Self Protecting Pirates and Black-Box Traitor Tracing

Aggelos Kiayias[1] and Moti Yung[2]

[1] Graduate Center, CUNY, NY USA,
akiayias@gc.cuny.edu
[2] CertCo, NY USA
moti@cs.columbia.edu

Abstract. We present a new generic black-box traitor tracing model in which the pirate-decoder employs a self-protection technique. This mechanism is simple, easy to implement in any (software or hardware) device and is a natural way by which a pirate (an adversary) which is black-box accessible, may try to evade detection. We present a necessary combinatorial condition for black-box traitor tracing of self-protecting devices. We constructively prove that any system that fails this condition, is incapable of tracing pirate-decoders that contain keys based on a superlogarithmic number of traitor keys. We then combine the above condition with specific properties of concrete systems. We show that the Boneh-Franklin (BF) scheme as well as the Kurosawa-Desmedt scheme have no black-box tracing capability in the self-protecting model when the number of traitors is superlogarithmic, unless the ciphertext size is as large as in a trivial system, namely linear in the number of users. This partially settles in the negative the open problem of Boneh and Franklin regarding the general black-box traceability of the BF scheme: at least for the case of superlogarithmic traitors. Our negative result does not apply to the Chor-Fiat-Naor (CFN) scheme (which, in fact, allows tracing in our self-protecting model); this separates CFN black-box traceability from that of BF. We also investigate a weaker form of black-box tracing called single-query "black-box confirmation." We show that, when suspicion is modeled as a confidence weight (which biases the uniform distribution of traitors), such single-query confirmation is essentially not possible against a self-protecting pirate-decoder that contains keys based on a superlogarithmic number of traitor keys.

1 Introduction

The problem of Traitor Tracing can be understood best in the context of Pay-TV. In such a system there are n subscribers, each one possessing a decryption box (decoder). The authority scrambles digital data and broadcasts it to all subscribers, who use their decryption boxes to descramble the data. It is possible for some of the users to collude and produce a pirate decoder: a device not registered with the authority that can decrypt the scrambled digital content. The goal of Traitor Tracing is to provide a method so that the authority, given

J. Kilian (Ed.): CRYPTO 2001, LNCS 2139, pp. 63–79, 2001.

a pirate decoder, is able to recover the identity of some of the legitimate users that participated in the construction of the decoder (traitors). In such a system piracy would be reduced due to the fear of exposure.

A standard assumption is that each user's decoder is "open" (to the user) so that the decryption key is recoverable. A set of users can combine their keys in order to construct a pirate decoder. It is immediately clear that each user should have a distinct private key, otherwise distinguishing traitors from non-traitors would be impossible. Given the contents of a pirate decoder the authority should be able to recover one of the traitors' keys. A scheme that allows this, is called a Traitor Tracing Scheme (TTS). A standard measure of the efficiency of a TTS is the size of the ciphertexts. Constructing a TTS with linear (in the number of users) ciphertexts is trivial; as a result the focus is on how to achieve traitor tracing when the ciphertext size is sublinear in the number of users. An additional requirement for TTSs is black-box traitor tracing, namely, a system where tracing is done using only black-box access to the pirate decoder (namely, only an input/ output access is allowed). To keep tracing cheap, it is extremely desirable that the tracing algorithm is black-box.

Previous Work

Let us first review the work of the various notions of traitor tracing. Traitor Tracing was introduced in [CFN94,CFNP00], with the presentation of a generic TTS. Explicit constructions based on combinatorial designs were given in [SW98b]. A useful variation of the [CFN94] scheme was presented in [NP98]. Public key Traitor Tracing Schemes based on ElGamal encryption were presented in [KD98, BF99]. In most settings (here also) it is assumed that the tracing authority is trusted (i.e. the authority does not need to obtain a *proof* that a certain user is a traitor); the case where the authority is not trusted was considered in [Pfi96, PS96,PW97]. An online approach to tracing, targeting pirate re-broadcasting (called dynamic traitor tracing) was presented in [FT99]. A method of discouraging users from sharing their decryption keys with other parties, called self-enforcement, was introduced in [DLN96]. A traitor tracing scheme along the lines of [KD98,BF99] combining self-enforcement and revocation capabilities was presented in [NP00]. Further combinatorial constructions of traitor tracing schemes in combination with revocation methods were discussed in [GSY99].

Previous work on black-box traitor tracing is as follows: a black-box traitor tracing scheme successful against any resettable[1] pirate decoder was presented in [CFN94,CFNP00]. In [BF99], a black-box traitor tracing scheme was presented against a restricted model called "single-key pirates": the pirate-decoder uses a single key for decryption *without* any other side computation (note that this single key could have been a combination of many traitors' keys). In the same paper, a weaker form of black-box traitor tracing was presented: "black-box confirmation." In this setting the tracer has a set of suspects and it wants to confirm that the traitors that constructed the pirate decoder are indeed included in the set of suspects. The work in [BF99] presented a single-query black-box confir-

[1] A pirate decoder is called resettable if the tracer has a means of resetting the device to its initial state for each trial.

mation method: using a single query to the pirate decoder the tracer solves the problem; multiple queries may be used to increase confidence. Black-box confirmation can be used for general tracing by trying all possible subsets. However the resulting traitor tracing algorithm needs exponential time (unless the number of traitors is a constant). In [Pfi96], a piracy prevention behavior was noted, dealing with the possibility of pirate decoders shutting down whenever an invalid ciphertext (used for tracing, perhaps) is detected. In [BF01] a combination of black-box confirmation and tracing appeared: extending the methods of [BF99] it was shown how one can trace within the suspect set (which is assumed to include all traitors) and recover one of the traitors. In addition, a new mode of black box tracing was considered in [BF01] called minimal access black box tracing: for any query to the pirate decoder, the tracer does not obtain the plaintext but merely whether the pirate-decoder can decrypt the ciphertext and "play" it or not (e.g. the case of a pirate cable-box incorporating a TV-set).

Our Results

The Model: Our perspective on black-box traitor tracing is as follows: under normal operation all users decrypt the same message; we say that in this case all users are *colored* in the same way. As we will see, in order to trace a pirate decoder in a black-box manner we have to disrupt this uniformity: color the users using more than one color. A ciphertext that induces such a coloring over the user population, will be called an "invalid" ciphertext. Tracing algorithms will have to probe with invalid ciphertexts (we assume our tracing methods to be aware of this fact). We consider a simple self-protection mechanism that can be used by any pirate decoder in order to detect tracing: *before decrypting, the pirate decoder computes the projection of the induced coloring onto the set of traitor keys (for some systems the stored keys can actually be combinations of traitor keys). If the traitor keys are colored by two colors or more, then the decoder knows that it is probed by the tracer, and can take actions to protect itself.* Computing the projection of the coloring onto the traitor keys is typically not a time-consuming operation and can be implemented within any software or hardware pirate decoder: prior to giving output the pirate decoder decrypts the given input with all available traitor keys (or combinations thereof) that are stored in its code. Since the decoder is black-box accessible, the presence of the keys internally, does not reduce its evasion power.

Necessary Combinatorial Condition and Negative Results: By adding the above simple self-protecting mechanism to the capabilities of pirate decoders together with an appropriate reaction mechanism we present a condition that has to be satisfied by any TTS in order to be able to black-box trace a pirate decoder that contains $\omega(\log n)$ traitor keys. Namely, the condition that *most users should be colored in the same way.* If this is not the case, we present a strategy that can be followed by a pirate decoder of any type (involving the previously stated self-protection mechanism) that defeats any black-box tracing method with high probability, assuming randomly chosen traitors.

