Improving the Round Complexity of Traitor Tracing Schemes

Aggelos Kiayias[1,⋆] and Serdar Pehlivanoglu[2,⋆⋆]

[1] Computer Science and Engineering,
University of Connecticut Storrs, CT, USA
aggelos@cse.uconn.edu
[2] Division of Mathematical Sciences
School of Physical and Mathematical Sciences
Nanyang Technological University, Singapore
pserdar@ntu.edu.sg

Abstract. A traitor tracing scheme is a multiuser encryption that has a built-in key leakage deterrence mechanism : the sender is capable of utilizing a tracing process that can interact with any adversarial decoder and reveal the identities of the users whose keys are employed by the decoder. A number of desired design goals have been put forth for traitor tracing schemes, notably the minimization of the length of the ciphertexts, the length of the encryption key and the storage for private keys. An important efficiency parameter that is not as widely investigated is the round complexity of the tracing process, i.e., the number of rounds of interaction that is required for the tracing process to terminate. In this work we provide (1) a general formalization of this important design consideration, (2) a novel tracing procedure that exhibits an asymptotic improvement over the previously known approaches. Our first result is achieved by casting the tracing process as a game between the tracer and the adversary where the objective of the tracer is to reveal the identity of the corrupted users while the adversary wishes to prevent that while still meeting a minimum functionality requirement. The second result involves a novel application of fingerprinting codes.

1 Introduction

The distribution of content to a set of subscribers is not served adequately by the mere employment of an encryption scheme. Indeed, in this setting — where many users are supposed to share the same decryption capability — it is plausible to expect that an adversary may take hold of some subscriber keys and spread them to other entities. In order to prevent this type of key-leakage some deterrence mechanisms can be built into the encryption scheme. In the multiuser encryption setting, the ability of an authority to interact with a rogue decoder and recover

⋆ Research partly supported by NSF Awards 0447808, 0831304, 0831306.
⋆⋆ Research is supported by The Singapore National Research Foundation under Research Grant NRF-CRP2-2007-03.

J. Zhou and M. Yung (Eds.): ACNS 2010, LNCS 6123, pp. 273–290, 2010.

at least one of the identities of the users whose keys have been leaked and utilized in the decoder gives rise to traitor tracing schemes.

These constructions were introduced in [6] and subsequently a great number of subsequent works improved various aspects of them [1,2,3,4,7,8,10,13,14,15, 17,18,19,20,21,22,23,24,26,25,27,28,29,30,31]. In this long sequence of works a number of important efficiency characteristics of traitor tracing schemes have been identified and have been stepwise refined : (i) the length of ciphertexts, (ii) the length of the encryption key, (iii) the size of storage required in the subscriber side. These quantities are typically expressed as functions of the number of users n, the failure probability of tracing ϵ and an upper bound w on the number of corrupted users.

An important aspect of traitor tracing schemes is the number of rounds of interaction that are required between the tracing authority (or simply the tracer) and a rogue device in order for the tracer to establish the desired identities. A possible reason for not pursuing this measure of optimization is that much fewer essentially different techniques to perform tracing exist compared to the large number of different constructions.

In fact the majority of schemes share the same tracing strategy that can be summarized in the following fashion : divide the users in N subsets of a certain size and perform a "walking procedure" of N steps that utilizes partially corrupted ciphertexts and identifies which one of the N subsets is overlapping with the corrupted users. Then repeat the procedure with a different set of subsets. In the end combine all the results to infer a corrupted user identity. In its most basic form, the subsets can be singletons, i.e., the users themselves and it is only needed to perform a single such walking procedure (cf. [22]).

We refer to this ubiquitous tracing strategy as "linear tracing"; it was utilized in the majority of the works cited above. Interestingly none of the previous works explicitly focused on the round complexity as an efficiency measure and on its proper formalization. In fact a proper formalization of the measure not only involves the number of steps/users n, the failure probability ϵ but also the minimum threshold σ that is required by a rogue decoder to meet when decrypting a valid ciphertext (as obviously if σ drops it becomes harder and harder to trace).

The only two known improvements of the linear tracing strategy were put forth in [26] called "binary search tracing" and "noisy binary search tracing" (the latter being an improvement inspired by [12]). These were the first asymptotic improvements of the round complexity of the linear tracing strategy as we highlight below.

The present work has two major contributions:

1. We present the first formalization of the round complexity of traitor tracing schemes. This is achieved by casting the tracing process as a protocol game between the tracer and the adversary (the rogue device). The goal of the tracer is perform identification of one of the traitor users while the goal of the adversary is to prevent this from happening while still maintaining the required functionality threshold. Of independent interest is the fact that

this formalization of tracing is the flip side of a class of privacy related interactions considered in [11] : there the adversary is in the tracer side and wishes to violate the privacy of the users.

2. We present a new tracing strategy that has an improved round complexity compared to the previous known approaches for small values of σ. Further, our strategy is the first one that can utilize an upper bound on the number of corrupted users and use that bound to curb the round complexity. Our strategy relies on a novel application of fingerprinting codes [3, 33] overimposed on the tracing process. An illustration of our results is shown in figure 1.

Tracing Technique In Use	Corruption Bound	Failure Probability	Adversary Threshold σ	Round Complexity
Linear [22, 6]	n	ε	$> 4n\varepsilon_p$	$O(n^3\sigma^{-2}\log\frac{1}{\varepsilon})$
Bin. Search [26]	n	ε	$> 4n\varepsilon_p$	$O(n^2\sigma^{-2}\log\frac{\log n}{\varepsilon}\log^3 n)$
Noisy Bin. Search [26, 12]	n	ε	$> \frac{2}{3}$	$O(n^2\log\frac{1}{\varepsilon}\log n)$
Our Result(1)	n	ε	$> 4n\varepsilon_p + 2\delta$	$O(n^2\delta^{-2}\log\frac{n^2\log n/\varepsilon}{\varepsilon})$
Our Result(2)	w	ε	$> 4n\varepsilon_p + 2\delta$	$O(w^2\delta^{-2}\log\frac{w^2\log n/\varepsilon}{\varepsilon})$

Fig. 1. A comparison of tracing techniques where σ is the required minimum success threshold for the adversary in decoding valid transmissions. The value ε_p corresponds to essentially the probability of building a "keyless" decoder. The value n corresponds to the number of users.

As it is evident by the table our first result matches the best previous tracing strategy of noisy binary searching for constant adversarial thresholds but are also capable of addressing smaller adversarial thresholds (e.g., $\sigma = O(\frac{1}{\log n})$ etc.) with comparable asymptotic behavior. In addition our second result improves the round complexity bound much further if one assumes that $w << n$. We note that the general approach we put forth here is fundamentally different than the one of all these previous works and it is not apparent how to utilize a bound in the number of corrupted users w in the previous strategies.

As final note, beyond [26], another previous work that identified the importance of round complexity was [16]; while the results of that paper exhibit a linear dependency on w they employ much stronger assumptions that go beyond the transmission capabilities of the majority of previous works: in particular, they assume that the content can be watermarked with an alphabet that is linear in w. This is not applicable in many settings where traitor tracing is being used (e.g., the distribution of cryptographic keys); further even in settings where watermarking is possible the alphabet size is preferably binary or a small constant independent of w. For this reason we do not include the results of [16] in the comparison above.

