Copyrighting Public-key Functions and Applications to Black-box Traitor Tracing

Aggelos Kiayias* Moti Yung†

Abstract

Copyrighting a function is the process of embedding hard-to-remove marks in the function's implementation while retaining its original functionality. Here we consider the above problem in the context of public-key encryption and we parallel the process of copyrighting a function to the process of designing traitor tracing schemes.

We derive two copyrighted public-key encryption functions for the 2-key setting, solving an open question left by earlier work with respect to copyrighting discrete-logarithm based functions. We then follow a modular design approach and show how to elevate the 2-key case to the multi-user setting, employing collusion secure codes. Our methodology provides a general framework for constructing public-key traitor tracing schemes that has the interesting property that the transmission rate remains constant if the plaintext size can be calibrated to reach an appropriate minimal length. Achieving a constant rate, i.e., constant expansion in the size of ciphertexts and keys, is an important open problem in the area of traitor tracing schemes. Our design shows how one can solve it for settings that accommodate the required plaintext calibration (e.g., when a bulk of symmetric cipher keys can be encrypted in one message).

Our constructions support "black-box traitor tracing", the setting where the tracer only accesses the decryption box in input/output queries/responses. For the first time here we provide a modeling of black-box traitor tracing that takes into account adversarially chosen plaintext distributions, a security notion we call *semantic black-box traceability*. In order to facilitate the design of schemes with semantic black-box traceability we introduce as part of our modular design approach a simpler notion called semantic user separability and we show that this notion implies semantic black-box traceability. In the multi-user setting our constructions also demonstrate how one can derive public-key traitor tracing by reducing the required "marking assumption" of collusion-secure codes to cryptographic hardness assumptions.

*Computer Science and Engineering Dept., University of Connecticut, Storrs, CT, USA, aggelos@cse.uconn.edu. Research partly supported by NSF CAREER Award CNS-0447808.

†RSA Laboratories, Bedford, MA, USA and Computer Science Dept., Columbia University, NY, USA moti@cs.columbia.edu

Contents

1 Introduction.

Copyrighting a function is the process that transforms a function to a function-family where each member (i.e., each function representation) has the same functionality but possesses discernible and unique markings. Such markings should be perceptible to the appropriate entities and moreover they should be resilient to removal even by adversaries commanding a sub-collection of the functions. Copyrighting functions has numerous applications in settings where the same function is used by a number of possibly adversarial entities, and where it is desired that the illegal redistribution or misuse of the function should be somehow deterred (due to the fear of exposure when the marking is detected).

Naccache, Shamir and Stern in [**?**] studied the problem of copyrighting symmetric encryption functions. The techniques presented in [**?**] applied to one-way (hash) functions and to symmetric encryption (and were implemented based on the RSA function used as a private key function). They left as an open question whether it is possible to achieve function copyright for discrete-log based systems. Perhaps more importantly, their work leaves another open question : is it possible to copyright public-key encryption functions (or decryption functions to be more precise)? Here, we provide answers to these questions. We argue that copyrighting a public-key encryption function, interpreted as copyrighting the decryption function (since the encryption is meant to be publicly available and not owned by any entity) parallels the design of public-key traitor tracing schemes. Using this observation as a starting point for the current work, we present a general framework for designing public-key traitor tracing schemes as well as two concrete instantiations of the framework which possess unique efficiency and security characteristics.

1.1 Traitor tracing

Traitor tracing [**?**] allows the distribution of digital content to a set of subscribers, so that if any misbehaving coalition thereof (up to a certain size) leaks its decryption keys to a pirate that constructs a "pirate-decoder", it is still possible to discover the identities of the misbehaving parties (a.k.a. traitors). Traitor tracing is attractive since it offers some redistribution resistance even in settings where the subscription keys can be reverse-engineered out of decoders; this is quite useful given that software obfuscation is limited and in fact does not appear to allow for cryptographically strong hiding (cf. [**?**]).

From the time of the primitive's introduction in [**?**], a series of works including [**?**, **?**, **?**, **?**, **?**, **?**, **?**, **?**, **?**, **?**, **?**, **?**, **?**, **?**, **?**] proposed more efficient and more robust schemes or schemes with advanced capabilities. Two desirable properties of a traitor tracing scheme are (i) Public-key traitor tracing, where any third party is able to send secure messages to the set of subscribers (increasing the functionality and scale of the service); and (ii) Black-box traitor tracing, which suggests that the tracing procedure can be accomplished with merely black-box access to the pirate-decoder. Black-box traitor tracing reduces tracing costs drastically as there is no need to reverse engineer the pirate-decoder and tracing can be performed remotely.

Traitor tracing has also its shortcomings: the size of the ciphertexts and keys used by traitor-tracing schemes depends on quantities such as the number of users and/or the maximum traitor collusion that is anticipated. Even though progress has been made from the initial scheme of [**?**] in reducing the "communications overhead" in traitor tracing schemes, so far there has not been a scheme in which the rate of the three main efficiency parameters of a traitor tracing scheme: ciphertext, encryption-key and user-key, is constant; the "rate" of a

parameter expresses the ratio of its size over the size of the plaintexts (note that the plaintext size is essentially the security parameter). We will refer to the sum of the rates of the three parameters collectively, as the "transmission rate" of a traitor tracing scheme. The reason we do not concentrate solely on the ciphertext rate, or the "per message" overhead, is that memory costs induced by the size of keys are equally important contributions to the "transmission" costs of a traitor tracing scheme (especially when implemented in dedicated or embedded systems). We note that minimizing the transmission rate has been, in fact, open since [**?**].

1.2 Our results

Our starting point is the notion of a multi-key public-key encryption scheme. Based on this primitive we provide a formal model for black-box traceability that takes into account adversarial plaintext distributions: we call this security notion "semantic black-box traceability." This means in particular that the pirate will be allowed to choose the plaintext distribution that it wishes the pirate decoder to operate on. In previous works that considered black-box traitor tracing, including [**?**, **?**, **?**], such adversarial capability was not considered. We stress that a traitor tracing scheme may be used to distribute varied material (possibly determined by a malicious content distributor), and thus it is important to consider adversarial plaintext distributions in the same way that it is important to consider semantic security when designing public-key encryption schemes. It is important to note that allowing the pirate to choose the plaintext distribution is tricky from a definitional viewpoint, as any such formulation should also take into account the success probability that is required of the pirate decoder (if not, "key-less" decoders can be feasible which, in turn, will render tracing impossible).

We continue to present a general methodology for designing public-key traitor tracing schemes with semantic black-box traceability, inspired by the copyrighting a function methodology of [**?**]. Our framework involves two major stages.

In the first step the focus is on copyrighting public-key functions in the 2-user setting. The 2-user setting, while retaining much of the complexity of the multi-user setting, is more readily attainable. An important aspect of our modular design methodology is the notion of **semantic user-separability** that we introduce. Semantic user separability is a crucial stepping stone for attaining semantic black-box traceability while being conceptually much simpler; we show that any semantic user-separable 2-key public-key encryption scheme satisfies semantic black-box traceability (in the 2-user setting).

We then present two explicit constructions of 2-key public-key encryptions that are user-separable. Therefore these constructions allow for black-box traceability in the 2-user setting. Our first construction is based on the DDH problem over composite groups and its traceability relies on the quadratic residuosity assumption. Our second construction is based on the DDH problem over prime order subgroups and its traceability relies on the DDH assumption.

In the second stage of our framework, we start with a 2-user public-key encryption scheme that achieves semantic user separability and we provide a general construction for obtaining a solution to the multi-user setting that involves a collusion secure code [**?**, **?**] and the concatenation of parallel composition of a number of instances of the underlying 2-user encryption. Note that the employment of collusion-secure codes in general requires a "marking assumption" that restricts the behavior of the pirates. Our 2-user semantic separability notion is a tool that, based on our parallel composition approach, effectively enforces such marking assumption under complexity theoretic assumptions. In our concrete designs the quadratic residuosity

assumption and the decisional Diffie-Hellman assumption are employed, respectively, for the two schemes.

