A Concise Bounded Anonymous Broadcast Yielding Combinatorial Trace-and-Revoke Schemes

No Author Given

No Institute Given

Abstract. Broadcast Encryption is a fundamental primitive, allowing sending a secure message to any chosen target set of N users.While many efficient constructions are known, understanding the efficiency possible for an Anonymous Broadcast Encryption (AnoBE), i.e., one which can hide the target set, is quite open. The best solutions by Barth, Boneh, and Waters ('06) and Libert, Paterson, and Quaglia ('12) are built on public key encryption (PKE) and their ciphertext sizes are, in fact, N times that of the underlying PKE. Kiayias and Samary ('12) also showed a lower bound stating that linear penalty is the best possible if N is an independent unbounded parameter. However, when considering certain constraints on the user set's size, the problem remains interesting. We consider the problem of comparing AnoBE with PKE under the same assumption. In the restricted case for 2-user only, Phan, Pointcheval and Strefler gave a 2-user AnoBE scheme relying on ElGamal PKE which reduced the rate from 2 to 1.5.

We first present an AnoBE construction for up to k users from LWE assumption, where k is bounded by the scheme security parameter. We call schemes where the number of users is bounded in this way *Anonymous Broadcast Encryption for Bounded Universe* – AnoBEB. Further, our scheme is as efficient as the underlying LWE public-key encryption, the rate is, in fact, 1 and thus optimal.

More interestingly, employing AnoBEB, we introduce a new approach to construct an efficient "Trace and Revoke scheme" which combines the functionalites of revocation and of tracing traitors. Note that, as was put forth by Kiayias and Yung (EUROCRYPT '02), combinatorial traitor tracing schemes can be constructed by combining a system for small universe, integrated via an outer traceability codes (collusion-secure code or identifying parent property (IPP) code). There were many efficient traitor tracing schemes from traceability codes but no known scheme supports revocation. Our new approach integrates our AnoBEB system with a Robust IPP code, introduced by Barg and Kabatiansky (IEEE IT '13). This shows an interesting use for robust IPP in cryptography. The robust IPP codes were only implicitly shown by an existence proof. In order to make our technique concrete, we propose two explicit instantiations for robust IPP codes. Our final construction gives the most efficient trace and revoke scheme in the bounded collusion model.

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

1 Introduction

Broadcast encryption is a fundamental 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 while the other users learn nothing about the content. The first constructions of broadcast encryption were proposed by Berkovits [6], and most notably by Fiat-Naor [19] who advocated that an efficient scheme should be more efficient than just repeating a single ciphertext per user. Thereafter, many interesting schemes were proposed, in particular Boneh, Gentry and Waters [8] introduced a scheme with a constant size ciphertext.

Privacy, and anonymity of receivers, in particular, are important in numerous real-life applications. Unfortunately, it turned out to be extremely difficult to hide the target set in broadcast encryption. To date, no efficient anonymous broadcast encryption has been constructed. Yet, anonymity in broadcast encryption system was considered by many, see: [5], [32], [18], [37], [31]. The state of the art constructions from by Barth *et al.* and Libert *et al.* [5, 32] start from a public-key encryption (PKE) and result in schemes with ciphertext size which is N times the ciphertext size of the underlying PKE scheme. Moreover, justifying the above results, Kiayias and Samari [29] proved lower bounds: 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 $\Omega(r + n)$ for any set of r revoked users. Note that it can be that $s = O(N)$ and $r = O(N)$. Hence, unfortunately, sub-linear complexity in the number of users is impossible.

However, in practice, the case where N is a constant has been largely employed. In fact, all combinatorial traitor tracing schemes start with a scheme of small bounded size (say 2-user for collusion-secure codes in Boneh-Shaw scheme [10] and in Kiayias and Yung scheme [30] and q-user for q-IPP codes in Chor-Fiat-Naor scheme [15] and in Phan-Safavi-To scheme [36]), and then combine these schemes to achieve a general one. So we ask here: What can be done for a user set whose size is not an unbounded independent parameter? does the ciphertext size of such an anonymous broadcast encryption scheme still grows linearly in the number of users, comparing to the single-user encryption, namely the corresponding public-key encryption from the same assumption? For $N = 2$, Phan *et al.* [37] provided a construction of anonymous broadcast encryption scheme in which the ciphertext length is about 1.5 times the ciphertext size of its underlying ElGamal encryption scheme. Here, we will consider the case where N is much larger but is bounded by another system parameter (namely the security parameter). We call this case “anonymous broadcast encryption for bounded universe,” or for short (AnoBEB).

Combination of AnoBEB *with* IPP *code.* From an AnoBEB, we will construct the first Trace and Revoke system that is based on a traceability code. Previous constructions from a traceability code only yielded traitor tracing scheme (TT) but with no revocation. We first explain, from the classical combinatorial method, any AnoBEB for q-user can be integrated with a $q-$ary IPP code to produce a traitor tracing scheme.

- As we know from [8, 11, 25] (actually, a flaw and a fix were recently given in [25]), any public-key anonymous broadcast encryption (in fact, they proved this for a more restricted case of anonymity, called augmented broadcast encryption) also supports tracing traitor. Therefore, any solution for AnoBEB directly implies a trace and revoke scheme for a small universe.
- Combinatorial methods of designing a traitor tracing consist of two steps: first, construct a small scheme, then combine these schemes to achieve a general one. This method was proposed in the very first traitor tracing paper of Chor-Fiat-Naor [15]. Kiayias and Yung [30] integrated a 2-user traitor tracing scheme with a collusion-secure code [10] into a TT scheme. It can be summarized as follows: First, 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 then essentially encrypts each sub-key twice with PKE and gets sub-ciphertexts. Each recipient, provided sub-keys associated with 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.

  Table 1 shows an example of a traitor tracing with binary collusion secure code. A legitimate user is assigned a codeword in the code. The authority will decompose a session key K into segments K_j according to the length ℓ of the code. In each sub-system, the segment of session key K_j will be encrypted twice alternately with public-keys $pk_{0,j}$ or $pk_{1,j}$. Each user i provided a secret-key $sk_{0,j}$ or $sk_{1,j}$ depending on the value of its codeword at position j. The user thus provided ℓ secret-keys and employs these secret-keys to recover the sub-session keys K_j, $j = 1, \ldots, \ell$ from one of two ciphertexts $c_{0,j}$ or $c_{1,j}$. Finally, the user combines them to obtain the original session key K. The tracing procedure consists of using the traceability in each 2-user

Key assignment :									
	Table 0	$pk_{0,1}$	$pk_{0,2}$	$pk_{0,3}$	$pk_{0,4}$	$pk_{0,5}$	...	$pk_{0,\ell}$	
	Table 1	$pk_{1,1}$	$pk_{1,2}$	$pk_{1,3}$	$pk_{1,4}$	$pk_{1,5}$	...	$pk_{1,\ell}$	
	Codeword i	1	0	0	1	0	...	1	
	user i	$sk_{1,1}$	$sk_{0,2}$	$sk_{0,3}$	$sk_{1,4}$	$sk_{0,5}$	...	$sk_{1,\ell}$	
Encryption :									
	Session Key	$K_1\oplus$	$K_2\oplus$	$K_3\oplus$	$K_4\oplus$	$K_5\oplus$	...	$\oplus K_\ell$	$= K$
	Ciphertext	$c_{0,1}$	$c_{0,2}$	$c_{0,3}$	$c_{0,4}$	$c_{0,5}$	...	$c_{0,\ell}$	
		$c_{1,1}$	$c_{1,2}$	$c_{1,3}$	$c_{1,4}$	$c_{1,5}$	...	$c_{1,\ell}$	

Table 1: Traitor tracing with binary collusion secure code

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.
- The above method is then generalized for $q-$ary identifiable parent property (IPP) code. A $q-$ary IPP code $\mathcal{C}$ is an code if whenever we are given a descendant (a word) that is generated by a subset of codewords (parents) of code $\mathcal{C}$, we are able to determine at least one of the parents. A traitor tracing scheme can then be obtained by applying q times PKE (instead of 2 times PKE when using binary collusion secure code) at each of ℓ positions associated with a $q-$ary IPP code [36]. Now, if we replace q times PKE by an AnoBEB for q-user, which is as efficient as the underlying PKE, we can save a factor q in ciphertext efficiency. Therefore, the design of an efficient AnoBEB has a direct impact on the IPP code-based TT.

While the application of an AnoBEB in constructing a traitor tracing is directly inherent from the classical combinatorial method, as explained above, we furthur investigate how it can help to construct trace and revoke systems. Note that traceability and revocation are very difficult to be combined. We refer to [11] for a discussion about the difficulties of combining these two "orthogonal" functionalities.

1.1 Contributions.

We present three main results:

1. First note that it was not known how to generalize a PKE to an anonymous BE scheme for, say, a bounded universe of N users (AnoBEB for short) with a ciphertext rate (between the anonymous BE scheme and the underlying PKE) strictly less than N, for any $N \neq 2$. We show a transformation from LWE PKE into an AnoBEB with an optimal rate. The security of our proposed schemes for k users relies on the $k-\mathsf{LWE}$ problem [33].
2. We then propose a new method for achieving a trace and revoke system from an AnoBEB, a secret sharing scheme, and a robust IPP code. It is worth remarking that robust IPP code, introduced by Barg *et al.* [4], is an interesting generalization of IPP code, but to the best of our knowledge, till today it has not found any application in cryptography.
3. We, finally, give a concrete construction of a trace and revoke system. In [4], only a proof of existence of robust IPP codes was given. We propose two explicit instantiations of their result, in adding a condition to deal with the revocation. Our final trace and revoke system also enjoys the more demanding "public traceability" property as in [11, 13, 36].

1.2 Techniques.

LWE- *based anonymous broadcast encryption for bounded universe.* Ling *et al.* [33] introduced the first lattice-based traitor tracing scheme (LPSS) based on the

$k - \mathsf{LWE}$ assumption. They showed a polynomial-time reduction from $k - \mathsf{LWE}$ to LWE, so their scheme is as efficient as the LWE encryption. A natural question is whether one can also rely on $k - \mathsf{LWE}$ to design an anonymous, revoke, or broadcast encryption scheme. Revoking users is a very difficult task and the following question is still open: for a constant number of revoked users, can we design a revoke scheme that is comparably efficient to the underlying encryption. Based on $k - \mathsf{LWE}$, 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 $k - \mathsf{LWE}$, 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 aim at an anonymous broadcast encryption for a small universe.

Surprisingly, our construction of a AnoBEB scheme comes from a basic "tweaking purpose" idea: switching the tracing procedure in [33] to be functional as a broadcast encryption. We first recall that in the LPSS traitor tracing scheme, the linear tracing technique [15] 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. In the subsequent steps, the tracer will disable, one by one, users in the group, preventing them from decrypting the ciphertext. We observe that if we switch the suspected users in LPSS scheme to be the legitimate users, and the removed users in the suspected set to the revoked users, then, in fact, in principle we get a broadcast encryption. Because the LPSS traitor tracing can deal with a bounded number of traitors, we actually get a broadcast encryption for a bounded number of users, that we call broadcast encryption for bounded universe.

The main remaining technical difficulty is to prove the anonymity property of this broadcast encryption. 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 [18], which only corrupts users outside both $\mathcal{S}_0, \mathcal{S}_1$, then the proof is direct because from the $k - \mathsf{LWE}$ assumption, the encryption for $\mathcal{S}_0$ and for $\mathcal{S}_1$, both, look like random ciphertexts to the adversary. It is more challenging to consider a general adversary which can also corrupt the key in the intersection of $\mathcal{S}_0$ and $\mathcal{S}_1$. Fortunately, we can exploit an intermediate theorem in [33] 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, our technique applies an hybrid argument which moves 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 users in $\mathcal{S}_1 \setminus \mathcal{S}_0$ (or in $\mathcal{S}_1 \setminus \mathcal{S}_0$, respectively).