Necessary Condition and Negative Results for Confirmation: The assumption above which underlies our negative result is that the choice of keys

available to the pirate is randomly distributed over the keys of the user population, i.e. the tracer has no a-priori idea about the identities of the traitors. In the context of black-box confirmation the situation is different because it is assumed that the tracer has a set of suspects, that are traitors with higher probability compared to a user chosen at random. We formalize this setting (differently from [BF01]) by assigning a "confidence level" function to the set of suspects that measures the amplification of the probability that a user is a traitor given that he belongs to the suspect set. Using this formalization we show that single-query black-box confirmation fails against any pirate-decoder, provided that the decoder contains a superlogarithmic number of traitor keys, and the confidence level of the tracer is below a certain (explicitely defined) threshold. We note that the confidence level exhibits a trade-off with the size of the suspect set, i.e. for small suspect sets, the confidence of the tracer should be very high in order to be successful in black-box confirmation. An immediate corollary of our result is that single-query black-box confirmation can be successful against decoders including a superlogarithmic number of traitor keys only in the case that the confidence level of the tracer is so high that *the probability that a user is a suspect given that it is a traitor is arbitrarily close to 1.* Note that in this case, confirmation becomes quite localized (the tracer knows already that the suspect set contains all traitors with very high probability; this type of confirmation is covered in [BF01]).

Applying the Results to Concrete Systems: We continue by combining our negative results with specific properties of concrete schemes which we analyze. First, we consider the Boneh-Franklin scheme [BF99] which possesses many attractive properties (based on public key, small ciphertext size, deterministic tracing). We show that the scheme is incapable of black-box traitor tracing when there are $\omega(\log n)$ traitors in the self-protecting model, unless the scheme becomes trivial (i.e. with ciphertexts of size linear in the number of users). This partially (for the $\omega(\log n)$ traitor case) settles in the negative the open problem from [BF99] who asked whether [BF99] traceability can be extended to the general black-box traitor tracing model of [CFN94,CFNP00] (i.e. black-box tracing of *any* resettable pirate decoder). Note that this is not an inconsistency with the black-box traitor tracing methods of [BF99], since they apply tracing against pirate decoders of an explicit construction or against a constant number of traitors. Similar negative results hold for the scheme of [KD98]. We note that our negative results do not apply to the black-box tracing methods of [CFN94,CFNP00] since their scheme is proved to work against any resettable pirate-decoder by (obviously not coincidentally) using colorings that satisfy the condition we show to be necessary (most users are colored in the same way). Thus, our work can be seen as retrofitting a design criterion for the early work of [CFN94] and it provides a separation with respect to black-box traceability between [CFN94, CFNP00] and [BF99,KD98]. Additionally, we show that black-box confirmation fails for both [BF99,KD98] against a superlogarithmic number of traitors unless the confidence level of the tracer is extremely high. Note again that this is not

an inconsistency with the black-box confirmation result of [BF99] which allows the differently modeled tracer's confidence to be quite large.

Organization. To state negative results, careful modeling is required. We define Multicast Encryption Schemes and non-black-box traitor tracing in section 2, whereas in section 3 we formalize the concepts of black-box tracing and coloring, and we provide the groundwork for the rest of the paper. In section 4 we prove the necessary condition for black-box traitor tracing (section 4.1), and we identify families of TTSs that are incapable for black-box tracing (section 4.2). Black-box confirmation is discussed in section 4.3. The negative results regarding the black-box traceability of the [BF99] and [KD98] schemes in the "self-protecting" pirate-decoder model, are proven in section 5.1 and section 5.2 respectively.

2 Multicast Encryption Schemes

Any traitor tracing scheme is based on a Multicast Encryption Scheme (MES) – a cryptographic primitive we formalize in this section. Let $\mathcal{U} := \{1, \ldots, n\}$ be the set of users. Let $\{\mathcal{G}_w\}_{w \in \mathbb{N}}$ be some a family of sets of elements of length w (e.g. $\mathcal{G}_w = \{0,1\}^w$). For a certain w, we fix the following sets: the message space $\mathcal{M} \subseteq \mathcal{G}_w$; the ciphertext-space $\mathcal{C} \subseteq \mathcal{G}_w^v$; the user key-space $\mathcal{D} \subseteq \mathcal{G}_w^u$; v, u express the dimension of ciphertext space and user key space respectively over the message space. Without loss of generality we will assume that $u \leq v$ i.e. a user key does not have to be "longer" than a ciphertext (this is justified by all concrete MESs in the literature). Note that in a concrete MES $\mathcal{M}, \mathcal{C}, \mathcal{D}$ may be of slightly different structure e.g. in the [BF99]-scheme $\mathcal{M} \subseteq G_q$, $\mathcal{C} \subseteq G_q^v$ but $\mathcal{D} \subseteq Z_q^{v-1}$ (see section 5.1), but these differences are of minor importance here. A function $\sigma(n)$ will be called *negligible* if $\sigma(n) < n^{-c}$ for all c, for sufficiently large n. For brevity we make the assumption that 1^w is polynomially related to n. A Multicast Encryption Scheme (MES) is a triple (G, E, D) of probabilistic polynomial time algorithms with the following properties:

- Key Generation. On input 1^w and n, G produces a pair (e, K) with $K \subseteq \mathcal{D}$, $|K| = n$.
- Encryption. $c \leftarrow E(1^w, m, e)$; $m \in \mathcal{M}$, $e : (e, K) \leftarrow G(1^w, n)$, $(c \in \mathcal{C})$.
- Decryption. For any $m \in \mathcal{M}, (e, K) \leftarrow G(1^w, n)$, if $c \leftarrow E(1^w, m, e)$, then the probabilities $\mathbf{Prob}[m' \neq m : m' \leftarrow D(1^w, d, c)]$ and $\mathbf{Prob}[m' \neq m'' : m' \leftarrow D(1^w, c, d), m' \leftarrow D(1^w, c, d')]$ are negligible, for any keys $d, d' \in K$. The first probability states that incorrect decryption event is negligible whereas the second probability states that all user keys decrypt the same word but with negligible error.

Note that the above scheme can be either public or secret key. It is easy to adapt the standard notions of semantic security or chosen-ciphertext security for MESs.

Let $\mathcal{F}$ be the set of functions of $(\mathbb{N} \to \mathbb{N})$ s.t. $f \in \mathcal{F}$ if and only if f is non-decreasing and constructible (i.e., there is an algorithm M s.t. on input n,

M outputs the string $0^{f(n)}$). Moreover, for any $f, g \in \mathcal{F}$ it holds that either (a) $\exists n_0\ \forall n \geq n_0\ (f(n) = g(n))$ (b) $\exists n_0\ \forall n \geq n_0\ (f(n) > g(n))$ (c) $\exists n_0\ \forall n \geq n_0\ (f(n) < g(n))$ (i.e. it is possible to define a total order over $\mathcal{F}$). Since we are interested only in functions less than n, we assume that $\forall f \in \mathcal{F}$ it holds that $\forall n (f(n) \leq n)$. To facilitate traitor tracing, some additional security requirements have to be imposed.

Non-Triviality of Decryption. For any probabilistic polynomial time algorithm A the following probability is negligible for almost all messages m: $\mathbf{Prob}[m = m' : m' \leftarrow A(1^w, c); c \leftarrow E(1^w, m, e)]$. This property ensures that there are no "shortcuts" in the decryption process. Namely, decryption without access to a key amounts to reversing a one-way function, thus for effective decryption one needs some or a combination of the designated user keys.