Our public-key traitor tracing schemes, have the interesting property of being the first to achieve constant transmission rate when operated in a setting where the distributor has the flexibility of adjusting the size of the plaintexts to accommodate tracing. Such flexibility is always possible in bulk data encryption (or in the public-key setting, bulky transmission of numerous session keys). Given that the basic block of plaintext is calibrated, the "calibration length" (i.e., the size of the basic block of plaintext) becomes an additional parameter. Overall, this introduces a tradeoff that allows us to have constant transmission and key size rate by increasing the calibration factor appropriately. Note that the calibration factor is a property of the basic block of the plaintext of the public encryption system and not a property of the entire cleartext. For example, the block may contain the keys for many pieces of content put together and subsequent blocks may be transmitted symmetrically encrypted.

Our first public-key traitor tracing scheme achieves a constant rate of 2 for ciphertexts and 1 for secret-keys when the plaintexts reach a minimal size (via calibration) that is $\mathcal{O}(lc^2 \log(n/\epsilon))$ where l is a security parameter, c is the maximum number traitors that can be tolerated, n is the number of users and ϵ is the error-probability of the tracing procedure. Note that simple (uncopyrighted) ElGamal encryption has identical transmission rate (but regardless of calibration). Our second public-key traitor tracing scheme has a ciphertext rate of 3 and a secret-key rate of 2, under the same calibration requirements. Our second scheme achieves black-box traceability for a larger family of pirates and adversarial plaintext distributions though.

Organization. In section **??** we present some basic notations, definitions and the formal model for black-box traceability. In particular in section **??** we discuss the intractability assumptions we employ, in section **??** we define public-key traitor tracing and our new definition for semantic black-box traceability. Then, in section **??** we present the copyrighted function setting of [**?**] together with our reformulation of the concept to apply to copyrighted public-key functions. In section **??**, we first present the notion of semantic user-separability (section **??**) and show that it implies black-box traceability in the 2-user setting. Then, we present two schemes of copyrighted public-key encryption for the 2-user setting. The generic construction for traitor tracing schemes based on the 2-user case is presented in section **??**. Finally, our new public-key traitor tracing schemes are described in a self-contained fashion in section **??**.

Remarks. This work is the full revised version of [**?**]. The present version includes additionally the new formal modeling of semantic black-box traceability and the reduction to semantic user separability. The present version also corrects a false claim stated in [**?**] regarding the number of queries required for the black-box traceability of one of the constructions. Further, in [**?**] the first scheme we presented was given without black-box traceability and it was left as an open problem to achieve it. This open problem is resolved in the affirmative herein.

Recent Works: In subsequent work that builds on our results, Chabanne, Phan and Pointcheval, [**?**] presented a scheme with asymptotic rate 1 and put forth the notion of public-traceability. Later, Phan, Naini and Tonien, [**?**] showed how to achieve rate 1 and maintain public-traceability at the same time (a question that was left open in the work of [**?**]).

2 Preliminaries and modeling

A function $\sigma : \mathbb{N} \to \mathbb{R}$ will be called negligible if for all $c \in \mathbb{N}$ there exists a $l_0 \in \mathbb{N}$ so that for all $l \geq l_0$ it holds that $\sigma(l) < l^{-c}$. Throughout the text l will denote a security parameter. All procedures that we consider unless noted otherwise are probabilistic polynomial-time (PPT) in l. If K is a set of objects and f is a procedure that samples an element of K, denote by $k \hookleftarrow_f K$ a random variable over K following the distribution induced by f; note that we may occasionally omit f or K from this notation if they are implied by the context. If K is a finite set of objects, let $\text{len}[k \in K]$ denote the maximum size that an object in K may have. As stated above $\text{len}[k \in K]$ is polynomial in the security parameter l and perhaps it may depend on other parameters as well. We use $|x|$ to denote the size of an object x, for example, $|x| = \lfloor \log_2 x \rfloor + 1$ if $x \in \mathbb{N}$; also let $[k]$ denote the set $\{1, \ldots, k\}$. If $f(b, v)$ is a function with real values, we write $f(b, v) \sim c$ where $c \in \mathbb{R}$ is a constant iff $\lim_{b,v \to \infty} f(b, v) = c$. The notation $a \in_U R$ stands for "a is sampled from R following the uniform distribution." If K_l is a set of objects of a certain size $k(l)$ where $l \in \mathbb{N}$ then we will denote by $\mathbb{K}$ the collection $\{K_l\}_{l \in \mathbb{N}}$. We will also use the notation $[\mathbb{K}]_l$ to refer to the l-th element of the collection.

A code $\mathcal{C}$ is a family of sets of strings over an alphabet Σ parameterized by $n \in \mathbb{N}$. We assume that the there exists a description polynomial-time function $\mathsf{desc}_{\mathcal{C}}$ that on input 1^n produces all codewords of $[\mathcal{C}]_n$. In cases where the length of each element of $[\mathcal{C}]_n$ is v and the alphabet size is d we refer to the code $\mathcal{C}$ as a $\langle n, v \rangle_d$-code. If $\omega \in [\mathcal{C}]_n$ then we write $\omega = (\omega)_1 \ldots (\omega)_v$ where each $(\omega)_\ell \in \Sigma$.

A family of probability distributions $\mathcal{D}$ parameterized by $l \in \mathbb{N}$ will be called a probability ensemble. We define the min-entropy of a probability ensemble $\mathcal{D}$ to be the $(\mathbb{N} \to \mathbb{R})$ function $\min_{a \in [\Omega]_l}(-\log \mathbf{Prob}[X = a])$ where $[\Omega]_l$ is the support set of $[\mathcal{D}]_l$ and X is a random variable distributed according to $\mathcal{D}$. Given two probability ensembles $\mathcal{D}_1, \mathcal{D}_2$ we define the statistical distance of them as a function in l that is equal to

$$\frac{1}{2} \sum_{a \in [\Omega]_l} |\mathbf{Prob}[X = a] - \mathbf{Prob}[Y = a]|$$

where X, Y are random variables distributed according to $[\mathcal{D}_1]_l$ and $[\mathcal{D}_2]_l$ respectively. Two probability ensembles are called statistically indistinguishable if their statistical distance is a negligible function in l. We say that two probability ensembles are computationally indistinguishable if it holds that $\max_{\mathcal{A}} |\mathbf{Prob}[\mathcal{A}(X) = 1] - \mathbf{Prob}[\mathcal{A}(Y) = 1]|$ is a negligible function in l where $\mathcal{A}$ is a PPT (probabilistic polynomial-time) predicate; note that such predicates will be called distinguishers between the two random variables X, Y.

Next we define the notion of a multi-key public-key encryption scheme :

Definition 1 *A multi-key public-key encryption scheme is a tuple $\langle \mathbb{P}, \mathbb{C}, \mathbb{K}, \mathbb{S}, G, E, D \rangle$ so that*

1. *$\mathbb{P}$ and $\mathbb{C}$ are the plaintext-space and ciphertext-space respectively. $\mathbb{K}$ and $\mathbb{S}$ is the public and secret-key space respectively. These are collections parameterized by the security parameter $l \in \mathbb{N}$.*

2. *Key Generation. The sampling algorithm G is used for key generation in the following manner : $\langle pk, sk_1, \ldots, sk_n, \tau \rangle \leftarrow G(1^l, 1^n)$; note that τ is some auxiliary private information that may be empty. In general we denote by $[\mathbb{S}]_{l,pk}$ the subset of $[\mathbb{S}]_l$ that contains all sk such that $\langle pk, \ldots, sk, \ldots \rangle$ can appear as an output of $G(1^l, 1^n)$ for some n (i.e., all possible secret-keys for a fixed public-key).*

3. *Encryption.* $E : ([\mathbb{K}]_l \times [\mathbb{P}]_l) \to [\mathbb{C}]_l$ *is a probabilistic polynomial-time procedure in* l.

