This paper appears
In Algorithmica, Springer, vol. 67, no. 3, Pages 418-448, 2013

Black-box Trace&Revoke Codes

Hung Q. Ngo[1], Duong Hieu Phan[2,3], and David Pointcheval[2]

[1] SUNY Buffalo
[2] ENS/CNRS/INRIA
[3] University Paris 8

Abstract. We address the problem of designing an efficient broadcast encryption scheme which is also capable of tracing traitors. We introduce a code framework to formalize the problem. Then, we give a probabilistic construction of a code which supports both traceability and revocation. Given N users with at most r revoked users and at most t traitors, our code construction gives rise to a Trace&Revoke system with private keys of size $O((r+t)\log N)$ (which can also be reduced to constant size based on an additional computational assumption), ciphertexts of size $O((r+t)\log N)$, and $O(1)$ decryption time. Our scheme can deal with certain classes of pirate decoders, which we believe are sufficiently powerful to capture practical pirate strategies.
In particular, our code construction is based on a combinatorial object called (r,s)-disjunct matrix, which is designed to capture both the classic traceability notion of disjunct matrix and the new requirement of revocation capability. We then probabilistically construct (r,s)-disjunct matrices which help design efficient Black-Box Trace&Revoke systems. For dealing with "smart" pirates, we introduce a tracing technique called "shadow group testing" that uses (close to) legitimate broadcast signals for tracing. Along the way, we also proved several bounds on the number of queries needed for black-box tracing under different assumptions about the pirate's strategies.

Keywords: Broadcast Encryption, Traitor Tracing, Disjunct Matrix, Shadow Group Testing

1 Introduction

In many real-world applications such as Pay-TV, satellite radio, and the distribution of copyright-protected materials, a content provider needs to broadcast digital information to a specific set of users (e.g., subscribers) who were given key(s) for decrypting the content. Two natural requirements arise in such setting. First, the broadcast system should be able to painlessly *revoke* the receiving rights of an arbitrary subset of subscribers, probably because they unsubscribed from the service or violated some rules. This is the essence of the *broadcast encryption* problem [Ber91,FN94]. Second, some users might collude to build a pirate decoder and distribute it, for a fee or not. Or, a pirate might achieve similar effects via hacking accounts of legitimate users. Either way, such users are called *traitors*. It is thus desirable for the system to be able to trace (and then revoke) at least one traitor by examining the pirate decoder. This is the *traitor tracing* problem [CFN94]. All in all, broadcast encryption systems which are capable of both tracing and revoking would be widely useful.

Many existing works studied the two problems separately, leading to inefficiency when applying the solution to one problem to cope with the other. For example, *collusion-secure codes* [BS95] provide a powerful tool against the illegal distribution of fingerprinted material in settings satisfying the so-called *Marking Assumption* [BS95,Tar03]. Though designed for fingerprinting large digital objects, these codes have been widely applied in the context of multi-user encryption for tracing traitor, motivated by the work of Kiayias and Yung [KY02] in which a black-box tracing scheme with constant ciphertext rate was proposed. The schemes in [FNP07,Sir07,BN08,BP08] belong to this class.

A drawback of employing collusion-secure codes for multi-user encryption is the resulting relatively large key size. For the marking assumption to be valid, each user – characterized by a codeword – should be assigned one out of two (or one out of many, with larger alphabet such as in IPP codes [SSW00]) keys for each position of the codeword. Indeed, the lower bound established in [Tar03] shows that the code length must be $\Omega(t^2\log(N/\epsilon))$ for a system of N users with at most t traitors and with the failure probability ϵ of the tracing procedure. This means the private key sizes of users cannot be improved beyond this bound. The construction in [Tar03] has a code length of about $100t^2\log(N/\epsilon)$, making the schemes inefficient for general practical usage.

Due to the quadratic dependency on the number of traitors, code-based schemes are only suitable for applications where the number of traitors is relatively small, say $t = O(\log N)$. For example, in the Pay-TV application the users' secret keys are stored in tamper-resistant smartcards, making it difficult and time-consuming to recover a key: recovering one key in a tamper-resistant smartcard does not necessarily help speed up the recover of another key in another tamper-resistant smartcard. In such applications, it is probably reasonable to assume a small value of t. However, even when the number of traitors is small, collusion-secure codes can still be quite inefficient. For instance, for a system of four millions users and at most $t = 22$ traitors, Tardos' code induces a system where each user's private key is composed of more than one million sub-keys.

Recently, schemes based on collusion-secure codes allowing erasure have been made more practical in terms of ciphertext size (as small as a constant, as shown in [BN08,BP08]). However, in this case the code length and thus the private key size should be even longer.

A relatively small number of proposed systems, called *trace&revoke* systems [NP00,NNL01,BW06], do address both traitor tracing and broadcast encryption. These schemes can roughly be classified into three categories: schemes based on some forms of polynomial interpolation [NP00,DF03], schemes in the tree-based subset-cover framework [NNL01,HS02,DF02], and pairing-based schemes that support full collusion [BW06].

However, there are still fundamental shortcomings. The schemes in [NP00,DF03] have to fix an a-priori bound of the size of the collusion and as soon as the pirate could collect more than this bound of keys, from traitors or revoked users, it can totally break the schemes (by reconstructing the master secret key). The schemes [NNL01,DF02] are rather efficient but they can only deal with a non-standard notion of traitor tracing where the tracer can either identify a traitor or render pirate decoder useless (this is mentioned in [NNL01]). The scheme [BW06] can deal with full collusion but with a large ciphertext size of $O(\sqrt{N})$ even when there is no traitor in the system.

The major objective of our paper is to propose a new code-based framework for constructing black-box Trace&Revoke systems which have small private key *and* ciphertext sizes. We will deal with *black-box tracing*, which means the traitors are traced by simple interactions with the pirate decoder. The decoder can be reset as many times as we want, i.e. it is stateless.

Contributions and paper outline. Section 2 formalizes the notion of revocable codes, and efficient conjunction revocable codes in particular. The Complete-Subtree scheme [NNL01] is a type of conjunction revocable code. The decoding possibility of a user (i.e. a codeword) is captured by a *predicate*, which is a binary relation indicating whether a codeword is capable of decrypting a signal.

Section 3 formalizes the traceability of codes. We introduce two notions in order to formalize the capability and the strategy of pirate decoders:

- Each pirate decoder, produced by a collusion C, is associated to a word in a *Useful Feasible Set of* C. Roughly, this set is defined such that if a signal can be decrypted by every member of C then it can also be decrypted by a word in the useful feasible set. This notion formalizes useful pirate decoders because the pirate decoder should assure the "minimum" capability of the collusion: when all members in the collusion can decrypt a signal, the pirate decoder should also be able to decrypt that same signal. However, Useful Feasible Set alone does not capture "smart" decoders in the sense that it does not use any strategy: whenever it can decrypt, it will decrypt.
- The pirate decoder could of course choose an anti-tracing strategy by refusing to decrypt signals that it considers abnormal or coming from a tracing procedure. We introduce the notion of *qualified signals* to formalize the anti-tracing strategies of pirates.

We borrow the notion of useful feasible set from collusion-secure codes to formalize the capability of the colluders. We note that, in collusion-secure codes, a word in the feasible set is also associated to a perfect decoder and in order to deal with smarter pirate decoders, one should use a collusion-secure code with an all-or-nothing transform [KY02] or one should use a robust collusion-secure codes that allow erasure [Sir07,BP08,BN08]. In our setting, the pirate's strategy is formalized by the notion of qualified signals.

In Section 3, we also indicate a simple connection between traceability of codes with the so-called *disjunct matrices*, a classic combinatorial object which has "built-in" tracing capability.

The problem with disjunct matrices is that they have no "built-in" efficient revocation capability. Indeed, disjunct matrices or equivalently *cover-free families* have been used for traitor tracing in [TSN06]. However, by following the tracing framework of [BF99] it cannot be used for revocation. We deal with this problem in Section 4 by introducing a new combinatorial object called (r, s)-*disjunct matrices*, which retains the tracing-capability of disjunct matrices while also supports revocation. As disjunct matrices have applications in diversely many areas [DH00], we believe that the new and stronger notion of (r, s)-disjunct matrices will be widely applicable as well.

Section 5 is the heart of the paper, where we present a method for constructing good (r, s)-disjunct matrices which allow for tracing and efficient revocation. The resulting code yields a Trace&Revoke scheme with private key size and ciphertext length $O((t + r) \log(N/(t + r)))$ for N users, at most r revoked users and at most t traitors. The constants hidden in the big-O are small (≤ 8). This randomized construction yields a key assignment scheme where users pick their keys independently from the same distribution and all keys have the same role. Thus, unlike the complete-subtree method which leads to a highly asymmetric key assignment making it not suitable for tracing smart pirate decoders, our code has better "built-in" support for traceability against non-trivial pirate strategies.

Rigorously, we deal with non-trivial pirates (that are characterized by some qualified predicates). For a probabilistic code where codewords are picked independently from the same distribution and all keys used in encryption have the same role, a non-trivial pirate can estimate the number of keys used in a normal encryption and can refuse to decrypt a ciphertext that contains too few or too many keys that lies outside its estimated interval. This strategy of pirate, called weight-limited pirate, is formalized under interval qualified predicate. To cope with this strategy of pirate decoder, we introduce a tracing technique called *shadow group testing* that uses (close to) legitimate broadcast signals for tracing. In particular, in one setting, we consider general pirate decoders that can used any strategy. We show that the problem of deciding whether a given signal is a legitimate broadcast signal (here, broadcast signal is a ciphertext targeted to all users) or a tracing signal is **NP**-hard. We thus establish that the shadow group testing technique can be used in conjunction with our code to construct revocable codes that are traceable against non-trivial pirates (modulo the computational hardness), in the conventional stateless setting where the pirate decoder could be resettable. This does not show that our trace&revoke code can be used to deal with any pirate decoder but it can be seen as a step toward this goal.

For tracing a particular type of pirate decoder which only decrypts signals of certain weights, we prove upper and lower bounds on the number of tests needed for a variant of group testing where each test must consist of a given number of items.

Last but not least, we prove upper- and lower-bounds for the number of black-box queries necessary in the information-theoretic limit when the pirate only decrypts signals which are legitimate broadcast signals (and when it has the keys). This result applies to arbitrary Trace&Revoke system, not just code-based ones like ours, as long as the pirate decryption assumption is valid.

Section 6 discusses the questions of how to optimize code lengths (using multi-user tracing families), private key size (down to a constant using Asano' method [Asa02]), and how to deal with unbounded number of traitors/revoked users.

2 Revocable Codes

2.1 General settings

Broadcast encryption (BE) schemes enable the sender of a message to specify a subset of the users the message will be sent to, called the *target set* or the *privileged set*. The complement of the target set is called the *revoked set*. To revoke the receiving rights of some desired subset of users, a BE scheme typically generates three pieces of data: (a) the *Id Header*, which is a bit-string that unambiguously identifies the target set/revoked set; (b) the *Key Header*, which encapsulates a session key for the privileged users; and (c) the *Message Body*, which contains the payload encrypted with the session key.

In what follows, we describe a BE scheme based on codes. Roughly speaking, each user is associated with a "codeword" which determines the private key(s) assigned to that user. To revoke a subset of

users, a code-based BE scheme generates a "signal" c which is a word (not necessarily a codeword) from which the Key Header will be constructed. The signal c will have to be "compatible" with the codewords assigned to privileged users and "incompatible" with revoked users so that only privileged users can decode. The notation of compatibility is captured by a Boolean predicate associated with the code. The formal definitions of broadcast encapsulation, public-key encryption and secret sharing schemes are given in Appendix A.1.

Definition 1 ((ℓ, N)-Code). Given a finite *alphabet* Σ, and positive integers ℓ and N, an (ℓ, N)-*code* Γ is an N-subset of Σ^ℓ, namely $\Gamma \subseteq \Sigma^\ell$ and $|\Gamma| = N$. Members of Σ^ℓ are called *words*. Members of Γ are called *codewords*. The quantity ℓ is called the *length*, and N the *size* of the code.

In order to build a BE scheme, we associate a codeword w to each user. Henceforth, without loss of generality we use codewords to identify users. Given a set $R \subseteq \Gamma$ of revoked users, the code-based BE scheme broadcasts by first generates a signal $c \in \Sigma^\ell$ which is not necessarily a codeword. The signal will be used to generate the session key for broadcasting. A user w is "compatible" with a signal w iff a corresponding predicate is true:

Definition 2 (Predicate). We associate a code Γ with a predicate $\mathsf{D} : \Sigma^\ell \times \Sigma^\ell \to \{0, 1\}$. The boolean value $\mathsf{D}(w, c)$ indicates whether the word/user w is compatible with the signal c. The practical semantic is that user w is able to recover the content associated with the signal c iff $\mathsf{D}(w, c) = 1$.