Key-User correspondence. It should be guaranteed that each user does not divulge its own key; more generally that a user is responsible when its key is being used for decryption. This should apply to collusions of users as well. More specifically, given $t \in \mathcal{F}$, there should be no probabilistic polynomial-time algorithm working with non-negligible success probability that given the keys of a set of subscribers $d_{i_1}, \ldots, d_{i_k}$ with $k \leq t(n)$, and all other public information, and is able to compute one additional private key d_j with $j \notin \{i_1, \ldots, i_k\}$.

Non-Ambiguity of Collusions. The user keys are drawn from a key-space $\mathcal{D}_e$ defined for each encryption key e; i.e. $\mathcal{D}_e \subseteq \mathcal{G}_w^u$ contains all d that can be used to invert e. Obviously $\mathcal{D}_e \supseteq K$, if $(e, K) \leftarrow G(1^w, n)$. Then, the following holds: Given $t \in \mathcal{F}$; let A, B be probabilistic polynomial algorithms. Given T_1, T_2 two disjoint subsets of K, of cardinality less or equal to $t(n)$. Let I_1, I_2 be all private and public information available to T_1, T_2 correspondingly. Then the following probability is negligible $\mathbf{Prob}[d = d' \wedge (d \in \mathcal{D}_e) : d \leftarrow A(T_1, I_1, 1^w), d' \leftarrow B(T_2, I_2, 1^w)]$.

Non-ambiguity of collusions requires that two disjoint sets of users cannot generate the same decryption key. It is an essential property of any traitor-tracing scheme, since if it fails it is immediately possible to generate instances where tracing is impossible due to ambiguity.

Definition 1. Traitor Tracing Scheme (non-black-box). *Given $t, f, v \in \mathcal{F}$, a MES satisfying non-triviality of decryption, key-user correspondence for $t(n)$, non-ambiguity of collusions for $t(n)$ and, in addition, has $wv(n)$ ciphertext size, is called a $\langle t(n), f(n), v(n)\rangle$-Traitor Tracing Scheme (TTS) if there exists a probabilistic polynomial time algorithm B (tracing algorithm) s.t. for any set $T \subseteq K$, $(e, K) \leftarrow G(1^w, n)$, with $|T| \leq t(n)$ and any probabilistic polynomial time algorithm A that given T and all public information outputs $d \in \mathcal{D}_e$, it holds that:* $\mathbf{Prob}[\tau \in T : \tau \leftarrow B(d, K, 1^w), d \leftarrow A(T, 1^w)] \geq 1/f(n)$.

Because of key-user correspondence, the recovery of τ is equivalent to exposing a traitor. Note that in the non-black-box setting it is assumed that the decoder is "open" and because of the non-triviality of decryption a decryption key is available to the tracer. Black-Box Traitor Tracing Schemes where the tracing algorithm does not have access to keys (but only black box access to devices) are discussed in the next section.

3 Black-Box Traitor Tracing: Preliminaries

3.1 Colorings

Consider an MES with given $w, (e, K)$. A coloring of the user population is a partition $\cup_i C_i$ of $\mathcal{U}$. Let $s \in \mathcal{G}_w^v$ (an element from the extended ciphertext space) induces a coloring over $\mathcal{U}$ as follows: Define a relation over K: $d \equiv d'$ iff $D(1^w, d, s) = D(1^w, d', s)$. Note that if D is deterministic then this is an equivalence relation. The coloring can be defined as the set of all the equivalence classes of $\equiv$. If D is probabilistic (with negligible error) we define $\equiv$ as $d \equiv d'$ iff $\mathbf{Prob}[D(1^w, d, s) \neq D(1^w, d', s)]$ is negligible.

If $c \leftarrow E(1^w, m, e)$ for some $m \in \mathcal{M}$ (i.e., c is a "real or valid ciphertext") then it holds that for all $d, d' \in K$, $D(1^w, d, c) = D(1^w, d', c)$ (with high probability if D is probabilistic), therefore there is only one equivalence class induced by c, i.e. all users are colored by the same color (we call such a coloring *trivial*). Let $\mathcal{X}_1$ be the subset of $\mathcal{G}_w^v$ s.t. $\forall s \in \mathcal{X}_1$, s induces a trivial coloring (with negligible error). Obviously the valid ciphertexts constitute a subset of $\mathcal{X}_1$.

We say that an MES can induce a coloring $\cup_i C_i$ if there is an algorithm that produces a string s s.t. the string s induces the coloring $\cup_i C_i$ over the user population. Note that a decryption algorithm of some sort may not necessarily return one of the "color labels" i.e. the elements of the set $\{D(1^w, d, s) \mid d \in K\}$ (this can happen if the decryption algorithm operates with some "compound" decryption key – that has been derived from combining more than one of the users' keys).

3.2 Black-Box Traitor Tracing Schemes

The black-box tracing algorithm $\mathcal{R}$ and the pirate decoder algorithm $\mathcal{B}$ are probabilistic polynomial-time Turing machines with communication and output tapes. $\mathcal{B}$ incorporates a correct decoding algorithm: i.e. given a valid ciphertext it decrypts it, by running the decryption algorithm D with some key d that inverts e (note that d is not necessarily one of the user keys, but it is an element of $\mathcal{D}_e$ by the non-triviality of decryption property; also note that d may change from one decryption to the next). In the terminology of the previous section this means that if all traitor keys are colored in the same way the pirate decoder is bound to decrypt properly. If $\mathcal{B}$, on the other hand, finds that something is wrong with the encryption it may take measures to protect itself, e.g. it may return a random word. The set of user keys that are employed in the construction of $\mathcal{B}$ is denoted by $\mathcal{T}$ (due to key-user correspondence the set $\mathcal{T}$ can be also defined to be the set of traitor users). The tracing algorithm $\mathcal{R}$ is allowed oracle access to $\mathcal{B}$, namely, $\mathcal{R}$ can adaptively generate input strings s (queries) for $\mathcal{B}$ and $\mathcal{B}$, in response, will return a value (which is a correct decryption if s is a valid ciphertext).

From now on we will use the following notation: $\cup_{i=1}^{k(n)} C_i^n$ denotes a coloring induced over the user population by some s of $\mathcal{G}_w^v$; $c_i(n)$ will denote the cardinality of C_i^n. Note that for any n, it holds that $k(n), c_i(n) \in \{1, \ldots, n\}$; with this in mind we will use standard asymptotic notation to express the relation

of these functions to n, e.g. $k(n) = \Theta(n)$ means that the number of colors is linear in n etc. We make the assumption that the functions $k(n), c_i(n)$ that are related to colorings produced by $\mathcal{R}$ are always in $\mathcal{F}$. Note that occasionally we may suppress "(n)" and write k instead of $k(n)$ etc.

Definition 2. *For $t, f \in \mathcal{F}$, we say that a polynomial-time (in n) probabilistic algorithm $\mathcal{R}$ is a $\langle t(n), f(n)\rangle$-tracer if for any set of traitors $\mathcal{T} \subseteq \mathcal{U}$ s.t. $|\mathcal{T}| \leq t(n)$ and for any polynomial-time pirate-decoder algorithm $\mathcal{B}$ that was created using the keys of $\mathcal{T}$, $\mathcal{R}^{\mathcal{B}}$ given all user keys, outputs a user with non-negligible probability in n, who is in the traitor set with probability at least $1/f(n)$.*

In this paper we consider tracers R which are non-ambiguous, i.e., when they probe the decoder they know that their queries are valid ciphertexts or invalid ones.