4. *Decryption.* $D : ([\mathbb{S}]_l \times [\mathbb{C}]_l) \to [\mathbb{P}]_l$ *is a deterministic polynomial-time procedure in* l *that satisfies the condition* $D(sk, E(pk, m)) = m$ *for all* $l \in \mathbb{N}$, $m \in [\mathbb{P}]_l$ *and* $pk \in [\mathbb{K}]_l$ *and* $sk \in [\mathbb{S}]_{l,pk}$.

5. *Semantic Security (in the sense of indistinguishability): we define the following game* G_n *that operates in conjunction with a* PPT *adversary* A: G_n *on input* 1^l *obtains* $\langle pk, sk_1, \ldots, sk_n, \tau \rangle$ *from* $G(1^l, 1^n)$ *and simulates* $A(1^l, pk)$; A *returns two messages* $m_0, m_1 \in [\mathbb{P}]_l$ *such that* $m_0 \neq m_1$ *as well as some auxiliary information* aux. G_n *samples* $c \hookleftarrow_{E(pk,m_b)} [\mathbb{C}]_l$ *where* $b \in_U \{0,1\}$ *and simulates* $A(1^l, aux, c)$ *to obtain output* b^*; *finally* G_n *returns* 1 *if and only if* $b^* = b$ *(otherwise 0). We define* $\mathsf{Adv}_A^n(l) = \mathbf{Prob}[\mathsf{G}_n(1^l) = 1]$ *in the above procedure. The multi-user encryption scheme is said to be semantically secure if* $2\mathsf{Adv}_A^n(l) - 1$ *is a negligible function in* l *for any* $n = \mathcal{O}(l^c)$ *with* c *an arbitrary constant.*

Note that in some cases we will want to stress that the multi-key scheme operates for a specific value of n. *In such cases we may write instead "*n*-key public-key encryption scheme."*

The above definition is a simple syntactic extension of the standard notion of public-key encryption with semantic security; since no requirement is imposed on the structure of the secret-keys, any public-key encryption scheme also fits the above definition (by e.g., setting $sk_1 = \ldots = sk_n$ for all $n \in \mathbb{N}$).

2.1 Intractability assumptions

The security of the schemes that we will develop will be based on the hardness of the Decisional Diffie Hellman (DDH) Problem over a multiplicative cyclic subgroup $\langle g \rangle$. Formally we define the assumption as follows:

Definition 2 Decision Diffie Hellman Assumption. *Let* desc *be a probabilistic algorithm that on input* 1^l *produces a string that contains (a) the description of a polynomial-time in* l *group operation* $\cdot : \mathcal{G} \times \mathcal{G} \to \mathcal{G}$, *(b) the description of a polynomial-time in* l *membership test "*$x \in \mathcal{G}$*", (c) an element* $g \in \mathcal{G}$, *(d) an integer* t. *We denote* $\vec{d}$ *an arbitrary output of* $\mathsf{desc}(1^l)$.

For a given $\vec{d}$, *we denote by* $\mathcal{D}_{\vec{d}}$ *the set* $\{\langle g, g^x, g^y, g^{xy} \rangle \mid x, y < t\}$, *and by* $\mathcal{R}_{\vec{d}}$ *the set* $\{\langle g, g^x, g^y, g^z \rangle \mid x, y, z < t\}$. *The* DDH *assumption for* $\mathsf{desc}(\cdot)$ *states that any poly-time distinguisher* D *for the uniform distributions over the sets* $\mathcal{D}_{\vec{d}}, \mathcal{R}_{\vec{d}}$ *will have a distance that is a negligible function in* l.

The DDH assumption has been used in a variety of settings and over many different group description generators; for an overview and applications the reader is referred to [**?**] [**?**]. The DDH assumption over subgroups of prime order within the multiplicative group $\mathbb{Z}_p^*$ where p is prime, is known to be equivalent to the security of regular ElGamal encryption, see [**?**]. We note here that ElGamal-like encryption with composite modulus has also been used extensively, e.g. [**?**, **?**]. In such case it can be that the order of g within $\mathcal{G}$ is not public and thus the value t can be chosen to be some integer that approximates it (without revealing much information about it). Note that the DDH would be expected to be hard independently of whether the factorization of the modulus is known, see [**?**].

In this work we will employ the DDH assumption over the following two group description generators:

- desc_{prime}: on input 1^l it samples two prime numbers p, q so that $q \mid p-1$ and $2^l > p > 2^{l-1}$ and $|q|$ is a predetermined function in l; it sets the group operation as multiplication modulo p, and selects g to be an element of $\mathbb{Z}_p^*$ that has order q; finally it sets $t = q$.

- desc_{compo}: on input 1^l it samples two prime numbers of the same size p, q so that $n = pq$ satisfies $2^l > n > 2^{l-1}$ and $p \equiv q \equiv 3 \bmod 4$; it sets the group operation as multiplication modulo n, and selects g to be an element of $\mathbb{Z}_n^*$ that has order $\rho = \frac{p-1}{2} \cdot \frac{q-1}{2}$; finally it sets $t = \lceil n/4 \rceil$.

We also utilize the Quadratic Residuosity (QR) Assumption [**?**]:

Definition 3 Quadratic Residuosity Assumption. *If $n = pq$ is selected as in $\mathsf{desc}_{compo}(1^l)$ then the Quadratic Residuosity assumption states that any polynomial-time distinguisher D for the uniform distribution over the sets $\mathcal{J}_n$ (Jacobi +1 elements mod n) and $\mathcal{Q}_n$ (quadratic residues mod n) has negligible success probability in l.*

We will employ also an equivalent variant of the assumption where the sets $\mathcal{J}_n - \mathcal{Q}_n$ and $\mathcal{Q}_n$ are assumed to be computationally indistinguishable.

2.2 Public-key traitor tracing schemes

In this section we define a public-key traitor tracing scheme extending the functionality of a multi-user public-key encryption scheme as defined in definition **??**. A public-key traitor tracing scheme will combine a multi-user public-key encryption with a traceability algorithm : this procedure will be capable of identifying secret-keys by observing a decryption algorithm. The intended functionality of a public-key traitor tracing scheme is in a multi-recipient encryption setting where many receivers want to invert the public-key pk. If an adversary (known as a pirate in this setting) uses t keys given by some users (known as traitors) to construct a decryption device (known as a pirate-box) for the purpose of implementing an illegal receiver in the system, the authority will be able to recover the identity of one of the traitor users given the pirate-box (by utilizing the traitor-tracing algorithm). Formally we have:

Definition 4 *A public-key traitor tracing scheme with c-traceability where c is a $(\mathbb{N} \to \mathbb{N})$ function on a parameter n, is a multi-user public-key encryption scheme $\langle \mathbb{P}, \mathbb{C}, \mathbb{K}, \mathbb{S}, G, E, D \rangle$ so that the following holds:*

There exists a procedure $\mathcal{T}$ so that for any $\epsilon \in (0,1)$, $n \in \mathbb{N}$, any $C = \{i_1, \ldots, i_t\} \subseteq \{1, \ldots, n\}$, with $t \leq c(n)$, any probabilistic polynomial-time procedure A, and any $\langle pk, sk_1, \ldots, sk_n, \tau \rangle \leftarrow G(1^l, 1^n)$, if A is given pk and $\{sk_{i_1}, \ldots, sk_{i_t}\}$ and A outputs $d \in [\mathbb{S}]_{l,pk}$, then $\mathbf{Prob}[\emptyset \subsetneq \mathcal{T}(1^n, 1^l, \tau, d) \subseteq C] \geq 1 - \epsilon$ *in time polynomial in* $\log(1/\epsilon) + n + l$.