Revocation from robust IPP *code.* We next explain why it is difficult to get revocation with code-based schemes and how we can overcome the problem. We recall that the binary collusion secure code is well suitable for traitor tracing, quite efficient with $\widetilde{O}(t^2n^2)$ ciphertext size; its shortcoming is the incapacity of supporting revocation. In fact, to revoke a group of users, the authority has to disable the ability to decrypt with 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 legitimate non-revoked users will be affected, and will not be able to decrypt anymore. A non-trivial remedy is for the designer of system to choose a code with big alphabet for example $q-$ary IPP code instead of a binary collusion secure code with alphabet size two. Revocation will decrease the number of valid keys slightly. Certainly, in this case, the possibility that legitimate users will be excluded from the system with revoked users must also be taken into account. A secret sharing scheme, in turn, is a mechanism that allows us to think about a solution: a legitimate user only needs to have a certain fraction (over the threshold) of the sub-keys to be able to recover the original message. However, this reduced requirement gives an advantage to the pirates as well: they become stronger as they do not need to put all sub-keys in the pirate decoder; namely, they are permitted to delete sub-keys. The introduction of robust IPP of Barg *et al.* [4] which allows the identification of parents even if some positions are intentionally erased, allows for a tool to deal with the above problem. We propose here a new generic method for designing a trace and revoke system from robust IPP codes and AnoBEB. 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 AnoBEB.

Finally, because robust IPP codes were only implicitly shown in [4], we propose two explicit constructions for robust IPP codes. Our final construction results in the most efficient trace and revoke scheme in the bounded collusion model.

1.3 Related works

As shown in the paper of Boneh and Waters (BW) at [11], traceability and revocation are very difficult to be combined. There exist 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 more recently by [44] (which embeds a collusion secure code into a broadcast system), Nishimaki, Wichs, and Zhandry (NWZ) [35], and by Agrawal *et al.* [1]. The BW and NWZ schemes are quite powerful in that they support malicious collusions of unbounded size, but, on the other hand, their ciphertexts are very large (in BW, the size grows proportionally to $\sqrt{N}$, where N is the total number of users and in NWZ, they use the inefficient general functional encryption schemes).

For bounded schemes where the number of traitors is small, the Agrawal *et al.*'s scheme [1], relying on learning with errors, is quite efficient with ciphertext size $\widetilde{O}(r+t+n)$ where r is the maximum number of revoked users, t the maximum number of traitors, and n the security parameter. But they only support a weak level of tracing: black-box confirmation with the assumption that the tracer gets a suspect set that contains all the traitors. Concerning black-box trace and revoke in bounded collusion model, the instantiation of the NWZ scheme also gives the most efficient construction. However, as stated in [1], the generic nature of their construction results in loss of concrete efficiency: when based on the bounded collusion FE of [23], the resulting scheme has a ciphertext size growing at least as $\widetilde{O}((r+t)^5\mathcal{P}oly(n))$; by relying on learning with errors, this blowup can be improved to $\widetilde{O}((r+t)^4\mathcal{P}oly(n))$, but at the cost of relying on heavy machinery such as attribute based encryption [24] and fully homomorphic encryption [22]. Our trace and revoke result, in contrast, achieves ciphertext size $\widetilde{O}((r+t^2)(n^3)\log N)$ with black-box tracing like in [35], which is the prevalent standard model for tracing and is by far more realistic and useful than the black-box confirmation as in [1].

Combinatorial schemes are considered in [34] and in [2]. In [34] only a weak form of black-box tracing is considered, 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 [30], [9], [34], [7].

2 Definitions and Preliminaries

2.1 Secret sharing schemes

A secret sharing scheme ($\mathcal{SSS}$) [41] distributes a secret amongst a group of users, each of whom keeps a share. The $\mathcal{SSS}$ contains two algorithms: Share and Combine, defined formally as follows:

Definition 1 ($(m,n)-$Secret Sharing Scheme).
Share(K,m,n)*: Takes as input a secret bit string K and positive integers m, n. It outputs 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.*

Correctness means that any m-subset of $\{(i,s_i)\}$ generated by Share(K,m,n), Combine outputs the string K generated by Share. Furthermore, when generated as part of Share then the bit string K must be uniformly distributed. *Security* means that any less than m shares yield no information about K.

2.2 Trace and Revoke Systems

We next recall the standard definition of a trace and revoke scheme. Let $\mathcal{PT}$ and $\mathcal{CT}$ denote the plaintext and ciphertext spaces, respectively. We also let $U(\mathcal{PT})$ denote the uniform distribution over plaintext space $\mathcal{PT}$.

Adapted from the definition of the trace and revoke system in [1], we will present a trace and revoke system for a universe $\mathcal{U}=\{1,\ldots,N\}$ in the black-box model. A Trace and Revoke (TR) system, in turn, consists of the following algorithms:

$\mathsf{Setup}(1^n, t, r)$**:** Takes as input the security parameter n, a maximum malicious coalition size t and the bound r on the number of revoked users. It outputs the global parameters param of the system, a public key ek and a master secret key MSK.

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

$\mathsf{Encrypt}(\mathsf{ek}, M, \mathcal{R})$**:** Takes as input the public key ek, a message $M \in \mathcal{PT}$ and a set of revoked users $\mathcal{R} \subset \mathcal{U}$ (cardinality $\leq r$), outputs a ciphertext $c \in \mathcal{CT}$.

$\mathsf{Decrypt}(\mathsf{ek}, \mathsf{dk}_i, c)$**:** Takes as input the public key ek, the decryption key dk_i of user i and a ciphertext $c \in \mathcal{CT}$. The algorithm outputs the message $M \in \mathcal{PT}$ or an invalid symbol $\perp$.

$\mathsf{Tracing}(\mathcal{D}, \mathcal{R}, \mathsf{ek})$**:** is a black-box tracing algorithm which takes as input a set $\mathcal{R}$ of $\leq r$ revoked users, public key ek and has access to a pirate decoder $\mathcal{D}$. The tracing algorithm outputs the identity of at least one user who participated in building $\mathcal{D}$ or an invalid symbol $\perp$.

The correctness requirement is that, with overwhelming probability over the randomness used by the algorithms, we have:

$$\forall M \in \mathcal{PT}, \forall i \notin \mathcal{R} : \mathsf{Decrypt}(\mathsf{ek}, \mathsf{dk}_i, \mathsf{Encrypt}(\mathsf{ek}, M, \mathcal{R})) = M,$$

for any set $\mathcal{R}$ of $\leq r$ revoked users.

Requirement on the pirate decoder

- The classical requirement is that the pirate decoder $\mathcal{D}$ is a device that is able to decrypt successfully any ciphertext with overwhelming probability and the pirate device is resettable, meaning that it should not maintain state during the tracing process. In [33], a strong model of pirate decoder was considered where the tracing algorithm is executing in minimal access black-box model and the pirate decoder is only required to have a non-negligible probability of success. More formally, the tracer is allowed to access $\mathcal{D}$ via an oracle $\mathcal{O}^{\mathcal{D}}$. It means that the oracle $\mathcal{O}^{\mathcal{D}}$ will be fed the input which has the form $(\mathbf{c}, M) \in (\mathcal{CT}, \mathcal{PT})$. 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. $\mathcal{D}(\mathbf{c}) = M$ and will get 0 in the other case. It requires that the pirate device $\mathcal{D}$ decrypts correctly with a non-negligible probability (ϵ) in the security parameter n, namely:

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

  for some constant $\alpha > 0$.
- In [26], the authors show a flaw in the transformation of an augmented broadcast encryption into traitor tracing and proposed a fix in which a very strong notion of Pirate Distinguisher [26, 35] was put forth, in place of the classical notion of pirate decoder. The Pirate Distinguisher is not required to

output entire message (or an indicator bit as in minimal access model) nor to decrypt with high probability every ciphertexts which are taken from random messages. Instead, it is enough that the pirate decoder can distinguish the encryption of two different messages M_0, M_1 of its choice. We call $\mathcal{D}$ is a ε-useful Pirate Distinguisher if

$$\left| \mathsf{Pr}\left[\mathcal{D}\left(\mathbf{c}_b\right) = b : \begin{array}{c} T \leftarrow \mathcal{A}(1^n); (\mathsf{MSK}, \mathsf{ek}) \leftarrow \mathsf{Setup}(1^n, N); \\ \{\mathsf{dk}_i \leftarrow \mathsf{Extract}(\mathsf{ek}, \mathsf{MSK}, i)\}_{i \in T}; \\ (\mathcal{D}, M_0, M_1) \leftarrow \mathcal{A}(\mathsf{ek}, \{\mathsf{dk}_i\}_{i \in T}); \\ b \xleftarrow{\$} \{0,1\}; \mathbf{c}_b \leftarrow \mathsf{Encrypt}(\mathsf{ek}, M_b, S); S \subset [N] \end{array} \right] - \frac{1}{2} \right| \geq \varepsilon,$$

In this work, we will deal with this notion of pirate distinguisher which is actually the strongest notion about the usefulness of pirate decoders.
Interestingly, in the case of bit encryption like in LPSS scheme [33] and in our scheme, the notion of pirate distinguisher is equivalent to the pirate decoder in the minimal access black-box model. Indeed, as there are only two messages 0 and 1, the requirement that the oracle $\mathcal{O}^{\mathcal{D}}$ (in the definiton of pirate decoder) can correctly decrypt ciphertexts of one of these two messages with non-negligible probability is equivalent to a pirate distinguisher that can distinguish the encryption of the two messages 0 and 1. Therefore, the LPSS scheme is also secure when considering the notion of pirate distinguisher. Inherently, our tracing algorithm can also deal with pirate distinguishers.

Semantic Security. The CPA security of a trace-and-revoke scheme TR is defined based on the following game.

- The challenger runs $\mathsf{Setup}(1^n, t, r)$ and gives the produced public key ek to the adversary $\mathcal{A}$.
- The adversary (adaptively) chooses a set $\mathcal{R} \subset \mathcal{U}$ of $\leq r$ revoked users. The challenger gives $\mathcal{A}$ all the dk_i for all $i \in \mathcal{R}$.
- The adversary then chooses two messages $M_0, M_1 \in \mathcal{PT}$ of equal length and gives them to the challenger.
- The challenger samples $b \hookleftarrow \{0,1\}$ and provides $c \hookleftarrow \mathsf{Encrypt}(\mathsf{ek}, M_b, \mathcal{R})$ to $\mathcal{A}$.
- Finally, the adversary returns its guess $b' \in \{0,1\}$ for the b chosen by the challenger. The adversary wins this game if $b = b'$.

We define $\mathsf{Succ}^{\mathsf{IND}}(\mathcal{A}) = \Pr[b' = b]$, the probability that $\mathcal{A}$ wins the game. We say that a TR system is semantically secure (IND) 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 $\mathsf{Adv}^{\mathsf{IND}}(\mathcal{A}) = |\mathsf{Succ}^{\mathsf{IND}}(\mathcal{A}) - \frac{1}{2}| = |\Pr[b' = b] - \frac{1}{2}|$.

Traceability. The tracing game between an attacker $\mathcal{A}$ and a challenger $\mathcal{B}$ is defined as following:

1. The challenger runs $\mathsf{Setup}(1^n, t, r)$ and gives ek to $\mathcal{A}$.

2. The adversary $\mathcal{A}$ outputs a set $\mathcal{T} \subset \{u_1, u_2, \ldots, u_t\} \subset \{1, \ldots, N\}$ of colluding users. We assume that $\mathcal{T} \cap \mathcal{R} = \emptyset$. The adversary sends t arbitrary key queries in an adaptive way to $\mathcal{B}$.
3. The challenger $\mathcal{B}$ responds to $\mathcal{A}$ decryption keys $\mathsf{dk}_1, \ldots, \mathsf{dk}_t$.
4. The adversary $\mathcal{A}$ outputs two messages M_0, M_1 and creates a pirate distinguishser $\mathcal{D}$ so that it can distinguishable correctly the encryptions of M_0, M_1 with probability at least ε.
5. The challenger $\mathcal{B}$ executes the procedure $\mathsf{Tracing}(\mathcal{D}, \mathcal{R}, \mathsf{ek})$. The adversary wins the game if $\mathcal{B}$ outputs $\perp$ or an user index that does not belong to $\mathcal{T}$.