We will refer to the function f as the *uncertainty* of the tracer. Obviously obtaining a tracer with $\Theta(n)$ uncertainty for any MES is very simple: merely output any user at random achieves that. The other extreme is a tracer with uncertainty $\Theta(1)$ (ideally uncertainty=1), that no matter how large is the user population it returns a traitor with constant probability of success.

Remark 3. Consider the tracing approach of accusing any user at random. As stated above this has linear uncertainty and is obviously not useful in any setting. Suppose now that we have a lower bound on the number of traitors $\omega(t'(n))$; the uncertainty of this tracing approach becomes $n/t'(n)$ which can be sublinear if $t'(n)$ is not a constant. Nevertheless because we would like to rule it out as a way of tracing we say that the uncertainty is still linear — and therefore not acceptable (but it is linear in $n' = n/t'(n)$ instead of n); abusing the notation we may continue to write that the uncertainty in this case is $\Theta(n)$).

Definition 4. *For some $t, f, v \in \mathcal{F}$, a $\langle t(n), f(n), v(n)\rangle$-Black-Box Traitor Tracing Scheme (BBTTS), is an MES that (1) satisfies key-user correspondence and non-ambiguity of collusions for $t(n)$, (2) satisfies non-triviality of decryption, (3) it has $v(n)w$ ciphertext size, and (4) there is an $\langle t(n), f(n)\rangle$-tracer so that all colorings used by the tracer can be induced by the MES.*

We say that an MES is *incapable of Black-Box Tracing collusions of size* $t(n)$ if any polynomial-time tracer $\mathcal{R}$ has linear uncertainty (i.e., it is a $\langle t(n), \Theta(n)\rangle$-tracer).

The proof technique for establishing the fact that a BBTTS is incapable of black-box traitor tracing is the following: for any tracer $\mathcal{R}$ that can be defined in the BBTTS there is another algorithm $\mathcal{R}'$ that operates *without* oracle access to $\mathcal{B}$ so that the outputs of $\mathcal{R}$ and $\mathcal{R}'$ are essentially identical (i.e. they can be different in at most a negligible fraction of all inputs). More specifically the oracle $\mathcal{B}$ can be simulated without knowing any information pertaining to $\mathcal{B}$. In such a case we will state that the tracer essentially operates without interacting with the decoder and as a result it will be immediate that it has linear uncertainty (similar to the fact that any algorithm trying to guess a result of a coin flip

without interacting with any agents which know the result of the coin cannot have probability greater than 1/2). A preliminary result on tracing follows; we show that strings that induce the trivial coloring over the user population are useless for tracing:

Proposition 5. *Queries which are elements of $\mathcal{X}_1$ do not help in reducing the uncertainty of a tracer.*

Proof. If the $\mathcal{R}$ algorithm uses an element of $\mathcal{X}_1$ for querying the pirate decoder then, the pirate decoder decrypts normally. This answer can be simulated by any decryption box. In particular, since the tracer is non-ambiguous it knows that it can generate the answer itself using any of the user keys (since it knows all user keys). □

We will assume that the number of traitors in any pirate decoder is sublinear in n, and as it is customary, we will give to the tracer the advantage of knowing a (sublinear) upper bound on the number of traitors. Additionally we would like to point out that our negative results on traitor tracing are not based on history-recording capabilities of the pirate decoder (i.e. $\mathcal{B}$ as an oracle does not have access to the previously asked queries). As a result the tracer is allowed to reset the decoder in its initial state after each query. In addition, our results apply even when the tracer has access to the randomness used by the pirate decoder.

4 Necessary Conditions for Black-Box Traitor Tracing

4.1 Combinatorial Condition

In this section we establish the fact that if the number of traitor keys is superlogarithmic in the user population size, it is not possible to trace without the decoder noticing it, unless queries of a specific type are used. We denote by $\cup_{i=1}^{k(n)} C_i^n \downarrow \mathcal{T}$ the projection of a coloring onto the traitor keys. Any pirate decoder can easily compute $\cup C_i^n \downarrow \mathcal{T}$; this is done by merely applying the decryption algorithm with each traitor key onto the given element s. Since this is a straightforward process we assume that any pirate decoder implements it. Obviously, if $\cup C_i^n \downarrow \mathcal{T}$ contains more than one color then the decoder "understands" it is being traced. In some systems, rather than projecting on individual traitor keys, one can project on combinations thereof (and thus reduce storage and computation requirements).

Theorem 6. *Suppose that a pirate decoder containing $t(n) = \omega(\log n)$ traitor keys, randomly distributed over all user keys, is given a query $s \in \mathcal{G}_w^v$ that induces a non-trivial coloring $\cup_{i=1}^{k(n)} C_i^n$ over the user population. Suppose further, that the coloring has the property $\neg(\exists i\ c_i(n) = n - o(n))$. Then, the probability that the pirate decoder does not detect it is being queried by the tracer is negligible in n.*

Proof. (recall that $|C_i^n| = c_i(n)$ for $i = 1, \ldots, k$; $c_1(n) + \ldots + c_k(n) = n$) Since $t(n)$ and $c_i(n)$ for $i = 1, \ldots, k$ are elements of $\mathcal{F}$, without loss of generality we assume that $c_i(n) \geq t(n)$ for all $i = 1, \ldots, \ell$ with $\ell \leq k$, for sufficiently large n. Obviously if $\ell = 0$ the decoder detects it is being traced.

Recall that we occasionally write t instead of $t(n)$ and c_i instead of $c_i(n)$. The total number of ways the pirate keys may be distributed over the user population are $\binom{n}{t}$. Similarly, the number of ways in which the decoder cannot detect that it is being traced is $\sum_{i=1}^{\ell} \binom{c_i}{t}$. The probability that the decoder cannot detect that it is being traced is $P := \frac{\sum_{i=1}^{\ell} \binom{c_i}{t}}{\binom{n}{t}} = \frac{(c_1)_t + \ldots + (c_\ell)_t}{(n)_t}$, where $(m)_v := m!/(m-v)!$. For sufficiently large n there will be a $m \in \{1, \ldots, \ell\}$ s.t. $c_m(n) \geq c_i(n)$ for all $i = 1, \ldots, k$.