We stress the weakness of the above definition with respect to the power of the tracer: the adversary is required to return some decryption key $d \in [\mathbb{S}]_{l,pk}$ where $[\mathbb{S}]_{l,pk}$ is the set of secret keys that can be produced together with pk from G. One can make various heuristic arguments that such a definition is satisfactory to some extent (e.g., by somehow arguing that the only possible way to decrypt a ciphertext encrypted with pk is using D on some element of $[\mathbb{S}]_{l,pk}$ and by assuming that such a key would be possible to be reverse-engineered from a

decryption device) however such arguments are typically ad-hoc. On the one hand the pirate may not use directly a certain decryption key $d \in S_{l,pk}$ but instead it can construct a simulator for the decryption operation that does not suggest necessarily an element d; on the other hand, such element may be hard to recover given the code of the simulator. To capture this type of adversarial behavior, we will strengthen the traitor tracing property below. We first start with a definition :

Definition 5 *A $[\mathcal{D}, \sigma]$-semantic-pirate against a multi-user public-key encryption scheme, $\langle \mathbb{P}, \mathbb{C}, \mathbb{K}, \mathbb{S}, G, E, D \rangle$, where $\mathcal{D}$ is a plaintext distribution over $\mathbb{P}$, and σ is a $(\mathbb{N} \to \mathbb{R})$ function, is a* PPT *$\mathcal{P}$ so that if $\langle pk, sk_1, \ldots, sk_n, \tau \rangle \leftarrow G(1^l, 1^n)$, then the pirate $\mathcal{P}$ on input $(1^n, 1^l, sk_{i_1}, \ldots, sk_{i_t})$ where $i_1, \ldots, i_t \in \{1, \ldots, n\}$, $t \leq n$, returns a* PPT *B satisfying:*

$$\mathbf{Prob}[B(E(pk, m)) = m] \geq \sigma(l)$$

when $m \hookleftarrow_{\mathcal{D}} [\mathbb{P}]_l$.

A few remarks are in place. First observe that the adversary returns a PPT, which amounts to returning the description of a probabilistic procedure (e.g., a Turing machine) paired with a polynomial function that acts as a time bound for simulating the pirate-box. Second, note that the above definition requires from the pirate $\mathcal{P}$ to always output a functional pirate-box. This is not really a restriction, as pirates that operate on traitor key sets of specific size/form, may easily be extended to pirates that operate on any size of traitor key input (the extended $\mathcal{P}'$ will simply return the description of $D(sk, \cdot)$ whenever $\mathcal{P}$ is undefined for an input of traitor keys that includes some key sk).

We next proceed to define semantic black-box traceability. The property will be parameterized by a family of plaintext ensembles Δ out of which the pirate may select one ensemble to construct a pirate-box, a function κ that will be the success threshold that is required to be reached at minimum by pirate-boxes, and a function c that will specify an upper bound on the number of traitor keys that will be available to the pirate. In general we would like to achieve semantic black-box traceability for a Δ that include as many ensembles as possible, for a κ that is as small as possible and for a c that is as close to n as possible.

Definition 6 *A public-key traitor tracing scheme with $[\Delta, \kappa, c]$-semantic-black-box-traceability where c is a $(\mathbb{N} \to \mathbb{N})$ function on a parameter n, Δ is a collection of plaintext probability ensembles over $\mathbb{P}$, and κ is a $(\mathbb{N} \to \mathbb{R})$ function, is a multi-user public-key encryption scheme $\langle \mathbb{P}, \mathbb{C}, \mathbb{K}, \mathbb{S}, G, E, D \rangle$ so that the following holds:*

There exists a procedure $\mathcal{N}$ so that for any $\epsilon \in (0, 1)$, $n \in \mathbb{N}$, any $C = \{i_1, \ldots, i_t\} \subseteq \{1, \ldots, n\}$, with $t \leq c(n)$, and any $[\mathcal{D}, \sigma]$-semantic-pirate $\mathcal{P}$, with $\mathcal{D} \in \Delta$ and $\sigma > \kappa$ (for all l), we have that: If $\langle pk, sk_1, \ldots, sk_n, \tau \rangle \leftarrow G(1^l, 1^n)$, and $B \leftarrow \mathcal{P}(1^n, 1^l, sk_{i_1}, \ldots, sk_{i_t})$ then it holds that: $\mathbf{Prob}[\emptyset \subsetneq \mathcal{N}^{B(\cdot)}(1^{l+n}, \tau) \subseteq C] \geq 1 - \epsilon$ and $\mathcal{N}$ has running time polynomial in $\ln(\epsilon^{-1}) + (\sigma(l) - \kappa(l))^{-1} + n + l$.

We note that for efficiency purposes we will typically require that there is a non-negligible bound between $\sigma(l)$ (the decryption success of the pirate-box) and $\kappa(l)$ the success bound that needs to be reached at minimum for plaintext distributions in Δ. In this way the black-box tracing procedure will be polynomial in n, l.

Note that in the definitions **??** and **??** of traceability and semantic-black-box traceability respectively, we required that the tracing and analyzer procedures terminate in time polynomial

in n; this may be restricted further to be $\log n$ for systems that expect a large number of users and more frequent tracing operations. Another possible extension is to allow further access of the pirate to secret-keys of uncorrupted users through the employment of a chosen ciphertext oracle.

In the following proposition we point to a simple natural limitation in the construction of schemes that satisfy black-box traceability : for tracing to succeed it should hold that the min-entropy of the pirate-distribution should not be below the negated logarithm of the successful decryption bound κ that is required from the pirate-box to exceed.

Proposition 7 *For any functions $c \in (\mathbb{N} \to \mathbb{N})$, $\kappa \in (\mathbb{N} \to \mathbb{R})$, the multi-user public-key encryption scheme $\langle \mathbb{P}, \mathbb{C}, \mathbb{K}, \mathbb{S}, G, E, D \rangle$ does not satisfy $[\Delta, \kappa, c]$-semantic-black-box-traceability if Δ includes a plaintext probability ensemble $\mathcal{D}$ so that the min-entropy μ of $\mathcal{D}$ satisfies $\mu \leq -\log \kappa$.*

Proof. Suppose that $\mathcal{D}$ has min-entropy μ and consider the following $[\mathcal{D}, 2^{-\mu}]$-pirate: $\mathcal{P}$ on any input returns a pirate-box B that always returns the plaintext $m_0 \in [\mathbb{P}]_l$ such that $\mathbf{Prob}[m = m_0] \geq 2^{-\mu}$ when $m \hookleftarrow_{\mathcal{D}} [\mathbb{P}]_l$. Clearly the success probability of $B(\cdot)$ is at least $2^{-\mu}$. Given that $\kappa(l) \leq 2^{-\mu}$, it is very easy to show that for any analyzer procedure $\mathcal{N}$ it is possible to find $\epsilon \in (0, 1)$ and $C = \{i_1, \ldots, i_n\}$ so that $\mathcal{N}$ fails to trace the pirate-box B as constructed above as long as $n > 1$ (and this holds true independently of time allowed for tracing). □

We complete this section with a definition that specifies the efficiency parameters of a given traitor tracing scheme.

Definition 8 Efficiency Parameters. *The three basic efficiency parameters of traitor tracing schemes are (i) the ciphertext rate $\frac{\text{len}[c \in [\mathbb{C}]_l]}{\text{len}[m \in [\mathbb{P}]_l]}$, (ii) the user-key rate $\frac{\text{len}[d \in [\mathbb{S}]_l]}{\text{len}[m \in [\mathbb{P}]_l]}$, and (iii) the encryption-key rate $\frac{\text{len}[pk \in [\mathbb{K}]_l]}{\text{len}[m \in \mathbb{P}]}$. The transmission rate of the scheme is defined as the sum of the three rates.*

3 Copyrighting a function

Nacacche, Shamir and Stern [**?**] introduced a technique for personalizing a certain function f to a set of users. This fingerprinting technique generates a number of personalized copies of f, so that $f_1(x) = \ldots = f_n(x) = f(x)$ for all x. The copies are drawn out of a keyed collection of different versions of f, denoted by $\{f_k\}_{k \in \mathcal{K}}$. It is assumed that there is a "generator" function $F(x, k) = f_k(x)$ for all $x, k \in \mathcal{K}$ that is publicly known and also that $\mathcal{K}$ can be sampled efficiently by some (secret) procedure $\mathcal{G}_{\mathcal{K}}$.