2.3 Anonymous Broadcast Encryption

A broadcast system is called anonymous (AnoBE for short) if it allows addressing a message to a subset of the users, without revealing this privileged set even to users who successfully decrypt the message. When the number of users in our system is bounded by the security parameter, we have the notion of *anonymous broadcast encryption for bounded universe* – AnoBEB. We follow the definition in [32]:

Let $\mathcal{PT}$ and $\mathcal{CT}$ denote the plaintext and ciphertext spaces, respectively. Let $\mathcal{U} = \{1, \ldots, N\}$ be the universe of users, where $N \leq k$ for some k bounded by a security parameter n. An anonymous broadcast encryption for bounded universe (AnoBEB) consists of the following algorithms:

$\mathsf{Setup}(1^n, N)$: Takes as input the security parameter n and the maximal number of users N. It outputs a public key ek and a master secret key MSK.

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

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

$\mathsf{Decrypt}(\mathsf{ek}, \mathsf{dk}_i, c)$: Takes as input the public key ek, the decryption key dk_i of user i and a ciphertext $c \in \mathcal{CT}$. The algorithm outputs the message $M \in \mathcal{PT}$ or an invalid symbol $\perp$.

The correctness requirement is that, with overwhelming probability over the randomness used by the algorithms, we have:

$$\forall M \in \mathcal{PT}, \forall i \in \mathcal{S} : \mathsf{Decrypt}(\mathsf{ek}, \mathsf{dk}_i, \mathsf{Encrypt}(\mathsf{ek}, M, \mathcal{S})) = M.$$

The CPA security of AnoBEB defined based on the following game between an adversary $\mathcal{A}$ and a challenger $\mathcal{B}$

- The challenger runs $\mathsf{Setup}(1^n, N)$ and gives the produced public key ek to the adversary $\mathcal{A}$.
- The adversary (adaptively) chooses indices $i \in \mathcal{U}$ to ask decryption keys. The challenger gives $\mathcal{A}$ all the dk_i for all required indices.

- The adversary then chooses two messages $M_0, M_1 \in \mathcal{PT}$ of equal length and a set $\mathcal{S} \subset \mathcal{U}$ of users with restriction that no index $i \in \mathcal{S}$ required decryption key before. It then gives M_0, M_1 and $\mathcal{S}$ to the challenger.
- The challenger samples $b \hookleftarrow \{0,1\}$ and provides $c \hookleftarrow \mathsf{Encrypt}(\mathsf{ek}, M_b, \mathcal{S})$ to $\mathcal{A}$.
- The adversary $\mathcal{A}$ continues asking for decryption keys for any index i outside $\mathcal{S}$.
- Finally, the adversary returns its guess $b' \in \{0,1\}$ for the b chosen by the challenger. The adversary wins this game if $b = b'$.

We define $\mathsf{Succ}^{\mathsf{IND}}(\mathcal{A}) = \Pr[b' = b]$, the probability that $\mathcal{A}$ wins the game. We say that AnoBEB is semantically secure (IND) 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

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

For anonymous game, the challenger $\mathcal{B}$ runs $\mathsf{Setup}(1^n, N)$ to obtain a public key ek and a 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 to $\mathcal{A}$ the 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 $\mathcal{S}_0, \mathcal{S}_1 \subset \mathcal{U}$ of users. We require that $\mathcal{A}$ has not issued key queries for any index $i \in \mathcal{S}_0 \triangle \mathcal{S}_1 = (\mathcal{S}_0 \setminus \mathcal{S}_1) \cup (\mathcal{S}_1 \setminus \mathcal{S}_0)$. The adversary $\mathcal{A}$ passes M and $\mathcal{S}_0, \mathcal{S}_1$ to the challenger $\mathcal{B}$. The challenger $\mathcal{B}$ randomly chooses a bit $b \in \{0,1\}$, computes $c = \mathsf{Encrypt}(\mathsf{ek}, M, \mathcal{S}_b)$ and sends c to $\mathcal{A}$.

Phase 2. $\mathcal{A}$ adaptively issues decryption key extraction queries on indices $i \notin \mathcal{S}_0 \triangle \mathcal{S}_1$ and obtains decryption keys dk_i.

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

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

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

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

2.4 Lattice and $k-\mathsf{LWE}$ problem

For two matrices A, B of compatible dimensions, let $(A\|B)$ (or sometimes $\begin{pmatrix} A \\ B \end{pmatrix}$) denote vertical concatenations of A and B. For $A \in \mathbb{Z}_q^{m\times n}$, define $\mathrm{Im}(A) =$

$\{A\mathbf{s} \mid \mathbf{s} \in \mathbb{Z}_q^n\} \subseteq \mathbb{Z}_q^m$. 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 $\{\mathbf{b} \in \mathbb{Z}_q^m \mid \forall \mathbf{c} \in X, \langle \mathbf{b}, \mathbf{c}\rangle = 0\}$.

Assume that D_1 and D_2 are distributions over a countable set X, their statistical distance is defined to be $\frac{1}{2}\sum_{x\in X}|D_1(x) - D_2(x)|$. We say that two distributions D_1 and D_2 (two ensembles of distributions indexed by n) are statistically close if their statistical distance is negligible in n. We use the notation $x \hookleftarrow D$ to refer that the element x is sampled from the distribution D. We also let $U(X)$ denote the uniform distribution over X. Let $\mathbf{B} = \{\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_n\} \subset \mathbb{R}^n$ consists of n linearly independent vectors. The n-dimensional lattice Λ generated by the basis $\mathbf{B}$ is

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

The length of a matrix $\mathbf{B}$ is defined as the norm of its longest column: $\|\mathbf{B}\| = \max_{1\le i\le n}\|\mathbf{b}_i\|$. Here we view a matrix as simply the set of its column vectors.

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 $D_{L,S}(\mathbf{b}) = \exp(-\pi\|S^{-1}\mathbf{b}\|^2)$ for all $\mathbf{b} \in L$.

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

Lemma 1 (Theorem 3.1, [3]). *There is a probabilistic polynomial-time algorithm that, on input positive integers $n, m, q \ge 2$, 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 $\le 3mq^{n/m}$.*

Lemma 2 (GPV algorithm, [21]). *There exists a probabilistic polynomial-time algorithm that given a basis $\mathbf{B}$ of an n-dimensional lattice $\Lambda = L(\mathbf{B})$, a parameter $s \ge \|\tilde{\mathbf{B}}\| \cdot \omega\left(\sqrt{\log n}\right)$ [1], outputs a sample from a distribution that is statistically close to $D_{\Lambda,s}$.*

Definition 2 ($k-\mathsf{LWE}$ problem, [33]). *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 $A \hookleftarrow U(\mathbb{Z}_q^{m\times n})$, $\mathbf{u} \hookleftarrow U(\mathbb{Z}_q^n)$ and $\mathbf{x}_i \hookleftarrow D_{\Lambda^\perp_{-\mathbf{u}}(A),S}$ for $i \le k \le m$, the goal is to distinguish between the distributions (over $\mathbb{T}^{m+1}$)*

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

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

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

In [33], 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 where $(k, S) - \mathsf{LWE}$ 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.

2.5 Projective Sampling

Inspired by the notion of projective hash family [16], Ling *et al.* [33] proposed a new concept called projective sampling family. A construction of projective sampling family from $k - \mathsf{LWE}$ problem was built as well. The major purpose of their construction is to switch a secret key traitor tracing scheme into a public key one, where tracing signals are sampled from a distribution of spanned spaces by secret keys $\mathbf{x}_j$. In their scheme, each secret key $\mathbf{x}_j \in \mathbb{Z}_q^m$ is associated with a public matrix H_j (projective key). Given the projective keys H_j, any entity in the system can simulate the tracing signal in a computationally indistinguishable way (under the $k - \mathsf{LWE}$ assumption) in the sense that the simulated signal $U(\cap_j \mathrm{Im}(H_j))$ is indistinguishable from original tracing signal $U\left(\mathrm{Span}_j(\mathbf{x}_j^+)^\perp\right)$ even for entities who know the secret keys $\mathbf{x}_j$. This implies that anyone in the system is allowed to execute the tracing procedure.

We recall the construction of H_j [33] as following:

1. Given a matrix $A \in \mathbb{Z}_q^{m\times n}$ and an invertible matrix $A \in \mathbb{Z}_q^{m\times m}$, sampling signals are taken from a spanned space $U\left(\mathrm{Span}_{j\leq k}(\mathbf{x}_j^+)^\perp\right) + \lfloor \nu_{\alpha q}\rceil^{m+1}$, where $\mathbf{x}_j \hookleftarrow D_{\Lambda^\perp_{-\mathbf{u}}(A),S}$. We call vectors $\mathbf{x}_j \in \mathbb{Z}_q^m$ secret keys.
2. Sample $H \hookleftarrow U\left(\mathbb{Z}_q^{m\times(m-n)}\right)$, conditioned on $\mathrm{Im}(H) \subset \mathrm{Im}(A)$. Define the public projected value of $\mathbf{x}_j$ on H as $\mathbf{h}_j = -H^t \cdot \mathbf{x}_j$.
3. Define $H_j = (\mathbf{h}_j^t \parallel H) \in \mathbb{Z}_q^{(m+1)\times(m-n)}$ as the public projected key of $\mathbf{x}_j$.

Simulated signals are now sampled from the distribution $U(\cap_{j\leq k}\mathrm{Im}(H_j)) + \lfloor \nu_{\alpha q}\rceil^{m+1}$. Under the $(k, S) - \mathsf{LWE}$ hardness assumptions, the following two distributions:

$$U\left(\mathrm{Span}_{j\leq k}(\mathbf{x}_j^+)^\perp\right) + \lfloor \nu_{\alpha q}\rceil^{m+1} \text{ and } U\left(\cap_{j\leq k}\mathrm{Im}(H_j)\right) + \lfloor \nu_{\alpha q}\rceil^{m+1}$$

are indistinguishable. This implies that given projected keys H_j, anyone can take samples from the distribution $U\left(\mathrm{Span}_{j\leq k}(\mathbf{x}_j^+)^\perp\right) + \lfloor \nu_{\alpha q}\rceil^{m+1}$ although he does not have the secret keys $\mathbf{x}_j$.

We restate an important result that is frequently used in our proofs. This result comes directly from Theorem 25 and Theorem 27 in [33].

Lemma 3. *We denote by $[t] = \{1, \ldots, t\}$ the set of the t first positive integers. Under the $k - \mathsf{LWE}$ assumption, for $k > t$, given t secret keys $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_t$, for any $j \notin [t]$, the distrisbutions*

$$U\left(\mathrm{Span}_{i\in[t]}(\mathbf{x}_i^+)^\perp\right) + \lfloor \nu_{\alpha q}\rceil^{m+1},\ U\left(\mathrm{Span}_{i\in[t]\cup\{j\}}(\mathbf{x}_i^+)^\perp\right) + \lfloor \nu_{\alpha q}\rceil^{m+1},$$

are indistinguishable (from Theorem 25 in [33]), and the distributions

$$U\left(\cap_{i\in[t]}\mathrm{Im}(H_i)\right) + \lfloor \nu_{\alpha q}\rceil^{m+1},\; U\left(\cap_{i\in[t]\cup\{j\}}\mathrm{Im}(H_i)\right) + \lfloor \nu_{\alpha q}\rceil^{m+1},$$

are indistinguishable as well (from Theorem 27 in [33]).