Given signal c, the predicate D specifies the target set (the set $\{w \mid w \in \Gamma \wedge \mathsf{D}(w, c) = 1\}$), or equivalently the revoked set. A subset of revoked users might somehow be able to collude, and combine their keys to recover the content. By combining their codewords, the colluders can generate new words (not necessarily codewords) which potentially can be used to decode signals which were not meant to be decodable by any one of them. We formalize this capability of a collusion by the following notion.

Definition 3 (Feasible Set). A collusion C of users can produce new words from their own codewords. This derivation of new words depends on the structure of the code. The set of words that can be derived from a subset C of codewords is called the **feasible set**, and is denoted by $\mathsf{F}(C; \Gamma)$. When there is no ambiguity, we omit Γ and use $\mathsf{F}(C)$ to denote the feasible set.

We can now define the notion of revocable code which is a basic building block for BE schemes.

Definition 4 ((Efficiently) Revocable Code). Let $\Gamma = \{w^1, \cdots, w^N\}$ be an (ℓ, N)-code. The code Γ is called r-**revocable** if there is a predicate D such that for all $R \subseteq \Gamma$ of size $|R| \leq r$, there exists a signal $c \in \Sigma^\ell$ satisfying the following conditions: $\forall u \in \Gamma - R : \mathsf{D}(u, c) = 1$ and $\forall v \in \mathsf{F}(R) : \mathsf{D}(v, c) = 0$. If, in addition, there is a poly-time algorithm $\mathsf{RevAlgo}$ that, given R, outputs a signal c satisfying the above condition, then the code is said to be **efficiently** r-**revocable**.

2.2 Conjunction Codes and Broadcast Encryption

In order to clarify the above formalism, we now present a specific family of binary revocable codes called k-*conjunction codes* and a BE scheme based on this family. For any two vectors $u, c \in \{0, 1\}^\ell$, let $u \wedge c$ (resp. $u \vee c$) denote the bitwise AND (resp. OR) of two vectors u and c. Let $\mathbf{w}_H(c)$ denote the Hamming weight of any word $c \in \{0, 1\}^\ell$.

Let $k \leq \ell$ be a positive integer, a k-conjunction code is a subset of $\{0, 1\}^\ell$ with the following associated predicate and feasible set.

Definition 5 (Predicate for k-Conjunction Codes). For any $u, c \in \{0, 1\}^\ell$, the predicate $\mathsf{D}_k(u, c) := (\mathbf{w}_H(u \wedge c) \geq k)$ is called the k-*conjunction predicate*.

Definition 6 (Feasible Set for k-Conjunction Codes). For any set $C = \{u^1, \cdots, u^c\} \subseteq \Gamma$. Define

$$\mathsf{F}(C) = F(C; \Gamma) = \left\{ w \in \{0, 1\}^\ell \quad s.t \quad \forall i \in [l], w_i \in \{0\} \cup \bigcup_{j=1}^{c} \{u_i^j\} \right\}.$$

Each word $w = (w_i)_{i=1}^{\ell} \in \{0,1\}^{\ell}$ can be thought of as a subset of $[\ell]$: the subset of indices i for which $w_i = 1$. Then, the above definitions can be translated as: $\mathsf{D}_k(u,c) = 1$ iff the intersection of u and c has size at least k, and $F(C)$ is the collection of all subsets of the union $\bigcup_{c \in C} c$.

The notion of revocable codes can now be made more precise for this family.

Definition 7 ((Efficiently) Revocable k-Conjunction Code). A binary (ℓ, N)-code Γ is called (r,s)-**Revocable k-Conjunction** if for all $R \subseteq \Gamma$ and $|R| \leq r$, there exists a signal $c \in \{0,1\}^{\ell}$, with $\mathbf{w}_H(c) \leq s$ satisfying the following conditions: $\forall u \in \Gamma - R : \mathsf{D}_k(u,c) = 1$ and $\forall v \in \mathsf{F}(R) : \mathsf{D}_k(v,c) = 0$. If in addition there is a polynomial time algorithm $\mathsf{RevAlgo}$ that computes c given R, then the code is said to be efficiently revocable.

Example 8 (The Complete-subtree scheme as a 1-conjunctive code). The Complete-Subtree scheme [NNL01] can be roughly described as follows. Imagine a full binary tree $\mathcal{T}$ with N nodes. Each user is associated with a leaf of the three. To each tree node there corresponds a distinguished encryption key. A user is given the set of keys corresponding to all the internal nodes from the leaf to the root of the tree. To revoke a subset R of users (i.e. leaves of the tree), we first construct a minimal subtree $\mathcal{T}'$ of $\mathcal{T}$ that spans R and the root. Let K be the set of nodes of $\mathcal{T}$ that are "hanging off" of $\mathcal{T}'$. Then, it is easy to see that each non-revoked user has some key in the set K, and no revoked user has any key in the set K. The system can then use the set K of keys to encrypt the broadcast message, whose length will be proportional to $|K|$, which can be shown to be $O(|R| \log(N/|R|))$.

The above key assignment scheme can be casted in terms of a 1-conjunction code as follows. There is a codeword for each leaf of the N-leaf full binary tree $\mathcal{T}$. The code length is $\ell = 2N-1$, each position (i.e. coordinate) of the code corresponds to a node of the tree. For each codeword w, there is a 1 in a position if the corresponding node is on the path from w to the root. We will refer to this code the *CS-code*.

Following the results in [NNL01] and our brief description above, the following proposition is straightforward.

Proposition 9. *For any $r \in [N]$, the CS-Code is a 1-conjunction $(r, r\log(N/r))$-revocable $(2N-1, N)$-code.*

We now present a broadcast encryption scheme that implements our above predicate for a general k-conjunction binary code. The new trick is to combine the secret sharing with the k-conjunction code such that only legitimate users possesses sufficient shares to be able to decrypt. This is a generalization of the previous schemes [NNL01] where the secret sharing is not involved. In fact, under our construction, the previous schemes [NNL01] correspond to the case $k = 1$ and the 1-out-of-m secret sharing becomes trivial.

Definition 10 (BE from Conjunction Codes). Let us be given a generator of Efficiently (r,s)-Revocable k-Conjunction binary (ℓ, N)-Codes, a secret sharing scheme $\mathcal{SSS}$, and a secure public-key encryption scheme $\mathcal{PKE}$. We build a BE scheme Π that can revoke up to r users in the following way.

- $\mathsf{Setup}(1^{\lambda}, N)$
  1. Run the code generating algorithm on (N,k,r) to obtain an Efficiently (r,s)-Revocable k-Conjunction (ℓ, N)-Code Γ.
  2. Run $\mathcal{PKE}.\mathsf{Setup}(1^{\lambda})$ to get the public parameters param for the encryption scheme;
  3. For $i = 1, \ldots, \ell$, run the key generation algorithm $\mathcal{PKE}.\mathsf{KeyGen}(\mathsf{param})$ to get the pair $(\mathsf{dk}_i, \mathsf{ek}_i)$.
  4. Set $\mathsf{MSK} = (\Gamma, \{\mathsf{dk}_i\})$, and $\mathsf{EK} = \{\mathsf{ek}_i\}$.
  5. For $i = 1, \ldots, N$, the user i is associated with the codeword $w^i \in \Gamma$, we write $w^i = w^i_1 \ldots w^i_{\ell}$ and set $\mathsf{usk}_i \leftarrow \{\mathsf{dk}_j / w^i_j = 1, j = 1, \ldots, \ell\}$.
- $\mathsf{Encaps}(\mathsf{EK}, R)$:
  1. For a revoked set R of size at most r, since the code Γ is efficiently (r,s)-revocable, one can find out a signal c of weight at most s, such that $\mathsf{D}_k(u,c) = 0$, for any $u \in F(R)$, and $\mathsf{D}(u,c) = 1$ for any $u \in [N] - R$. We denote by $m = \mathbf{w}_H(c)$ this weight;

2. Denote by $i_1, \ldots, i_m$ the positions of m 1-bits in c, *i.e.,* $c_{i_j} = 1$, for $j = 1, \ldots, m$;
3. Call $\mathsf{Share}(\kappa, k, m)$, a k-out-of-m secret sharing scheme of a κ-bit secret. The $\mathsf{Share}(\kappa, k, m)$ algorithm outputs a secret $K \xleftarrow{\$} \{0,1\}^\kappa$ and m shares $s_1, \ldots s_m$;
4. Set $e_{i_j} = \mathcal{PKE}.\mathsf{Enc}(\mathsf{pk}_{i_j}, s_j)$, for $j = 1, \ldots, m$.
5. Output K and $H = (c, (e_{i_j}), j = 1, \ldots, m)$.

– $\mathsf{Decaps}(\mathsf{usk}_j, R, H)$:
1. If j is in $[N] - R$, then $\mathsf{D}_k(w^j, c) = 1$. This means $\mathbf{w}_H(w^j \wedge c) \geq k$.
2. Denote by $i_1, \ldots, i_k$ the positions of the first k 1-bit in $w^j \wedge c$.
3. Compute $s_j = \mathcal{PKE}.\mathsf{Dec}(\mathsf{sk}_{i_j}, e_{i_j})$, for $j = 1, \ldots, k$. With the $\mathsf{Combine}$ algorithm on these k shares, reconstruct K.

The case $k = 1$ is the simplest case: the secret sharing scheme simply consists in choosing a random $K \xleftarrow{\$} \{0,1\}^\kappa$, and then to set $s_i = K$ for all i.

3 Traceable Codes

Traitors are users who collude to build (and distribute) a *pirate decoder*. The goal of *traitor tracing schemes* is to allow an authority to trace back, from a pirate decoder, at least one codeword which was used and thus at least one traitor. More precisely, let $T \subset \Sigma^\ell$ denote the set of (codewords of) at most t traitors. From this set of codewords the "pirate" produces a *pirate decoder* $\mathbb{D}$ that efficiently decrypts some broadcast signals $c \in \{0,1\}^\ell$. We view $\mathbb{D}$ as boolean function on predicates $c \in \{0,1\}^\ell$: $\mathbb{D}(c) = 1$ means $\mathbb{D}$ correctly decrypts c, and $\mathbb{D}(c) = 0$ otherwise.

The main task of the traitor tracing scheme is to identify at least one codeword in T (i.e. one traitor) by "examining" the pirate decoder. In this paper we are only concerned with the commonly used "blackbox tracing" model, where the tracer can only query the decoder function $\mathbb{D}$ [CFN94,KY02]. In reality, querying $\mathbb{D}$ is roughly equivalent to the act of sending a broadcast signal to the physical decoding device, examining its output (whether it decrypts the content correctly), and then resetting the device. While it is true that some devices may not be stateless (say, some data is stored in a ROM), the blackbox tracing model is still a reasonably practical model.

Our plan is as follows. We first define the notion of a traceable code. The code is designed such that from a word w belonging to the "useful feasible set" of words, the code allows us to trace back at least one traitor. Recall that the feasible set $\mathsf{F}(T)$ is the set of all words that the traitors can derive. (We can roughly think of $\mathsf{F}(T)$ as the set of decryption keys that the traitors can derive from their private keys.) The useful feasible set $\mathsf{UF}(T)$ is a subset of the feasible set, whose meaning is define below. Then, we give a specific family of 1-conjunction traceable codes based on disjunct matrices. Finally, we describe how we might possibly get a hold of a word from the $\mathsf{UF}(T)$. This task is highly dependent on the "anti-tracing" strategy used by the pirate, and thus we will model the anti-tracing strategy with the notion of "qualified signals."

3.1 Traceable Codes

The feasible set models the collection of all words which can be "constructed" by the traitors from their codewords. A word in the feasible set $\mathsf{F}(T)$ may or may not be useful for decrypting broadcast contents. We consider a word w from a collusion T *useful* if it satisfies the following condition: for any signal, if every user in T is able to decrypt it, then w should be also able to decrypt it.

Definition 11 (Useful Feasible Set). The **useful feasible set**, denoted by $\mathsf{UF}(T; \Gamma)$ (or $\mathsf{UF}(T)$) from a collusion T is the subset of words w in $\mathsf{F}(T)$ such that, for any signal $c \in \Sigma^\ell$, if $\mathsf{D}(u, c) = 1, \forall u \in T$, then $\mathsf{D}(w, c) = 1$.