The probability P is then: $\frac{(c_1)_t + \ldots + (c_\ell)_t}{(n)_t} \leq \frac{\ell (c_m)_t}{(n)_t} \leq \frac{n (c_m)_t}{(n)_t}$. Therefore we only need to show that $(c_m(n))_t/(n)_t$ is negligible in n. We consider two sub-cases:
(i) There exists a real number $\alpha > 1$ such that $n \geq \alpha c_m(n)$ for sufficiently large n. Then, $(c_m)_t/(n)_t \leq (c_m)_t/(\alpha c_m)_t$. It holds that $\frac{c_m - i}{\alpha c_m - i} \leq \frac{1}{\alpha}$ for any $i = 0, \ldots, t-1$, (recall that $c_m \geq t$). Then $(c_m)_t/(n)_t \leq 1/\alpha^t$ which obviously is negligible since $\alpha > 1$ and $t = \omega(\log n)$: in details, $1/\alpha^t < 1/n^d$ for any constant d and sufficiently large n; equivalently $n^d < \alpha^t$ or $\alpha^{d \log_\alpha n} < \alpha^t$ or $t := t(n) > d \log_\alpha n$, which is true since $t(n) = \omega(\log n)$.
(ii) There is no $\alpha > 1$ with $n \geq \alpha c_m(n)$. Since $c_m(n) \leq n$ though, there has to be a function $f(n) \in \mathcal{F}$ s.t. $c_m(n) = n - f(n)$. If $f(n) = \Theta(n)$ there is a $0 < \beta \leq 1$ s.t. $f(n) \geq \beta n$. The case $\beta = 1$ is not possible since we deal with elements which induce coloring and $c_m = 0$ is impossible. In the case $\beta < 1$ we have that $n - f(n) \leq n - \beta n$ or equivalently $n \geq 1/(\beta - 1) \cdot c_m(n)$ therefore we are in case (i) since $1/(\beta - 1) > 1$ (i.e. $\alpha := 1/(\beta - 1)$). Finally if $f(n) = o(n)$ we fall into the case excluded by the theorem. □

The Theorem asserts that a decoder detects that it is being queried unless *most users are colored in the same way.* Namely, the negation of the Theorem's condition $\neg(\exists i \; c_i(n) = n - o(n))$ is that there is an i s.t. almost all users are colored in the same way ($c_i(n) = n - o(n)$). By "almost all" we mean that $c_i(n)/n \to 1$ when $n \to \infty$.

4.2 Negative Results

In this section we discuss how a pirate decoder can take advantage of Theorem 6 in order to protect itself. Specifically we show that there is a deterministic self-protecting strategy for any pirate decoder: *when the pirate decoder detects tracing it returns "0" (a predetermined output).* This strategy is successful for decoders containing enough traitor keys. The next Theorem asserts that any BBTTS whose underlying MES can only produce ciphertexts that are either valid or do not color most users in the same way (as discussed in the previous section) has $\Theta(n)$ uncertainty for any pirate decoder that incorporates $t(n) = \omega(\log n)$ traitor keys.

Theorem 7. *Given an $\langle t(n), f(n), v(n)\rangle$-BBTTS s.t. the underlying MES can only induce colorings $\cup_{i=1}^{k(n)} C_i^n$ with the property $(k(n) = 1) \vee \neg(\exists i\ c_i(n) = n - o(n))$ then it holds that if $t(n) = \omega(\log n)$ then $f(n) = \Theta(n)$.*

Proof. Assume that the decoder employs $t(n)$ traitor keys. The algorithm followed by the decoder is the following: before decrypting, it computes $\cup C_i^n \downarrow \mathcal{T}$. If all traitor keys are colored in the same way, it decrypts using any key. If there is more than one color the decoder returns "0".

The coloring conditions on the MES assures that an invalid ciphertext will be detected by the pirate decoder based on Theorem 6. Consequently the decoder on an invalid ciphertext will return "0" with overwhelming probability. On the other hand, any element in $\mathcal{X}_1$ will be properly decrypted. Since the tracer is non-ambiguous, the oracle can be simulated with overwhelming probability. So the tracer essentially operates without interacting with the decoder. By remark 3 the uncertainty of the scheme is $\Theta(n)$. □

The pirate decoder strategy used in the proof above can be defeated by a tracer that is able to produce colorings s.t. $n - o(n)$ users are colored in the same way. This is achieved in the MES of [CFNP00], and a black-box traitor tracing method which uses such colorings is presented there.

4.3 Negative Results for Black-Box Confirmation

Black-Box Confirmation is an alternative form of revealing some information about the keys hidden in the pirate decoder. Suppose that the tracer has some information that traitors are included in a set of suspects $\mathcal{S}$ and wants to confirm this. The fact that the tracer has some information about the traitor keys means that they are not randomly distributed over all users' keys and therefore Theorem 6 is not applicable (in fact, biasing the distribution of a potential adversary is, at times, a way to model suspicion). Under such modeling, we can show a strong negative result for single-query black-box confirmation, i.e. when a single query is sent to the pirate-decoder that induces the same color on the suspects and different color(s) on other users. If the pirate decoder returns the color label associated to the suspect set then the suspicion is confirmed (note that this is exactly the black-box confirmation method used in [BF99]).

The change of the distribution of the traitor keys can be modeled as follows: the probability $\mathbf{Prob}[i \in \mathcal{T} | i \in \mathcal{S}] = \alpha(n)\mathbf{Prob}[i \in \mathcal{T}]$ where $\alpha(n) > 1$ for sufficiently large n; note that when the tracer has no information it holds that $\alpha(n) = 1$. Let us fix t the size of the traitor set. We will denote the distribution of t-sets of potential traitor keys by $\mathcal{D}_{\mathcal{S},\alpha}$, and refer to $\alpha(n)$ as the advantage of the tracer. For example, for $t = 1$ the probability of all $\mathcal{T}$ inside $\mathcal{S}$ is α/n, whereas the probability of all other $\mathcal{T}$'s is $\frac{n-\alpha s}{n(n-s)}$. As usual, we allow the tracer to know an upper bound on the number of traitors' keys and therefore $|\mathcal{S}| \geq |\mathcal{T}|$.

Lemma 8. *Let $\mathcal{S}$ be a set of users such that $s(n) := |\mathcal{S}|$ and an $\alpha(n) \in \mathcal{F}$ such that $s(n)\alpha(n) \leq cn$ for some $c \in (0, 1)$. Suppose that a pirate decoder employing*

$t(n) = \omega(\log n)$ traitor keys, distributed according to $\mathcal{D}_{\mathcal{S},\alpha}$, is given a query that induces the following coloring over the user population: the users in $\mathcal{S}$ are colored in the same way and the remaining users in different color(s). Then, the probability that the traitor set is included in the suspect set is negligible in n.

Proof. For simplicity we write s, α instead of $s(n), \alpha(n)$ respectively. We show that the probability $\mathbf{Prob}[\mathcal{T} \subseteq \mathcal{S}]$, when $\mathcal{T}$ is distributed according to $\mathcal{D}_{\mathcal{S},\alpha}$, is negligible.

It is easy to see that $\mathbf{Prob}[\mathcal{T}] = \alpha^t / \binom{n}{t}$ (when $\mathcal{T}$ is distributed according to $\mathcal{D}_{\mathcal{S},\alpha}$ and $\mathcal{T} \subseteq \mathcal{S}$), and as a result $\mathbf{Prob}[\mathcal{T} \subseteq \mathcal{S}] = \alpha^t \binom{s}{t} / \binom{n}{t}$. The fact that $s\alpha \leq cn$ implies $\frac{\alpha(s-i)}{n-i} \leq c$ for any $i > 0$; as a result it holds that $\alpha^t \binom{s}{t} / \binom{n}{t} < c^t$. Since $0 < c < 1$ and $t = \omega(\log n)$ the probability is negligible. □

Theorem 9. *Single-Query Black-Box Confirmation with a suspect set $\mathcal{S}$ and confidence $\alpha(n)$ is not possible against any pirate-decoder which contains $t(n) = \omega(\log n)$ traitor keys, provided that $|\mathcal{S}|\alpha(n) \leq cn$ for some constant $c \in (0, 1)$.*

Proof. Suppose that the pirate decoder returns "0" when it detects an invalid ciphertext. Then, by lemma 8 with overwhelming probability not all the traitors are in the suspect set, thus the pirate decoder will return the color label of the suspect set with negligible probability in n. As a result single-query black-box confirmation will fail. □