Paper Organization. In section 2 we present our basic primitive, the multiuser encryption scheme. In section 2.1 we put forth the simple linear length multiuser encryption scheme as our basic work paradigm. For simplicity we present our

results over this scheme but they can be easily generalized to other settings where we have sequences of subsets of users etc. In section 3 we present our formalization of a tracing game that makes explicit the notion of tracing round complexity. In section 4 we present our novel tracing algorithm. Given that our formalization of the notion of tracing round complexity is new, we present first a complete analysis of the standard linear tracing strategy in our model that is then followed by our new tracing strategy based on fingerprinting codes. In the appendix we present the proofs of all our claims.

2 Multiuser Encryption Schemes

Any traitor tracing scheme is based on an underlying encryption mechanism called a multi-user encryption scheme (ME).

A multi-user encryption scheme ME is a triple (**KeyDist**, **Transmit**, **Receive**). The parameter of the scheme is n, the number of receivers and is associated with three sets $\mathsf{K}, \mathsf{M}, \mathsf{C}$ corresponding to the sets of keys, plaintexts and ciphertexts respectively. We describe the I/O of these procedures below:

- **KeyDist.** It is a probabilistic algorithm that on input 1^n, it produces $(tk, ek, sk_1, \ldots, sk_n)$. The decryption key sk_i is to be assigned to the i-th user while ek is the encryption key. The tracing key tk is some auxiliary information to be used for tracing that may be empty.
- **Transmit.** It is a probabilistic algorithm that given a message $m \in \mathsf{M}$, it prepares an element $c \in \mathsf{C}$. We will write the following to denote the distribution of the output: $c \leftarrow \mathbf{Transmit}(ek, m)$
- **Receive.** It is a deterministic algorithm that on input c sampled from $\mathbf{Transmit}(ek, \langle m \rangle)$ and a user-key sk_i for some $i \in [n]$ where $(tk, ek, sk_1, \ldots, sk_n) \leftarrow \mathbf{KeyDist}(1^n)$, it either outputs m or fails. Note that **Receive** can also be generalized to be a probabilistic algorithm but we will not take advantage of this here.

We can consider a variant of the multi-user encryption scheme that is stateful where the algorithm **Transmit** is parameterized by a set of states denoted by States. In a stateful multi-user encryption scheme, the **Transmit** algorithm prepares an element $c \in \mathsf{C}$ as a function of the current state and updates the state after each transmission. In this chapter, unless stated otherwise, we will be discussing stateless multiuser encryption schemes.

The above determine the syntax of the algorithms that define a multiuser encryption scheme ME. We expect from such a scheme to satisfy correctness in the usual sense. In particular we require that:

Definition 1. ***Correctness.*** *We say a multiuser encryption scheme* ME *is correct if for any $n \in \mathbb{N}$, for any message $m \in \mathsf{M}$ and for any $u \in [n]$, it holds that*

$$\mathbf{Prob}[\mathbf{Receive}(\mathbf{Transmit}(ek, M), sk_u) \in \{m\}] = 1$$

where $(tk, ek, sk_1, \ldots, sk_n)$ is distributed according to $\mathbf{KeyDist}(1^n)$.

Security. In a setting where the hybrid encryption approach is employed, the content transmission operates at two levels: first, a one-time content key k is selected and encrypted with the multiuser encryption scheme. Second, the actual message will be encrypted with the key k and will be transmitted alongside the encrypted key. It follows that a minimum requirement would be that the scheme ME should be sufficiently secure to carry a cryptographic key k. As an encryption mechanism this is known in the context of public key cryptography as a "Key Encapsulation Mechanism" [9]. The security model we present in this section will take this formalization approach, i.e., it will focus on the type of security that needs to be satisfied by a multiuser encryption scheme in order to be used as a key encapsulation mechanism. The figure 2 is the standard security game that captures the key encapsulation we require from the multi-user encryption scheme.

$\mathsf{TransmitOracle}(m)$	$\mathsf{ReceiveOracle}(c, u)$
retrieve ek;	retrieve sk_u;
$c \leftarrow \mathbf{Transmit}(ek, m)$;	return $\mathbf{Receive}(c, sk_u)$;
return c;	

Experiment $\mathbf{Exp}_{\mathcal{A}}^{ME}(1^n)$

$(tk, ek, sk_1, \ldots, sk_n) \leftarrow \mathbf{KeyDist}(1^n)$

$aux \leftarrow \mathcal{A}^{\mathsf{TransmitOracle}(\cdot),\mathsf{ReceiveOracle}(\cdot)}(1^n)$

$m_0, m_1 \overset{R}{\leftarrow} \mathsf{M}$

$b \overset{R}{\leftarrow} \{0,1\}$; $c \leftarrow \mathbf{Transmit}(ek, m_1)$

$b' \leftarrow \mathcal{A}^{\mathsf{TransmitOracle}(\cdot)}(aux, m_b, c)$

return 1 if and only if $b = b'$

Fig. 2. The CCA-1 security game for multi user encryption scheme between the adversary and the challenger

Definition 2. *We say a multiuser encryption scheme* ME *is CCA-1 ε-insecure if for any probabilistic polynomial time algorithm $\mathcal{A}$, it holds that*

$$\mathbf{Adv}_{\mathcal{A}}^{kem}(1^n) = \mathbf{Prob}[\mathbf{Exp}_{\mathcal{A}}^{kem}(1^n) = 1] - \frac{1}{2}] \leq \varepsilon$$

where the experiment $\mathbf{Exp}_{\mathcal{A}}^{kem}$ is defined as in figure 2.

We note that ε in general is not supposed to be a function of n, i.e. the security property should hold independently of the number of users present in the recipient list. It is possible to define CPA version of the above security definition/game by not letting the attacker to access the ReceiveOracle on the second line of the security game. In such case we say the multiuser encryption scheme is CPA ε-insecure, if the above condition given in the definition holds for a CPA adversary.

Note that typically key encapsulation mechanisms are defined without any input beyond the encryption key (i.e., there is no plaintext part). For convenience

we take a different approach where we provide the input. In effect, we state above that the encryption mechanism of the multiuser encryption matches the syntax of regular encryption and is supposed to satisfy the security requirements of a key encapsulation mechanism.

2.1 Linear Length Multiuser Encryption Scheme

We will now, present a straightforward multiuser encryption scheme that produces a ciphertext of length linear in number of receivers. We will name this scheme by $\mathtt{ME}_L$ which will be transmitting a single encrypted message to each of the receivers. It is parameterized by an encryption scheme $(\mathtt{E}, \mathtt{D})$.

- **KeyDist**$_L$: Given 1^n it produces a set of keys $\{k_1, \ldots, k_n\} \subseteq \mathsf{K}$, sk_i is set to be the key k_i for $i = 1, \ldots, n$, and sets $ek = \langle k_1, \ldots, k_n \rangle$. The tracing key tk is empty.
- **Transmit**$_L$: On input $m \in \mathsf{M}$ and the encryption key $ek = \langle k_1, \ldots, k_n \rangle$, it transmits the encryption of the message m with ek by employing the encryption scheme $(\mathtt{E}, \mathtt{D})$ as follows:

$$\langle \mathtt{E}_{k_1}(m), \ldots, \mathtt{E}_{k_n}(m) \rangle$$

- **Receive**$_L$: Given the key $sk_i = k_i$ for some $i \in [n]$ and a transmission of the form $\langle c_1, \ldots, c_n \rangle$, it returns $\mathtt{D}_{k_i}(c_i)$.