In this section we give a brief overview of the results of [**?**]. The following definition is from [**?**], slightly amended (see below for comments):

Definition 9 *A keyed collection $\{f_k\}_{k \in \mathcal{K}}$ is called:*
(i) c-copyrighted against passive *adversaries, if there is an analyzer procedure $\mathcal{T}$ so that: an adversary given c elements of $\mathcal{K}$ constructs another element sk_0 of $\mathcal{K}$; then, $\mathcal{T}$ given sk_0 and possibly some trapdoor information τ associated to $\mathcal{K}$, is able to reconstruct at least one of the c elements that were given to the adversary.*
(ii) c-copyrighted against active *adversaries, if there is an analyzer procedure $\mathcal{N}$ so that: an adversary given c elements of $\mathcal{K}$ produces a simulator $\mathcal{S}$ that agrees with $f_k(x)$ for almost all*

inputs x, then $\mathcal{N}$ with oracle access to $\mathcal{S}$ and also possibly given some trapdoor information τ associated to $\mathcal{K}$, is capable of recovering at least one of the c elements that were given to the adversary.

We note that in the original definition of [**?**] the notion of copyrighted function was more restricted. In particular, in the case of passive adversaries the adversary was supposed to produce one of the keys of the collection $\mathcal{K}$, and in the case of active adversaries the analyzer was given the code of the implementation instead of merely black-box access as formulated above. It is easy to see that the formulation presented above captures the intended functionality of a copyrighted function and in fact it strengthens it. Finally, the original definition did not make explicit the need of a trapdoor information for the tracing procedure (that nevertheless was necessary for the construction presented there). The original formulation of copyrighted against passive adversaries given in [**?**] will be given below and reformulated as copyrighted in the "strong sense" (as it implies our formulation above):

Definition 10 *A keyed collection $\{f_k\}_{k\in\mathcal{K}}$ is called:*
(i') c-copyrighted against passive *adversaries* in the strong-sense*, if given c elements of $\mathcal{K}$ it is computationally impossible to find another element of $\mathcal{K}$.*

In [**?**] a method was presented that allowed copyrighting a hash function based on RSA-encryption. The basic design paradigm of [**?**] solved the two-user case first and then the multi-user case was addressed by employing collusion-secure codes [**?**]. Note that in [**?**] a method to copyright the RSA-encryption function was given, but only as a symmetric-encryption function, since no public-components were allowed. In [**?**] it was left as an open question whether it is possible to achieve a copyright mechanism based on the Discrete-Logarithm Problem. Later on we will answer this question in the affirmative.

Still, the most important question that arises from the work of [**?**], who showed how to copyright symmetric encryption functions, is whether it is possible to copyright a public-key encryption function. We consider this question in the following subsection and we find it to be essentially equivalent to the notion of constructing public-key traitor tracing schemes.

Copyrighting a Public-Key Encryption Function. Given that a public-key encryption scheme has an encryption function that is meant to be publicly available to any sender, the interpretation of "copyrighting a public-key encryption" we adopt will be to copyright the decryption function of a public-key encryption scheme.

Following the general approach of copyrighting a function from [**?**] we deduce that a public-key encryption function $\langle \mathbb{P}, \mathbb{C}, \mathbb{K}, \mathbb{S}, G, E, D\rangle$ can be copyrighted if there is a way to produce a number of variants (say n) of the decryption function D keyed by elements $k_1, \ldots, k_n$ so that each variant is capable of inverting the encryption function E. In particular:

Definition 11 *A n-key, c-copyrighted Public-Key Encryption Scheme against passive (resp. active) adversaries is a tuple $\langle \mathbb{P}, \mathbb{C}, \mathbb{K}, \mathbb{S}, G, E, D\rangle$ so that*
(i) $\langle \mathbb{P}, \mathbb{C}, \mathbb{K}, \mathbb{S}, G, E, D\rangle$ is a n-key public-key encryption scheme.
(ii) for any $pk \in \mathbb{K}_l$, the function collection $\{D(sk, \cdot) : \mathbb{C} \to \mathbb{P}\}_{sk\in\mathbb{S}_{l,pk}}$ is c-copyrighted against passive (resp. active) adversaries.

From the above it can be seen that an n-key, c-copyrighted public-key encryption scheme against passive (resp. active) adversaries is conceptually equivalent to a public-key traitor tracing scheme where the c-copyrighted parameter corresponds to the resilience of the traceability

procedure. More specifically, if the scheme is c-copyrighted against passive adversaries this will give rise to a public-key encryption with c-resilient traceability, whereas if the scheme is copyrighted against active adversaries this will give rise to a public-key encryption with black-box traceability. This conceptual equivalence will motivate the remaining of the paper that focuses on the design of public-key traitor tracing schemes in a modular fashion starting from the 2-user case and extending to the multi-user setting through appropriate code constructions.

4 The two-user case: from user separation to black-box traceability

In this section we concentrate on 2-key public-key encryption. We will show that semantic black-box traceability follows from the notion of semantic user separability that we formalize next. This result will enable us to infer semantic black-box traceability for explicit constructions by simply arguing that they satisfy semantic user-separability, a conceptually simpler notion.

4.1 Semantic user separability

In this section we introduce the concept of *(semantic) user separation* for a 2-key public-key encryption scheme. In the remaining of the section, we note that the two keys of the 2-key scheme will be referred to as "key 0" and "key 1" or sk_0, sk_1 respectively.

Definition 12 *A 2-key public-key encryption scheme $\langle \mathbb{P}, \mathbb{C}, \mathbb{K}, \mathbb{S}, G, E, D\rangle$ is called* $[\Delta, \kappa]$-user-separable *where Δ is a family of plaintext ensembles and κ a function $(\mathbb{N} \rightarrow \mathbb{R}^+)$, if there exists a* PPT *procedure $E_0 : \mathbb{P} \rightarrow \mathbb{C}$ that takes as additional input the tracing trapdoor τ, has identical functionality to E in the view of the user holding key 0 and has the following two additional properties:*

US1. *Ciphertexts produced by E_0 are computationally computationally indistinguishable from regular ciphertexts produced by E for an adversary holding the key sk_0; in particular we will denote the distinguishing probability by $\epsilon_0(l)$ that will be a negligible $(\mathbb{N} \rightarrow \mathbb{R}^+)$ function in the security parameter l.*

Specifically, we have that for all $\mathcal{D} \in \Delta$, the two ensembles (1) $\langle sk_0, E(pk, m)\rangle$ where $(pk, sk_0, sk_1, \tau) \leftarrow G(1^l)$, with $m \hookleftarrow_{\mathcal{D}} [\mathbb{P}]_l$ and (2) $\langle sk_0, E_0(pk, \tau, m)\rangle$ where $(pk, sk_0, sk_1, \tau) \leftarrow G(1^l)$ with $m \hookleftarrow_{\mathcal{D}} [\mathbb{P}]_l$, have distinguishing probability at most $\epsilon_0(l)$ for any polynomial-time bounded distinguisher.

US2. *Any adversary holding the key sk_1, that decrypts valid E ciphertexts for any plaintext drawn from an ensemble $\mathcal{D} \in \Delta$ with probability at least $\kappa(l) + \alpha(l)$ where $\alpha(l)$ is a non-negligible $(\mathbb{N} \rightarrow \mathbb{R}^+)$ function in l, when given a special ciphertext encrypted by E_0 on a plaintext drawn from distribution $\mathcal{D}$, it can decrypt it correctly (i.e., in the way user 0 does) with probability at most $\kappa(l)$.*

Specifically, we have that for all PPT *$\mathcal{A}$, $\mathcal{D} \in \Delta$, if $(pk, sk_0, sk_1, \tau) \leftarrow G(1^l)$ and it holds that $\mathbf{Prob}[\mathcal{A}(pk, sk_1, E(m)) = m] \geq \kappa(l) + \alpha(l)$ with α a non-negligible function and $m \hookleftarrow_{\mathcal{D}} [\mathbb{P}]_l$ then it holds that $\mathbf{Prob}[\mathcal{A}(pk, sk_1, c) = D(sk_0, c)] \leq \kappa(l)$ where c is a ciphertext sampled as $c \hookleftarrow_{E_0(pk,\tau,m)} [\mathbb{C}]_l$.*

It follows that if a 2-key public-key encryption scheme is $[\Delta, \kappa]$-user-separable then it is possible to encrypt plaintexts that the user holding the key 0 finds it difficult to distinguish from regular transmissions but user 1 cannot decrypt them with probability better than κ as long as it behaves as a correct decryptor for valid ciphertexts with probability bounded by a non-negligible fraction above κ. A scheme that is $[\Delta, \kappa]$-user-separable will be said to satisfy semantic user separability for the family of ensembles Δ.