3 Anonymous Broadcast Encryption for Bounded Universe

We now construct an anonymous broadcast encryption for bounded universe scheme (AnoBEB) from $k-\mathsf{LWE}$ problem. Let N be the maximal number of users (receivers are implicitly represented by integers in $\mathcal{U}=\{1,\dots,N\}$). Given a security parameter n, we assert that parameters q,m,α,S are chosen so that the $(k,S)-\mathsf{LWE}$ problem is hard to solve as presented in [33]. Since the adversary can corrupt any user, we require that $N\le k$ (the system's bounded universe constraint).

$\mathsf{Setup}(1^n,N)$**:** Takes as input the security parameter n and maximal number of users N. It uses Lemma 1 to generate 2 matrices $(A,T)\in\mathbb{Z}_q^{m\times n}\times\mathbb{Z}^{m\times m}$ and picks $\mathbf{u}$ uniformly in $\mathbb{Z}_q^n$. We set a master secret key $\mathsf{MSK}=(A,T)$ and a public key $\mathsf{ek}=\{A^+,(H_j)_{j\le N}\}$, where $A^+=(\mathbf{u}^t\|A)$ and the projected keys H_j (corresponding to the secret keys $\mathbf{x}_j$, defined in Section 2.5) are added each time a secret key $\mathbf{x}_j$ is generated by the $\mathsf{Extract}$. For a system of N users, one can run N times $\mathsf{Extract}$ inside the Setup to generate N secret keys.

$\mathsf{Extract}(\mathsf{ek},\mathsf{MSK},j)$**:** Takes as input the public key ek, the master secret key MSK and a user index $j\in\mathcal{U}$, the algorithm calls the GPV algorithm (Lemma 2) using the basis $\Lambda^\perp(A)$ consisting of the rows of T and the standard deviation matrix S. It obtains a sample $\mathbf{x}_j$ from $D_{\Lambda^\perp_{-\mathbf{u}}(A),S}$. The algorithm outputs decryption key $\mathsf{dk}_j=\mathbf{x}_j^+:=(1\|\mathbf{x}_j)\in\mathbb{Z}^{m+1}$ for user j.

$\mathsf{Encrypt}(\mathsf{ek},M,\mathcal{S})$**:** Takes 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, one chooses a vector $\mathbf{y}\in\mathbb{Z}_q^{m+1}$ from the distribution $U(\cap_{i\in S}\mathrm{Im}(H_i))$, $\mathbf{e}\hookleftarrow\lfloor\nu_{\alpha q}\rceil^{m+1}$ and outputs $\mathbf{c}\in\mathcal{CT}$, which is broadcasted to every member of $\mathcal{S}$ as follows:

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

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

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

Correctness. We require that for a given subset $\mathcal{S}\subseteq\mathcal{U}$ and all $j\in\mathcal{S}$, if $\mathbf{c}=\mathsf{Encrypt}(\mathsf{ek},m,\mathcal{S})$ and dk_j is the decryption key for user $j\in\mathcal{S}$, we then recover $M=\mathsf{Decrypt}(\mathsf{ek},\mathsf{dk}_j,\mathbf{c})$ with overwhelming probability. Indeed, since

$\cap_{i\in S}\mathrm{Im}(H_i) \subseteq \mathrm{Span}_{i\in\mathcal{S}}(\mathbf{x}_i^+)^\perp$, for each user $j \in \mathcal{S}$ and $\mathbf{y} \hookleftarrow U(\cap_{i\in S}\mathrm{Im}(H_i))$, we have $\langle \mathbf{x}_j^+, \mathbf{y}\rangle = 0$. Therefore,

$$\begin{aligned}\langle \mathbf{x}_j^+, \mathbf{c}\rangle &= \langle \mathbf{x}_j^+, \mathbf{y}\rangle + \langle \mathbf{x}_j^+, \mathbf{e}\rangle + \langle \mathbf{x}_j^+, \begin{pmatrix} M\lfloor q/2\rfloor \\ \mathbf{0}\end{pmatrix}\rangle \mod q \\ &= \langle \mathbf{x}_j^+, \mathbf{e}\rangle + M\lfloor q/2\rfloor \mod q,\end{aligned}$$

where $\mathbf{e} \hookleftarrow \lfloor \nu_{\alpha q}\rceil^{m+1}$. According to [33], the quantity $\langle \mathbf{x}_j^+, \mathbf{e}\rangle$ is relatively small modulo q with overwhelming probability. The procedure $\mathsf{Decrypt}$ returns the original message with overwhelming probability. Therefore, every user in $\mathcal{S}$ can decrypt successfully.

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

Theorem 1. *Under the $k-\mathsf{LWE}$ hardness assumption, for any $N \le k$, the AnoBEB scheme Π constructed as above is IND-secure.*

Proof. We consider the sequence of the following games between a challenger $\mathcal{B}$ and an attacker $\mathcal{A}$.

Game G_0**:** This is the real world game, security as defined in the security model. The interaction between the challenger $\mathcal{B}$ and the adversary $\mathcal{A}$ takes place as follows:

Setup. The challenger generates matrix $A \hookleftarrow U(\mathbb{Z}_q^{m\times n})$ and $\mathbf{u} \hookleftarrow U(\mathbb{Z}_q^n)$. The challenger sends public key $\mathsf{ek} = \{A^+, (H_j)_{j\le N}\}$, where each H_j is the projected key associated with a secret key $\mathbf{x}_j$ and $A^+ = (\mathbf{u}^t \| A)$. The public key then sent to $\mathcal{A}$.

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

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 be outside $\mathcal{S}$. $\mathcal{A}$ then sends M_0, M_1 and $\mathcal{S}$ to $\mathcal{B}$. The challenger picks at random a bit $b \hookleftarrow U(\{0,1\})$, outputs a challenge ciphertext (of the message M_b) sampled from one of two following distributions:

$$\begin{aligned}\mathcal{D}_0 &= U\left(\cap_{i\in\mathcal{S}}\mathrm{Im}(H_i)\right) + \lfloor \nu_{\alpha q}\rceil^{m+1} + \begin{pmatrix} M_0\lfloor q/2\rfloor \\ 0\end{pmatrix}, \\ \mathcal{D}_1 &= U\left(\cap_{i\in\mathcal{S}}\mathrm{Im}(H_i)\right) + \lfloor \nu_{\alpha q}\rceil^{m+1} + \begin{pmatrix} M_1\lfloor q/2\rfloor \\ 0\end{pmatrix}.\end{aligned}$$

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

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

Game G_1**:** The challenger now makes one small change to the previous game. Namely, every steps in this game coincides with a corresponding step in the previous one, but the challenge ciphertext sampled from one of two distributions $\mathcal{D}_0^1$ and $\mathcal{D}_1^1$.

$$\mathcal{D}_0^1 = U\left(\cap_{i \in \mathcal{S}\setminus\{j\}} \mathrm{Im}(H_i)\right) + \lfloor \nu_{\alpha q} \rceil^{m+1} + \binom{M_0\lfloor q/2 \rfloor}{0},$$

$$\mathcal{D}_1^1 = U\left(\cap_{i \in \mathcal{S}\setminus\{j\}} \mathrm{Im}(H_i)\right) + \lfloor \nu_{\alpha q} \rceil^{m+1} + \binom{M_1\lfloor q/2 \rfloor}{0},$$

whereas $j \in \mathcal{S}$. Applying Lemma 3, within the view of $\mathcal{A}$, there are two pairs of distributions

$$\mathcal{D}_0 = U\left(\cap_{i \in \mathcal{S}} \mathrm{Im}(H_i)\right) + \lfloor \nu_{\alpha q} \rceil^{m+1} + \binom{M_0\lfloor q/2 \rfloor}{0},$$

$$\mathcal{D}_0^1 = U\left(\cap_{i \in \mathcal{S}\setminus\{j\}} \mathrm{Im}(H_i)\right) + \lfloor \nu_{\alpha q} \rceil^{m+1} + \binom{M_0\lfloor q/2 \rfloor}{0}$$

and

$$\mathcal{D}_1 = U\left(\cap_{i \in \mathcal{S}} \mathrm{Im}(H_i)\right) + \lfloor \nu_{\alpha q} \rceil^{m+1} + \binom{M_1\lfloor q/2 \rfloor}{0},$$

$$\mathcal{D}_1^1 = U\left(\cap_{i \in \mathcal{S}\setminus\{j\}} \mathrm{Im}(H_i)\right) + \lfloor \nu_{\alpha q} \rceil^{m+1} + \binom{M_1\lfloor q/2 \rfloor}{0}$$

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

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

Game G_ℓ**:** The challenger also makes one small change to the previous games, while every step in this game coincides with the previous one, but for the challenge ciphertext sampled from one of two distributions $\mathcal{D}_0^\ell$ and $\mathcal{D}_1^\ell$, as follows:

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

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

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

To summarize, we have a sequence of games where the final game **Game** G_ℓ has zero-advantage and the difference of each two successive games **Game** $\mathrm{G}_{\mathrm{i}-1}$, **Game** G_i, for all $2 \leq i \leq \ell$, is negligible, and ℓ is polynomial. Therefore, the scheme Π is $\mathsf{IND}-$secure.

We next consider anonymity of the AnoBEB scheme (our main Theorem for this section):

Theorem 2. *Under the* $k-\mathsf{LWE}$ *hardness, for any* $N \leq k$*, our scheme is* ANO*-secure.*

Proof. Intuitively, 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 [18], which only corrupts users outside both $\mathcal{S}_0, \mathcal{S}_1$, then the proof is direct because from the $k-\mathsf{LWE}$ assumption, the encryption for $\mathcal{S}_0$ and for $\mathcal{S}_1$, both, look like random ciphertexts to the adversary. It is more challenging to consider a general adversary which can also corrupt the key in the intersection of $\mathcal{S}_0$ and $\mathcal{S}_1$. Fortunately, by applying Lemma 3 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 $\mathcal{S}$. We then apply a hybrid argument which moves 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 users in $\mathcal{S}_1 \setminus \mathcal{S}_0$ (or in $\mathcal{S}_1 \setminus \mathcal{S}_0$, respectively).

We will prove the above by considering a sequence of games, as following:

Game G_0**:** This is the real world game, security defined in the security model. We repeat the interaction between the challenger $\mathcal{B}$ and the adversary $\mathcal{A}$ as following:

Setup. The challenger generates a matrix $A \hookleftarrow U(\mathbb{Z}_q^{m\times n})$ and picks $\mathbf{u}$ uniformly in $\mathbb{Z}_q^n$. Then the public key is set to $\mathsf{ek} = \{A^+, (H_j)_{j\leq k}\}$, with $A^+ = (\mathbf{u}^t \| A)$, and given to $\mathcal{A}$.

Phase 1. When $\mathcal{A}$ asks for the decryption key for user i, $\mathcal{B}$ replies with $\mathbf{x}_i^+ = (1||\mathbf{x}_i)$, where $\mathbf{x}_i \hookleftarrow D_{\Lambda^{\perp}_{-\mathbf{u}}(A),S}$.