Note the trade-off between the size of $\mathcal{S}$ and the advantage $\alpha(n)$. How large should be the advantage of the tracer so that single-query black-box confirmation is possible? it should hold that $\alpha(n)|\mathcal{S}| = n - o(n)$. In this case it holds that $\mathbf{Prob}[i \in \mathcal{S} | i \in \mathcal{T}] = \mathbf{Prob}[(i \in \mathcal{S}) \wedge (i \in \mathcal{T})] / \mathbf{Prob}[i \in \mathcal{T}] = \alpha(n)\mathbf{Prob}[(i \in \mathcal{S}) \wedge (i \in \mathcal{T})] / \mathbf{Prob}[i \in \mathcal{T} | i \in \mathcal{S}] = \alpha(n)\mathbf{Prob}[i \in \mathcal{S}] \to 1$, when $n \to \infty$ (under the condition that $\alpha(n)|\mathcal{S}| = n - o(n)$). This, together with the above Theorem imply:

Corollary 10. *Single-query Black-box confirmation is impossible against any pirate decoder that includes $t(n) = \omega(\log n)$ traitor keys, unless the probability that a user is a suspect given that it is a traitor is arbitrarily close to 1.*

Some remarks should be placed herein: (1) $\mathbf{Prob}[i \in \mathcal{S} | i \in \mathcal{T}]$ is arbitrarily close to 1, means that the confidence level of the tracer is so high that it "forces" $\mathcal{T}$ to be a subset of $\mathcal{S}$ (for more discussion on confirmation in this case and the relation to the black-box confirmation results of [BF01] see subsection 5.1). (2) We do not rule-out black-box confirmation with smaller confidence levels in different models or by multiple-queries that do not directly color the suspect set in a single color and the remaining users differently.

5 From Necessary Conditions to Concrete Systems

In this section, we apply our generic necessary condition results to concrete systems. We actually analyze specific properties of the schemes of [BF99,KD98];

these properties in combination with the generic results reveal inherent black-box tracing limitations of these schemes in the self-protecting model. This demonstrates that these schemes are, in fact, sensitive to the self-protection property of our model and the number of traitors. This shows the power of the self-protecting pirate model, since in more restricted pirate models (restricting the power of the pirate decoder or the number of traitors) tracing was shown possible, whereas we get negative results for the more general model defined here. We note that below we will assume that self-protection involves decryption with traitor keys. However, achieving self-protection using a linear combination of traitor keys is possible as well; in which case the actual traitor keys are not necessarily stored and the storage as well as the computation of the pirate can be reduced.

Our results can be seen as a separation of the schemes of [BF99,KD98] and the scheme of [CFN94,CFNP00] with respect to black-box traceability. In the latter scheme our self-protection method fails to evade tracing, since the ciphertext messages induce colorings which fall into the exception case of Theorem 6 and the tracing method, in fact, employs such ciphertexts.

5.1 The [BF99]-Scheme

Description. We present the basic idea of the Boneh and Franklin scheme [BF99]. All base operations are done in a multiplicative group G_q in which finding discrete logs is presumed hard, whereas exponent operations are done in Z_q. Vectors (denoted in bold face) are in Z_q^v and $\boldsymbol{a} \cdot \boldsymbol{b}$ denotes the inner product of $\boldsymbol{a}$ and $\boldsymbol{b}$. Given a set $\Gamma := \{\boldsymbol{\gamma_1}, \ldots, \boldsymbol{\gamma_n}\}$ where $\boldsymbol{\gamma_i}$ is a vector of length v, and given random $\boldsymbol{r} := \langle r_1, \ldots, r_v \rangle$ and $c \in Z_q$, we select $\boldsymbol{d_i} = \theta_i \boldsymbol{\gamma_i}$, $i = 1, \ldots, n$ such that $\forall i \;\; \boldsymbol{r} \cdot \boldsymbol{d_i} = c$, where n is the number of users (i.e. we select $\theta_i := c/(\boldsymbol{r} \cdot \boldsymbol{\gamma_i})$). The vector $\boldsymbol{\gamma_i}$ is selected as the i-th row of an $(n \times v)$-matrix B where the columns of B form a base for the null space of A, where A is an $(n-v) \times n$ matrix where the i-th row of A is the vector $\langle 1^i, 2^i, \ldots, n^i \rangle$, $i = 0, \ldots, n-v-1$.

The public key is $\langle y, h_1, \ldots, h_v \rangle$, where $h_j = g^{r_j}$ and $y = g^c$, where g is a generator of G_q. Note that all vectors $\boldsymbol{d_i}$ are representations of y w.r.t the base $h_1, \ldots, h_v$. Vector $\boldsymbol{d_i}$ is the secret key of user i. Encryption is done as follows: given a message $M \in G_q$, a random $a \in Z_q$ is selected and the ciphertext is $\langle My^a, h_1^a, \ldots, h_v^a \rangle$. Given a ciphertext, decryption is done by applying $\boldsymbol{d_i}$ to the "tail" of the ciphertext: $h_1^a, \ldots, h_v^a$ pointwise, in order to obtain y^a by multiplication of the resulting points, and then M is recoverable by division (cf. ElGamal encryption). In [BF99] a tracing algorithm is presented showing that the scheme described above is a $\langle t(n), 1, 2t(n) \rangle$-TTS. It is also shown that their scheme is black-box against pirate decoders of specific implementations ("single-key pirate", "arbitrary pirates"). We next investigate further black-box capabilities of the [BF99]-scheme.

Negative Results. Suppose that we want to induce a coloring $\cup_{i=1}^{k(n)} C_i^n$ in the [BF99] scheme. Given a (possibly invalid) ciphertext $\langle C, g^{r_1 x_1}, \ldots, g^{r_v x_v} \rangle$, user i decrypts as follows: $C/g^{r_1 x_1 (\boldsymbol{d_i})_1 + \ldots + r_v x_v (\boldsymbol{d_i})_v}$. Thus, we can color user i by the color label $C/g^{\theta_i c_i}$ (the value of the decryption by the user) provided that we

find the $x_1, \ldots, x_v$ such that $r_1 x_1 (\boldsymbol{d_i})_1 + \ldots + r_v x_v (\boldsymbol{d_i})_v = \theta_i c_i$. This can be done by finding a $\boldsymbol{z} := \langle z_1, \ldots, z_v \rangle$ s.t. $\boldsymbol{\gamma_i} \cdot \boldsymbol{z} = c_i$ for all $i = 1, \ldots, n$. Given such a $\boldsymbol{z}$ we can compute the appropriate x-values to use in the ciphertext as follows: $x_j = z_j (r_j)^{-1}$ for $j = 1, \ldots, v$. Note that for valid ciphertexts it holds that $\boldsymbol{z} = a\boldsymbol{r}$ for some $a \in Z_q$ (and as a result $x_1 = \ldots = x_v = a$).

Next we present a property of the Boneh-Franklin scheme, showing that an invalid ciphertext (namely, a ciphertext which induces more than one color), cannot color too many users by the same color.

Theorem 11. *In the [BF99]-MES, given a (possibly invalid) ciphertext that induces a coloring over the user population so that v users are labelled by the same color then all users are labelled by the same color.*

Proof. Suppose that the ciphertext $\langle C, g^{r_1 x_1}, \ldots, g^{r_v x_v} \rangle$ colors user i by label $C/g^{\theta_i c_i}$ to user i, and that v users are colored by the same label. Let $c'_i := \boldsymbol{r} \cdot \boldsymbol{\gamma_i}$, for $i = 1, \ldots, n$. Without loss of generality assume that users $1, \ldots, v$ are colored by the same label. Then it holds that $\theta_1 c_1 = \ldots = \theta_v c_v$ or equivalently $c_1/c'_1 = \ldots = c_v/c'_v$. Let $a := c_1/c'_1$. Then we have that $c_1 = ac'_1, \ldots, c_v = ac'_v$.