It follows from the definition that $T' \subseteq T$ implies $\mathsf{UF}(T') \subseteq \mathsf{UF}(T)$. To see this, consider a word $w \in \mathsf{UF}(T') \subseteq \mathsf{F}(T)$ and an arbitrary signal c. If $\mathsf{D}(u, c) = 1 \forall u \in T$, then $\mathsf{D}(u, c) = 1 \forall u \in T'$, which in turns implies $\mathsf{D}(w, c) = 1$. Hence, $w \in \mathsf{UF}(T)$. Roughly, adding more traitors to a traitor set leads to more useful words. Intuitively, the pirate should at least be able to decrypt signals that all traitors are able of decrypting. Hence, the above definition is in some sense the weakest requirement of being useful.

Useful Feasible Set of 1*-conjunction codes* To illustrate the concept of useful feasible set, let us characterize the useful feasible sets of 1-conjunction codes. Let Γ be a 1-conjunction (ℓ, N)-code, and $T \subseteq \Gamma$ be an arbitrary set of traitors. Recall that the alphabet is binary in this case. Each word $w = (w_i)_{i=1}^{\ell} \in \{0,1\}^{\ell}$ can naturally be viewed as a subset of $[\ell]$: the set of all positions $i \in [\ell]$ for which $w_i = 1$. This way, the intersection and union of words are also words in $\{0,1\}^{\ell}$. Also, we can write $w \subset v$ for two words $w, v \in \{0,1\}^{\ell}$ without ambiguity.

Proposition 12. *Let $T \subseteq \Gamma$ be an arbitrary non-empty set of codewords of a* 1*-conjunction (ℓ, N)-code. Then,*

$$\mathsf{UF}(T) = \{w \in \mathsf{F}(T) \mid \exists u \subset T, u \subseteq w\}\,.$$

In words, a word w in the feasible set $\mathsf{F}(T)$ is useful if w contains some member of T.

Proof. The fact that $\{w \in \mathsf{F}(T) \mid \exists u \subset T, u \subseteq w\} \subseteq \mathsf{UF}(T)$ is straightforward from definitions. We prove the converse. Assume to the contrary that there is some $w \in \mathsf{UF}(T)$ what does not contain any $u \in T$. Let c be a signal, viewed as a subset of $[\ell]$, constructed by collecting arbitrarily one member from each of the set $u \setminus w$ for each $u \in T$. Then, $\mathsf{D}_1(u, c) = 1$ for all $u \in T$ yet $\mathsf{D}_1(w, c) = 0$ because $w \cap c = \emptyset$. This is a contradiction.

We are now ready to formalize the notion of traceable codes.

Definition 13 ((Efficiently) Traceable Code). An (ℓ, N)-code Γ is **t-traceable** if from any collusion $T \subset \Gamma$ of size at most t, and any word w in the useful feasible set $\mathsf{UF}(T)$, there is an algorithm that on input w outputs a codeword in T. This algorithm is a *tracing algorithm*. If there is a polynomial time tracing algorithm, then the code is said to be *efficiently traceable*.

We have not specified how one might be able to construct a traceable code, efficiently or not, even in the 1-conjunction code case. Given a generic 1-conjunction code, and a word $w \in \mathsf{UF}(T)$, we can construct a set T_w of candidate codewords which are all codewords u which contains w. However, we can not be sure which of the words in T_w belong to the traitor set T. We will enlist the help of disjunct matrices to construct 1-conjunction traceable codes.

3.2 Traceable codes from disjunct matrices

The classic combinatorial structure allowing for a very common type of tracing is the so-called *disjunct matrices* [DH00]. Roughly speaking, an r-disjunct matrix is a binary matrix satisfying the following property: given the (boolean) union of at most r unknown columns of the matrix we can identify *all* the unknown columns in time linear in the size of the matrix. Disjunct matrices turn out to be very useful in constructing efficiently traceable 1-conjunction codes, so we formally define and discuss their properties next.

Let $\mathbf{M}$ be an $\ell \times N$ binary matrix. As in the previous section, we will also view each column of $\mathbf{M}$ as a subset of $[\ell]$, In particular, the set of columns of $\mathbf{M}$ is a family of subsets of $[\ell]$. Similarly, the rows of $\mathbf{M}$ form a family of subsets of $[N]$.

Definition 14 (r-Disjunct Matrix). An $\ell \times N$ binary matrix $\mathbf{M}$ is said to be **r-disjunct** if no column (viewed as a subset of $[\ell]$) is contained in the union of any r other columns.

This concept is equivalent to the notion of r-cover-free set family [EFF85]. Disjunct matrices are used to design non-adaptive group tests in the following sense. There is a set of at most r *positive items* in a population of N items. The rest of the items are *negatives*. We must identify the positives using as few non-adaptive "tests" as possible. Each test is a subset of items. A test returns positive iff at least one positive item is contained in the test. In the original group testing application [Dor43], each item is a blood sample, and a test is a pool of blood samples which indicates if any sample in the pool is positive for syphilis. That application explains the "positive" and "negative" terms.

Associate each column of an $\ell \times N$ binary matrix $\mathbf{M}$ with an item. Each row of $\mathbf{M}$ represents a test, which consists of all columns with a 1 on the row. The test outcome vector is precisely the union of the positive columns, where 1 represents positive test outcome and 0 negative. It is well-known [DH00] that, if $\mathbf{M}$ is r-disjunct, then there is a $O(\ell N)$-time algorithm that identifies *all* the positives given the test outcome vector. The algorithm eliminates all items that participate in a negative test. The remaining items are identified as positives. The matrix is r-disjunct if and only if this elimination procedure returns the correct positive set, for an arbitrary set of at most r positives.

The following proposition explains a slightly stronger capability of disjunct matrices. The proposition allows for a disjunct matrix to identify a subset of positives when the outcome vector which is not necessarily an exact union of some positives.

Proposition 15. *Let $\mathbf{M}$ be an r-disjunct matrix with dimension $\ell \times N$. Let $\mathbf{M}^{(j)}$ denote the jth column of $\mathbf{M}$. Let T be any (unknown) subset of at most r columns of $\mathbf{M}$. Let $\emptyset \neq S \subset T$ and $w \in \{0,1\}^\ell$ be a vector such that $\bigcup_{j\in S} \mathbf{M}^{(j)} \subseteq w \subseteq \bigcup_{j\in T} \mathbf{M}^{(j)}$. Then, from the vector $\mathbf{w}$ we can identify a set of columns U in time $O(\ell N)$ such that $S \subseteq U \subseteq T$.*

Proof. Let $\mathbf{M} = (m_{ij})$, and $w = (w_i)_{i=1}^{\ell}$. Remove any column j such that $m_{ij} = 1$ and $w_i = 0$ for some $i \in [\ell]$. Let U be the set of remaining columns. We claim that $S \subseteq U \subseteq T$. First, consider any column $j \notin T$. By the definition of r-disjunctness, column $\mathbf{M}^{(j)}$ is not contained in the union of columns in T. In particular, $\mathbf{M}^{(j)}$ is not contained in $\mathbf{w}$. Thus, there is some row $i \in [\ell]$ such that $m_{ij} = 1$ and $w_i = 0$. Column j is thus removed by the algorithm. We conclude that $U \subseteq T$. Next, consider any column $j \in S$. Since $\mathbf{M}^{(j)} \subseteq \mathbf{w}$ column j is not removed. Consequently, $S \subseteq U$ as desired.

We can also think of an $\ell \times N$ binary matrix as an (ℓ, N)-code where the codewords are defined to be the columns of the matrix. The following corollary follows from Propositions 15 and 12.

Corollary 16. *Let Γ be the collection of columns of an $\ell \times N$ t-disjunct matrix. Then, Γ is an efficiently t-traceable 1-conjunction code.*

3.3 Black-box Traceability and Decoders' Strategies

So far, we have defined traceable codes and specified how to obtain efficiently traceable 1-conjunction codes from disjunct matrices. Traceable codes allow us to pin-point at least one traitor from any given useful feasible word. So the remaining question is how to get a hold of a useful feasible word.

Our ability to identify a useful feasible word depends intimately on the "anti-tracing" strategy adopted by the pirate (decoder). The anti-tracing strategy attempts to decrypt some signal while ignores others. We call the set of signals that the pirate (decoder) attempts to decrypt the "qualified signals," and this set is modeled with a relation as in the following definition.

Definition 17 (Qualified signals). Let $T \subset \Gamma$ be the set of traitors. Let $\mathcal{Q}$ to be the binary relation $\mathcal{Q}$ over $T \subset \Gamma$ and signals $c \in \{0,1\}^\ell$ defined by $\mathcal{Q}(T, c) = 1$ if the pirate decoder attempts to decrypt broadcast messages associated with the signal c, and $\mathcal{Q}(T, c) = 0$ otherwise.

Recall that $\mathbb{D}$ denotes the pirate decoder, which is a boolean function, where $\mathbb{D}(c) = 1$ iff the decoder successfully decode signal c. For any subset T of words, define $\mathsf{D}(T, c) = \wedge_{u\in T} D(u, c)$. We certainly can not hope to trace pirate decoders that do not decode any signal at all. Our aim is to be able to trace decoders which decode signals that it aims to decode with a non-negligible probability.

Definition 18 (Effective Decoder). A pirate decoder $\mathbb{D}$ is called $(\mathcal{Q}, p)$-effective if for any signal $c \in \Sigma^\ell$, $\Pr[\mathbb{D}(c) = 1 \mid \mathsf{D}(T, c) = 1 \text{ and } \mathcal{Q}(T, c) = 1] \geq p$, where T is the set of codewords (traitors) used to build the decoder.

We will consider black-box tracing procedures, which trace traitors by simple interactions with the pirate decoder. We assume that the decoder can be reset as many time as one wants, thus is it stateless and can be modeled with the function $\mathbb{D}$ as described earlier. However, the decoder can have a specific strategy $\mathcal{Q}$. The only thing that is going for us is that the pirate decoder has to be $(\mathcal{Q}, p)$-effective.

Definition 19 (Black-box (Efficiently) Traceable Code). An (ℓ, N)-code Γ is $(\mathcal{Q}, p, \delta)$**-blackbox** t**-traceable** if there exists a tracing algorithm **Trace** such that, for any collusion T of size at most t, and any $(\mathcal{Q}, p)$-effective decoder $\mathbb{D}$, the tracing algorithm $\mathbf{Trace}^{\mathbb{D}}$, with oracle access to the decoder, outputs a traitor in T with probability at least δ. If there is a tracing algorithm which runs in time $\text{poly}(N)$, then the code is said to be $(\mathcal{Q}, p, \delta)$-*blackbox efficiently* t-*traceable code*. In particular, such an algorithm can only issue $\text{poly}(N)$ queries to the decoder.

To illustrate the above concepts, let us consider several anti-tracing strategies.

Example 20 (Naive Decoder). We first consider the case when the pirate has no strategy at all. The following decoder was called a "perfect decoder" in [BN08]. A **naive decoder** is a decoder that tries to decrypt any word c with no strategy: $\mathcal{Q}(T, c) = 1$ for any collusion T and any signal c.

This is of course the weakest adversary for a tracing algorithm. For example, the CS-Code [NNL01] described in Example 8 and the disjunct matrix-based code described in Corollary 16 are both black-box efficiently traceable. Assuming the decoder $\mathbb{D}$ is naive, the tracing algorithm works as follows. It queries the decoder $\mathbb{D}$ with all weight-1 signals $c \in \{0,1\}^\ell$. These are the signals with 1 in some coordinate and 0s elsewhere. By repeating the queries many times, we can amplify the success probability (to be more than δ, see Lemma 21 below) of identifying the set of positions $i \in [\ell]$ at which some codeword in T has a 1. From these positions, we obtain a useful feasible word w for which $u \subseteq w$ for every codeword $u \in T$. This useful feasible word w is precisely the union of the traitor codewords. Hence, the set of all traitors can be traced if the code is based on a t-disjunct matrix.

For the CS-code, we apply the tracing algorithm described in [NNL01]. Recall that in the CS-code codewords are constructed from leaves of a full binary tree with N leaves. Each codeword is of length $\ell = 2N - 1$, one position for each node in the tree. A codeword has 1s in the positions corresponding to the nodes from the associated leaf up to the root. Now, from the feasible word w above, we know of all the paths from the traitors to the root and thus we can easily identify the traitors.

The following simple lemma shows us how to amplify the success probability of a tracing procedure. Each query to a pirate decoder is some signal c, and we would like to know whether $\mathbb{D}(c) = 1$ or 0 with high confidence.