Challenger phase. $\mathcal{A}$ chooses a message M, two subsets $\mathcal{S}_0, \mathcal{S}_1$ with the restriction that no asked query is in $\mathcal{U} \setminus (\mathcal{S}_0 \bigtriangleup \mathcal{S}_1)$ and sends it to $\mathcal{B}$. The challenger picks randomly $b \in \{0,1\}$ and gives $\mathcal{A}$ a ciphertext $\mathbf{c}$ taken from one of two distributions (distribution $\mathcal{D}_b$, over $\mathbb{T}^{m+1}$): $\mathcal{D}_0 = U\left(\cap_{i\in\mathcal{S}_0}\mathrm{Im}(H_i)\right) + \lfloor \nu_{\alpha q}\rceil^{m+1} + \begin{pmatrix} M\lfloor q/2 \rfloor \\ \mathbf{0}\end{pmatrix}$, $\mathcal{D}_1 = U\left(\cap_{i\in\mathcal{S}_1}\mathrm{Im}(H_i)\right) + \lfloor \nu_{\alpha q}\rceil^{m+1} + \begin{pmatrix} M\lfloor q/2 \rfloor \\ \mathbf{0}\end{pmatrix}$.

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

Guess. $\mathcal{A}$ guesses b' for b.

Game G_1**:** In this game, the inputs and the settings of this game are identical to the ones of **Game** G_0. In the challenger phase, the adversary $\mathcal{A}$ received a ciphertext from one of the two following distributions: $\mathcal{D}_0 = U\left(\cap_{i\in\mathcal{S}_0}\mathrm{Im}(H_i)\right) + \lfloor \nu_{\alpha q}\rceil^{m+1} + \begin{pmatrix} M\lfloor q/2 \rfloor \\ \mathbf{0}\end{pmatrix}$, or $\mathcal{D}_1^1 = U\left(\cap_{i\in\mathcal{S}_1\cup\{j_1\}}\mathrm{Im}(H_i)\right) + \lfloor \nu_{\alpha q}\rceil^{m+1} + \begin{pmatrix} M\lfloor q/2 \rfloor \\ \mathbf{0}\end{pmatrix}$, where the projected key H_{j_1} corresponds to the secret key $\mathbf{x}_{j_1}^+ \hookleftarrow D_{\Lambda^{\perp}_{-\mathbf{u}}(A),S}$, $j_1 \in \mathcal{S}_0 \setminus \mathcal{S}_1$.

Here we notice that the adversary $\mathcal{A}$ does not know the key $\mathbf{x}_{j_1}^+$ because $\mathcal{A}$ can only choose the keys with index in $\mathcal{U}\setminus(\mathcal{S}_0 \triangle \mathcal{S}_1)$. Since $k-\mathsf{LWE}$ is hard, by applying Lemma 3, the two distributions $\mathcal{D}_1 = U\left(\cap_{i\in\mathcal{S}_1}\mathrm{Im}(H_i)\right) + \lfloor \nu_{\alpha q}\rceil^{m+1} + \begin{pmatrix} M\lfloor q/2 \rfloor \\ \mathbf{0}\end{pmatrix}$, $\mathcal{D}_1^1 = U\left(\cap_{i\in\mathcal{S}_1\cup\{j_1\}}\mathrm{Im}(H_i)\right) + \lfloor \nu_{\alpha q}\rceil^{m+1} + \begin{pmatrix} M\lfloor q/2 \rfloor \\ \mathbf{0}\end{pmatrix}$ are indistinguishable. This means that the difference between the advantage of $\mathcal{A}$ in **Game** G_1 and **Game** G_0 is negligible.

Game G_τ**:** We assume that $\kappa = |\mathcal{S}_0 \setminus \mathcal{S}_1|$ and $\mathcal{S}_0 \setminus \mathcal{S}_1 = \{j_1, j_2, \ldots, j_\kappa\}$. For each $2 \leq \tau \leq \kappa$, we consider a game in 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 that $\mathcal{T}_\kappa = \mathcal{S}_0 \cup \mathcal{S}_1$. In each game in this sequence, the inputs and the settings are identical to the ones of previous games. In the challenger phase, the adversary $\mathcal{A}$ receives a ciphertext from one of the two following distributions: $\mathcal{D}_0 = U\left(\cap_{i \in \mathcal{S}_0} \mathrm{Im}(H_i)\right) + \lfloor \nu_{\alpha q} \rceil^{m+1} + \binom{M\lfloor q/2 \rfloor}{\mathbf{0}}$ and $\mathcal{D}_1^\tau = U\left(\cap_{i \in \mathcal{T}_\tau} \mathrm{Im}(H_i)\right) + \lfloor \nu_{\alpha q} \rceil^{m+1} + \binom{M\lfloor q/2 \rfloor}{\mathbf{0}}$.

Since adversary $\mathcal{A}$ does not know any key $\mathbf{x}_{j_\tau}^+$ in the set $\mathcal{S}_0 \setminus \mathcal{S}_1$ and the $k - \mathsf{LWE}$ problem is hard, we apply Lemma 3, the two distributions: $\mathcal{D}_1^{\tau-1} = U\left(\cap_{i \in \mathcal{T}_{\tau-1}} \mathrm{Im}(H_i)\right) + \lfloor \nu_{\alpha q} \rceil^{m+1} + \binom{M\lfloor q/2 \rfloor}{\mathbf{0}}$ and $\mathcal{D}_1^\tau = U\left(\cap_{i \in \mathcal{T}_\tau} \mathrm{Im}(H_i)\right) + \lfloor \nu_{\alpha q} \rceil^{m+1} + \binom{M\lfloor q/2 \rfloor}{\mathbf{0}}$, are indistinguishable for each τ. This means that the difference between the advantage of $\mathcal{A}$ in any transition in the sequence of games **Game** G_τ, $1 \leq \tau \leq \kappa$ is negligible.

Game $\mathrm{G}_{\kappa+\eta}$**:** We assume that $\iota = |\mathcal{S}_1 \setminus \mathcal{S}_0|$ and $\mathcal{S}_1 \setminus \mathcal{S}_0 = \{j_1, j_2, \ldots, j_\iota\}$. For each $1 \leq \eta \leq \iota$, we consider a game in 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 that $\mathcal{T'}_\iota = \mathcal{S}_0 \cup \mathcal{S}_1$. In each game in this sequence, the inputs and the settings are identical to the ones of previous games. In challenger phase, the adversary $\mathcal{A}$ receives a ciphertext from one of following two distributions: $\mathcal{D}_0^\eta = U\left(\cap_{i \in \mathcal{T'}_\eta} \mathrm{Im}(H_i)\right) + \lfloor \nu_{\alpha q} \rceil^{m+1} + \binom{M\lfloor q/2 \rfloor}{\mathbf{0}}$, and $\mathcal{D}_1^\kappa = U\left(\cap_{i \in (\mathcal{S}_0 \cup \mathcal{S}_1)} \mathrm{Im}(H_i)\right) + \lfloor \nu_{\alpha q} \rceil^{m+1} + \binom{M\lfloor q/2 \rfloor}{\mathbf{0}}$. It means that we keep fix the distribution $\mathcal{D}_1^\kappa$ and replace the distribution $\mathcal{D}_0$ by $\mathcal{D}_0^\eta = U\left(\cap_{i \in \mathcal{T'}_\eta} \mathrm{Im}(H_i)\right) + \lfloor \nu_{\alpha q} \rceil^{m+1} + \binom{M\lfloor q/2 \rfloor}{\mathbf{0}}$, where we set $\mathcal{D}_0^\eta = \mathcal{D}_0$ in case $\eta = 0$. 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 the two following distributions $\mathcal{D}_0^{\eta-1} = U\left(\cap_{i \in \mathcal{T'}_{\eta-1}} \mathrm{Im}(H_i)\right) + \lfloor \nu_{\alpha q} \rceil^{m+1} + \binom{M\lfloor q/2 \rfloor}{\mathbf{0}}$, and $\mathcal{D}_0^\eta = U\left(\cap_{i \in \mathcal{T'}_\eta} \mathrm{Im}(H_i)\right) + \lfloor \nu_{\alpha q} \rceil^{m+1} + \binom{M\lfloor q/2 \rfloor}{\mathbf{0}}$ are indistinguishable for each $1 \leq \eta \leq \iota$. Therefore the difference between the advantage of $\mathcal{A}$ in the transitions of the sequence of games **Game** $\mathrm{G}_{\eta+\kappa}$, $1 \leq \eta \leq \iota$ is negligible. We recall that in the last game ($\eta = \iota$), $\mathcal{A}$ will receive a challenger ciphertext taken from $U\left(\cap_{i \in (\mathcal{S}_0 \cup \mathcal{S}_1)} \mathrm{Im}(H_i)\right) + \lfloor \nu_{\alpha q} \rceil^{m+1} + \binom{M\lfloor q/2 \rfloor}{\mathbf{0}}$, or $U\left(\cap_{i \in (\mathcal{S}_0 \cup \mathcal{S}_1)} \mathrm{Im}(H_i)\right) + \lfloor \nu_{\alpha q} \rceil^{m+1} + \binom{M\lfloor q/2 \rfloor}{\mathbf{0}}$. Obviously, the advantage of adversary $\mathcal{A}$ in this game is equal to zero since these distributions are identical.

We conclude (as all sequences are polynomial size) that our scheme AnoBEB is ANO-secure under the hardness of $k-\mathsf{LWE}$ problem.

Concerning efficiency, our scheme AnoBEB is exactly as efficient as the Ling *et al.*'s traitor tracing scheme in [33] which was shown in [33] to be as efficient as the standard LWE encryption.

Finally, we also note that, as shown in [33], example parameters are $k = m/10$, $\sigma = \widetilde{\Theta}(n)$, $q = \widetilde{\Theta}(n^5)$ and $m = \Theta(n \log n)$. We can therefore set our parameters to: $N = k$ and the efficiency of the AnoBEB scheme is approximately as efficient as the underlying LWE-PKE, inherently from the fact the LPSS k-LWE traitor tracing has approximately the same efficiency as the underlying LWE-PKE, as shown in [33].

4 Trace and Revoke System from AnoBEB and Robust IPP Codes

Our goal now is to construct a Trace and Revoke (TR) scheme from AnoBEB. The formal definition of a TR scheme is provided in Section 2.2. In our approach, we combine a robust t-IPP code with an $n-$bit AnoBEB scheme. We also give explicit instantiations for robust t-IPP code at the end of this section. We will start this section by presenting the formal definition of robust IPP code [4].

Robust IPP *codes.* Let $\mathcal{C} = \{w_1, \ldots, w_N\} \subset \Sigma^\ell$ be a $q-$ary code of size N and length ℓ, 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 $w_i = (w_{i,1}, \ldots, w_{i,\ell})$. 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. 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 $D(X)$. The set of descendants of X, denoted $\mathsf{desc}(X)$, is defined by

$$\mathsf{desc}(X) = \Big\{x = (x_1, \ldots, x_\ell) \in \Sigma^\ell \mid x_j \in X_j, 1 \leq j \leq \ell\Big\}.$$

We call codewords in the coalition X are parents of the set $\mathsf{desc}(X)$. Define a $t-$descendant of the code $\mathcal{C}$, denoted $\mathsf{desc}_t(\mathcal{C})$, $\mathsf{desc}_t(\mathcal{C}) = \bigcup_{X \subset \mathcal{C}, |X| \leq t} \mathsf{desc}(X)$. 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 next.

Definition 3. *Given a code $\mathcal{C} = (\ell, N, q)$, let $t \geq 2$ be an integer. The code $\mathcal{C}$ is called a t-IPP code if for all $x \in \mathsf{desc}_t(\mathcal{C})$, it holds that* $\bigcap_{x \in \mathsf{desc}(X), X \subset \mathcal{C}, |X| \leq t} X \neq \emptyset$.