4.2 Semantic user separability implies semantic black-box traceability

In the remaining of this section we will establish the fact that any user-separable 2-key public-key encryption scheme has a corresponding analyzer procedure $\mathcal{N}$ that makes it a semantic black-box traitor tracing scheme. We first start with a preparatory lemma.

Lemma 13 *Let $\mathcal{D}_1, \mathcal{D}_2$ be two probability distributions over $\{0,1\}$ defined as follows: if $v \hookleftarrow_{\mathcal{D}_1} \{0,1\}$ it holds that $v = 1$ with probability at least δ_1 whereas if $v \hookleftarrow_{\mathcal{D}_2} \{0,1\}$ it holds that $v = 1$ with probability at most δ_2. Moreover it holds that there exist $\psi, \gamma \in (0,1)$ such that $\psi \cdot \delta_1 \geq \delta_2 + \gamma$. Then, there exists a deterministic polynomial-time T that given $\vec{v} \hookleftarrow \underbrace{\mathcal{D}_b \times \ldots \times \mathcal{D}_b}_{K}$ it returns $b \in \{1,2\}$ with probability at least $1 - \epsilon$ for an appropriate choice of $K = \mathsf{poly}(\ln(\epsilon^{-1}), (1-\psi)^{-1}, \delta_1^{-1}, \gamma^{-1})$.*

Proof. We define T as follows: let K' be the Hamming weight of $\vec{v}$. If it holds that $K'/K > \psi \cdot \delta_1$ then return 1 else return 2. We next compute the success probability of T in predicting the probability distribution.

Suppose that the given bitstring is drawn from distribution $\mathcal{D}_1$. The probability of the event $K'/K > \psi\delta_1$ can be bounded from below using the Chernoff bound as follows: let μ_1 be the probability of having a 1 in a given location of v; we have that, $\mathbf{Prob}[K' \leq \mu_1 K - d] \leq e^{-d^2/2K}$ for any $d > 0$. From this we obtain that, $\mathbf{Prob}[K'/K \leq \delta_1 - d/K] \leq \mathbf{Prob}[K'/K \leq \mu_1 - d/K] \leq e^{-d^2/2K}$, for any $d > 0$. We set now $d = K(\delta_1 - \psi\delta_1) = (1-\psi)K\delta_1$ and we obtain that $\mathbf{Prob}[K'/K \leq \psi\delta_1] \leq e^{-K(1-\psi)^2\delta_1^2/2}$, which suggests that T will return the correct answer in this case with probability at least $1 - e^{-K(1-\psi)^2\delta_1^2/2}$.

Now suppose that the given bitstring is drawn from the distribution $\mathcal{D}_2$ and $\mu_2 \leq \delta_2$ is the probability of obtaining a 1 in v. Using the Chernoff bound again we have that $\mathbf{Prob}[K' - \mu_2 K \geq d] \leq e^{-d^2/2K}$ from which we obtain that $\mathbf{Prob}[K' \geq \delta_2 K + d] \leq \mathbf{Prob}[K' \geq \mu_2 K + d] \leq e^{-d^2/2K}$ and by setting $d = K(\psi\delta_1 - \delta_2)$ we obtain that $\mathbf{Prob}[K'/K \geq \psi\delta_1] \leq e^{-K(\psi\delta_1 - \delta_2)^2/2}$. This suggests that the correct answer will be returned with probability at least $1 - e^{-K(\psi\delta_1 - \delta_2)^2/2}$.

In both cases we want that the probability is at least $1 - \epsilon$. It follows that due to $1 - e^{-K(1-\psi)^2\delta_1^2/2} \geq 1 - \epsilon$ we obtain that $K \geq \ln(\epsilon)^{-2}/((1-\psi)\delta_1)^2$ and similarly that $K \geq \ln(\epsilon)^{-2}/(\psi\delta_1 - \delta_2)^2$ from the second relation.

Now given that for $\psi, \gamma \in (0,1)$ it holds that $\psi\delta_1 \geq \delta_2 + \gamma$, we have that $\psi\delta_1 - \delta_2 \geq \gamma$. From this it follows that K is a polynomial function in $\ln(\epsilon^{-1}) + (1-\psi)^{-1} + \delta_1^{-1} + \gamma^{-1}$. □

In figure **??** we present the black-box traitor tracing procedure for a pirate-box $\mathcal{B}$ that operates on a pirate distribution $\mathcal{D}$ and is successful with probability σ. It employs the alternative encryption E_0 that is suggested by the semantic user-separability property. The parameter ψ employed in the figure will be clarified in the theorem below :

Analyzer $\mathcal{N}$ for given plaintext distribution $\mathcal{D}$
Input : tracing trapdoor τ; public-key pk; security parameter 1^l
Oracle access: pirate-box $\mathcal{B}$
Parameters: $K \in \mathbb{N}, \psi, \sigma \in (0,1)$
$K' = 0$;
For $i = 1, \ldots, K$
    sample $m_i \hookleftarrow_{\mathcal{D}} [\mathbb{P}]_l$;
    set $C_i \leftarrow E_0(pk, \tau, m)$;
    $m'_i \leftarrow \mathcal{B}(C_i)$;
    if $(m_i = m'_i)$ then $K' = K' + 1$;
if $K'/K \geq \psi \cdot (\sigma - \epsilon_0)$ output 0 else output 1;

Figure 1: Black-Box Analyzer Procedure for a 2-user public-key encryption.

Theorem 14 *The $[\Delta, \kappa]$-user-separable 2-key public-key encryption scheme $\langle \mathbb{P}, \mathbb{C}, \mathbb{K}, \mathbb{S}, G, E, D \rangle$ satisfies $[\Delta, \kappa, 1]$-semantic-black-box traceability using the analyzer $\mathcal{N}$ presented in figure **??**.*

In particular for any $\langle \Delta, \sigma \rangle$-semantic-pirate $\mathcal{P}$, such that $\sigma(l) \geq \kappa(l) + \alpha(l)$ where $\alpha(l)$ is a non-negligible function in l, it holds that (1) $\mathcal{N}$ has success probability at least $1 - \epsilon$, (2) the running time of $\mathcal{N}$ is polynomial time in $\ln(\epsilon^{-1}) + \alpha(l)^{-1} + \sigma(l)^{-1}$.

Proof. In the remaining of the proof we drop the (l) from $\sigma(l), \alpha(l), \mu(l), \epsilon_0(l)$.

We set $\psi = 1 - \frac{\alpha}{2(\sigma - \epsilon_0)}, \gamma = \alpha/2 - \epsilon_0$. Observe that $\psi, \gamma \in (0,1)$ for sufficiently large values of l (note that using the fact that ϵ_0 is negligible if of essence here). Let pk, sk_0, sk_1, τ be the public and secret-keys of the system as well as the tracing trapdoor. Let $\langle \mathcal{B}, \mathcal{D}, C \subseteq \{0,1\} \rangle$ be the pirate box, the pirate plaintext distribution and the traitor set with $|C| = 1$. The analyzer $\mathcal{N}$ defined in figure **??** operates as follows: it samples K plaintexts following the pirate distribution $\mathcal{D}$, applies to them the special encryption function E_0 using the tracing trapdoor τ, and then applies $\mathcal{B}$ on them to collect the number of successful decryptions K'. Then, if K'/K is sufficiently large, the analyzer returns 0 otherwise it returns 1. Based on the result of lemma **??** we observe that the analyzer $\mathcal{N}$ operates the deterministic test T defined there that is successful with probability $1 - \epsilon$ (for an appropriate selection of K); the two probability distributions $\mathcal{D}_1, \mathcal{D}_2$ over $\{0,1\}$ that are defined there are produced by the equality test $m_i = m'_i$ and they vary depending on whether $\mathcal{B}$ had access to the key of the traitor set $\{0\}$ or the key of the traitor set $\{1\}$. It remains below to determine δ_1, δ_2 for the two cases and verify that $\psi \cdot \delta_1 \geq \delta_2 + \gamma$.

