT}$.

Decrypt$(\mathsf{ek}, \mathsf{dk}_i, c)$**:** Take as input the public key ek, the decryption key dk_i of user i and a ciphertext $c \in \mathcal{CT}$. If $i \in \mathcal{U} - \mathcal{S}$ in the sense that the user is outside $\mathcal{S}$, the algorithm outputs a message $M \in \mathcal{PT}$.

Tracing$(\mathcal{D}, \mathsf{ek}, \mathsf{TK})$**:** It takes as input the public key ek, the tracing key TK and has access to a pirate decoder $\mathcal{D}$. The tracing algorithm outputs the identity of at least one user that participated in building $\mathcal{D}$.

In this paper, we shall assume that pirate devices are resettable, meaning that they do not maintain state during the tracing process. Moreover, we also assume that the tracing procedure is being considered in the minimal black-box access model. It means that the tracer has access to $\mathcal{D}$ by using an oracle $\mathcal{O}^{\mathcal{D}}$. The oracle $\mathcal{O}^{\mathcal{D}}$ will be fed the input which has the form $(c, M) \in (\mathcal{CT}, \mathcal{PT})$. The tracer will get 1 from the output $\mathcal{O}^{\mathcal{D}}$ in the case that the decoder decrypts correctly the ciphertext c, i.e. $\mathcal{D}(c) = M$ and will get 0 in the other case. We require that the pirate device $\mathcal{D}$ decrypts correctly with a non-negligible probability.

$$\Pr_{\substack{M \hookleftarrow U(\mathcal{PT}) \\ c \hookleftarrow \mathsf{Encrypt}(m)}} \left[\mathcal{O}^{\mathcal{D}}(c, M) = 1\right] \geq \frac{1}{|\mathcal{PT}|} + \frac{1}{\lambda^c},$$

for some constant $c > 0$.

For a given N, t, λ, we define security of TR system using the following game

The traceability is defined via the following game between an attacker $\mathcal{A}$ and a challenger $\mathcal{B}$:

Tracing Game

1. The adversary $\mathcal{A}$ outputs a set $\mathcal{T} = \{u_1, u_2, \ldots, u_t\} \subset \{1, \ldots, N\}$ of colluding users.
2. The challenger $\mathcal{B}$ runs $\mathsf{Setup}(\lambda, N)$ and sends all public keys ek, tracing key TK and decryption keys $\mathsf{dk}_{u_1}, \ldots, \mathsf{dk}_{u_t}$ to the adversary $\mathcal{A}$.
3. The adversary $\mathcal{A}$ creates a resettable pirate decoder $\mathcal{D}$ so that the pirate decoder decrypts correctly the ciphertexts with at least ε.
4. The challenger $\mathcal{B}$ executes the procedure $\mathsf{Tracing}(\mathcal{D}, \mathsf{ek}, \mathsf{TK})$ and outputs a set $\mathcal{L} \subset \mathcal{T}$ that is accused.

We say that the adversary $\mathcal{A}$ wins the game if the set $\mathcal{L}$ is either empty, or is not a subset of $\mathcal{T}$ and $\Pi = (\mathsf{Setup}, \mathsf{Extract}, \mathsf{Encrypt}, \mathsf{Decrypt}, \mathsf{Tracing})$ is a TR scheme with tracing success probability α against t-coalition ε-pirates if no polynomial time attacker $\mathcal{A}$ can win the game described above with probability more than $1 - \alpha$. We denote by $\mathsf{Adv}_{\mathsf{TR}}$ the probability that adversary $\mathcal{A}$ wins this game.

We remake that any Anonymous Broadcast Encryption is also a TR system. Indeed, it is relatively straightforward to see that revocation is implied directly from the semantic security of BE system. About traceability of Anonymous BE systems, we can apply the linear tracing technique for the Anonymous BE system to capture this ability. We do not present arguments in the general case. Instead, we will apply directly this technique in AnoBES context.

1. For $i = 0$ to t, do the following:
   (a) Let $\mathsf{cnt} \leftarrow 0$.
   (b) Repeat the following steps $W \leftarrow 8\lambda(t/\varepsilon)^2$ times:
      i. $M \hookleftarrow U(\mathcal{PT})$
      ii. $c \hookleftarrow U\left(\mathrm{Span}(\boldsymbol{x}_1^+, \ldots, \boldsymbol{x}_i^+)^\perp\right) + \lfloor \nu_{\alpha q} \rceil^{m+1} + \left(\frac{M\lfloor q/2 \rfloor}{0}\right)$
      iii. Call oracle $\mathcal{O}^{\mathcal{D}}$ on input c and if $\mathcal{O}^{\mathcal{D}}(c, M) = 1$ then $\mathsf{cnt} \leftarrow \mathsf{cnt} + 1$

(c) Let $\tilde{p}_i$ be the fraction of times that $\mathcal{D}$ decrypted the ciphertexts correctly. We have $\tilde{p}_i = \mathsf{cnt}/W$.

2. Let $\mathcal{L}$ be the set of all $i \in \{1, \cdots, t\}$ for which $\tilde{p}_i - \tilde{p}_{i-1} \geq \varepsilon/4t$.
3. Output the set $\mathcal{L}$ as the set of guilty colluders.

We define p_i as the probability the pirate decoder $\mathcal{D}$ decrypts correctly the ciphertext for $i \in [0, t]$.

$$p_i \;=\; \Pr_{\substack{\boldsymbol{c} \hookleftarrow Tr_i \\ M \hookleftarrow U(\{0,1\})}} \left[\mathcal{O}^{\mathcal{D}} \left(c + \left[\frac{M \cdot \lfloor q/2 \rfloor}{\mathbf{0}} \right], M \right) = 1 \right],$$

where

$$Tr_i = U\left(\mathrm{Span}(\boldsymbol{x}_1^+, \ldots, \boldsymbol{x}_i^+)^\perp\right) + \lfloor \nu_{\alpha q} \rceil^{m+1}.$$

A gap between p_{i-1} and p_i is meant to indicate that u_i is a traitor.

We will compute an approximation $\tilde{p}_i$ of the probabilities p_i using $W = 8\lambda(t/\varepsilon)^2$ samples. Applying the additive form of the Chernoff bound $\Pr\left[|p - \tilde{p}| > \varepsilon\right] < 2e^{-W\varepsilon^2}$, we have

$$\begin{aligned} \Pr\left[|p_i - \tilde{p}_i| > \varepsilon/(16t)\right] &< 2e^{-2W(\varepsilon/(16t))^2} \\ &= 2e^{-2\cdot 8\lambda(t/\varepsilon)^2(\varepsilon/(16t))^2} \\ &= 2e^{-\lambda/16} < 2e^{-\lambda/64}, \end{aligned}$$

which is negligible in the security parameter λ for all $i = 1, \ldots, t$. Therefore, we may assume from here on that $|p_i - \tilde{p}_i| \leq \varepsilon/(16t)$ for all $i = 1, \ldots, t$.

The confirmation and soundness properties of the tracing algorithm followed directly from Theorem 24, 25 in [19].

However, public tracing is more challenging. Fortunately, our system also supports public traceability. It comes directly from the [19]. We thus achieve a Public TR for small Universe.