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

In [10], Boneh and Shaw considered a more general coalition, called wide-sense envelope of the coalition X. The set of descendants in their fingerprinting code is defined as follows:

$$\Big\{x = (x_1, \ldots, x_\ell) \in (\Sigma \cup \{*\})^\ell \;\mid\; \text{if } j \notin D(X) \text{ then } x_j \in X_j\Big\},$$

where $D(X)$ consists of detectable coordinates of the coalition X. 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 (*marking assumption*). The notion Robust IPP code is a concept that allows a limited number of coordinates to not follow their parents. These coordinates are 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 there is a descendant x in the set $\mathsf{desc}(X)$, following the marking assumption rule except εn coordinates that can deviate from this rule. Call a coordinate i of $x \in \mathsf{desc}(X)$ 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 one replaced by an 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 4. *Code $\mathcal{C} \subset \Sigma^\ell$ is a (t, ε)-IPP code (robust t-IPP code) if for all $x \in \mathsf{desc}(X)_\varepsilon$, where $X \subset \mathcal{C}$ and $|X| \leq t$, it holds that* $\displaystyle\bigcap_{X \subset \mathcal{C}, |X| \leq t, x \in \mathsf{desc}(X)_\varepsilon} X \neq \emptyset.$

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 $\varepsilon = 0$, a robust IPP becomes an IPP code.

A robust IPP code is said to have the traceability property if for any $x \in \mathsf{desc}_\varepsilon(X)$, the codeword $c \in \mathcal{C}$ closest to x by the Hamming distance is always one of the parents of x, i.e., $c \in \displaystyle\bigcap_{X \subset \mathcal{C}, |X| \leq t, x \in \mathsf{desc}(X)_\varepsilon} X$. This implies that a pirate can be provably identified by finding any vector $c \in \mathcal{C}$ such that the distance from c to x is the shortest. A robust IPP code with traceability property is called robust TA code. We shall use robust IPP with traceability property.

Secret Sharing. We also choose a $(\rho\ell, \ell)$-secret sharing scheme, where $\rho = 1 - \varepsilon$ so that any non-revoked users can decrypt. The formal definition of secret sharing scheme, which consists of two algorithms Share and $\mathsf{Combine}$, is given in Appendix A.1. Let r be maximum number of revoked users. For the correctness, given the parameter r on the bound of number of revoked users, we require that the distance Δ is set to verify the condition:

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

Construction of a TR *scheme.* 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 to be a sequence of ℓ subsets of Σ, i.e. $S_i \subseteq \Sigma$. Given a vector $\omega = (\omega_1, \ldots, \omega_\ell) \in \Sigma^\ell$, the *agreement* between ω and a mixture S is defined to be the number of positions $i \in [\ell]$ for which $\omega_i \in S_i$: $\text{AGR}(\omega, S) = \sum_{i=1}^{\ell} \mathbf{1}_{\omega_i \in S_i}$, where $\mathbf{1}_{\omega_i \in S_i} = 1$ if $\omega_i \in S_i$ and $\mathbf{1}_{\omega_i \in S_i} = 0$ if otherwise.

We will construct a TR 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 coordinate of the codeword $w_i \in \mathcal{C}$. By assigning each user i in Γ to a set with ℓ sub-keys, the decryption key for the user i has form $\mathsf{dk}_i = (\mathsf{sk}_{1,w_{i,1}}, \ldots, \mathsf{sk}_{j,w_{i,j}}, \ldots, \mathsf{sk}_{\ell,w_{i,\ell}})$, where each sub-key generated by the $\mathsf{Extract}$ algorithm of AnoBEB.

We consider an arbitrary group of decryption keys. At any coordinate component of the group, there are at most q sub-keys. We have a one-to-one correspondence between the set of q sub-keys and the set of decryption keys of q users in AnoBEB system. Consequently, to broadcast a message K (will be splitted into ℓ shares $K_1, \ldots, K_\ell$) to the set of N users, we apply the $\mathsf{Share}(K, \rho\ell, \ell)$ of $(\rho\ell, \ell)-$secret sharing scheme and we encrypt each j^{th}-share K_j with AnoBEB. Note that the message K is then often used as a session key to encrypt the data via a data encapsulation mechanism.

Formally, to build a TR system for N users, we concatenate ℓ instantiations of the scheme AnoBEB (for q users) according to an $q-$ary code $\mathcal{C}$. In particular, we will combine AnoBEB with robust IPP code $\mathcal{C}$. Our construction consists of 5 algorithms: Setup, $\mathsf{Extract}$, $\mathsf{Encrypt}$, $\mathsf{Decrypt}$ and $\mathsf{Tracing}$.

$\mathsf{Setup}(1^n, t, r)$**:** Takes as input the security parameter n, a maximum malicious coalition size t and the bound r on the number of revoked users. Let $\mathcal{C}$ be a t-IPP robust code size N over alphabet $\Sigma = [q]$. By calling ℓ times the procedure $\mathsf{AnoBEB.Setup}(1^n, q)$, where ℓ is the length of the code $\mathcal{C}$, we obtain public keys ek_j and master secret keys MSK_j, $j = 1, \ldots, \ell$. We set $\mathsf{ek} = (\mathsf{ek}_1, \ldots, \mathsf{ek}_\ell)$ and $\mathsf{MSK} = (\mathsf{MSK}_1, \ldots, \mathsf{MSK}_\ell)$.

$\mathsf{Extract}(\mathsf{ek}, \mathsf{MSK}, i)$**:** Takes as index $i \in [N]$ for each user, we use 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 $w_{i,j}$ is the value at position j of codeword w_i. Here,

$$\mathsf{sk}_{j,w_{i,j}} = \mathsf{AnoBEB.Extract}(\mathsf{ek}_j, \mathsf{MSK}_j, w_{i,j}), j \in [\ell].$$

$\mathsf{Encrypt}(\mathsf{ek}, K, \mathcal{R})$**:** Takes as input a set of revoked users $\mathcal{R} \subset \mathcal{C}$, where the cardinality of $\mathcal{R}$ is at most r. The message $K \in \mathcal{PT}$, where $\mathcal{PT}$ is the plaintext domain, will be broadcasted to the target set $\mathcal{C} \setminus \mathcal{R}$. We call the procedure $\mathsf{Share}(K, \rho\ell, \ell)$ of $(\rho\ell, \ell)$-secret sharing scheme and obtain ℓ shares $K_1, \ldots, K_\ell$ in which at least $\rho\ell$ of the shares are needed to recover the message K. We consider the following mixture $\mathcal{M} = (\mathcal{M}_1, \ldots, \mathcal{M}_\ell) = (\Sigma \setminus \mathcal{R}[1], \ldots, \Sigma \setminus \mathcal{R}[\ell])$, where $\mathcal{R}[j] = \cup_{i \in \mathcal{R}} w_{i,j}$. Set $c_i = \Big(\mathsf{AnoBEB.Encrypt}(\mathsf{ek}_i, K_i, \mathcal{M}_i)\Big)$ for each $i = 1, \ldots, \ell$. The ciphertext is $\mathbf{c} = (c_1, \ldots, c_\ell) \in \mathcal{CT}^\ell$, where $\mathcal{CT}$ is the ciphertext domain of AnoBEB.

$\mathsf{Decrypt}(\mathsf{ek}, \mathsf{dk}_i, \mathbf{c})$**:** Takes as input ciphertext $\mathbf{c} \in \mathcal{CT}^\ell$ and a decryption key dk_i of user i. The user i calls the decryption function $\mathsf{AnoBEB.Decrypt}(\mathsf{ek}_j, \mathsf{sk}_{j,w_{i,j}}, c_j)$ of the AnoBEB scheme on sub-keys $\mathsf{sk}_{j,w_{i,j}}$ for each $j = 1, \ldots, \ell$. If $i \in \mathcal{R}$ then i cannot decrypt any c_i and cannot recover K (will be proved in the part of semantic security of Theorem 3). Otherwise, $i \notin \mathcal{R}$, the user obtains at least $\rho\ell$ values among the shared values K_j (as will be proved in the correctness). By calling the function $\mathsf{Combine}$ of the secret sharing scheme over pairs $\{(j, K_j)\}$, the user recovers the original message K.

$\mathsf{Tracing}(\mathcal{D}, \mathcal{R}, \mathsf{ek})$**:** Takes as input a set $\mathcal{R}$ of $\leq r$ revoked users, a public key ek and has access to a pirate distinguisher $\mathcal{D}$. We consider the mixture $\mathcal{M}$ as in $\mathsf{Encrypt}$ procedure. Let $\mathcal{T}$ be the subset of $\mathcal{U} \setminus \mathcal{R}$ with at most t elements (traitors). The pirate distinguisher outputs two messages K^0 and K^1 and then sends to the Tracer. We assume that the pirate distinguisher is an ϵ-useful in the sense that it can distinguish, with a non-negligible probability ϵ, ciphertexts in the form $\mathbf{c} = (c_1, \ldots, c_\ell)$, where $c_i = \Big(\mathsf{AnoBEB.Encrypt}(\mathsf{ek}_i, K_i^b, \mathcal{M}_i)\Big)$ for each K_i^b is i-th component of the message K^b, $b \leftarrow \{0, 1\}$. We denote here $\mathcal{M}_j = \{j_\iota\}_{\iota \in Q}$, $Q \subseteq [q]$ or $\mathcal{M}_j = \emptyset$ for all $j = 1, \ldots, \ell$. We consider the tracing procedure as follows:

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. $c_j = \mathsf{AnoBEB.Encrypt}(\mathsf{ek}_j, K_j^b, \mathcal{M}_j)$.
      ii. Call the pirate distinguisher $\mathcal{D}$ on input $\mathbf{c} = (c_1, \ldots, c_j, \ldots, c_\ell)$. If $\mathcal{D}(\mathbf{c}) = b$ then $\mathsf{cnt} \leftarrow \mathsf{cnt} + 1$.
   (c) Let $\tilde{p}_{j,j_\iota}$ be the fraction of times that $\mathcal{D}$ outputs b correctly. We have $\tilde{p}_{j,j_\iota} = \mathsf{cnt}/W$.
   (d) $\mathcal{M}_j = \mathcal{M}_j \setminus \{j_\iota\}$.
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) $c_j = \mathsf{AnoBEB.Encrypt}(\mathsf{ek}_j, K_j^b, \mathcal{M}_j)$
   else $c_j = \mathsf{random}$ and $\omega_j = *$.

End for.

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

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

We prove that the tracing algorithm returns at least $\rho\ell$ keys. Indeed, if the output of the algorithm provides $t < \rho\ell$ keys then the ciphertext in the final iteration step ℓ will appear as t normal components and the pirate device will be able to still correctly decrypt the ciphertext. This is a contradiction because in the setting of our system, by using $(\rho\ell, \ell)-$secret sharing scheme, it is impossible for any decoder device to successfully decrypt the ciphertext with less than $\rho\ell$ normal components. Therefore, our tracing algorithm will output at least $\rho\ell$ pirate keys. We thus get at the end of Step 2 a pirate word with $\rho\ell$ components without $*$. Since the scheme Γ employs robust IPP code $\mathcal{C}$, the tracer uses the property of robust IPP for the pirate word which was found from the black-box tracing to identify at least one user who contributed to build the pirate device.

Since the tracing procedure uses the tracing procedure in robust IPP codes, which does not require any secret information (like IPP codes), and we only use the procedure AnoBEB.Encrypt to produce the tracing signals, the combined scheme Γ supports public traceability.

Correctness and Security of the TR system.

Theorem 3. *Given*

- $\mathcal{C} = (\ell, N, q)$, *a robust t-IPP code of Hamming distance Δ and $0 < \varepsilon < (t+1)^{-1}$;*
- *a $(\rho\ell, \ell)-$secret sharing scheme, where $\rho = 1 - \varepsilon$;*
- *an anonymous broadcast encryption for q users AnoBEB;*

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 successfully at least one traitor from any coalition of up to t traitors. Moreover, assume that the scheme AnoBEB is IND-secure, then the scheme Γ is also an IND-secure scheme.

Proof. The proof of the semantic security can be found in Appendix A.2. We focus onf the revocation and the traceability.