Define $\boldsymbol{z} = \langle z_1, \ldots, z_v \rangle$ s.t. $z_j = r_j x_j$ for $j = 1, \ldots, v$. It follows that $\boldsymbol{\gamma_i} \cdot \boldsymbol{z} = c_i$, for $i = 1, \ldots, n$ (we call this system of equations system 1). Because it holds that $\boldsymbol{\gamma_i} \cdot (a\boldsymbol{r}) = ac'_i$ for $i = 1, \ldots, v$ (and this will hold for any v users) it follows that $\boldsymbol{z} = a\boldsymbol{r}$ provided (which we show next) that $\boldsymbol{\gamma_1}, \ldots, \boldsymbol{\gamma_v}$ are linearly independent (since in this case system 1 is of full rank, and as a result it has a unique solution). Since $\boldsymbol{z} = a\boldsymbol{r}$ it follows that $x_1 = \ldots = x_v = a$, i.e. the ciphertext $\langle C, g^{r_1 x_1}, \ldots, g^{r_v x_v} \rangle$ is valid.

To complete the proof we have to show that any v vectors of $\Gamma = \{\boldsymbol{\gamma_1}, \ldots, \boldsymbol{\gamma_n}\}$ are linearly independent. Suppose, for the sake of contradiction, that $\boldsymbol{\gamma_1}, \ldots, \boldsymbol{\gamma_v}$ are linearly dependent. Recall that $\boldsymbol{\gamma_i}$ is the i-th row of a $(n \times v)-$matrix B where the columns of B constitute a base of the null space of the $(n - v) \times n$-matrix A. Let us construct another base as follows: the null space of A contains all n-vectors $\boldsymbol{x} := \langle x_1, \ldots, x_n \rangle$ such that $A\boldsymbol{x}^T = 0$. Choose $x_1, \ldots, x_v$ as free variables and solve the system $A\boldsymbol{x}^T = 0$ (the system is solvable since if we exclude any v columns of A the matrix becomes the transpose of a Vandermonde matrix of size $n - v$; due to this fact the choice of the "first" v $\boldsymbol{\gamma}$ vectors is without loss of generality). Solving the system like this will generate a base B' for the null space of A so that the first v rows of B' contain the identity matrix of size v. But then it is easy to see that there are vectors in the span of B' that do not belong in the span of B, a contradiction. As a result $\boldsymbol{\gamma_1}, \ldots, \boldsymbol{\gamma_v}$ should be linearly independent. The same argument holds for any other v vectors of Γ. □

By theorem 7 we know that almost all users $(n - o(n))$ should be colored in the same way in order for the pirate-decoder to be unable to detect tracing. However, by the previous Theorem it holds that at most $v-1$ users can be colored in the same way (otherwise the coloring becomes trivial which means that the ciphertext does not constitute a query which helps in tracing by Proposition 5). As a result it should hold that $v = n - o(n)$; note that in this case $v/n \to 1$ if $n \to \infty$. As a result we obtain the following corollary:

Corollary 12. *Let $\langle t(n), f(n), v(n)\rangle$-BBTTS be a scheme based on the [BF99]-MES. If $t(n) = \omega(\log n)$ then it holds that either $f(n) = \Theta(n)$ or that $v(n) = n - o(n)$.*

Essentially this means that the [BF99]-scheme is incapable of black-box tracing superlogarithmic self-protecting traitor collusions unless the ciphertext size is linear in the number of users.

Regarding single-query black-box confirmation (introduced in [BF99]) we showed that when suspicion is modeled as biasing the uniform distribution, where suspects are distinguished by increasing the probabilistic confidence in them being traitors, then as a result of section 4.3 it holds that:

Corollary 13. *In the [BF99]-scheme, Single-query Black-box confirmation is impossible against a pirate decoder which includes $t(n) = \omega(\log n)$ traitor keys, unless the probability that a user is a suspect given that it is a traitor, is arbitrarily close to 1.*

Note: in [BF01], a more sophisticated combination of black-box confirmation with traitor tracing is presented. The scheme is a single-query black-box confirmation in principle, but multiple queries that induce different colorings *within* the suspect set are employed, until a traitor is pinned down. Our negative results for black-box confirmation (in the self-protecting model variant) apply to this setting as well. The arguments in [BF01] are plausible in the "arbitrary pirates" model (including self-protecting one). For the method to work, however, they assume "compactness" (called confirmation requirement), namely that it is *given* that all traitors are within the suspect list. Our results point out that without this compactness, relying solely on likelihood (modeled as probability), successful confirmation is unlikely unless there is a very high confidence level (which will enforce the "compactness condition" almost always). Our results do not dispute black-box confirmation under compactness but rather point to the fact that obtaining (namely, biasing a uniform distribution to get) a "tight" suspect set $\mathcal{S}$ which satisfies compactness at the same time can be hard.

5.2 The [KD98]-Scheme

Description. The scheme of Kurosawa and Desmedt is defined as follows: a random secret polynomial $f(x) = a_0 + a_1 x + \ldots a_v x^v$ is chosen and the values $g^{a_0}, \ldots, g^{a_v}$ are publicized (the public key of the system). User i is given $f(i)$ as its secret key. A message s is encrypted as follows: $\langle g^r, sg^{ra_0}, g^{ra_1}, \ldots, g^{ra_v}\rangle$, were r is chosen at random. User i decrypts as follows: $sg^{ra_0}g^{ra_1 i} \ldots g^{ra_v i^v}/g^{rf(i)} = s$. It is more convenient to think of the secret key of user i as $\langle f(i), \boldsymbol{i}\rangle$ where $\boldsymbol{i} := \langle 1, i, i^2, \ldots, i^v\rangle$. In [KD98] it was proven that their scheme satisfies key-user correspondence for collusions of up to v users provided the discrete-log problem is hard. However non-ambiguity of collusions was overlooked, something pointed out in [SW98a] and in [BF99].

The problem arises from the fact that the set of possible keys used also includes linear combinations of user keys: $\langle \sum_{m=1}^{t} \alpha_m f(i_m), \sum_{m=1}^{t} \alpha_m \boldsymbol{i_m}\rangle$ where

$\alpha_m \in Z_q$ with $\sum_{m=1}^{t} \alpha_m = 1$ and $i_1, \ldots, i_t \in \{1, \ldots, n\}$. This tuple can also be used for decryption since: given $\langle g^r, sg^{ra_0}, g^{ra_1}, \ldots, g^{ra_v} \rangle$, one may compute

$$sg^{ra_0} g^{ra_1 \sum_{m=1}^{t} \alpha_m (\boldsymbol{i}_m)_1} \ldots g^{ra_v \sum_{m=1}^{t} \alpha_m (\boldsymbol{i}_m)_v} / g^{r \sum_{m=1}^{t} \alpha_m f(i_m)} = s$$

To achieve non-ambiguity of collusions we would like to show that given any two subsets of users $i_1, \ldots, i_t$ and $j_1, \ldots, j_t$ it should hold that $\{\sum_{m=1}^{t} \alpha_m \boldsymbol{i}_m \mid \alpha_1, \ldots, \alpha_m\} \cap \{\sum_{m=1}^{t} \alpha_m \boldsymbol{j}_m \mid \alpha_1, \ldots, \alpha_m\} = \emptyset$. Something that can be true only if $v \geq 2t$ i.e. v should be twice the size of the biggest traitor collusion allowed. In the light of this, it is not known if it is possible to trace traitors in this scheme (even in the non-black-box setting). The only known approach is the brute-force "black-box confirmation" for all possible traitor subsets suggested in [BF99] that needs exponential time (unless the number of traitors is assumed to be a constant). Despite this shortcoming the [KD98]-scheme is a very elegant public-key MES that inspired further work as seen in the schemes of [BF99, NP00]. In the rest of the section we show that the [KD98]-scheme has similar black-box traitor tracing limitations as the [BF99]-scheme.