We require the broadcast encryption scheme to be capable of transmitting a cryptographic key. We will ask that this same requirement should also be satisfied by the underlying cryptographic primitive $(\mathtt{E}, \mathtt{D})$, i.e., a cryptographic key should be encapsulated safely by the underlying encryption primitive.

We formalize the security requirement as the following game: for a random choice of the key k, the adversary $\mathcal{A}$ can adaptively choose plaintexts and see how $\mathtt{E}_k$ encrypts them; similarly, is capable of observing the output of decryption procedure $\mathtt{D}_k$. The adversary is challenged with a pair (c, m) for which it holds that either $c \leftarrow \mathtt{E}_k(m)$ or $c \leftarrow \mathtt{E}_k(m')$ where m, m' are selected randomly from the message space. The goal of the adversary is to distinguish between the two cases. This models a CCA1 type of encryption security, or what is known as a security against lunch-time attacks.

Experiment $\mathbf{Exp}^{kem}_{\mathcal{A}}$

Select k at random.
$aux \leftarrow \mathcal{A}^{\mathtt{E}_k(), \mathtt{D}_k()}()$
$m_0, m_1 \xleftarrow{R} \mathsf{M}$; $b \xleftarrow{R} \{0, 1\}$; $c = \mathtt{E}_k(m_1)$
$b' \leftarrow \mathcal{A}^{\mathtt{E}_k()}(aux, c, m_b)$
return 1 if and only if $b = b'$;

Fig. 3. The security game of CCA1 secure key encapsulation for an encryption scheme

Definition 3. *We say the symmetric encryption scheme* $(\mathtt{E}, \mathtt{D})$ *is* ε*-insecure if it holds that for any probabilistic polynomial depth circuit* $\mathcal{A}$

$$\mathbf{Adv}_{\mathcal{A}}^{kem} = |\mathbf{Prob}[\mathbf{Exp}_{\mathcal{A}}^{kem} = 1] - \frac{1}{2}| \leq \varepsilon$$

Observe that the above requirement is weaker that one would typically expect from an encryption scheme that may be desired to protect the plaintext even if it is arbitrarily distributed. We note though that the key encapsulation security requirement will still force the encryption function to be probabilistic: indeed, in the deterministic case, the adversary can easily break security by encrypting m_b and testing the resulting ciphertext for equality to c. Further, since we are only interested in key encapsulation we can require the encryption oracle to only return encryptions of random plaintexts (as opposed to have them adaptively selected by the adversary).

Theorem 1. *A linear length multiuser encryption scheme* $\mathtt{ME}_L$ *satisfies correctness (cf. Definition 1) i.e., we assume that for all* $k, m \in \mathsf{K}, \mathsf{M}$ *it holds* $\mathtt{D}_k(\mathtt{E}_k(m)) = m$*. It is, further, CCA-1* ε*-insecure in the sense of Definition 2 with* $\varepsilon \leq 2n \cdot \varepsilon_p$ *where the underlying encryption scheme* $(\mathtt{E}, \mathtt{D})$ *is* ε_p*-insecure in the sense of Definition 3.*

The obvious shortcoming of the above scheme is that the ciphertext length is linear in n. A nontrivial approach to reduce the transmission overhead involves the usage of fingerprinting codes, see [5,1,6,7,33] for a non-inclusive citation list. As the scope of this paper is not improving the transmission overhead we will not discuss such constructions. We find the description of linear length traitor tracing scheme sufficient to discuss the new formalization and to introduce the new technique in identifying a traitor. This new technique is readily applicable to all traitor tracing mechanisms involving some walking argument, i.e. querying the adversary with special transmissions that randomizes the ciphertexts one by one.

3 Tracing Game: Definitions

We define the property that is the characteristic of a traitor tracing scheme. We first introduce the concept of the tracing game.

Definition 4. *A* tracing game *is specified by any triple* $\langle \mathbf{KeyDist}, \mathcal{Q}, \mathcal{R} \rangle$*, where* **KeyDist** *is a probabilistic algorithm that on input* 1^n*, it produces a tuple* $(tk, ek, sk_1, \ldots, sk_n)$*,* $\mathcal{R}$ *is a predicate and* $\mathcal{Q}$ *is a set of random variables.*

We now explain the meaning of the tracing game. A tracing game is an interaction between two parties: the adversary and the tracer. The tracer has at its disposal the encryption and the tracing key (resp. tk, ek) while the adversary has a set of user keys, that is a subsequence of $sk_1, \ldots, sk_n$. The ultimate objective of the tracer is to identify one of the keys that the adversary controls.

Next we set the general rules of engagement between the tracer and the adversary that are determined by $\mathcal{Q}, \mathcal{R}$. The essence on the constraint that we will place on the interaction is the following: as long as the tracer follows a certain query distribution as defined in $\mathcal{Q}$ the adversary is obliged to formulate its responses in a way that they will satisfy the predicate $\mathcal{R}$ with sufficient probability.

More specifically the pair $\langle \mathcal{A}, \mathcal{T} \rangle$ will be said to be σ-*admissible* according to a tracing game $\langle \mathbf{KeyDist}, \mathcal{Q}, \mathcal{R} \rangle$ with a parameter t provided that $\mathcal{A}, \mathcal{T}$ follow the proper rules of engagement. More specifically, σ-admissible would be a pair of interacting algorithms that when $\mathcal{T}$ sends some message to $\mathcal{A}$ that follows a random variable of $\mathcal{Q}$ then $\mathcal{A}$ as to provide a response that satisfies the predicate $\mathcal{R}$. Formally we have the following definition.

Definition 5. *Let $\langle \mathcal{A}, \mathcal{T} \rangle$ be a pair of interacting algorithms (the adversary and the tracer) and let $\langle \mathbf{KeyDist}, \mathcal{Q}, \mathcal{R} \rangle$ be a tracing game. For $n \in \mathbb{N}$ and any $\mathsf{C} \subseteq [n]$ assume the following regarding the interaction of $\langle \mathcal{A}, \mathcal{T} \rangle$:*

- *Initialization. The tuple $(tk, ek, sk_1, \ldots, sk_n) \leftarrow \mathbf{KeyDist}(1^n)$ is sampled and the adversary $\mathcal{A}$ is given input $\{sk_j\}_{j \in \mathsf{C}}$ while the tracer $\mathcal{T}$ is given tk, ek.*
- *Round actions. $\mathcal{A}$ and $\mathcal{T}$ exchange messages in rounds until the tracer $\mathcal{T}$ terminates. In the i-th round $\mathcal{T}$ goes first transmitting a value q_i and then party $\mathcal{A}$ responds by a value a_i. In case $\mathcal{A}$ produces no output at a certain round i the value a_i is defined to be $\perp$.*

We say the pair $\langle \mathcal{A}, \mathcal{T} \rangle$ is σ-admissible *for the tracing game $\langle \mathbf{KeyDist}, \mathcal{Q}, \mathcal{R} \rangle$ for t-coalitions, where $t \in \mathbb{N}$ if for any $n \in \mathbb{N}, \mathsf{C} \subseteq [n]$, in case, q_i is distributed according to some member of $\mathcal{Q}$ it holds that for any $\mathsf{C} \subseteq [n]$ with $|\mathsf{C}| \leq t$,*

$$\mathbf{Prob}[\mathcal{R}(\mathsf{C}, tk, ek, sk_1, \ldots, sk_n, q_i, a_i) = 1] \geq \sigma$$

where a_i is the response of $\mathcal{A}$ to the query q_i on input $\{sk_j\}_{j \in \mathsf{C}}$. We denote the random variable that is the output of the tracer $\mathcal{T}$ after interacting with $\mathcal{A}$ by $\langle \mathcal{A}, \mathcal{T} \rangle (tk, ek, sk_1, \ldots, sk_n, \mathsf{C})$. We denote the maximum number of rounds that take place before $\mathcal{T}$ terminates the protocol by $r_{\langle \mathcal{A}, \mathcal{T} \rangle}$.