Given a ciphertext $\mathbf{c}$, any users $i \in [N] \setminus \mathcal{R}$ can decrypt it successfully. Indeed, since $\mathcal{C}$ is the code having the minimum Hamming distance that satisfies inequality (2) above, it implies that (1) is true. Therefore, for any user i in $[N] \setminus \mathcal{R}$, we have $\text{AGR}(w_i, \mathcal{M}) \geq \ell - r(\ell - \Delta) \geq \rho\ell$. This implies that the user i has at least $\rho\ell$ sub-keys that agree with the mixture $\mathcal{M}$ and recovers at least $\rho\ell$ sub-messages K_i. By calling the function Decrypt on $\mathbf{c}$, the user i will recover the underlying original message. In contrast, any revoked user in $\mathcal{R}$ gets no any sub-key and thus cannot decrypt the ciphhertext $\mathbf{c}$.

Concerning traceability, at the end of the black-box tracing procedure (after finishing the procedure Tracing), we get a pirate word. With a given pirate word, to ensure that the identify algorithm can return efficiently at least a traitor from any $t-$collusion as explained in the tracing algorithm, it remains to ensure that the robust IPP codes have the traceability property.

According to Proposition 3.1 in [4], in order to achieve the traceability in robust IPP codes, the minimum Hamming distance of the code must satisfy: $\Delta/\ell > 1 - \left(\frac{1-\varepsilon}{t^2} - \frac{\varepsilon}{t}\right)$, whereas $0 < \varepsilon < (t+1)^{-1}$. However, we can not directly use thus Proposition 3.1 in [4] because of two reasons:

- In [4], a proof of existence of robust IPP codes was given, but there was no immediate explicit construction given there, neither was any analysis of the length of such a code presented.
- Robust IPP codes only deal with the number of traitor. In our scheme, we need, moreover, to take into account of the number of the revoked users. The condition (2) captures both the condition on the revocation and traceability, so that in total we have an extended code requirement to consider (namely, robust IPP code supporting revocations).

Two explicit constructions of Robust IPP *codes.* We now present two explicite instantiations of robust IPP codes verifying the condition (2).

Construction 1. The relative distance of the code $\mathcal{C}$ is defined by $\delta := \Delta/\ell$. We will consider a code with δ satisfying the Gilbert-Varshamov bound. 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 the Gilbert-Varshamov theorem (Theorem 4.10, [39]), 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 the derandomization procedure of Porat-Rothschild [38]. This means that we give an explicit construction for the code $\mathcal{C}$. It progresses as following:
We choose $\delta = 1 - \frac{1}{d+1}$. Obviously, we do not want large δ because that can only increase 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 $1 - H_q(\delta)$. Below, we will use the fact that $\log(1+x) \approx x$ for small x extensively.

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

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 short, we obtain $q = \Theta(d)$ and $\ell = O(d \log N)$.

Construction 2. The above construction 1 is interesting in the theoretical point of view as the code length is optimal. However, due to the derandomization that makes the construction explicite, the decoding algorithm is in exponential time in the dimension of the code, as it relies on the decoding of Porat-Rothschild code. In this second construction, we rely on the Reed-Solomon code, and thus obtain a polynomial-time decoding. The efficiency is not as optimal as the construction 1 but the cost is only $\log N$.

We also pick $d = \max\left\{\dfrac{r}{1-\rho}, \dfrac{t^2}{(1-\epsilon)-\epsilon t}\right\}$. The Reed-Solomon code has $\delta = \frac{\ell-k+1}{\ell} = 1 - \frac{k}{\ell} + \frac{1}{\ell}$, whereas k is the 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 $\ell = O\left(\frac{2d\log N}{\log(d\log N)}\right)$.

Ciphertext Size of the TR System. We now consider the ciphertext size of scheme Γ, which is the size of an AnoBEB ciphertext times the length of the Robust IPP code. By relying on the Construction 2 of the IPP robust code, our trace and revoke achieves the ciphertext size complexity of $\widetilde{O}((r+t^2)(n^2)\log N)$ which is the code length multiplied by the LWE ciphertext size. This is an LWE-based scheme and thus a bit-encryption, as in [33].

From bit encryption to multi-bit encryption. As we want to encrypt an n-bit size session key, we need to repeat our scheme n times and therefore, the ciphertext size becomes $\widetilde{O}((r+t^2)(n^3)\log N)$, which is still the most efficient trace and revoke scheme for standard black-box tracing in the bounded collusion model.

Efficiency Comparison with other TR Systems in Bounded Collusion Model. For bounded schemes where the number of traitors is small, the Agrawal *et al.*'s scheme [1], relying on learning with errors, is very efficient with ciphertext size $\widetilde{O}(r+t+n)$ where r is the maximum number of revoked users, t the maximum number of traitors, and n the security parameter. But they only support a weak level of tracing: black-box confirmation with the assumption that the tracer gets a suspect set that contains all the traitors. Converting black-box confirmation into black-box tracing requires an exponential time complexityin the number of traitors. Concerning black-box trace and revoke in bounded collusion model, up to now, the instantiation of the NWZ scheme gives the most efficient construction. However, as stated in [1], the generic nature of their construction results in loss of concrete efficiency: when based on the bounded collusion FE of [23], the resulting scheme has a ciphertext size growing at least as $\widetilde{O}((r+t)^5\mathcal{P}oly(n))$; by relying on learning with errors, this blowup can be improved to $\widetilde{O}((r+t)^4\mathcal{P}oly(n))$, but at the cost of relying on heavy machinery such as attribute based encryption [24] and fully homomorphic encryption [22]. Our trace and revoke result, in contrast, achieves ciphertext size $\widetilde{O}((r+t^2)(n^3)\log N)$ with black-box tracing like in [35], which is the prevalent standard model for tracing and is by far more realistic than the black-box confirmation as in [1]. The following Table 2 resumes the comparison between Trace and Revoke schemes in bounded collusion model.

Trace & Revoke Schemes	Ciphertext Size	Type of Tracing Algorithm	Type of Pirate
ABPSY [1]	$\tilde{O}(r+t+n)$	Black-box confirmation	Decoder
NWZ [35]	$\tilde{O}\left((r+t)^4\mathsf{Poly}(N)\right)$	Black-box tracing	Distinguisher
Ours	$\tilde{O}\left((r+t^2)n^3\log N\right)$	Black-box tracing	Distinguisher

Table 2: Comparison between Trace and Revoke schemes in bounded collusion model. n is the security parameter, N is the total number of recipients and r, t are respectively the bounds on the number of revoked users and traitors

5 Discussion and Conclusion

Let us discuss a few points of interest. We first compare our scheme with IPP and collusion secure code-based schemes:

- We note that all known code-based schemes, *e.g,* [7, 9, 10, 13, 14, 17, 20, 28, 30, 36, 37, 40, 42], relying on collusion-secure code or on IPP code, only support traitor tracing while we target the more challenging case of trace and revoke construction.
- The length of our proposed robust IPP codes is approximately $O((r+t^2)\log N)$. It is essentially the length of the best collusion secure code, namely the Tardos code [43] which is $O(t^2(\log \frac{N}{\theta}))$, where θ is the error probability in identifying traitors (we note that an interesting property in IPP and robust IPP codes is that one achieves zero error in identifying traitors and that collusion secure code does not support revocation). As far as one can construct an AnoBEB which is as efficient as the underlying PKE (which is the case for LWE encryption as we achieve in this work), then one gets a robust IPP code based trace and revoke scheme from our method which has the same ciphertext size as the collusion secure code based traitor tracing schemes. Concerning IPP code schemes, by using our LWE-based AnoBEB, we save a factor q in ciphertext efficiency in comparing to the IPP code traitor tracing in [36] when instantiating the PKE with LWE encryption.

- Boneh-Naor [9] and Billet-Phan [7] provided solutions to tracing traitors from imperfect pirate device, with short ciphertext size. Their schemes were built from robust collusion secure codes and PKE. The main idea is to randomly choose a position in the code and then encrypt the session key twice with the two keys at the chosen position. We can completely follow these methods to obtain a traitor tracing scheme from a robust IPP code and an AnoBEB with short ciphertext size. In particular, when instantiating PKE with LWE, their schemes of double size of the standard LWE encryption while we can get a scheme with the same size of the standard LWE encryption, thus saving a factor 2 in efficiency. Note also that, unlike our case, collusion secure code based scheme do not support public tracing as all known methods for tracing in collusion secure code require the knowledge of the secret information.

A few open questions remain:

– In trace and revoke systems, there are two main approaches to tackle the problem:
  - restrict to bounded collusion model (motivated by the fact that this is a practical scenario) and give efficient solutions;
  - consider the full collusion setting (all users can become traitors) and improve theoretical results as there are actually no efficient scheme, say, of ciphertext size which depends on $polylog(N)$, from the standard assumptions without relying on general iO or multi-linear maps [12] or positional witness encryption [27] (for which there are currently no algebraic implementations that are widely accepted as secure).

  Recently, at STOC '18, Goyal, Koppula and Waters [25], relying on Mixed Functional Encryption with Attribute-Based Encryption, gave a traitor tracing scheme for full collusion from the LWE assumption with $polylog(N)$ ciphertext size. This avoids the use of iO or multi-linear maps in Boneh-Zhandry scheme from CRYPTO '14 [12]. However, this scheme support traitor tracing only. It is an interesting open question to construct a polylog size trace and revoke scheme for full collusion from a standard assumption, since combining tracing and revoking functionalities is always a difficult problem.

– In this paper we provided an LWE-based construction of AnoBEB which is as efficient as the underlying LWE PKE. We raise an open question of constructing AnoBEB schemes from other standard and well established encryptions, namely ElGamal, RSA, Paillier encryptions, without a significant loss in efficiency. This seems to us to suggest an interesting and a challenging problem, even for the simplest case of a system of $N = 2$ users. The solution will directly give the most efficient trace and revoke systems for bounded collusion model (by instantiating our trace and revoke scheme of Section 4) from DDH, RSA and DCR assumptions, respectively.

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 Bhavani M. Thuraisingham, David Evans, Tal Malkin, and Dongyan Xu, editors, *ACM CCS 17*, pages 2277–2293. ACM Press, October / November 2017.
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`.
3. J. Alwen and C. Peikert. Generating shorter bases for hard random lattices. *Theor. Comput. Science*, 48(3):535–553, 2011.
4. Alexander Barg and Grigory Kabatiansky. Robust parent-identifying codes and combinatorial arrays. *IEEE Trans. Information Theory*, 59(2):994–1003, 2013.
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.
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.
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.
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.
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.
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.
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.
12. Dan Boneh and Mark Zhandry. Multiparty key exchange, efficient traitor tracing, and more from indistinguishability obfuscation. In Juan A. Garay and Rosario Gennaro, editors, *CRYPTO 2014, Part I*, volume 8616 of *LNCS*, pages 480–499. Springer, Heidelberg, August 2014.
13. Hervé Chabanne, Duong Hieu Phan, and David Pointcheval. Public traceability in traitor tracing schemes. In Ronald Cramer, editor, *EUROCRYPT 2005*, volume 3494 of *LNCS*, pages 542–558. Springer, Heidelberg, May 2005.
14. Yi-Ruei Chen and Wen-Guey Tzeng. A public-key traitor tracing scheme with an optimal transmission rate. In Sihan Qing, Chris J. Mitchell, and Guilin Wang, editors, *ICICS 09*, volume 5927 of *LNCS*, pages 121–134. Springer, Heidelberg, December 2009.