Lemma 21. *Consider a probabilistic pirate decoder that, for each query c, if it is able to decrypt c then it gives the correct answer $\mathbb{D}(c) = 1$ with probability at least p, and if it cannot decrypt c then it always outputs $\mathbb{D}(c) = 0$. Suppose we want to issue q different queries $c_1, \cdots, c_q$ to this pirate decoder. Then, by repeating* each *query $O\left(\frac{\ln\left(\frac{q}{1-\delta}\right)}{\ln\left(\frac{1}{1-p}\right)}\right)$ times, we will obtain all q correct answers with probability more than δ. (If $p = 1$ then each query is issued once only, for a total of q queries.)*

Proof. Suppose we repeat each query m times, and outputs 1 only if at least one of the m copies of the query returns $\mathbb{D}(c) = 1$. Then, we will be wrong with probability at most $(1-p)^m$. Hence, for q different queries $c_1, \cdots, c_q$ we will be wrong on some of them with probability at most $q(1-p)^m$ by the union bound. By picking $m = O\left(\frac{\ln\left(\frac{q}{1-\delta}\right)}{\ln\left(\frac{1}{1-p}\right)}\right)$ we then can ensure that the probability that we are wrong is less than $1 - \delta$.

We have been relatively brief in the above description on the naive decoder because the decoder is too simple to spend much space on. In practice, we certainly cannot assume that a decoder will accept to decrypt any signal, even if it could. For example, against the CS-code the pirate decoder can employ the following strategy: it does not decrypt any weight-1 signal where the 1 is in the position of a traitor leaf node. Under this strategy, the CS-Code is not blackbox-traceable, unless with error probability greater the 1/2 because no tracing algorithm can distinguish between a traitor (a leaf node) and its sibling in the full binary tree. Note that the sibling may very well be a non-traitor. The CS-Code cannot deal with this type of pirate's strategy because the code has a rigid structure where each position in the code plays

a specific role and corresponds to a subset of users of different sizes. For probabilistic constructions of codes where all the code positions have the same role, the strategy of refusing to decrypt some position has no significant impact on the tracing algorithm. Our probabilistic constructions, described in the next sections, can deal with the above pirate's strategy against CS-Code for that reason.

However, the pirate's strategy can certainly be smarter than rejecting some position(s) of the code. For example, for a probabilistic code where the codewords are chosen independently from the same distribution and all positions play the same role, a non-trivial pirate can estimate the (Hamming) weight of signals used in broadcast encryption and can refuse to decrypt a ciphertext that correspond to a signal containing too few or too many 1s. This strategy of pirate, called the "weight-limited pirate," is formalized as follow:

Definition 22 (Weight-Limited Decoder). A **Weight-Limited Decoder** is a decoder that only decrypts signals c with Hamming weight in an interval $[a, b]$: $\mathcal{Q}(T, c) := (\mathbf{w}_H(c) \in [a, b])$.

If the tracing algorithm works for a weight-limited decoder with interval $[a, a]$, then it *a fortiori* works for a weight-limited decoder with interval $[u, v]$, for any $u \leq a \leq v$. Therefore, the most interesting case is a singleton interval. We will denote $\mathcal{Q}_a$ the Weight-Limited Strategy $\mathcal{Q}$ for the singleton interval $[a, a]$. The following simple proposition should help clarify the concepts of weight-limited decoder and blackbox efficiently traceable codes.

Proposition 23. *The column set of a t-disjunct $\ell \times N$ matrix is a 1-conjunction code which is $(\mathcal{Q}_1, p, \delta)$-blackbox efficiently t-traceable code, where the number of queries the tracer issues to the decoder is $O\left(\ell \frac{\ln\left(\frac{\ell}{1-\delta}\right)}{\ln\left(\frac{1}{1-p}\right)}\right)$.*

Proof. We issue ℓ different weight-1 queries (with repetitions) to a $(\mathcal{Q}_1, p)$-effective decoder $\mathbb{D}$. From Lemma 21, the total number of queries issued is $O\left(\ell \frac{\ln\left(\frac{\ell}{1-\delta}\right)}{\ln\left(\frac{1}{1-p}\right)}\right)$. From the results of these queries, we will be able to recover a useful feasible word w which is the boolean union of all the traitors participated in constructing the decoder. From Proposition 15, we can identify all the traitors in $O(\ell N)$ time.

4 Trace&Revoke Codes

4.1 1-Conjunction Trace&Revoke Codes and (r, s)-disjunct matrices

Sections 2 and 3 defined and presented basic examples of revocable and traceable codes. This section defines *trace&revoke codes* which are capable of both revoking users and tracing traitors. Roughly, in a trace and revoke code, one can still trace traitors even after revoking a set of users. The codes we discuss from this section on will be 1-conjunction codes.

As usual, we interchangeably think of $w \in \{0, 1\}^\ell$ as a subset of $[\ell]$. In particular, for any position $j \in [\ell]$, we write $j \in w$ iff $w_j = 1$ and $j \notin w$ otherwise. Let Γ be an (ℓ, N)-code. For any subset $P \subset [\ell]$, let $\Gamma_P \subseteq \Sigma^P$ denote the restrictions of all codewords in Γ onto positions in P; namely $\Gamma_P = \{w|_P \ : \ w \in \Gamma\}$, where $w|_P$ denotes the projection of w onto positions in P.

Let $R \subseteq \Gamma$ be any set of codewords, then $P(R)$ denotes the positions $i \in [\ell]$ for which $w_i = 1$ for some $w \in R$. Let $\bar{P}(R) = [\ell] - R$. Then, define $\Gamma_{\bar{R}} = (\Gamma - R)_{\bar{P}(R)}$. In words, $\Gamma_{\bar{R}}$ is the set of all codewords not in R restricted to the positions not in $P(R)$. We can think of $\Gamma_{\bar{R}}$ as Γ with R being "modded out."

Definition 24 (1-Conjunction Trace&Revoke Code). An (ℓ, N)-code Γ is called an (efficient) 1-conjunction (r, s, t)-trace&revoke if:

1. Γ is a 1-conjunction (efficiently) (r, s)-revocable code;

2. For any subset $R \in \Gamma$ of at most r codewords, the code $\Gamma_{\bar{R}}$ is a (efficiently) t-traceable code.

The main intuition behind the definition should be clear: after revoking users in R, the remaining codewords form a t-traceable code so that we can trace after revoke. But how do we construct (r, s, t)-trace&revoke codes? We will impose an extra constraint on disjunct matrices to add revoking capability to disjunct matrices.

In Section 3.2, we motivated the use of disjunct matrix for tracing. However, general disjunct matrices do not necessarily have efficient revocation capability. For example, the identity matrix is r-disjunct for any r, but it would represent a horrible 1-conjunction revoke code because a broadcast signal must have weight $\Omega(N)$ leading to large broadcast messages.

This section introduces "(r, s)-disjunct matrices" that retain the traceability of disjunct matrices yet also attain efficient revocation capability. Constructions of good (r, s)-disjunct matrices are presented in Section 5.

Let $R \subseteq [N]$ be a non-empty subset of columns of a binary matrix $\mathbf{M}$ with ℓ rows and N columns. A row set $I \subseteq [\ell]$ is said to *eliminate* R if the union of the rows in I is precisely $[N] - R$.

Definition 25 ((r, s)-disjunct). An $\ell \times N$ matrix $\mathbf{M}$ is said to be (r, s)-disjunct if it satisfies the following property. Given an arbitrary set R of up to r columns of $\mathbf{M}$, there is a set $I \subseteq [\ell]$ of at most s rows which eliminates R.

Finally, we show how the new notion of disjunct matrices leads to 1-conjunction trace and revoke codes.

Theorem 26. *For any $(r+t, s)$-disjunct matrix $\mathbf{M}$ with ℓ rows and N columns. Then, the set of columns of $\mathbf{M}$ forms a 1-conjunction (r, s, t)-trace&revoke code.*

Proof. Let Γ denote the set of columns of $\mathbf{M}$. Let R be any subset of columns of $\mathbf{M}$ with size $|R| \leq r < r + t$. Then, there is a subset $I \subseteq [\ell]$ of at most s rows that eliminates R because $\mathbf{M}$ is $(r + t, s)$-disjunct. Let $c \in \{0, 1\}^\ell$ be the characteristic vector of I, i.e. $c_i = 1$ for $i \in I$ and $c_i = 0$ for $i \in [\ell] - I$. Then, c is a signal for which $\mathsf{D}_1(c, w) = 0$ for all $w \in R$ and $\mathsf{D}_1(c, w) = 1$ for all $w \notin R$. Furthermore, the Hamming weight of c is at most s. Consequently, Γ is an (r, s)-revocable code.

Next, we show that for any set R of at most r columns of $\mathbf{M}$, $\Gamma_{\bar{R}}$ is t-traceable. Let $\mathbf{M}_{\bar{R}}$ denote the matrix obtained from $\mathbf{M}$ by removing all columns in R and all rows in $P(R)$. Then, by Corollary 16 it is sufficient to show that $\mathbf{M}_{\bar{R}}$ is t-disjunct. Let T be an arbitrary set of at most t columns of $\mathbf{M}_{\bar{R}}$, which by extension is also a set of at most t columns of $\mathbf{M}$. Let w be a column of $\mathbf{M}_{\bar{R}}$ not in T. If w – as a column of $\mathbf{M}_{\bar{R}}$ is contained in the union of columns in T, then w – as a column of $\mathbf{M}$ – is contained in the union of all columns in $R \cup T$ because all positions in $P(R)$ are covered by columns in R. This means $\mathbf{M}$ is not $(r + t)$-disjunct; and in particular $\mathbf{M}$ is not $(r + t, s)$-disjunct, a contradiction!

4.2 Trace&revoke schemes from 1-conjunction blackbox trace&revoke codes

Finally, we incorporate the notion of blackbox tracing and pirate strategy $\mathcal{Q}$ into the code definition.

Definition 27 (Black-box 1-Conjunction Trace&Revoke Code). An (ℓ, N)-code Γ is $(r, s, \mathcal{Q}, p, \delta, t)$-*blackbox (efficient) trace&revoke* if

1. Γ is a 1-conjunction (efficient) (r, s)-revocable code, and
2. For any set R of at most r codewords, the code $\Gamma_{\bar{R}}$ is a 1-conjunction $(\mathcal{Q}, p, \delta)$-blackbox (efficiently) t-traceable code.

Given such a blackbox trace and revoke code, we can transform it into a trace and revoke system in a similar fashion to the the one from a revocable code to a broadcast encryption in Definition 10. The details (definition of a trace&revoke scheme and the transformation) can be found in Appendix A.1 (Definitions 37 and 41).

5 Constructions of and tracing with (r, s)-disjunct matrices

We have shown that (r, s)-disjunct matrices give rise to trace and revoke codes. In this section we first show how to construct (r, s)-disjunct matrices probabilistically. Then, we use these matrices to construct blackbox trace and revoke codes.

5.1 Constructions of 1-Conjunction (r, s, t)-trace&revoke codes

A distribution of binary matrices. Let N, b, n be arbitrary positive integers. Let $\ell = nb$, and $\mathcal{M}(N, b, n)$ denote the distribution of $\ell \times N$ binary matrices generated as follows. Partition the set $[\ell]$ into n *parts*, each part has b "bins." The parts are $P_1 = \{1, \cdots, b\}$, $\cdots$, $P_n = \{(n-1)b+1, \cdots, nb\}$.

To generate a matrix $\mathbf{M} \in \mathcal{M}(N, b, n)$ with $\ell = bn$ rows and N columns, we generate columns of $\mathbf{M}$ independently in the following way. Each column of $\mathbf{M}$, viewed as a subset of $[\ell]$, is chosen by picking uniformly (with probability $1/b$) exactly one bin from each part. In particular, each column of $\mathbf{M}$ has exactly n elements.

We can think of each column as a "ball," and each part is a collection of b bins. The distribution $\mathcal{M}(N, b, n)$ is defined by throwing N balls to b bins belonging to a part, and repeat that experiment n times, one for each part. This type of matrix distribution is used in constructing compressed sensing matrices. The resulting random matrix can also be thought of as the incidence matrix of concatenating a random code of length n with the identity code [NPR12].

Construction of (r, s)-disjunct matrices. Given two integer parameters $1 \le r < N$, our goal is to (randomly) construct a $\ell \times N$ binary matrix $\mathbf{M}$ which is (r, s)-disjunct with s as small as possible. The idea is to choose a matrix $\mathbf{M}$ at random from $\mathcal{M}(N, b, n)$ with suitably chosen parameters n and b, and show that $\mathbf{M}$ is (r, s)-disjunct with high probability.