The definition of σ-admissibility in plain words it says that as long as the tracer $\mathcal{T}$ follows some of the specified valid moves in $\mathcal{Q}$ the party $\mathcal{A}$ has to oblige and satisfy with its response the predicate $\mathcal{R}$ with probability σ. We observe that the predicate $\mathcal{R}$ takes into account the total information that is available to both players and thus it is not something that the tracer $\mathcal{T}$ is necessarily capable of computing by itself. In the coming section we will formulate various tracing games that are resulting from interactions based on multiuser encryption schemes.

We will also condider the following variations of the definition:

1. *Stateful Tracing.* Consider the set $\{\mathcal{Q}_h\}_{h \in \{0,1\}^*}$, a collection of sets of random variables. In this specification, for the adversary to oblige and satisfy the $\mathcal{R}$ predicate we require the tracing queries of $\mathcal{T}$ to be consistent with

the history of previous queries. More specifically, we define, for any $i > 0$, $h = \langle q_1, \ldots, q_{i-1} \rangle$ to be the history of the queries posed by $\mathcal{T}$ (it is empty if $i = 1$), the next query q_i should be distributed according to some member of $\mathcal{Q}_h$ in order to impose the σ lower bound in the satisfaction of the predicate $\mathcal{R}$ for $\mathcal{A}$. Note that the predicate $\mathcal{R}$ will also take the history of the queries into account while producing a result. Stateful tracing is thus places a further possible restriction on the tracer's side as it drops any compliance requirements on the part of $\mathcal{A}$ when the tracer becomes inconsistent with the query history.

2. *Alfresco.* When tracing alfresco the tracer needs to form every query he makes to be statistically indistinguishable from members of $\mathcal{Q}$ (or from $\mathcal{Q}_h$ in case of stateful tracing). More specifically, when the tracer has submitted $h = \langle q_1, \ldots, q_{i-1} \rangle$ queries, in the i-th round it must choose a query that is statistically indistinguishable from a member of $\mathcal{Q}$ (from a member of $\mathcal{Q}_h$ in case of stateful tracing). Note that this is a different type of restriction on the tracer $\mathcal{T}$. Without this restriction the tracer has the flexibility to provide queries to $\mathcal{T}$ that are outside of $\mathcal{Q}$ but they may perhaps be computationally indistinguishable to some random variables of $\mathcal{Q}$ for $\mathcal{A}$; depending on the case this may carry substantial advantages for the tracer that are stripped in the case of alfresco tracing.
3. *Tracing with Resetting.* The adversary $\mathcal{A}$ is not allowed to maintain state from one round to the next, i.e., in each round the tracer can "reset" the adversary. This model weakens the adversary $\mathcal{A}$ as it is prohibited from keeping knowledge of previous queries. This can be taken advantage of by the tracer $\mathcal{T}$. Alternatively, when the adversary maintains state across rounds (i.e., the default in the definition above) we say we deal with *history-recording* adversaries.
4. *Abrupt adversaries.* This is a strengthening of the adversarial model that enables $\mathcal{A}$ to finish the game at a moment of its choosing. This means that $\mathcal{A}$ may produce a special symbol as a response upon receiving which the tracer $\mathcal{T}$ is not allowed to submit any further queries. We note that this is not in violation of the basic tenet of being admissible : $\mathcal{A}$ when it forms a σ-admissible pair with the adversary is still supposed to satisfy the $\mathcal{R}$ with probability at least σ. If $\mathcal{A}$ is abrupt though it may decide to stop the tracing game with probability as high as $1 - \sigma$ if given a query from $\mathcal{Q}$ (and in fact with even higher probability when given queries from outside of this set).

Now that we have defined the rules of engagement between the two players of the tracing we will specify when game is winnable by the tracer.

Definition 6. *We say that the tracing game* $\mathtt{TG} = \langle \mathbf{KeyDist}, \mathcal{Q}, \mathcal{R} \rangle$ *is winnable with probability* α *for* σ*-threshold and* t*-coalitions if there exists a tracer* $\mathcal{T}$ *such that for all* $\mathcal{A}$ *for which the pair* $\langle \mathcal{A}, \mathcal{T} \rangle$ *is* σ*-admissible it holds that for all* $n \in \mathbb{N}$, $\mathsf{C} \subseteq [n]$, $|\mathsf{C}| \leq t$,

$$\mathbf{Prob}[\emptyset \neq \langle \mathcal{A}, \mathcal{T} \rangle (tk, ek, sk_1, \ldots, sk_n, \mathsf{C}) \subseteq \mathsf{C}] \geq \alpha$$

where $(tk, ek, sk_1, \ldots, sk_n)$ *is distributed according to* $\mathbf{KeyDist}(1^n)$.

Provided that TG *is winnable with a tracer* $\mathcal{T}$*, we define the tracing round complexity* $\mathsf{roundc}[\mathtt{TG}, \mathcal{T}]$ *of the tracing game* TG *as the supremum of all* $r_{\langle \mathcal{A}, \mathcal{T} \rangle}$*; note that* $\mathsf{roundc}[\mathtt{TG}, \mathcal{T}]$ *is a function of* t, σ *and possibly of other parameters as well.*

It is interesting to note that the tracing game can be thought of as a privacy game if we flip the semantics of what side is adversarial. In this alternative interpretation, the good side is sitting on the adversary side and attempts to output some data that meet some "usefulness criterion" determined by $\mathcal{R}$ and are based on the private information of the users. On the other hand, the adversary is sitting on the tracer side and attempts to violate the privacy of the users. This perspective is in line with a recent work by Dwork et.al. [11] for which we do not pursue this parallel further here.

In this work, we are interested in tracing games whose **KeyDist** algorithm and the query distribution $\mathcal{Q}$ are related to multiuser encryption schemes. For an s-ary multiuser encryption scheme $\mathtt{ME} = (\mathbf{KeyDist}, \mathbf{Transmit}, \mathbf{Receive})$, the set of random variables $\mathcal{Q}$ is defined as the collection of all encryptions of plaintexts that can be transmitted, i.e. denoting it by $\mathcal{Q}^{\mathbf{Transmit}}$, it contains the random variables $\mathbf{Transmit}(ek, m)$ for any $m \in \mathsf{M}$ where $(tk, ek, sk_1, \ldots, sk_n) \leftarrow \mathbf{KeyDist}(1^n)$. In case the scheme ME is stateful over as a set of States, $\mathcal{Q}$ is parameterized with States as well. We note that depending on the actual use of a multiuser encryption scheme one may define the tracing game considering only specific distributions of plaintext.

With respect to the predicate $\mathcal{R}$ there are more than one ways to define it. Recall that intuitively we use this predicate to measure the success of the adversary in responding to the transmissions in a normal mode of operation as we define the query space $\mathcal{Q}$ to be sampled from **Transmit**. Indeed a σ-admissible tracer-adversary pair is one for which $\mathbf{Prob}[\mathcal{R}(\mathsf{C}, tk, ek, sk_1, \ldots, sk_n, q, a) = 1] \geq \sigma$ holds if a is the response of the adversary to the query q that is sampled as an element of $\mathcal{Q}$.