Note that we are guaranteed that $\mathcal{B}$ successfully decodes valid ciphertexts. Given property US1 of definition **??**, we have that the user holding the key sk_0 will be unable to distinguish special ciphertexts produced by E_0 compared to valid ciphertexts. It follows that a pirate-box produced with the key of the traitor set $C = \{0\}$ will have a successful probability of decryption of at least $\delta_1 = \sigma - \epsilon_0$ (recall that ϵ_0 is a negligible function defined in property US1).

On the other hand, observe that based on property US2 of definition **??**, a pirate-box $\mathcal{B}$ produced with the key of the traitor set $C = \{1\}$ that has correct decryption probability $\sigma \geq \kappa + \alpha$ and will correctly decrypt an E_0 ciphertext with probability at most $\delta_2 = \kappa$.

It follows by the definition of ψ, γ that $\psi \cdot \delta_1 \geq \delta_2 + \gamma$; by applying lemma **??** to the above to conclude that $\mathcal{N}$ will be successful with probability $1 - \epsilon$ and will have time-complexity

polynomial in $\ln(\epsilon)^{-1} + 2(\sigma - \epsilon_0)\alpha^{-1} + (\sigma - \epsilon_0)^{-1} + 2\alpha^{-1}$ from which the desired result follows taking into account that ϵ_0 is a negligible function in the security parameter l. □

Remark. It is relatively simple to derive a 2-user scheme with semantic user-separability. In particular if $\langle \mathbb{P}, \mathbb{C}, \mathbb{K}, \mathbb{S}, G, E, D \rangle$ is a public-key encryption scheme then we can generate two independent instances of it and encrypt plaintexts $m \in \mathbb{P}$ as pairs $\langle E_{\mathsf{user0}}(m), E_{\mathsf{user1}}(m) \rangle$; user 0 will hold the secret-key of the first coordinate where user 1 will hold the secret-key of the second coordinate. While user-separability follows easily in this case, the transmission rate of such a construction is not very favorable: for example, if it is instantiated with ElGamal encryption [**?**] we have that the ciphertext rate will be about 4. The schemes that we will present in the following two sections have better ciphertext rates.

5 Two concrete two-user schemes

In this section we present two 2-key public-key encryption schemes and prove that they satisfy semantic-black-box-traceability.

Scheme 1 is based on the Decision-Diffie Hellman over the quadratic residues in $\mathbb{Z}_N^*$ and the Quadratic Residuosity Assumption. Scheme 2 is based on the Decision-Diffie Hellman over a group of prime order $\mathcal{G}$.

5.1 Scheme 1

Consider the following 2-key public-key encryption function $\langle \mathbb{P}, \mathbb{C}, \mathbb{K}, \mathbb{S}, G, E, D \rangle$: The function G given 1^l simulates desc_{compo} with the following modification: the two primes p, q produced are required to satisfy the relations $p \equiv 3(\text{mod}8), q \equiv 7(\text{mod}8)$ which imply that $\frac{p-1}{2} \equiv 1(\text{mod } 4), \frac{q-1}{2} \equiv 3(\text{mod}4)$. Note that this type of composite numbers satisfy (i) $(\frac{2}{n}) = -1$, and (ii) $-1 \in \mathcal{J}_n - \mathcal{Q}_n$.

In addition, G, following desc_{compo} outputs $t = \lceil n/4 \rceil$ and also samples g an element of $\mathbb{Z}_n^*$ of order $\rho = \frac{p-1}{2}\frac{q-1}{2}$. Then, G sets $pk = \langle n, t, g, h \rangle$ where $h = g^\alpha \bmod n$ and h is selected according to the uniform distribution over $\langle g \rangle$. We will return to the selection of the secret-keys after we describe the encryption function.

The plaintext space $[\mathbb{P}]_l$ will be defined as $\{0,1\}^b$ where $b = |n| - 2 = l - 2$. Given a plaintext $m \in [\mathbb{P}]_l$ the probabilistic function E returns the pair $\langle A, B \rangle = \langle g^r \bmod n, h^r \cdot enc(m) \bmod n \rangle$ where r is sampled uniformly from $[t]$ and $enc(\cdot)$ is an invertible mapping from the set $\{0,1\}^b$ to $\mathcal{J}_n$ (we will describe such a mapping shortly).

Given the definition of E it is clear that given $\langle A, B \rangle$ the plaintext m can be recovered by the computation $enc^{-1}(A^{-\alpha}B \bmod n)$. From this it is also evident that for any public-key $\langle n, t, g, h \rangle$ we can define as the corresponding secret-key space, the set $\mathbb{S}_{l,pk} = \{x \in \mathbb{N} \mid x = \alpha(\text{mod}\rho)\}$: indeed, it holds that any $x \in \mathbb{S}_{l,pk}$ can be used as a secret-key for the decryption of the ciphertext $\langle A, B \rangle$.

Next we specify how G produces two distinct secret-keys and thus the encryption function is a 2-key public-key encryption scheme: the keys selected are the following $\{\alpha, \alpha + \rho\}$ and user 0 receives the even key from this set, while user 1 receives the odd key from this set (note that addition is performed over $\mathbb{Z}$ and the keys are of different parities since ρ is an odd number by the selection of p, q). The tracing trapdoor τ of the encryption scheme will be empty (i.e., no special system information will be needed to trace in scheme 1).

We complete our description of the 2-key public-key encryption by describing the encoding function $enc(\cdot)$. Recall that the plaintext-space for the encryption operation is $\{0,1\}^b$ with $b = |n| - 2$. Given a plaintext $m =_{\text{df}} m_0 m_1 m_2 \dots m_{b-1} \in \{0,1\}^b$ let $M' =_{\text{df}} m_0 + 2m_1 + 2^2 m_2 + \dots + 2^{b-2} m_{b-2} + \lceil \frac{n}{8} \rceil$. It is easy to see that $\frac{n}{8} \leq M' < \frac{n}{4}$. Now if $(\frac{M'}{n}) = 1$ the encoding of m would be defined to be $(-1)^{m_{b-1}} M' \bmod n$, else if $(\frac{M'}{n}) = -1$ then the encoding of m is defined as $(-1)^{m_{b-1}} \cdot 2 \cdot M' \bmod n$. This completes the description of *enc*. It is easy to verify that $enc(\{0,1\}^b) \subseteq \mathcal{J}_n$: this is due to the fact that $(\frac{2}{n}) = -1$ as well as that $-1 \in \mathcal{J}_n$ due to the choice of n.

The encoding function can be inverted as follows: given $c = enc(m)$ for some $m = m_0 \dots m_{b-1}$, we first compute m_{b-1} by checking whether $c < n/2$. We then evaluate $c' = (-1)^{m_{b-1}} \cdot c \bmod n$, and compute M' so that $M' =_{\text{df}} c'$ if $c' < n/4$, or $M' =_{\text{df}} c'/2$ (over $\mathbb{Z}$) if $c > n/4$. The decoding of $enc(M)$ is the binary representation of $M' - \lceil \frac{n}{8} \rceil$ concatenated by the bit m_{b-1} as recovered above.

A property of our invertible encoding $enc(\cdot)$ that we will make a non-trivial use for traceability is the following:

Proposition 15 *For all $l \in \mathbb{N}$ it holds that $enc(m_0 \dots m_{b-1}) \equiv_n -enc(m_0 \dots, m_{b-2}\overline{m_{b-1}})$ where if $\mathsf{b} \in \{0,1\}$ it holds that $\overline{\mathsf{b}} = 1 - \mathsf{b}$.*

Proof. The proof is immediate from the definition of the encoding function $enc(\cdot)$; simply recall that the encoding is always multiplied by the factor $(-1)^{m_{b-1}}$. □

Note that the choice of the $(b-1)$-th bit of the plaintext was arbitrary and we may had just as well performed an invertible encoding that relied in any other bit of the plaintext. Finally, we note that the tracing trapdoor τ of scheme 1 will be empty : in particular, this means that no additional information beyond what is publicly known will be employed by the tracer. In [**?**] this notion was called public-traceability.