Theorem 28. *Let $1 \le r < N$ be given positive integers. Let z, b, n be positive integers such that $r < b$ and $z \mid n$. Let $\mathbf{M}$ be a matrix chosen from the distribution $\mathcal{M}(N, b, n)$. (Recall that $\mathbf{M}$ has $\ell = nb$ rows and N columns. And, each column of $\mathbf{M}$ has weight exactly n.) Then, with probability at least*

$$1 - \left(\frac{Ne}{r}\right)^r N^{n/z}(r/b)^n$$

the matrix $\mathbf{M}$ satisfies both of the following conditions:

(i) let R be an arbitrary set of at most r columns of $\mathbf{M}$. Then there is a set I of rows which eliminates R, where $|I| \le zb$ and $I \subseteq \{zb(i-1)+1, \cdots, zbi\}$ for some $i \in \{1, \cdots, n/z\}$. In particular, $\mathbf{M}$ is (r, zb)-disjunct.
(ii) finding I given R takes time at most $O(\ell N)$.

Proof. Recall that $[\ell]$ is partitioned into n parts, each part has b "bins": $P_j = \{(j-1)b+1, \cdots, jb\}$, $j \in [n]$. Fix a set R of at most r columns of $\mathbf{M}$. Let $\overline{R} \subseteq [\ell]$ be the union of columns in R. Define

$$I = P_1 \cup \cdots \cup P_z \setminus \overline{R}.$$

In other words, I is the set of bins in the first z parts which contain none of the columns in R. In each of the first z parts, the columns in R can be in at most r bins. Hence, $z(b-r) \le |I| \le zb$.

We bound the probability that I does not eliminate R, which happens if some column in $[N] - R$ belongs to no bin in I. A fixed column in $[N] - R$ belongs to no bin in I with probability at most $(r/b)^z$. Hence, by the union bound the probability that some column in $[N] - R$ belongs to no bin in I is at most $(N-r)(r/b)^z < N(r/b)^z$. In other words, I does not eliminate R with probability at most $N(r/b)^z$.

Now, if we define I to be

$$I = P_{z+1} \cup \cdots \cup P_{2z} \setminus \overline{R},$$

then by the same reasoning the probability that I does not eliminate R is also at most $N(r/b)^z$. The same conclusion holds for the next group of z parts, and so forth. Since n parts can be partitioned in to n/z groups of z parts each, and they are all independent, the probability that R cannot be eliminated by any one of these I is at most $(N(r/b)^z)^{n/z} = N^{n/z}(r/b)^n$.

Finally, by the union bound over all choices of R (including $R = \emptyset$) we conclude that $\mathbf{M}$ does not satisfy property (i) with probability at most

$$\sum_{j=0}^{r} \binom{N}{j} N^{n/z}(r/b)^n \leq \left(\frac{Ne}{r}\right)^r N^{n/z}(r/b)^n.$$

Property (ii) follows straightforwardly from the above analysis, because we can simply check each block of z consecutive parts, one by one, and verify if I satisfies the desired property. □

Corollary 29 (Concrete parameters for an (r, s, t)-trace&revoke code). *For any $1 \leq r + t < N$, there exists an efficient 1-conjunction (r, s, t)-trace&revoke (ℓ, N)-code of length $\ell = 2(2(r + t)^2 + r + t)(\log_2 N + 1)$, where $s = (4r + 4t + 2)(\log_2 N + 1)$.*

The above corollary was obtain from Theorem 26 and Theorem 28 by setting $n = (r+t)\log_2(N^2e/(r+t))$, $b = 2(r + t) + 1$, and $z = n/(r + t)$. However, the corollary only shows the existence of such codes, it does not give an efficient strategy for constructing such codes. There are several directions one can take.

- Deterministically, in exponential time we can easily construct a matrix satisfying all conditions in the theorem with the parameters in the corollary because the theorem shows that such a matrix exists.
- Probabilistically, by slightly worsen some parameters, the theorem implies that we can construct probabilistically a good (r, s)-disjunct matrix with overwhelmingly large probability. For example, by setting $n = (r+t)\log_2(N^2e/(r+t))$, $b = 4(r+t)$, and $z = n/(r+t)$, the probability that a random matrix from $\mathcal{M}(N, b, n)$ satisfies all properties in the theorem is at least $1-\left(\frac{(r+t)}{N^2e}\right)^{r+t}$. In this case, we obtain with extremely high probability an efficient 1-conjunction $(r+t, 8(r+t)(\log_2 N+1))$-revocable code of size N and length $\ell = 8(r + t)^2(\log_2 N + 1)$.
- If desired, we can obtain a deterministic construction with a Las Vegas algorithm by repeating the above experiment independently multiple times. In each iteration, we can check in time $O(N^{O((r+t))})$ whether the randomly selected matrix satisfies the properties.

5.2 Combinatorial Group Testing with Prescribed-Weight Tests

From Corollary 29, we knows how to construct a trace and revoke code where after revoking at most r users R, we can identify at least one out of t traitors if we have access to a useful feasible word of the code $\Gamma_{\bar{R}}$. The identification of a useful feasible word, of course, depends intimately on the anti-tracing strategy of the pirate decoder. In this section, we develop the *shadow group testing* technique for identifying a useful feasible word when the pirate decoder is a limited-weight decoder (recall Definition 22). As we have observed earlier, it is sufficient to deal with constant-weight decoder strategy $\mathcal{Q}_a$. To describe the shadow group testing technique, we first derive some results regarding group testing where all pools have the same given size.

The non-adaptive case. Given positive integers r, z, N where $r + z \leq N$, the following gives upper and lower bounds the optimal number of non-adaptive tests for identifying $\leq r$ unknown items from the population of N items, where each test must have weight exactly z. Furthermore, the proof also presents two methods for constructing the tests achieving the upper-bound, one deterministic and the other randomized.

Theorem 30. *Let $\mathbf{M}$ be an $\ell \times N$ matrix which is r-disjunct with a uniform row weight of z. Then,*

$$\ell \geq \frac{N - r}{z} \cdot e^{zr/N}. \tag{1}$$

Given parameters $r + z \leq N$, there is a deterministic algorithm which constructs a $\ell \times N$ matrix which is r-disjunct with a uniform row weight of z, where

$$\ell = \frac{N-r}{z} e^{\frac{rz}{N-r-z+1}} \left(1 + r + \ln z + r \ln\left(\frac{(N-z)}{r}\right)\right). \tag{2}$$

Furthermore, by choosing weight-z rows uniformly at random, we can also construct such a matrix with success probability $\geq 1 - \epsilon$, for any given $\epsilon \in (0,1)$, where

$$\ell = O\left(\frac{N}{z} r \ln\left(\frac{N}{\epsilon r}\right) e^{\frac{rz}{N-r-z+1}}\right). \tag{3}$$

Proof. Let X be the set of all pairs (j, A) where $A \subset [N]$ is an r-subset of $[N]$, and $j \in [N] - A$. Note that $|X| = \binom{N}{r}(N-r) = N\binom{N-1}{r}$.

To prove the lower bound (1), fix an $\ell \times N$ r-disjunct matrix $\mathbf{M} = (m_{ij})$ with constant row weight z. A row i of $\mathbf{M}$ is said to *mask* a member $(j, A) \in X$ if row i "intersects" column j (i.e. $m_{ij} = 1$) and does not intersect *any* column $j' \in A$ (i.e. $m_{ij'} = 0, \forall j' \in A$). In order for $\mathbf{M} = (m_{ij})$ to be r-disjunct, every member of X must be masked by some row. A weight-z row masks exactly $z\binom{N-z}{r}$ members of X. Thus,

$$\begin{aligned}
\ell &\geq \frac{(N-r)\binom{N}{r}}{z\binom{N-z}{r}} = \frac{N-r}{z} \cdot \frac{N}{N-z} \cdot \frac{N-1}{N-z-1} \cdots \frac{N-r+1}{N-z-r+1} \\
&> \frac{N-r}{z} \cdot \left(\frac{N}{N-z}\right)^r = \frac{N-r}{z} \cdot \frac{1}{(1-z/N)^r} > \frac{N-r}{z} \cdot e^{zr/N}
\end{aligned}$$

We next derive an upperbound, we present two methods of constructing constant row-weight disjunct matrices, one deterministic and one probabilistic.

The deterministic construction works by casting the problem as an instance of the SET COVER problem and using the well-known greedy algorithm for SET COVER. The construction problem can be viewed as a set cover instance as follows. The universe to be covered is X. Each "set" is represented by a weight-z row $\mathbf{s}$. The elements in the universe that belong to the set $\mathbf{s}$ are precisely the members of X which $\mathbf{s}$ masks. Thus, each "set" $\mathbf{s}$ contains exactly $z\binom{N-z}{r}$ elements. A member $(j, A) \in X$ is covered by exactly $\binom{N-r-1}{z-1}$ sets. A classic result by Lovasz [Lov75] (and independently by Chvatal [Chv79]) implies that the greedy algorithm finds a set cover for X of size at most

$$\begin{aligned}
\ell &\leq \frac{\binom{N}{z}}{\binom{N-r-1}{z-1}} \left(1 + \ln\left(z\binom{N-z}{r}\right)\right) \\
&= \frac{N-r}{z} \cdot \frac{N}{N-r} \cdot \frac{N-1}{N-r-1} \cdots \frac{N-z+1}{N-r-z+1} \left(1 + \ln\left(z\binom{N-z}{r}\right)\right) \\
&< \frac{N-r}{z} \left(\frac{N-z+1}{N-r-z+1}\right)^z \left(1 + \ln z + r \ln\left(\frac{e(N-z)}{r}\right)\right) \\
&= \frac{N-r}{z} \left(1 + \frac{r}{N-r-z+1}\right)^z \left(1 + r + \ln z + r \ln\left(\frac{(N-z)}{r}\right)\right) \\
&< \frac{N-r}{z} e^{\frac{rz}{N-r-z+1}} \left(1 + r + \ln z + r \ln\left(\frac{(N-z)}{r}\right)\right).
\end{aligned}$$

This fact can also be seen from the *dual-fitting* analysis of the greedy algorithm for SET COVER. This set cover is exactly the set of rows of the r-disjunct matrix we are looking for. The final expression might seem a little unwieldy. Note, however, that compared to the lower bound (1), we are only off by a factor of about $O(r \ln(N/r))$. For most meaningful ranges of z and r, the factor $\left(1 + r + \ln z + r \ln\left(\frac{(N-z)}{r}\right)\right)$ can safely be thought of as $O(r \ln(N/r))$. Last but not least, if $rz = O(N)$ then $e^{\frac{rz}{N-r-z+1}} = O(1)$ and the number of rows l is not exponential. Also, when $rz = \Theta(N)$ the overall cost is $l = O(r^2 \log(N/r))$,

matching the best known bound for disjunct matrices. This optimality only applies when we are free to choose z in terms of N and r; in particular, when we have this freedom we will pick $z = \Theta(N/r)$.

The probabilistic construction works as follows. We think of members of X as "coupons" and the weight-z row vectors as boxes of coupons. Each box has precisely $z\binom{N-z}{r}$ different coupons in it. We want to collect as few boxes as possible to have a complete coupon collection. Let's pick the boxes uniformly at random, one by one. The probability that a given coupon is chosen in each round is $\frac{\binom{N-r-1}{z-1}}{\binom{N}{z}}$. Hence, by the union bound, after ℓ rounds the probability that at least one coupon is not collected is at most

$$|X|\left(1-\frac{\binom{N-r-1}{z-1}}{\binom{N}{z}}\right)^{\ell} \leq N\binom{N-1}{r}e^{-\ell\frac{\binom{N-r-1}{z-1}}{\binom{N}{z}}}.$$

This is an upper bound on our failure probability. If we want a guarantee of at most $\epsilon < 1$ failure probability, then we can simply choose ℓ such that

$$N\binom{N-1}{r}e^{-t\frac{\binom{N-r-1}{z-1}}{\binom{N}{z}}} \leq \epsilon.$$

Similar to the above analysis,

$$\ell = O\left(\frac{N}{z}r\ln\left(\frac{N}{\epsilon r}\right)e^{\frac{rz}{N-r-z+1}}\right)$$

is sufficient. □

The adaptive case. One might hope that adaptive tests may help overcome the $\tilde{\Omega}(e^{rz/N})$-barrier. Unfortunately, such is not the case.

Theorem 31. *Any adaptive group testing strategies for a population of N items with less than r positives in which each test has weight z must use $\Omega(e^{rz/N})$ tests. Furthermore, there exists a randomized testing strategy which uses, in expectation, $e^{rz/(N-r)} + N - z$ tests.*

Proof. We will show that if there are $\ell < \frac{1}{2}e^{rz/N}$ adaptive tests, then there are at least two different sets of positive items which give identical (adaptive) test results and thus the adaptive tests cannot distinguish between these two sets of positive items.