The exact choice of the way predicate $\mathcal{R}$ works will provide a classification of types for tracing games. Jumping forward we give a glimpse to possible definitions: for example, the adversary may return a key from the key space (not necessarily among the $\{sk_i\}_{i \in [n]}$) that successfully decrypts the query/transmission $q \leftarrow \mathbf{Transmit}(ek, m)$ with σ probability for any choice of $m \in \mathsf{M}$. Alternatively the predicate $\mathcal{R}$ checks to see if the response a is the plaintext transmitted in the encrypted form q. We note that $\mathcal{R}$ is capable of performing these checks in polynomial-time as it has access to the full information of keys and the adversarial input.

The exact choice of the predicate $\mathcal{R}$ and the restrictions on the queries posed by the tracer $\mathcal{T}$ yield different types of tracing. Formalizing three types of tracing games very roughly, our interest in this work is black-box traitor tracing.

Black-Box Tracing Game. In this setting, the tracer has merely black-box access to the pirate decoder. Black-box traitor tracing may in some cases allow tracing to be performed remotely without the physical availability of the pirate decoder.

The major challenge in the black-box traitor setting is to extract information regarding the original keys utilised in the construction of the pirate decoder. The tracer will communicate with the pirate decoder using a set of specially crafted queries. These queries will not be necessarily normal transmissions as the tracing center is allowed to communicate with the decoder in an arbitrary way. The response of the decoder may be equal to the decrypted plaintext, or be simply of binary form, essentially "yes", in case of returning the content in the cleartext form, or "no", in case of responding arbitrarily or jamming.

In our exposition, we will use the threshold σ to impose the adversarial constraint related to the success probability of the pirate decoder in decrypting regular transmissions. This is of particular importance, since tracing would be impossible against a pirate decoder that is not required to operate correctly at least some of the time.

Definition 7. *A multi-user encryption scheme* $\mathtt{ME} = (\mathbf{KeyDist}, \mathbf{Transmit}, \mathbf{Receive})$ *is a* black box traitor tracing scheme for t-coalitions with success probability α against σ-pirates *if the black-box tracing game* $\mathtt{TG} = \langle \mathbf{KeyDist}, \mathcal{Q}^{\mathbf{Transmit}}, \mathcal{R}^{\mathtt{BB}} \rangle$ *against t-coalitions is winnable with probability α against σ-adversaries.*

The $\mathcal{R}^{\mathtt{BB}}$ *is defined as follows:* $\mathcal{R}^{\mathtt{BB}}(\mathsf{C}, tk, ek, sk_1, \ldots, sk_n, q, a)$ *is equal to 1 if and only if* $a = m$ *whenever* $q = \mathbf{Transmit}(ek, m)$.

We define the tracing round complexity $\mathsf{roundc}[\mathtt{ME}]$ *of the multiuser encryption scheme as the infimum of all* $\mathsf{roundc}[\mathtt{TG}, \mathcal{T}]$ *for which* ME *is winnable with the tracer* $\mathcal{T}$.

One may also consider a more general view of the black box tracing model, that is related to the case that the pirate decoder is a tamper resistant box, such as a music player and the response of the decoder is not the exact decryption of the transmission but rather the actual rendering of the cleartext transmission on a display device. In such case, the tracer can still extract useful information by observing whether the given ciphertext results in music being played or not. It is possible to address such definitions in our framework by introducing a *filter* algorithm that restricts some of the information contained in a_i and having the tracer have access to a_i only through the filter.

Among the different variations of the tracing game described in Definition 4, tracing with resetting is relevant to black-box tracing as it is defining the capabilities of the pirate decoder the tracer has access to. We would like to motivate this case briefly in the following paragraph for the context of the present section.

A pirate decoder is said to be resettable if the tracer has the capability to reset the pirate decoder to its initial state and the decoder is available for a new query. This gives the tracer the advantage of asking queries that will be handled independently during the tracing process, i.e., effectively preventing the decoder from using previous querying information submitted by the tracer in order to decide its present action. In contrast, a *history recording* pirate decoder "remembers" the previous queries made by the tracer and because the tracing procedure is public, the history recording capability can be used by the decoder to evade tracing.

Other Tracing Games In the setting for non-black box tracing game, the adversary is considered to be a pirate decoder that is capable of receiving the transmission through some key material that is made out of traitor keys. The non-black box tracing game refers to the case where the response of the adversary is defined to be the key material that makes the decoder succesful. This response is not necessarily a real reaction of the adversary but rather an abstract notion that in reality may include physical tampering on behalf of the tracer. In many settings by "reverse-engineering" a decoder, it might be possible to retrieve the key employed within. We note that a decryption key from the key-space K should be available to the tracer because of the unlikelihood of performing decryption without a key.

The scenario for pirate rebroadcasting aims to capture a different setting compared to black-box tracing : when the tracer not only does not have access to the pirate decoder physically but in fact it is incapable of performing tracing outside of normal broadcast to the users. In this scenario, the adversary first decrypts the content by using its traitor key material and then once it is in clear text form, it rebroadcasts the content. Clearly a traitor tracing scheme with merely black-box tracing capability is useless against a pirate rebroadcast attack. The tracer is entirely powerless in handling such an attacker as the output of the rebroadcast itself will potentially provide no information about the traitor keys. It is easy to see that the restrictions imposed to the tracer in the pirate rebroadcasting setting are captured by the notion of tracing alfresco as described in the different variations of the tracing game in Definition 4. This is the case as the tracer needs to perform its task by making queries that are statistically indistinguishable from members of $\mathcal{Q}_h$ where h is the previous queries of the tracer.

4 Traceability in Multiuser Encryption Schemes

We will now show that the linear length multiuser encryption scheme $\mathtt{ME}_L$ is a black box traitor tracing scheme against resettable pirate decoders. Resettable pirate decoders allow the tracer to 'reset' the adversary and to receive 'fresh' responses from the pirate that forgets the history of the interaction. This is the key fact in our choice of tracing queries; i.e. we will deviate from the normal set of random variables $\mathcal{Q}^{\mathbf{Transmit}}$ and query the decoder with some special tracing ciphertexts that will yield the identification of a traitor involved in the piracy.

We will first give a high level description of the old technique in Section 4.1 and present our new technique which decreases the number of rounds. The new technique is more like a replacement of the old technique based on walking argument. Particularly, the improvement we made here is readily applicable to the subset cover framework of [24] and any other traitor tracing schemes based on fingerprinting codes (cf. [1,6]).

4.1 Formal Analysis of Linear Tracing Strategy

We will now show that the linear length multiuser encryption scheme $\mathtt{ME}_L$ is a black box traitor tracing scheme against resettable pirate decoders. Recall that resettable pirate decoders allow the tracer to reset the adversary and to receive fresh responses forgetful of the history of the interaction. This is the key fact in our choice of tracing queries; in particular we will deviate from the normal set of random variables $\mathcal{Q}^{\mathbf{Transmit}}$ and query the decoder with some special tracing ciphertexts that will yield the identification of a traitor involved in piracy.