15. 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.
16. Ronald Cramer and Victor Shoup. Universal hash proofs and a paradigm for adaptive chosen ciphertext secure public-key encryption. In Lars R. Knudsen, editor, *EUROCRYPT 2002*, volume 2332 of *LNCS*, pages 45–64. Springer, Heidelberg, April / May 2002.
17. Nelly Fazio, Antonio Nicolosi, and Duong Hieu Phan. Traitor tracing with optimal transmission rate. In Juan A. Garay, Arjen K. Lenstra, Masahiro Mambo, and René Peralta, editors, *ISC 2007*, volume 4779 of *LNCS*, pages 71–88. Springer, Heidelberg, October 2007.
18. 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.
19. 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.
20. Caroline Fontaine, Sébastien Gambs, Julien Lolive, and Cristina Onete. Private asymmetric fingerprinting: A protocol with optimal traitor tracing using tardos codes. In Diego F. Aranha and Alfred Menezes, editors, *LATINCRYPT 2014*, volume 8895 of *LNCS*, pages 199–218. Springer, Heidelberg, September 2015.
21. 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`.
22. Shafi Goldwasser, Yael Tauman Kalai, Raluca A. Popa, Vinod Vaikuntanathan, and Nickolai Zeldovich. Reusable garbled circuits and succinct functional encryption. In Dan Boneh, Tim Roughgarden, and Joan Feigenbaum, editors, *45th ACM STOC*, pages 555–564. ACM Press, June 2013.
23. Sergey Gorbunov, Vinod Vaikuntanathan, and Hoeteck Wee. Functional encryption with bounded collusions via multi-party computation. In Reihaneh Safavi-Naini and Ran Canetti, editors, *CRYPTO 2012*, volume 7417 of *LNCS*, pages 162–179. Springer, Heidelberg, August 2012.
24. Sergey Gorbunov, Vinod Vaikuntanathan, and Hoeteck Wee. Attribute-based encryption for circuits. In Dan Boneh, Tim Roughgarden, and Joan Feigenbaum, editors, *45th ACM STOC*, pages 545–554. ACM Press, June 2013.
25. Rishab Goyal, Venkata Koppula, and Brent Waters. Collusion resistant traitor tracing from learning with errors. In Ilias Diakonikolas, David Kempe, and Monika Henzinger, editors, *50th ACM STOC*, pages 551–563. ACM Press, June 2018.
26. Rishab Goyal, Venkata Koppula, and Brent Waters. Collusion resistant traitor tracing from learning with errors. In Ilias Diakonikolas, David Kempe, and Monika Henzinger, editors, *50th ACM STOC*, pages 660–670. ACM Press, June 2018.
27. Rishab Goyal, Satyanarayana Vusirikala, and Brent Waters. Collusion resistant broadcast and trace from positional witness encryption. Cryptology ePrint Archive, Report 2019/031, To appear in PKC 2019. `https://eprint.iacr.org/2019/031`.
28. Fuchun Guo, Yi Mu, and Willy Susilo. Identity-based traitor tracing with short private key and short ciphertext. In Sara Foresti, Moti Yung, and Fabio Martinelli, editors, *ESORICS 2012*, volume 7459 of *LNCS*, pages 609–626. Springer, Heidelberg, September 2012.

29. 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.
30. 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.
31. Jiangtao Li and Junqing Gong. Improved anonymous broadcast encryptions - tight security and shorter ciphertext. In Bart Preneel and Frederik Vercauteren, editors, *ACNS 18*, volume 10892 of *LNCS*, pages 497–515. Springer, Heidelberg, July 2018.
32. 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.
33. 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.
34. Hung Q. Ngo, Duong Hieu Phan, and David Pointcheval. Black-box trace and revoke codes. 67(3):418–448, November 2013.
35. 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.
36. 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.
37. 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.
38. 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.
39. Ron Roth. *Introduction to Coding Theory*. Cambridge University Press, New York, NY, USA, 2006.
40. Reihaneh Safavi-Naini and Yejing Wang. Traitor tracing for shortened and corrupted fingerprints. In *Digital Rights Management Workshop*, volume 2696 of *Lecture Notes in Computer Science*, pages 81–100. Springer, 2002.
41. Adi Shamir. How to share a secret. *Communications of the Association for Computing Machinery*, 22(11):612–613, November 1979.
42. Thomas Sirvent. Traitor tracing scheme with constant ciphertext rate against powerful pirates. Cryptology ePrint Archive, Report 2006/383, 2006. `http://eprint.iacr.org/2006/383`.
43. Gábor Tardos. Optimal probabilistic fingerprint codes. In *35th ACM STOC*, pages 116–125. ACM Press, June 2003.
44. 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.

A Appendix

A.1 Proof of the Semantic Security in Theorem 3

Proof. We consider a sequence of games starting with **Game** G_0 as following:

Game 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 he then gives to the adversary $\mathcal{A}_\Gamma$. In **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 AnoBEB (denoted by Π) by calling algorithm

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

In the **Challenger phase**, the adversary selects two messages $K^0, K^1 \in \mathcal{PT}^\ell$ and a subset of revoked users $\mathcal{R} \subset \mathcal{C}$. The challenger picks at random a $b \hookleftarrow \{0,1\}$, calls the procedure $\mathsf{Share}(K^b, \rho\ell, \ell)$ to get ℓ shares $K_1^b, \ldots, K_\ell^b$ for the message K^b and outputs a ciphertext $\Pi.\mathsf{Encrypt}(\mathsf{ek}_j, K_j^b, \mathcal{M}_j)_{j=1}^{\ell}$, 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 **Phase 2**, $\mathcal{A}_\Gamma$ received the ciphertext, sampled from one of two computationally indistinguishable distributions

$$\mathcal{D}_0 = \Big(\Pi.\mathsf{Encrypt}(\mathsf{ek}_1, K_1^0, \mathcal{M}_1), \ldots, \Pi.\mathsf{Encrypt}(\mathsf{ek}_\ell, K_\ell^0, \mathcal{M}_\ell)\Big)$$
$$\mathcal{D}_1 = \Big(\Pi.\mathsf{Encrypt}(\mathsf{ek}_1, K_1^1, \mathcal{M}_1), \ldots, \Pi.\mathsf{Encrypt}(\mathsf{ek}_\ell, K_\ell^1, \mathcal{M}_\ell)\Big),$$

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

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

Game G_1**:** The challenger now makes one small change to the **Game** G_0. Namely, instead of encrypting the first share K_1^0 with mixture $\mathcal{M}_1$, we encrypt K_1^1 with the mixture $\mathcal{M}_1$. This 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 the same as in **Game** G_0 except as mentioned about the 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

$$\begin{aligned}\mathcal{D}_0^1 = \Big(&\Pi.\mathsf{Encrypt}(\mathsf{ek}_1, K_1^1, \mathcal{M}_1), \Pi.\mathsf{Encrypt}(\mathsf{ek}_2, K_2^0, \mathcal{M}_2) \\ &, \ldots, \Pi.\mathsf{Encrypt}(\mathsf{ek}_\ell, K_\ell^0, \mathcal{M}_\ell)\Big).\end{aligned}$$

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

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

Therefore, we have

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

where ε_1 is a quantity, defined by

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

Claim 1. We assume that ε_1 is bounded by an avantage of the attacker in Π scheme, namely

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

Indeed, assume the contrary, that there exists a polynomial time attacker $\mathcal{A}_{\mathsf{DIST}}$ which is able to distinguish between the two distributions $\mathcal{D}_0$ and $\mathcal{D}_0^1$ with a non-negligible probability. We then build a simulator $\mathcal{S}$ to break the Π scheme as follows:

The simulator takes as input a public key ek_Π and generates $(\ell - 1)$ pairs of public key and secret key $\{\mathsf{ek}_i, \mathsf{MSK}_i\}_{i=2}^{\ell}$. $\mathcal{S}$ 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 $\{K_1^0, \ldots, K_\ell^0\}$, $\{K_1^1, \ldots, K_\ell^1\}$ and the family of mixture

$$\{\mathcal{M}_1, \mathcal{M}_2, \ldots, \mathcal{M}_\ell\}.$$

By querying the challenger of the scheme Π with the shares K_1^0, K_1^1 and the mixture $\mathcal{M}_1$, it receives a ciphertext of the form,

$$\mathsf{Encrypt}(\mathsf{ek}_1, K_1^b, \mathcal{M}_1),$$

where bit b was chosen randomly by the challenger. The others ciphertexts $\{\mathsf{Encrypt}(\mathsf{ek}_j, K_j^0, \mathcal{M}_j)\}_{j=2}^{\ell}$ generated by the simulator as well to establish a full ciphertext

$$\Big(\mathsf{Encrypt}(\mathsf{ek}_1, K_1^b, \mathcal{M}_1), \mathsf{Encrypt}(\mathsf{ek}_2, K_2^0, \mathcal{M}_2), \ldots, \\ \mathsf{Encrypt}(\mathsf{ek}_\ell, K_\ell^0, \mathcal{M}_\ell)\Big).$$

By our assumption, $\mathcal{A}^{\mathsf{DIST}}$ can distinguish efficiently the two distributions above, as soon as $\mathcal{A}^{\mathsf{DIST}}$ outputs bit b, the simulator $\mathcal{S}$ will return the same value b. We see that if $K_1^0 = K_1^1$, the two distributions $\mathcal{D}_0$ and $\mathcal{D}_0^1$ coincide.

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

Game G_2**:** This game is identical with **Game** G_1 with the difference being that the challenger changes the second coordinate in $\mathcal{D}_0^1$ by

$$\Pi.\mathsf{Encrypt}(\mathsf{ek}_2, K_2^1, \mathcal{M}_2)$$

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

$$\mathcal{D}_0^2 = \Big(\Pi.\mathsf{Encrypt}(\mathsf{ek}_1, K_1^1, \mathcal{M}_1), \Pi.\mathsf{Encrypt}(\mathsf{ek}_2, K_2^1, \mathcal{M}_2), \ldots, \\ \Pi.\mathsf{Encrypt}(\mathsf{ek}_\ell, K_\ell^0, \mathcal{M}_\ell)\Big).$$

We denote the advantage of the adversary in this game by $\mathsf{Adv}_{\mathcal{A}_\Gamma}^{\textbf{Game } \mathrm{G}_2}(\mathcal{D}_0^2, \mathcal{D}_1)$. And from this, we have

$$\mathsf{Adv}_{\mathcal{A}_\Gamma}^{\textbf{Game } \mathrm{G}_1}(\mathcal{D}_0^1, \mathcal{D}_1) \leq \mathsf{Adv}_{\mathcal{A}_\Gamma}^{\textbf{Game } \mathrm{G}_2}(\mathcal{D}_0^2, \mathcal{D}_1) + \varepsilon_2,$$

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 that of **Claim 1**, we get

$$\varepsilon_2 \leq \mathsf{Adv}_\Pi.$$

Game G_ℓ**:** We substitute the ℓ^{th} coordinate of the distribution $\mathcal{D}_0^\ell$ by $\Pi.\mathsf{Encrypt}(\mathsf{ek}_\ell, K_\ell^1, \mathcal{M}_\ell)$ and still introduce no change to 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 the advantage of the adversary in this game by $\mathsf{Adv}_{\mathcal{A}_\Gamma}^{\textbf{Game } \mathrm{G}_\ell}(\mathcal{D}_0^\ell, \mathcal{D}_1)$. Then, from this, we have

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

where ε_ℓ is a quantity, defined by

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

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

$$\Big|\mathsf{Adv}_{\mathcal{A}_\Gamma}^{\textbf{Game } \mathrm{G}_0}(\mathcal{D}_0, \mathcal{D}_1)\Big| \\ \Big|\mathsf{Adv}_{\mathcal{A}_\Gamma}^{\textbf{Game } \mathrm{G}_0}(\mathcal{D}_0, \mathcal{D}_1) - \mathsf{Adv}_{\mathcal{A}_\Gamma}^{\textbf{Game } \mathrm{G}_\ell}(\mathcal{D}_0^\ell, \mathcal{D}_1)\Big| \\ \leq \sum_{i=1}^{\ell} \varepsilon_\ell \leq \ell.\mathsf{Adv}_\Pi.$$