Negative Results. Suppose we want to induce the coloring $\cup_{i=1}^{k} C_i^n$ in the [KD98]-MES. Given a (possibly invalid) ciphertext $\langle g^r, sg^{x_0 a_0}, g^{x_1 a_1}, \ldots, g^{x_v a_v} \rangle$, user i applies $\langle f(i), \boldsymbol{i} \rangle$ to obtain $sg^{\sum_{j=0}^{v} x_j a_j (\boldsymbol{i})_j - rf(i)} = sg^{\sum_{j=0}^{v} (x_j - r) a_j (\boldsymbol{i})_j}$. So we can color each user by a color-label sg^{c_i}, if we find a $\boldsymbol{z}$ s.t. $\boldsymbol{z} \cdot \boldsymbol{i} = c_i$ for all $i = 1, \ldots, n$; given such a $\boldsymbol{z}$ we can compute the appropriate $x_0, \ldots, x_v$ values to use in the ciphertext as follows: $x_j = z_j (a_j)^{-1} + r$ for $j = 0, \ldots, v$. The set of all valid ciphertexts corresponds to the choice $\boldsymbol{z} = \boldsymbol{0}$ (and in this case it follows that $x_0 = \ldots = x_r$), nevertheless the choice of $\boldsymbol{z} = \langle a, 0, \ldots, 0 \rangle$ also colors all users in the same way although in this case the decryption yields sg^a (instead of s).

Next we present a property of the Kurosawa-Desmedt scheme, showing that an invalid ciphertext (which induces more than one color), cannot color too many users by the same color.

Theorem 14. *In the [KD98]-MES, given a (possibly invalid) ciphertext that induces a coloring over the user population so that $v+1$ users are labelled by the same color then all users are labelled by the same color.*

Proof. Suppose that the ciphertext $\langle g^r, sg^{x_0 a_0}, g^{x_1 a_1}, \ldots, g^{x_v a_v} \rangle$ induces a color-label sg^{c_i} on user i so that $v+1$ users are colored in the same way. Without loss of generality we assume that $c_1 = \ldots = c_{v+1}$. Define $z_j := (x_j - r) a_j$ for $j = 0, \ldots, v$. It follows that $\boldsymbol{i} \cdot \boldsymbol{z} = c_i$ for $i = 1, \ldots, n$. Seen as a linear system with $\boldsymbol{z}$ as the unknown vector the equations $\boldsymbol{i} \cdot \boldsymbol{z} = c_i$ for $i = 1, \ldots, n$ suggest that $\boldsymbol{z}$ corresponds to the coefficients of a polynomial $p(x) := z_0 + z_1 x + \ldots z_v x^v$ such that $p(i) = c_i$ for $i = 1, \ldots, n$. Because $p(1) = \ldots = p(v+1)$ and the degree of p is at most v it follows immediately that p has to be a constant polynomial, i.e. $\boldsymbol{z} = \langle a, 0, \ldots, 0 \rangle$ with $a = p(1) = \ldots = p(v+1)$. (Any $v+1$ equal value points on the polynomial will imply the above, which justifies the arbitrary choice of users). If follows immediately that user i receives the color label $sg^{c_i} = sg^{\boldsymbol{i} \cdot \boldsymbol{z}} = sg^a$ and as a result all users are labeled by the same color. □

With similar arguments as in section 5.1 we conclude:

Corollary 15. *Let $\langle t(n), f(n), v(n)\rangle$-BBTTS be a scheme based on the [KD98]-MES. If $t(n) = \omega(\log n)$ then it holds that either $f(n) = \Theta(n)$ or that $v(n) = n - o(n)$.*

Essentially this means that the [KD98]-scheme is incapable of black-box tracing superlogarithmic self-protecting traitor collusions unless the ciphertext size is linear in the number of users.

Corollary 16. *In the [KD98]-scheme, Single-query Black-box confirmation is impossible against a pirate decoder which includes $t(n) = \omega(\log n)$ traitor keys, unless the probability that a user is a suspect given that it is a traitor, is arbitrarily close to 1.*

References

[BF99] Dan Boneh and Matthew Franklin, *An Efficient Public Key Traitor Tracing Scheme*, CRYPTO 1999.

[BF01] Dan Boneh and Matthew Franklin, *An Efficient Public Key Traitor Tracing Scheme*, manuscript, full-version of [BF99], 2001.

[CFN94] Benny Chor, Amos Fiat, and Moni Naor, *Tracing Traitors*, CRYPTO 1994.

[CFNP00] Benny Chor, Amos Fiat, and Moni Naor, and Benny Pinkas, *Tracing Traitors*, IEEE Transactions on Information Theory, Vol. 46, no. 3, pp. 893-910, 2000. (journal version of [CFN94,NP98]).

[DLN96] Cynthia Dwork, Jeff Lotspiech and Moni Naor, *Digital Signets: Self-Enforcing Protection of Digital Content*, STOC 1996.

[FT99] Amos Fiat and T. Tassa, *Dynamic Traitor Tracing*, CRYPTO 1999.

[GSY99] Eli Gafni, Jessica Staddon and Yiqun Lisa Yin, *Efficient Methods for Integrating Traceability and Broadcast Encryption*, CRYPTO 1999.

[KD98] Kaoru Kurosawa and Yvo Desmedt, *Optimum Traitor Tracing and Asymmetric Schemes*, Eurocrypt 1998.

[NP98] Moni Naor and Benny Pinkas, *Threshold Traitor Tracing*, CRYPTO 1998.

[NP00] Moni Naor and Benny Pinkas, *Efficient Trace and Revoke Schemes* , In the Proceedings of Financial Crypto '2000, Anguilla, February 2000.

[Pfi96] Birgit Pfitzmann, *Trials of Traced Traitors*, Information Hiding Workshop, Spring LNCS 1174, pp. 49-63, 1996.

[PS96] Birgit Pfitzmann and Matthias Schunter, *Asymmetric Fingerprinting*, Eurocrypt 1996.

[PW97] Birgit Pfitzmann and M. Waidner, *Asymmetric fingerprinting for larger collusions*, in proc. ACM Conference on Computer and Communication Security, pp. 151–160, 1997.

[SW98a] Douglas Stinson and Ruizhong Wei, *Key preassigned traceability schemes for broadcast encryption*, In the Proceedings of SAC'98, Lecture Notes in Computer Science 1556, Springer Verlag, pp.144–156, 1998.

[SW98b] Douglas R. Stinson and R. Wei, *Combinatorial Properties and Constructions of Traceability Schemes and Frameproof Codes*, SIAM J. on Discrete Math, Vol. 11, no. 1, 1998.