Recall that the linear length multiuser encryption scheme $\mathtt{ME}_L$ transmits a vector of ciphertext $\langle \mathtt{E}_{k_1}(m), \ldots, \mathtt{E}_{k_n}(m) \rangle$ where $(\mathtt{E}, \mathtt{D})$ is the underlying symmetric encryption scheme of $\mathtt{ME}_L$ and the key k_i is available to the i-th receiver. The tracing queries consist of the special transmission $\mathbf{Transmit}^s_L(ek, m)$ for $s = 0, 1, \ldots, n$ by substituting the first s ciphertexts with random strings.

$$\mathbf{Transmit}^s_L(ek, m) = \langle \mathtt{E}_{k_1}(R_1), \mathtt{E}_{k_2}(R_2), \ldots \mathtt{E}_{k_s}(R_s), \mathtt{E}_{k_{s+1}}(m), \ldots \mathtt{E}_{k_n}(m) \rangle \quad (1)$$

where R_i, for $i = 1, \ldots, s$, is a random string of the same length as the message m. Given that the adversary-tracer pair is σ-admissible the adversary will be required to respond the queries of type $\mathbf{Transmit}^0_L(ek, m)$ such that the predicate $\mathcal{R}^{BB}$ is satisfied with probability at least σ. On the other hand note that the predicate necessarily fails with overwhelming probability for queries of type $\mathbf{Transmit}^n_L(ek, m)$. This suggests that the tracer can progressively randomize the pattern of the ciphertext until a position is identified that the pirate-box fails to decrypt successfully whenever it queries the tracing ciphertext.

The soundness of the above argumentation is supported by the following lemma:

Lemma 1. *Assuming that $s \notin \mathsf{C}$, any probabilistic polynomial time adversary $\mathcal{A}$, on input $\{k_i\}_{i \in \mathsf{C}}$, distinguishes the distributions $\mathbf{Transmit}^{s-1}_L(ek, m)$ and $\mathbf{Transmit}^s_L(ek, m)$ with probability at most $2\varepsilon_p$ where $(tk, ek, sk_1, \ldots, sk_n) \leftarrow \mathbf{KeyDist}(1^n)$ assuming that the underlying encryption scheme $(\mathtt{E}, \mathtt{D})$ is ε_p-insecure in the sense of Definition 3.*

Let us define p_s as the probability that the box decodes the special tracing ciphertext $\mathbf{Transmit}^s_L(ek, m)$. Suppose, now, that the key k_s is not available to the adversary, it holds that the pirate decoder can distinguish between the probability spaces $\mathbf{Transmit}^s_L(ek, m)$ and $\mathbf{Transmit}^{s-1}_L(ek, m)$ only with small probability that is related to the insecurity of the underlying encryption scheme $\mathtt{E}(\cdot)$. As a result, it holds that $|p_{s-1} - p_s|$ is sufficiently small with respect to the advantage ε_p in being succesful in security game of the underlying primitive.

On the other hand, for a succesful pirate decoder it holds that $p_0 \geq \sigma$ due to the $\mathcal{A}$-constraint described in the tracing game, while $p_n < \frac{1}{|\mathsf{M}|}$ (M denotes the plaintext space.). Hence there must be some $0 < s \leq n$ for which $|p_{s-1} - p_s| \geq \frac{\sigma - 1/|\mathsf{M}|}{n}$ by the triangular inequality. If it turns out that $\frac{\sigma - 1/|\mathsf{M}|}{n} > 2\varepsilon_p$, this leads the accusation of the user possessing k_s due to the above lemma 1.

Having discussed roughly the interaction of a tracer exploiting the walking argument, we want to note that there are two different ways to locate the probability drop between successive tracing queries. One is due to [22] which results the number of rounds to be $O(n^3\sigma^{-2}\log 1/\varepsilon)$ where ε is the security parameter. The other interaction methodology is given by Naor, Naor and Lotspiech in [24, 26] which resembles like a binary search of the unit gap where the probability drops sufficiently enough to make the accusation. This methodology results an interaction between the tracer and the adversary which has $O(n^2\sigma^{-2}\log\frac{1}{\varepsilon}\log^3 n)$ number of rounds.

For the sake of reference, we state(leaving the proof for full-version of the work) that the $\mathtt{ME}_L$ is a black-box traitor tracing scheme for which the corresponding tracing game for n-coalitions is winnable by a tracer exploits the walking argument of [22] against any probabilistic polynomial time adversarial algorithm $\mathcal{A}$. The tracer that we will construct queries the special transmission of the form $\mathbf{Transmit}^s_L(ek, m)$ for $s = 0, 1, \ldots, n$.

Theorem 2. *Consider a multiuser encryption scheme $\mathtt{ME}_L$ that employs a symmetric encryption scheme that is ε_p-insecure in the sense of Definition 3. $\mathtt{ME}_L$, on input (1^n), is a black box traitor tracing scheme for n-coalitions with probability $1-\varepsilon$ against resettable σ-pirates with $\sigma > 4n\varepsilon_p + \frac{1}{|\mathsf{M}|}$. It further holds that $\mathsf{roundc}[\mathtt{ME}_L, \mathcal{T}_S] \leq \frac{48n^3\cdot\ln(8/\varepsilon)}{(\sigma-1/|\mathsf{M}|)^2}$.*

4.2 Our New Tracing Technique Based on Fingerprinting Codes

Preliminaries on Fingerprinting Codes A fingerprinting code is a pair of algorithms $(\mathbf{CodeGen}, \mathbf{Identify})$ that is defined as follows: **CodeGen** is an algorithm that is given input 1^n, it samples a pair $(\mathcal{C}, tk) \leftarrow \mathbf{CodeGen}(1^n)$ where $\mathcal{C}$ is an (ℓ, n, q)-code defined over an alphabet Q with ℓ as a function of n and q, and the identifying key tk is some auxiliary information to be used for identifying that may be empty. We may use ℓ as a superscript, i.e. denoting by $\mathbf{CodeGen}^\ell$, to emphasize the fact that ℓ might be a function of n, q and some other parameters.

Identify is a deterministic algorithm that on input pair $(\mathcal{C}, tk)$, sampled by $\mathbf{CodeGen}(1^n)$, and a codeword $\mathsf{c} \in Q^\ell$, it outputs a codeword-index $t \in [n]$ or fails. Informally speaking, we say a fingerprinting code is (α, w)-identifier if c is constructed by a traitor coalition size of at most w by combining their codewords, the **Identify** algorithm outputs a traitor with a failure probability of at most α.

Tracing Queries. The tracing queries of the new technique consist of the special transmission $\mathbf{Transmit}^{\mathsf{S}}_L(ek, m)$ for any $\mathsf{S} \in [n]$ by substituting the ciphertexts $\mathtt{E}_{k_i}(m)$ with encryption of random strings if $i \in \mathsf{S}$.

$$\mathbf{Transmit}^{\mathsf{S}}_L(ek, m) = \langle \mathtt{E}_{k_1}(e_1), \mathtt{E}_{k_2}(e_2), \ldots, \mathtt{E}_{k_n}(e_n)\rangle \quad \text{and} \quad e_i = \begin{cases} R_i, & i \in \mathsf{S} \\ m, & i \notin \mathsf{S} \end{cases} \tag{2}$$

where R_i, for $i = 1, \ldots, n$, is a random string of the same length as the message m. Given that the adversary-tracer pair is σ-admissible the adversary will be required to respond the queries of type $\mathbf{Transmit}_L^{\emptyset}(ek, m) = \mathbf{Transmit}_L(ek, m)$ such that the predicate $\mathcal{R}^{BB}$ is satisfied with probability at least σ. The predicate is satisfied on responses of the queries of type $\mathbf{Transmit}_L^{\mathsf{S}}(ek, m)$ either with probability at least $\sigma/2$ or less than. Both of these cases will result in existence of a traitor in either S or $[n] \setminus \mathsf{S}$ if the probability σ is sufficiently large, i.e. $\sigma \geq 4n\varepsilon_p$.