Each adaptive test is a z-subset $F_i \subseteq [N]$. Consider any sequence of ℓ adaptive tests $F_1, \cdots, F_\ell$, where $\ell < \frac{1}{2}e^{rz/N}$. We will show that there are two different r-sets of items S and T such that S intersects all of the F_i and T also intersects all of the F_i. Thus, if S were the set of positives then all of the tests return positive. And, if T were the set of positives then all of the tests also return positive. Consequently, ℓ tests are not sufficient to distinguish between S and T. (Remark: to be a little more rigorous, we could model the adaptive test strategy as a binary tree, where each node represents a z-subset, and the two children of a node correspond to whether a test turns positive or not. Here, the sequence $F_1, \cdots, F_\ell$ corresponds to the all-positive branch of the tree.)

We use the probabilistic method. We pick a subset S of size r of $[N]$ uniformly at random, and show that the probability that S intersects all of the F_i is at least $2/\binom{N}{z}$, which would establish the claim.

For a fixed F_i, we have

$$\Pr[S \cap F_i = \emptyset] = \frac{\binom{N-z}{r}}{\binom{N}{r}} = \frac{N-z}{N}\cdot\frac{N-z-1}{N-1}\cdots\frac{N-z-r+1}{N-r+1} \leq \left(\frac{N-z}{N}\right)^r < e^{-zr/N}.$$

By the union bound,

$$\Pr\left[S \cap F_i = \emptyset, \text{ for some } i\right] < \ell \cdot e^{-rz/N} \leq \frac{1}{2}.$$

Thus,

$$\Pr\left[S \cap F_i \neq \emptyset, \text{ for all } i\right] > \frac{1}{2} \geq \frac{2}{\binom{N}{r}}.$$

For the upper-bound, we can pick a random test T (of size z) until the test returns negative. Let S be the set of positive items, then

$$\Pr[T \text{ returns negative}] = \Pr[S \cap T = \emptyset] \geq \frac{\binom{N-r}{z}}{\binom{N}{z}}.$$

Hence, the expected number of random tests we need is at most

$$\frac{\binom{N}{z}}{\binom{N-r}{z}} = \frac{N}{N-r} \cdots \frac{N-z+1}{N-r-z+1} \leq \left(1 + \frac{r}{N-r}\right)^z \leq e^{zr/(N-r)}.$$

After the negative test T is obtained, we can then construct new tests by replacing a negative item in T with an item outside of it one at a time until we found all of the positive items. □

5.3 The shadow group testing technique and 1-Conjunction $\mathcal{Q}_a$-blackbox trace&revoke codes

When the pirate decoder only answers queries with weight a (this is the meaning of the $\mathcal{Q}_a$ qualified signals), we will only use weight-a queries to trace because other signals do not give reliable answers. The following theorem explains how we can trace such pirate decoders.

Theorem 32 ($\mathcal{Q}_a$ tracing with shadow group tests). *Let $1 \leq t < N$ and ℓ be integers, and let Γ be the 1-conjunction code formed by the columns of a t-disjunct matrix with ℓ rows and N columns. Additionally, suppose the union of any t codewords has weight at most D. Then, the code Γ is also a $(\mathcal{Q}_a, 1, 1, t)$-traceable code where the number of queries the tracer issues to the decoder is at most $O\left(\frac{\ell}{a} D \log\left(\frac{\ell}{t}\right) e^{\frac{Da}{\ell - D - a + 1}}\right)$.*

Proof. Thanks to Proposition 15, instead of identifying a traitor, we can just identify a vector $w \in \{0,1\}^\ell$ that is contained in the union of all traitors' codewords and that contains at least one traitor's codeword. Such a vector is in the useful feasible set (see Proposition 12) of the code.

Identifying $w = (w_1, \cdots, w_\ell)$ is equivalent to identifying all the coordinates i of w for which $w_i = 1$. Thus, there is a subset $U \subseteq [\ell]$ of at most D unknown coordinates that we want to identify. We need to query the pirate decoder with weight-a signals c to identify U. Each query c is the characteristic vector of a subset of size a of $[\ell]$. So we think of each query as an a-subset A of $[\ell]$. The decoder is able to decrypt query A if and only if there is at least one traitor whose codeword intersects A. In other words, each query A is a *group test* for the "positives" U in the population $[\ell]$. We thus have a group testing problem "inside" another group testing problem. We refer to the "inner" group tests as the *shadow tests*, because they are not used to identify the traitors directly; rather, they are used to identify the shadow U of the traitors.

Finally, we directly apply Theorem 30 and use the non-adaptive (but deterministic) construction in the theorem (equation (2)) to attain the desired number of queries. □

Now, we incorporate the shadow group testing technique to construct blackbox trace and revoke codes. From Corollary 29 and the discussion after that we know how to construct a 1-conjunction (r, s, t)-trace&revoke code Γ. This code can easily be turned into a trace&revoke system as described in Section 4.2. Let R be any set of at most r traitors, we know that $\Gamma_{\bar{R}}$ is a 1-conjunction t-traceable code. In fact, from the construction of the $(r+t, s)$-disjunct matrix that leads to the (r, s, t)-trace&revoke code, we know that the Hamming weight of each codeword is $n = O((r+t)\log(N/(r+t)))$; hence, the Hamming weight of the union of at most t columns is $D = tn = O(t(r+t)\log(N/(r+t)))$. Now, if the pirate decoder applies the $\mathcal{Q}_a$ anti-tracing strategy, then we will only trace with weight-a signals using the shadow group test technique from Theorem 32. The number of queries the tracer needs is going to be $q = O\left(\frac{\ell}{a} D \log\left(\frac{\ell}{t}\right) e^{\frac{Da}{\ell - D - a + 1}}\right)$.

The question is, which value of weight a makes the most sense to the pirate? The answer to this question depends on how we broadcast after revoking the users in R. Looking back at the proof of Theorem 28, we chose a subset I that eliminates R where the size of I is at most bz. In fact, once I eliminates R we can add more elements to I so that $|I| = bz$. We use the set I to construct a broadcast signal c; I is the support set of c. Hence, all broadcast signals will have weight $s = bz$ (before or after revoking R). In other words, it will only make sense for the pirate decoder with the anti-tracing strategy $\mathcal{Q}_a$ to set $a = s = \Theta((r+t)\log(N/(r+t)))$. Also recall that $\ell = \Theta((r+t)^2\log(N/(r+t)))$. Hence, $\ell/a = O(r+t)$, and $e^{\frac{Da}{\ell-D-a+1}} = O((N/(r+t))^{O(t)})$. Combining with Lemma 21, we obtain the following corollary.

Corollary 33. *For any $1 \leq r+t < N$, there exists an $(r, s, \mathcal{Q}_s, p, \delta, t)$-blackbox trace&revoke code Γ, where $s = O((r+t)\log N)$ is the weight of broadcast signals, and the number of queries issued to the pirate decoder is $O\left(q \cdot \frac{\ln\left(\frac{q}{1-\delta}\right)}{\ln\left(\frac{1}{1-p}\right)}\right)$, where*

$$q = O\left(t(t+r)^2\log(N/(r+t))\log\left(\frac{(r+t)^2\log((N/(r+t))}{t}\right)\cdot\left(\frac{N}{r+t}\right)^{O(t)}\right).$$

5.4 Toward Traceability Against Arbitrary Pirate Decoders

We now discuss the case of arbitrary pirate decoders. Our goal is to be able to deal with any qualified predicate $\mathcal{Q}$. We propose a slightly modified version of the main scheme described in Corollary 33 for this aim. However, we have to introduce an additional assumption: the pirate decoder should decrypt (with non-negligible probability) broadcast signals that correspond to the case of non-revocation, i.e., when $R = \emptyset$. The tracing process in our Trace&Revoke scheme is thus not in the standard model but it seems still practical as it requires the pirate decoder to be able to decrypt ciphertexts in the most usual case (at least in some applications) where there is no detected malicious users in the system. Anyway, our scheme satisfies both the definition of a traitor tracing scheme (in which the revoked set is by definition an empty set) for arbitrary pirate decoders and the definition of a revoke system (in which the scheme can revoke users and does not require a tracing procedure).

As shown in Theorem 30, we can ask random queries to the pirate decoder, the number of queries is asymptotically similar to the deterministic case. Each random query consists of a set of $\beta 4r$ rows randomly chosen from ℓ rows of the matrix $\mathbf{M}$. In our code based trace&revoke scheme (Definition 41 in Appendix A.1), the number of chosen rows in a ciphertext (the weight of the signal c) varies from $4r$ (when there is no revoked users, $4r$ rows from any block cover all users) to $8r\log(N/r)$ (when there are r revoked users).

Assume the pirate knows our strategy of choosing rows in the encapsulation and *if* it can detect whether a query – though corresponding to an encapsulation for some R – cannot be an output of the $\mathsf{Encaps}(\mathsf{EK}, R)$, then it will not decrypt. The key point here is to issue (random) tracing queries so that it is computationally hard for the pirate to distinguish between a given tracing query and an output of $\mathsf{Encaps}(\mathsf{EK}, \emptyset)$ (i.e. broadcast signals in the non-revoke mode).

We now described the modified scheme. To impose the computational hardness on the pirate, we first permute randomly the rows of the matrix $\mathbf{M}$ to obtain a matrix $\mathbf{M}^*$. We will use the matrix $\mathbf{M}^*$ as the code, instead of $\mathbf{M}$, as the $(\ell, N)-$code in our encryption in Definition 41 (appendix A.1). We also slightly change the $\mathsf{Encaps}(\mathsf{EK}, R)$ in Definition 41 when there is no revoked users, *i.e.,* $R = \emptyset$. In this case, $4r$ rows from a block cover all users and the $\mathsf{Encaps}(\mathsf{EK}, \emptyset)$ procedure will normally use these rows for encryption. However, in this case, we will add $(\beta - 1)4r$ more random rows and let $\mathsf{Encaps}(\mathsf{EK}, \emptyset)$ procedure uses the total $\beta 4r$ rows for encryption. Note that the $(\beta-1)4r$ additional rows have no impact on the validity of the broadcast because the original $4r$ rows already covered all users.

We argue that no pirate decoder can distinguish a broadcast signal and a signal in the tracing procedure. In fact, in order to distinguish between a broadcast signal constructed this way and a random

set of $\beta 4r$ rows, the pirate must be able to tell whether a given set A of $\beta 4r$ rows contains a subset $B \subseteq A$ of $4r$ rows which might come from the same row block. We next argue that, even if the pirate has as many as $b = 4r = \omega(t)$ codewords, it is computationally hard to detect whether a given set of $\beta 4r$ rows contains $4r$ rows from the same block.

Consider the case where the matrix $\mathbf{M}^*$ is not public and the pirate decoder is resettable, thus can be reset to the initial state after each query. The pirate has a set T of $b = 4r$ codewords (each codeword is a column of the matrix $\mathbf{M}^* = (m_{ij})$). Each row i in the set A of βb rows of the random query corresponds to a subset of the set T: the subset of traitors j in T for which $m_{ij} = 1$. The problem of detecting whether there are b rows of A belonging to the same block thus becomes precisely an instance of the **NP**-hard EXACT COVER problem [GJ79] which is like SET COVER but each element in the universe is to be covered exactly once. To cover b columns we do not need more than b rows, and hence a solution to the EXACT COVER problem with additional empty rows will be a solution to the pirate's problem. Conversely, a solution to the pirate's problem obviously is a solution to the EXACT COVER instance. It is not hard to derive a reduction from 3-SAT to EXACT COVER. (See, e.g., [BM08].) Furthermore, if we start from 3-SAT-5 (known to be **NP**-hard [Fei98]) where each clause has at most 3 literals and each literals appear in at most 5 clauses, then we obtain instances of EXACT COVER where the number of sets is bounded by a constant (β in our case) times the size of the universe. Therefore, there is no known algorithm in polynomial time of $b = 4r$ that can solve this problem, even when β is a constant (about 15 or so).

We should however notice that our assumption requires in fact the average case hardness. On the other hand, we also notice that the pirate possesses only t users' keys but not b keys, and $b = 4r = \omega(t)$. Therefore, we believe that our assumption is reasonable.

5.5 Trace&Revoke in the Information-Theoretic Limit

In the tracing procedure of our scheme, as well as in the tracing procedures of almost all known schemes in the literature, we rely on the fact that the pirate accepts to decrypt the tracing queries, as it is hard to distinguish a query in tracing mode and a normal ciphertext. A natural question is, without any computational assumption about the pirate, can we still trace?