The soundness of the above argumentation is based on the following lemma and we will elaborate more on after the lemma:

Lemma 2. *Let $\mathsf{S} \subseteq [n]$ satisfies $\mathsf{C} \cap \mathsf{S} = \emptyset$ for some set $\mathsf{C} \subseteq [n]$. Any probabilistic polynomial time adversary $\mathcal{A}$, given $\{k_i\}_{i \in \mathsf{C}}$, distinguishes the distributions $\mathbf{Transmit}_L^{\emptyset}(ek, m)$ and $\mathbf{Transmit}_L^{\mathsf{S}}(ek, m)$ with probability at most $2n\varepsilon_p$ where $(tk, ek, sk_1, \ldots, sk_n) \leftarrow \mathbf{KeyDist}(1^n)$ assuming that the underlying encryption scheme $(\mathtt{E}, \mathtt{D})$ is ε_p-insecure in the sense of Definition 3.*

The proof of the above lemma can be structured as a general case of the security theorem 1 of the linear length multiuser encryption scheme. In the statement above we considered the worst case where $\mathsf{S} = [n]$ for which the probability difference is bounded by $2n\varepsilon_p$ as the Theorem 2 suggests.

Let us define p_{S} as the probability that the box decodes the special tracing ciphertext $\mathbf{Transmit}_L^{\mathsf{S}}(ek, m)$. Suppose, now, that no key k_i for $i \in \mathsf{S}$ is available to the adversary, it holds that the pirate decoder can distinguish between the distributions $\mathbf{Transmit}_L^{\emptyset}(ek, m)$ and $\mathbf{Transmit}_L^{\mathsf{S}}(ek, m)$ only with probability at most $2n\varepsilon_p$ due to the Lemma 2, i.e. it holds that $|p_{\emptyset} - p_{\mathsf{S}}| \leq 2n\varepsilon_p$ where ε_p is the insecurity of the underlying encryption scheme $\mathtt{E}(\cdot)$.

On the other hand, if any key k_i available to the adversary satisfies for $i \in \mathsf{S}$, then it holds that the pirate decoder can decrypt the tracing ciphertexts of the form $\mathbf{Transmit}_L^{\mathsf{S}}(ek, m)$ with probability at most $2n\varepsilon_p$ due to the security theorem put forth by the Theorem 1, i.e. $p_{\mathsf{S}} \leq 2n\varepsilon_p$.

Suppose now that we have $\sigma = p_{\emptyset} > 4n\varepsilon_p$. We query the adversary with the tracing ciphertexts of the form $\mathbf{Transmit}_L^{\mathsf{S}}(ek, m)$ to approximate the success probability p_{S}. If $p_{\mathsf{S}} > 2n\varepsilon_p$ then we obtain the fact that there exists a traitor in the set $[n] \setminus \mathsf{S}$. Otherwise it holds that $\sigma - p_{\mathsf{S}} > 2n\varepsilon_p$ for which case we observe the existence of a traitor in the set S due to the Lemma 2. The following theorem 3 suggests that the failure in approximation of p_{S} will be bounded by ε' for $O(\frac{\log 1/\varepsilon'}{2\delta^2})$ many queries with $\sigma > 4n\varepsilon_p + 2\delta$.

The new tracing technique is parameterized by a binary fingerprinting code $\mathtt{F} = \langle \mathbf{CodeGen}, \mathbf{Identify} \rangle$ that generates an $(\ell, n, 2)$-code $\mathcal{C} = \{\mathsf{c}^1, \ldots, \mathsf{c}^n\}$. We then define the set $\mathsf{S}_j = \{i : \mathsf{c}_j^i = 0\}$. Observe that $[n] \setminus \mathsf{S}_j = \{i : \mathsf{c}_j^i = 1\}$ hold.

We submit the adversary with the tracing queries of type $\mathbf{Transmit}_L^{\mathsf{S}_j}(ek, m)$ for $j = 1, \ldots, \ell$. If it happens that $\sigma > 4n\varepsilon_p$ then for each $j = 1, \ldots, \ell$ a traitor exists in the set S_j or in the set $[n] \setminus \mathsf{S}_j$. We let $\mathsf{p}_j = 0$ if the S_j happens to contain a traitor and $\mathsf{p}_j = 1$ holds otherwise. We finally obtain a pirate codeword p that is produced by a coalition of traitors. Running the identification algorithm over

the pirate codeword p outputs a receiver index that is found to be traitor. The overall failure probability is bounded by $\varepsilon_f + \ell\varepsilon'$ (for the failure in identification and failure in approximation respectively).

Theorem 3. *Consider a multiuser encryption scheme* $\mathtt{ME}_L$ *that employs a symmetric encryption scheme that is* ε_p*-insecure in the sense of Definition 3.* $\mathtt{ME}_L$*, on input* (1^n)*, is a black box traitor tracing scheme for* w*-coalitions with probability* $1-\varepsilon_f-\ell\varepsilon'$ *against resettable* σ*-pirates with* $\sigma > 4n\varepsilon_p + 2\delta$*. It further holds that* $\mathsf{roundc}[\mathtt{ME}_L, \mathcal{T}^{\mathtt{F}}_{KP}] \leq \frac{3\ell\cdot\ln(2/\varepsilon')}{\delta^2}$ *where* ℓ *is the length of the binary fingerprinting code* $\mathtt{F} = (\mathbf{CodeGen}^{\ell}, \mathbf{Identify})$ *which is a* (ε_f, w)*-identifier.*

Note that the tracing round complexity is a function of ℓ, the length of the underlying fingerprinting code, the parameter δ required for the adversary's admissibility to approximate the success in decryption, and finally ε', the security parameter of the tracing scheme.

Instantiation 1. In our first instantiation of the tracer, we consider the optimal codes of [33]. This provides a tracing against any size of traitor coalition for $O(n^2\delta^{-2}\log\frac{1}{\varepsilon'}\log\frac{n}{\varepsilon_f})$ number of queries submitted by the tracer where n is the number of receivers and ε_f is the security parameter of the Tardos code. The bound follows from Theorem 3 and the fact that the length of Tardos' codes is $O(n^2\log(n/\varepsilon_f))$ where n is the number of codewords (that in our setting matches the number of receivers). Note that this scheme tolerates an *unlimited* number of traitors and revocations.

Instantiation 2. Our second instantiation employs again Tardos' codes but assuming an upper bound on the number of traitors w. This provides for code length of $O(w^2\log\frac{n}{\varepsilon_f})$ and given that in our setting we have that n is the number of codewords that should be equal to the number of receivers using theorem 3 we obtain a tracing round complexity of $O(w^2\delta^{-2}\log\frac{1}{\varepsilon'}\log\frac{n}{\varepsilon_f})$, i.e., with only logarithmic dependency on the number of receivers in the system.

In the above, we can set $\varepsilon_f = \varepsilon/2$ and $\varepsilon = \frac{\varepsilon'}{2\ell}$, thus obtaining the round complexity of $\frac{3\ell\cdot\ln(4\ell/\varepsilon)}{\delta^2}$. Such selection will yield the results of the table 1(by neglecting $\log n/\varepsilon_f$ as the asymptotic complexity already includes $\frac{n^2\log n/\varepsilon_f}{\varepsilon_f}$).

References

1. Boneh, D., Naor, M.: Traitor Tracing with Constant Size Ciphertext. In: ACM Conference on Computer and Communications Security, pp. 501–510. ACM, New York (2008)
2. Berkman, O., Parnas, M., Sgall, J.: Efficient dynamic traitor tracing. In: SODA 2000, pp. 586–595 (2000)
3. Boneh, D., Franklin, M.: An Efficient Public-Key Traitor Tracing Scheme. In: Wiener, M. (ed.) CRYPTO 1999. LNCS, vol. 1666, pp. 338–353. Springer, Heidelberg (1999)

4. Boneh, D., Sahai, A., Waters, B.: Fully Collusion Resistant Traitor Tracing with Short Ciphertexts and Private Keys. In: Vaudenay, S. (ed.) EUROCRYPT 2006. LNCS, vol. 4004, pp. 573–592. Springer, Heidelberg (2006)
5. Boneh, D., Shaw, J.: Collusion-Secure Fingerprinting for Digital Data. IEEE Transactions on Information Theory 44(5), 1897–1905 (1998)
6. Chor, B., Fiat, A., Naor, M.: Tracing traitors. In: Desmedt, Y.G. (ed.) CRYPTO 1994. LNCS, vol. 839, pp. 257–270. Springer, Heidelberg (1994)
7. Chor, B., Fiat, A., Naor, M., Pinkas, B.: Tracing Traitors. IEEE Transactions on Information Theory 46(3), 893–910 (2000)
8. Chabanne, H., Hieu Phan, D., Pointcheval, D.: Public Traceability in Traitor Tracing Schemes. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 542–558. Springer, Heidelberg (2005)
9. Cramer, R., Shoup, V.: Universal Hash Proofs and a Paradigm for Adaptive Chosen Ciphertext Secure Public-Key Encryption. In: Knudsen, L.R. (ed.) EUROCRYPT 2002. LNCS, vol. 2332, pp. 45–64. Springer, Heidelberg (2002)
10. Dodis, Y., Fazio, N., Kiayias, A., Yung, M.: Scalable public-key tracing and revoking. In: PODC 2003, Proceedings of the Twenty-Second ACM Symposium on Principles of Distributed Computing (PODC 2003), Boston, Massachusetts, July 13-16 (2003)
11. Dwork, C., Naor, M., Reingold, O., Rothblum, G.N., Vadhan, S.P.: On the complexity of differentially private data release: efficient algorithms and hardness results. In: STOC 2009, pp. 381–390 (2009)
12. Feige, U., Raghavan, P., Peleg, D., Upfal, E.: Computing with Noisy Information. SIAM J. Comput. 23(5), 1001–1018 (1994)
13. Fiat, A., Tassa, T.: Dynamic Traitor Tracing. Journal of Cryptology 4(3), 211–223 (2001)
14. Gafni, E., Staddon, J., Lisa Yin, Y.: Efficient Methods for Integrating Traceability and Broadcast Encryption. In: Wiener, M. (ed.) CRYPTO 1999. LNCS, vol. 1666, pp. 372–387. Springer, Heidelberg (1999)
15. Jin, H., Lotspiech, J.: Renewable Traitor Tracing: A Trace-Revoke-Trace System For Anonymous Attack. In: Biskup, J., López, J. (eds.) ESORICS 2007. LNCS, vol. 4734, pp. 563–577. Springer, Heidelberg (2007)
16. Jin, H., Lotspiech, J.B.: Unifying Broadcast Encryption and Traitor Tracing for Content Protection. In: ACSAC 2009, pp. 139–148 (2009)
17. Jin, H., Lotspiech, J., Nusser, S.: Traitor tracing for prerecorded and recordable media. In: Digital Rights Management Workshop 2004, pp. 83–90 (2004)
18. Kiayias, A., Pehlivanoglu, S.: Pirate Evolution: How to Make the Most of Your Traitor Keys. In: Menezes, A. (ed.) CRYPTO 2007. LNCS, vol. 4622, pp. 448–465. Springer, Heidelberg (2007)
19. Kiayias, A., Pehlivanoglu, S.: Tracing and Revoking Pirate Rebroadcasts. In: Abdalla, M., Pointcheval, D., Fouque, P.-A., Vergnaud, D. (eds.) ACNS 2009. LNCS, vol. 5536, pp. 253–271. Springer, Heidelberg (2009)
20. Kiayias, A., Yung, M.: Self Protecting Pirates and Black-Box Traitor Tracing. In: Kilian, J. (ed.) CRYPTO 2001. LNCS, vol. 2139, pp. 63–79. Springer, Heidelberg (2001)
21. Kiayias, A., Yung, M.: On Crafty Pirates and Foxy Tracers. In: Sander, T. (ed.) DRM 2001. LNCS, vol. 2320, pp. 22–39. Springer, Heidelberg (2002)
22. Kiayias, A., Yung, M.: Traitor Tracing with Constant Transmission Rate. In: Knudsen, L.R. (ed.) EUROCRYPT 2002. LNCS, vol. 2332, pp. 450–465. Springer, Heidelberg (2002)

23. Kurosawa, K., Desmedt, Y.: Optimum Traitor Tracing and Asymmetric Schemes. In: Nyberg, K. (ed.) EUROCRYPT 1998. LNCS, vol. 1403, pp. 145–157. Springer, Heidelberg (1998)
24. Naor, D., Naor, M., Lotspiech, J.B.: Revocation and Tracing Schemes for Stateless Receivers. In: Kilian, J. (ed.) CRYPTO 2001. LNCS, vol. 2139, pp. 41–62. Springer, Heidelberg (2001)
25. Naor, M., Pinkas, B.: Threshold Traitor Tracing. In: Krawczyk, H. (ed.) CRYPTO 1998. LNCS, vol. 1462, pp. 502–517. Springer, Heidelberg (1998)
26. Naor, D., Naor, M., Lotspiech, J.B.: Revocation and Tracing Schemes for Stateless Receivers. In: Electronic Colloquium on Computational Complexity (ECCC), vol. 43 (2002)
27. Naor, M., Pinkas, B.: Efficient Trace and Revoke Schemes. In: Frankel, Y. (ed.) FC 2000. LNCS, vol. 1962, pp. 1–20. Springer, Heidelberg (2001)
28. Hieu Phan, D., Safavi-Naini, R., Tonien, D.: Generic Construction of Hybrid Public Key Traitor Tracing with Full- Public-Traceability, pp. 264–275.
29. Safavi-Naini, R., Wang, Y.: Sequential Traitor Tracing. In: Bellare, M. (ed.) CRYPTO 2000. LNCS, vol. 1880, pp. 316–332. Springer, Heidelberg (2000)
30. Safavi-Naini, R., Wang, Y.: Traitor Tracing for Shortened and Corrupted Fingerprints. In: Feigenbaum, J. (ed.) DRM 2002. LNCS, vol. 2696, pp. 81–100. Springer, Heidelberg (2003)
31. Silverberg, A., Staddon, J., Walker, J.L.: Efficient Traitor Tracing Algorithms Using List Decoding. In: Boyd, C. (ed.) ASIACRYPT 2001. LNCS, vol. 2248, pp. 175–192. Springer, Heidelberg (2001)
32. Staddon, J.N., Stinson, D.R., Wei, R.: Combinatorial Properties of Frameproof and Traceability Codes. IEEE Transactions on Information Theory 47(3), 1042–1049 (2001)
33. Tardos, G.: Optimal probabilistic fingerprint codes. In: ACM 2003, pp. 116–125 (2003)