To be concrete, assume the pirate can always verify whether a query comes from $\mathsf{Encaps}(\mathsf{EK}, R)$, for some R, and that the pirate decoder only decrypts if and only if this is indeed the case. We can prove that, for this powerful pirate, the number of queries can no longer be $\mathrm{poly}(N)$ for most reasonable values of r and t. In particular, with $q \leq \binom{N}{t}(1 - \frac{t}{N})/\binom{r}{t}$ queries it is not possible to always identify correctly at least one traitor. Moreover, in this model of pirate, we present a randomized tracing strategy that matches this bound in expectation. Note that our lower-bounds apply to any Trace&Revoke systems given the powerful pirate assumption.

Considering a trace&revoke schemes against the pirates who decrypt, if possible, a ciphertext if and only if this ciphertext is an output of the algorithm $\mathsf{Encaps}(\mathsf{EK}, R)$ for some R. We then have a lower bound of the number of queries for the tracing algorithm.

Theorem 34. *Let $t \leq r < N$ such that $t + r < N$. If we pose at most $q \leq \left(\binom{N}{t} - 2\right)/\binom{r}{t}$ queries, then it is not possible to always identify correctly the entire traitor set T. If we pose at most $q \leq \frac{\binom{N}{t}}{\binom{r}{t}}(1 - \frac{t}{N})$ queries, then it is not possible to always identify correctly at least one traitor.*

Proof. Let $c = \mathsf{Encaps}(\mathsf{EK}, R)$ for some R. Let T be the set of traitors. If $T \subseteq R$ then no traitor can decode c. If $T \setminus R \neq \emptyset$, then some traitor will be able to decode. Thus, the blackbox query can be represented by subsets $R \subseteq [N], |R| \leq r$. A query returns YES if $T \setminus R \neq \emptyset$ and NO otherwise.

Let us first consider the case when we want to identify all traitors in T, $|T| \leq t$. Suppose we pose q adaptive (!) black-box queries $F_1, \cdots, F_q$. Recall that each query can be represented by a set $F_i \subseteq [N]$, where $|F_i| \leq r$. We will prove that if $q \leq \frac{\binom{N}{t}-2}{\binom{r}{t}}$, then there are two distinct traitor sets T_1 and T_2 such that the series of queries $F_1, \cdots, F_q$ all return YES, and thus no algorithm can distinguish between T_1 and T_2.

To this end, fix *arbitrary* $F_1, \cdots, F_q$ and pick a set $T \subseteq [N]$ of size t uniformly at random. For each $i \in [q]$, $\Pr[T \subset F_i] \leq \frac{\binom{r}{t}}{\binom{N}{t}}$. Thus, by the union bound $\Pr_T[\text{some query } F_i \text{ returns NO}] \leq q\frac{\binom{r}{t}}{\binom{N}{t}}$. Put it another way,

$$\Pr_T[\text{all queries } F_i \text{ return YES}] \geq 1 - q\frac{\binom{r}{t}}{\binom{N}{t}} \geq \frac{2}{\binom{N}{t}}.$$

Thus, there exist two different sets T_1, T_2 such that all queries return YES.

Next, we consider the case when at least one traitor in T need to be identified. Our strategy is to show that, for an arbitrary query sequence $F_1, \cdots, F_q$, there are potential traitor sets $T_1, \cdots, T_k$, all of size t, satisfying the following conditions: (i) for each T_j, all queries F_i returns YES, (ii) there is no single user that belongs to all the T_j. Thus, the identification of some user in some T_j will be *wrong* for some other $T_{j'}$.

To guarantee (ii), we simply pick $k = \binom{N-1}{t-1} + 1$. Because $\binom{N-1}{t-1}$ is the maximum number of t-subsets of $[N]$ which contain a given element. To guarantee (ii), we use the probabilistic argument as above: pick T uniformly at random.

$$\Pr_T[\text{all queries } F_i \text{ return YES}] \geq 1 - q\frac{\binom{r}{t}}{\binom{N}{t}} \geq \frac{t}{N} = \frac{\binom{N-1}{t-1}}{\binom{N}{t}}.$$

The last inequality holds because $q \leq \frac{\binom{N}{t}}{\binom{r}{t}}(1 - \frac{t}{N})$. □

The following upper-bound is asymptotically as good as the above lower-bounds for most practically meaningful ranges of r and t. However, it is an upper-bound in expectation only.

Theorem 35. *Let $t \leq r < N$ such that $t + r \leq N$. There is an adaptive strategy which uses on average at most $q = \frac{\binom{N}{t}}{\binom{r}{t}} + N - r$ queries.*

Proof. We pose random "queries" R of size r to the blackbox decoder. (In reality, the queries are actually $\mathsf{Encaps}(\mathsf{EK}, R)$) until the answer is **no**. The expected number of such queries is the inverse of the probability that R contains the traitor set T, which is

$$\frac{\binom{N}{r}}{\binom{N-t}{r-t}} = \frac{N}{r}\frac{N-1}{r-1}\cdots\frac{N-t+1}{r-t+1} = \frac{\binom{N}{t}}{\binom{r}{t}}.$$

After some R containing T is found, we can test each $j \in R$ individually as follows. Fix $j' \in [N] - R$ Let $R' = R - \{j\} \cup \{j'\}$. If the answer to query R' is YES then $j \in T$. This way, we will be able to identify all members of T with an additional $N - r$ queries. □

6 Discussions

Optimization of the Code Length. Given N and $r < N$, we presented a randomized construction of $\ell \times N$ (r, s)-disjunct matrices with $s = O(r \log(N/r)$ and $\ell = O(r^2 \log(N/r)$. Note that, even for r-disjunct matrices which do not address revocation, no other known construction, including randomized ones, has asymptotically smaller number of rows. It is also known that $\ell = \Omega(r^2 \log_r N)$ for any r-disjunct matrix. Hence, if we use this matrix for broadcasting then the broadcaster's key size (or the total number of keys that will be attributed to users) is $O(r^2 \log(N/r))$. In this setting, we can trace back all "active" traitors (the traitors included in the pirate's codeword).

We can reduce the broadcaster's key size by using a related notion called *multiple user tracing* (MUT) families. Given positive integers $u \leq r$, an (r, u)-*MUT family* is a non-adaptive group testing matrix which, given the test outcomes imposed by an arbitrary set of $v \leq r$ positives, there is a decoding algorithm that outputs at least $\min(u, v)$ out of the v positives. It is known [AA07] that, given N, r, u, an $\ell \times N$

(r, u)-MUT matrix exists with $\ell = O\left((r + u^2) \log N\right)$. There is a method for constructing the MUT matrix so that even sublinear-time decoding is also reachable. Hence, we can use MUT families to design broadcast codes which can be used for tracing (assuming the naive pirate) up to $\sqrt{r}$ of the traitors while keeping the broadcaster's key size $O(r \log N)$. Note that in Complete Subtree, the broadcaster's key size is $2N - 1$.

Optimization of the Private Key Size. In the above construction, the users' private key size is linear in the weight of its associated codeword. We can optimize this, for the case of $1-$conjunction revocable code, by using Asano's method [Asa02]. The users' private key size then becomes constant. In this case, the security should also be based on a computational assumption that the RSA inversion is hard. This optimization is described in Appendix A.1 (Definition 40).

On the *a priori*-Bounds of Revoked Users and Traitors. Our scheme assumed an *a priori*-bounds r, t of revoked users and traitors. If the bound is unknown, a natural way to get around the problem is to "stack" on top of each other (r, s)-disjunct matrices for different values of r. This way, the resulting matrix will serve as (r, s)-codes for different r. The sacrifice is in code length. In the encryption mode, depending on the number of revoked users, we can use the appropriate matrix.

References

[AA07] Noga Alon and Vera Asodi. Tracing many users with almost no rate penalty. *IEEE Trans. Inform. Theory*, 53(1):437–439, 2007.

[Asa02] Tomoyuki Asano. A revocation scheme with minimal storage at receivers. In Yuliang Zheng, editor, *Advances in Cryptology – ASIACRYPT 2002*, volume 2501 of *Lecture Notes in Computer Science*, pages 433–450. Springer, December 2002.

[Ber91] Shimshon Berkovits. How to broadcast a secret (rump session). In Donald W. Davies, editor, *Advances in Cryptology – EUROCRYPT'91*, volume 547 of *Lecture Notes in Computer Science*, pages 535–541. Springer, April 1991.

[BF99] Dan Boneh and Matthew K. Franklin. An efficient public key traitor tracing scheme. In Michael J. Wiener, editor, *Advances in Cryptology – CRYPTO'99*, volume 1666 of *Lecture Notes in Computer Science*, pages 338–353. Springer, August 1999.

[BM08] J. A. Bondy and U. S. R. Murty. *Graph theory*, volume 244 of *Graduate Texts in Mathematics*. Springer, New York, 2008.

[BN08] Dan Boneh and Moni Naor. Traitor tracing with constant size ciphertext. In Peng Ning, Paul F. Syverson, and Somesh Jha, editors, *ACM CCS 08: 15th Conference on Computer and Communications Security*, pages 501–510. ACM Press, October 2008.

[BP08] Olivier Billet and Duong Hieu Phan. Efficient Traitor Tracing from Collusion Secure Codes. In Reihaneh Safavi-Naini, editor, *Information Theoretic Security—ICITS 2008*, volume 5155 of *Lecture Notes in Computer Science*, pages 171–182. Springer, 2008.

[BS95] Dan Boneh and James Shaw. Collusion-secure fingerprinting for digital data (extended abstract). In Don Coppersmith, editor, *Advances in Cryptology – CRYPTO'95*, volume 963 of *Lecture Notes in Computer Science*, pages 452–465. Springer, August 1995.

[BW06] 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: 13th Conference on Computer and Communications Security*, pages 211–220. ACM Press, October / November 2006.

[CFN94] Benny Chor, Amos Fiat, and Moni Naor. Tracing traitors. In Yvo Desmedt, editor, *Advances in Cryptology – CRYPTO'94*, volume 839 of *Lecture Notes in Computer Science*, pages 257–270. Springer, August 1994.

[Chv79] V. Chvátal. A greedy heuristic for the set-covering problem. *Math. Oper. Res.*, 4(3):233–235, 1979.

[DF02] Yevgeniy Dodis and Nelly Fazio. Public key broadcast encryption for stateless receivers. In *In Digital Rights Management 02, volume 2696 of LNCS*, pages 61–80. Springer, 2002.

[DF03] Yevgeniy Dodis and Nelly Fazio. Public key trace and revoke scheme secure against adaptive chosen ciphertext attack. In Yvo Desmedt, editor, *PKC 2003: 6th International Workshop on Theory and Practice in Public Key Cryptography*, volume 2567 of *Lecture Notes in Computer Science*, pages 100–115. Springer, January 2003.

[DH00] Ding-Zhu Du and Frank K. Hwang. *Combinatorial group testing and its applications*, volume 12 of *Series on Applied Mathematics*. World Scientific Publishing Co. Inc., River Edge, NJ, second edition, 2000.

[Dor43] R. Dorfman. The detection of defective members of large populations. *The Annals of Mathematical Statistics*, 14(4):436–440, 1943.

[EFF85] P. Erdős, P. Frankl, and Z. Füredi. Families of finite sets in which no set is covered by the union of r others. *Israel J. Math.*, 51(1-2):79–89, 1985.

[Fei98] Uriel Feige. A threshold of $\ln n$ for approximating set cover. *J. ACM*, 45(4):634–652, 1998.

[FN94] Amos Fiat and Moni Naor. Broadcast encryption. In Douglas R. Stinson, editor, *Advances in Cryptology – CRYPTO'93*, volume 773 of *Lecture Notes in Computer Science*, pages 480–491. Springer, August 1994.

[FNP07] 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: 10th International Conference on Information Security*, volume 4779 of *Lecture Notes in Computer Science*, pages 71–88. Springer, October 2007.

[GJ79] Michael R. Garey and David S. Johnson. *Computers and intractability*. W. H. Freeman and Co., San Francisco, Calif., 1979. A guide to the theory of NP-completeness, A Series of Books in the Mathematical Sciences.

[HS02] Dani Halevy and Adi Shamir. The LSD broadcast encryption scheme. In Moti Yung, editor, *Advances in Cryptology – CRYPTO 2002*, volume 2442 of *Lecture Notes in Computer Science*, pages 47–60. Springer, August 2002.

[KY02] Aggelos Kiayias and Moti Yung. Traitor tracing with constant transmission rate. In Lars R. Knudsen, editor, *Advances in Cryptology – EUROCRYPT 2002*, volume 2332 of *Lecture Notes in Computer Science*, pages 450–465. Springer, April / May 2002.

[Lov75] L. Lovász. On the ratio of optimal integral and fractional covers. *Discrete Math.*, 13(4):383–390, 1975.

[NNL01] Dalit Naor, Moni Naor, and Jeffery Lotspiech. Revocation and tracing schemes for stateless receivers. In Joe Kilian, editor, *Advances in Cryptology – CRYPTO 2001*, volume 2139 of *Lecture Notes in Computer Science*, pages 41–62. Springer, August 2001.

[NP00] Moni Naor and Benny Pinkas. Efficient trace and revoke schemes. In Yair Frankel, editor, *Financial Cryptography 2000*, volume 1962 of *Lecture Notes in Computer Science*, pages 1–20. Springer, February 2000.

[NPR12] Hung Q. Ngo, Ely Porat, and Atri Rudra. Efficiently decodable compressed sensing by list-recoverable codes and recursion. In *Proceedings of the 29th International Symposium on Theoretical Aspects of Computer Science (STACS)*, pages 230–241, 2012.

[Sir07] Thomas Sirvent. Traitor tracing scheme with constant ciphertext rate against powerful pirates. In Daniel Augot, Nicolas Sendrier, and Jean-Pierre Tillich, editors, *Workshop on Coding and Cryptography—WCC '07*, pages 379–388, April 2007.

[SSW00] J.N. Staddon, D.R. Stinson, and Ruizhong Wei. Combinatorial properties of frameproof and traceability codes. *IEEE Transactions on Information Theory*, 47:1042–1049, 2000.

[Tar03] Gábor Tardos. Optimal probabilistic fingerprint codes. In *35th Annual ACM Symposium on Theory of Computing*, pages 116–125. ACM Press, June 2003.

[TSN06] Dongvu Tonien and Reihaneh Safavi-Naini. An efficient single-key pirates tracing scheme using cover-free families. In Jianying Zhou, Moti Yung, and Feng Bao, editors, *ACNS 06: 4th International Conference on Applied Cryptography and Network Security*, volume 3989 of *Lecture Notes in Computer Science*, pages 82–97. Springer, June 2006.

A Definitions

A.1 Basic Definition

Definition 36 (Broadcast Encapsulation). A broadcast encapsulation scheme is a 3-tuple of algorithms $\mathcal{DBE} = (\mathsf{Setup}, \mathsf{Encaps}, \mathsf{Decaps})$:

- $\mathsf{Setup}(1^k, N)$, where k is the security parameter, and N the number of users, it generates the global parameters param of the system (omitted in the following); and returns a master secret key MSK and an encryption key EK. It also generates users' keys upk_i, for $i = 1, \ldots, N$.
- $\mathsf{Encaps}(\mathsf{EK}, R)$ takes as input a revoked set R and outputs a key header H and a session key $K \in \{0,1\}^k$.
- $\mathsf{Decaps}(\mathsf{usk}_i, R, H)$ takes as input the revoked set R and a user secret key. If $i \in [N] - R$, outputs the session key K.

The correctness requirement is that for any revoked set R and for any user $i \in [N] - R$ then the decapsulation algorithm gives back the ephemeral session key.

Definition 37 (Trace&Revoke Encapsulation). A trace&revoke encapsulation scheme is a broadcast encapsulation scheme with an additional tracing algorithm $\mathbf{Trace}^{\mathbb{D}}(R_{\mathbb{D}}, \mathsf{pk}, \mathsf{msk})$: the traitor tracing algorithm interacts in a black-box manner with a pirate decoder $\mathbb{D}$ that is built from a certain set T of traitors. The algorithm takes as input a subset $R_{\mathbb{D}} \subset [N]$ (could be adversarially chosen), the public key pk, the master key msk and outputs a set $T_{\mathbb{D}} \subseteq [N]$.

More precisely, under the conditions:

- there are at most t traitors: $|T| \leq t$;
- The minimal revoked set does not contain all the traitors: $T \not\subseteq R_{\mathbb{D}}$, or equivalently $S_{\mathbb{D}} = ([N] - R_{\mathbb{D}})$ contains at least a traitor;
- $\mathbb{D}$ is "efficient" to decrypt ciphertexts (*i.e.,* decrypts with some non-negligible probability) for some revoked sets R that include the minimal revoked set $R_{\mathbb{D}}$ but do not contain all the traitors ($R_{\mathbb{D}} \subseteq R$ but $T \not\subseteq R$);

then the tracing algorithm outputs at least a traitor in $S_{\mathbb{D}}$, *i.e.,* : $\emptyset \neq T_{\mathbb{D}} \subseteq T \cap S_{\mathbb{D}}$.

Definition 38 (Public-Key Encryption Scheme). $\mathcal{PKE} = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$:

- $\mathsf{Setup}(1^k)$, where k is the security parameter, generates the global parameters param of the system;
- $\mathsf{KeyGen}(\mathsf{param})$ generates a pair of keys, the public (encryption) key ek and the associated private (decryption) key dk;
- $\mathsf{Enc}(\mathsf{ek}, m; r)$ produces a ciphertext c on the input message m and the public key ek, using the random coins r (we may omit r when the notation is obvious);
- $\mathsf{Dec}(\mathsf{dk}, c)$ decrypts the ciphertext c under the private key dk. It outputs the plaintext, or $\perp$ if the ciphertext is invalid.

Definition 39 (Secret Sharing Scheme). $\mathcal{SSS} = (\mathsf{Share}, \mathsf{Combine})$:

- $\mathsf{Share}(k, m, n)$, outputs a secret bit string K of length k, as well as n shares s_1, ..., s_n, so that any m of them will allow to recover K.
- $\mathsf{Combine}(\{(i, s_i)\})$, from m pairs (i, s_i), it recovers the bit string K.

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

Definition 40 (Constant Size Private Key). Suppose there exists an algorithm that generates $1-$Conjunction $(r, s)-$Revocable $(\ell, N)-$Code.We build a BE scheme Π that can revoke up to r users in the following way.

- $\mathsf{Setup}(1^\lambda, N)$
  1. Run the Code generating algorithm on (N, r, d) to obtain a $1-$Conjunction $(r, s)-$Revocable $(\ell, N)-$Code Γ.
  2. Generate two large primes of the same size p, q and publish $M = pq$
  3. Generate ℓ pairs $(\mathsf{dk}_i, \mathsf{ek}_i), i = 1, \ldots, \ell$ such that $e_k d_k = 1(\mod (p-1)(q-1))$;
  4. Choose a random $X \xleftarrow{\$} \mathbb{Z}_M^*$
  5. Set $\mathsf{MSK} = (\Gamma, X, \{\mathsf{dk}_i\})$, $\mathsf{EK} = (N, \{\mathsf{ek}_i\})$, and $Reg = \emptyset$.
- $\mathsf{Join}(\mathsf{MSK}, i)$
  1. The user i is associated with the codeword $w^i \in \Gamma$.
  2. Set $\mathsf{usk}_i \leftarrow X^{\prod_{j=1}^{\ell} \mathsf{dk}_j^{w_j^i}}$; $\mathsf{upk}_i \leftarrow i$; $Reg \leftarrow Reg \cup \{i\}$.
- $\mathsf{Encaps}(\mathsf{EK}, R)$:
  1. The revoked set R should contain at most r users;
  2. Because Γ is $1-$conjunction $(r, s)-$revocable, one can find out an word c such that $\mathsf{D}_1(R, c) = 0$ and $\mathsf{D}_1(u, c) = 1$ for any $u \in [N] - R$, and $m = H(c) \leq d$.
  3. Denote by $i_1, \ldots, i_m$ the positions of m bits 1 in c, *i.e.,* $c_{i_j} = 1$, for $j = 1, \ldots, m$
  4. Set $e_{i_j} = X^{\mathsf{dk}_{i_j}}$, for $j = 1, \ldots, m$.
  5. Output $\mathcal{K}_e$ and $Header = (c, (e_{i_j}), j = 1, \ldots, m)$.
- $\mathsf{Decaps}(\mathsf{usk}_j, R, Header)$:
  1. If j is in $[N]-R$, then $\mathsf{D}_1(w^j, c) = 1$. There exists thus an index $1 \leq z \leq m$ such that $c_{i_z} = w_{i_z}^j = 1$
  2. Compute $s_{i_z} = \mathsf{usk}_j^{\prod_{s=1, s\neq i_z}^{\ell} \mathsf{ek}_j^{w_s^i}}$. From the s_{i_z}, reconstruct $\mathcal{K}_e$

Definition 41 (Trace&Revoke System from 1-Conjunction Trace&Revoke Codes). Let us be given a generator of $(r, s, \mathcal{Q}, p, \delta, t, \tau)$-blackbox Trace&Revoke 1-Conjunction (ℓ, N)-Codes, and a secure public-key encryption scheme $\mathcal{PKE}$. We build a Trace&Revoke encapsulation scheme Π that can revoke up to r users, and tracing traitor for a pirate decoder having up to t traitors' keys, in the following way.

- $\mathsf{Setup}(1^\lambda, N)$
  1. Run the code generating algorithm on $(\mathcal{Q}, N, k, r, t, s)$ to obtain an $(r, s, \mathcal{Q}, p, \delta, t, \tau)$-blackbox Trace&Revoke 1-Conjunction (ℓ, N)-Codes.
  2. Run $\mathcal{PKE}.\mathsf{Setup}(1^\lambda)$ to get the public parameters param for the encryption scheme;
  3. For $i = 1, \ldots, \ell$, run the key generation algorithm $\mathcal{PKE}.\mathsf{KeyGen}(\mathsf{param})$ to get the pair $(\mathsf{dk}_i, \mathsf{ek}_i)$.
  4. Set $\mathsf{MSK} = (\Gamma, \{\mathsf{dk}_i\})$, and $\mathsf{EK} = \{\mathsf{ek}_i\}$.
  5. For $i = 1, \ldots, N$, the user i is associated with the codeword $w^i \in \Gamma$: we set $\mathsf{usk}_i \leftarrow \{\mathsf{dk}_j / w_j^i = 1, j = 1, \ldots, \ell\}$.
- $\mathsf{Encaps}(\mathsf{EK}, R)$:
  1. For a revoked set R of size at most r, since the code Γ is efficiently (r, s)-revocable, one can find out a signal c of weight at most s, such that $\mathsf{D}_1(u, c) = 0$, for any $u \in F(R)$, and $\mathsf{D}_1(u, c) = 1$ for any $u \in [N] - R$. We denote by $m = \mathbf{w}_H(c)$ this weight;
  2. Denote by $i_1, \ldots, i_m$ the positions of m 1-bits in c, *i.e.,* $c_{i_j} = 1$, for $j = 1, \ldots, m$;
  3. Choose a random session key $K \xleftarrow{\$} \{0,1\}^\kappa$.
  4. Set $e_{i_j} = \mathcal{PKE}.\mathsf{Enc}(\mathsf{pk}_{i_j}, K)$, for $j = 1, \ldots, m$.
  5. Output K and $H = (c, (e_{i_j}), j = 1, \ldots, m)$.
- $\mathsf{Decaps}(\mathsf{usk}_j, R, H)$:
  1. If j is in $[N] - R$, then $\mathsf{D}_1(w^j, c) = 1$. This means $\mathbf{w}_H(w^j \wedge c) \geq 1$ and there exists thus i_j in $w^j \wedge c$
  2. Compute $K = \mathcal{PKE}.\mathsf{Enc}(\mathsf{sk}_{i_j}, e_{i_j})$.
- $\mathbf{Trace}^{\mathbb{D}}(R_{\mathbb{D}}, \mathsf{pk}, \mathsf{msk})$: Running the tracing algorithm for the code, each time the tracer asks a qualified query c to the pirate decoder, we do as follows: run $\mathsf{Encaps}(\mathsf{EK}, R)$ but directly use the signal c (in fact, the revoked set R in this case corresponds to the set of the users that cannot decrypt c) and query the pirate decoder on the R, H. If the pirate decoder exactly recovers the session key K, we return 1 to the tracer for the code, and otherwise we return 0.

It is straightforward that if the pirate decoder $R_{\mathbb{D}}$ answers all ciphertexts constructed from qualified signal c for $\mathcal{Q}$, then the tracing procedure can be directly reduced to the tracing for the code, and thus we can identify traitors, as in the $(r, s, \mathcal{Q}, p, \delta, t, \tau)$-blackbox Trace&Revoke.

In the particular case of Q_a, the fact that the pirate decoder $R_{\mathbb{D}}$ answers all ciphertexts constructed from qualified signal c for $\mathcal{Q}_a$ implies that the pirate decoder decrypts all ciphertext with a header H containing a encapsulations of the session key, each is encrypted by a key at a row of the matrix.