In figure **??** we present the efficiency properties of scheme 1. The maximum traceable collusion will be 1 user and this will be proved in the next two sections.

Plaintext Space	Ciphertext Rate	User-Key Rate	Public-Key Rate	Max Traceable Collusion
$\{0,1\}^b$	$\frac{2(b+2)}{b} \sim 2$	$\frac{(b+1)}{b} \sim 1$	$\frac{3(b+2)}{b} \sim 3$	1

Figure 2: Efficiency Parameters of Scheme 1 (Two-User Setting)

5.1.1 Semantic security and security against passive adversaries

We proceed to the security analysis of the construction. In the following standard lemma we show that the sampling performed by the sender during encryption introduces a negligible statistical distance compared to a uniform group distribution. This will come handy in the security analysis.

Lemma 16 *The uniform distribution over $\langle g \rangle$ is statistically indistinguishable from the distribution $\mathcal{D}$ induced over $\langle g \rangle$ by the mapping $k \leftarrow g^k$ where $k \in_U [t]$.*

Proof. Recall that $n = (2k+1)(2k'+1)$ for some integers k, k'. It follows that $n = 4kk' + 2(k + k') + 1$ and as a result it holds that $\lceil n/4 \rceil = \lceil kk' + \frac{k+k'}{2} + 1/4 \rceil$. Based on the selection of g we have that the order of g is equal to kk'. Now observe the following: for some $u, U \in \mathbb{Z}$, the statistical distance of the uniform probability distribution over $\mathbb{Z}_u$ compared to the probability distribution of $\omega \bmod u$ where $\omega \in_R \{0, 1, \ldots, U-1\}$ is equal to

$$\frac{1}{2}\Big(v \cdot |\frac{\pi+1}{U} - \frac{1}{u}| + (u-v)|\frac{\pi}{U} - \frac{1}{u}|\Big)$$

where $U = \pi u + v$ and $0 \leq v < u$. It follows that the statistical distance is equal to $v(u-v)/(U \cdot u)$ which is less than v/U. Applying this argument to the present scenario for $U = \lceil n/4 \rceil$ and $u = kk'$ we have that the statistical distance of the two distributions is less than $\frac{2(k+k')+1}{\lceil n/4 \rceil}$ which is a negligible quantity bounded by $2^{-l/2+2}$. □

Theorem 17 *The 2-key public-key encryption function described above is*
(i) Semantically Secure under the DDH *Assumption over $\mathcal{Q}_n$ and the QR Assumption in $\mathcal{Q}_n$.*
(ii) 1-copyrighted against passive adversaries (in the strong sense): given the public-key pk and a key α_x of $\{\alpha_0, \alpha_1\}$ it is computationally infeasible to construct another key in $\mathbb{S}_{l,pk}$ under the assumption that factoring n is hard.

Proof. (i) Recall the definition of semantic security in the sense of message indistinguishability as a game G_0 that is defined as follows: an adversary $\mathcal{A}$ is given the parameter 1^l and the public-key pk of the system; it returns two messages $m_0, m_1 \in \mathbb{P}_l$ such that $m_0 \neq m_1$. Then a coin is flipped, $b \in_R \{0, 1\}$ and $\mathcal{A}$ receives a ciphertext drawn from $\mathbb{C}_l$ according to $E(pk, m_b)$. Finally, $\mathcal{A}$ terminates by returning b^*. The adversary wins the game in the case of the event $(b = b^*)$.

In the game above, the ciphertext given to the adversary after the challenge is produced is of the form $\langle A, B \rangle = \langle g^r, h^r \cdot enc(m_b) \rangle$. Suppose we modify the game into game G_1 so that the value B is calculated in a different way as follows : $B = h^{r'} \cdot enc(m_b)$ where $r' \in_R [t]$. It follows by a standard argument that the statistical distance in the winning probability of the adversary will be bounded by the advantage of distinguishing DDH triples over the subgroup $\langle g \rangle$.

In the modified game G_1 the value B is a random element of $\mathcal{G}$ multiplied by the encoding of the message m_b. Note that $enc(m_b)$ does not necessarily belong to the subgroup $\langle g \rangle$. In fact $\langle g \rangle$ is the group of quadratic residues $\mathcal{Q}_n$ while $enc(m) \in \mathcal{J}_n$, the subgroup of Jacobi +1 symbols in $\mathbb{Z}_n^*$ that strictly subsumes $\mathcal{Q}_n$.

Next we modify the game further into game G_2 as follows: in case $enc(m_b) \in \mathcal{Q}_n$ we compute B as $h^{r'}$ (i.e., ignoring the $enc(m_b)$); on the other hand, if $enc(m_b) \in \mathcal{J}_n - \mathcal{Q}_n$ we compute B as $-h^{r'}$. Recall that $(\frac{-1}{n}) = (\frac{-1}{p}) \cdot (\frac{-1}{q})$ and $(\frac{-1}{p}) = -1$ and $(\frac{-1}{q}) = -1$ since $p, q = 3 (\bmod 4)$, as a result $-1 \in \mathcal{J}_n - \mathcal{Q}_n$. It is easy to see that the probability distribution of B in the two games is identical and thus there will be no difference in the success probability of the adversary.

Consider now the event Same to include all coin tosses for which m_0, m_1 have their encodings both in $\mathcal{Q}_n$ or both in $\mathcal{J}_n - \mathcal{Q}_n$.

It is apparent that as long as the event Same happens the success probability of the above modified game is equal to 1/2 since no information about b is retained in the input given to the adversary in the second stage. Next we consider the conditional space on the event ¬Same;

in this case we have that the success probability of the game is bounded by the advantage of a QR distinguisher since the element B is with $1/2$ probability either a random element of $\mathcal{Q}_n$ or a random element of $\mathcal{J}_n - \mathcal{Q}_n$. From the above we conclude that the success probability of $\mathcal{A}$ in the modified game G_2 is only by a negligible fraction different from $1/2$ (under the QR assumption).

(ii) First it is apparent that given any two keys $\alpha_0, \alpha_1 \in \mathbb{S}_{l,pk}$ one can immediately recover $L = (\text{mult}) \cdot \rho$. Such integer can be used to factor n as follows: first we compute m to be the largest odd divisor of L. Since L is a multiple of $(p-1)(q-1)$ and by selection it holds that $p \equiv 3(\text{mod}4)$ we have that m must be a multiple of $\frac{p-1}{2}$. We similarly argue for q. Then we select $a \in_R \mathbb{Z}_n$. If $a \notin \mathbb{Z}_n^*$ then n can be factored immediately (but this is a negligible probability event). Otherwise assume that $a \in \mathbb{Z}_n^*$. Consider now the probability event that either ($a \in \mathcal{Q}_p$ and $a \notin \mathcal{Q}_q$) or ($a \notin \mathcal{Q}_p$ and $a \in \mathcal{Q}_q$); this is a $1/2$ probability event conditioning on "$a \in \mathbb{Z}_n^*$." Note that this implies that either $a^{\frac{p-1}{2}} = 1 \bmod p$ or $a^{\frac{q-1}{2}} = 1 \bmod q$. If we compute now $b = a^m$ in $\mathbb{Z}$, it follows that $b+1$ will be a multiple of one of p, q but not of both simultaneously. Indeed, if it was a multiple of both simultaneously it would hold that $a^m \equiv -1 \bmod n$ something that contradicts the fact that either $a^m \equiv 1 \bmod p$ or $a^m \equiv 1 \bmod q$. Using this fact we obtain easily the factorization of n.

Given the above arguments, it suffices now to show that we can simulate the key-assignment process without possessing $\phi(n)$ (in this fashion we can turn any adversary against the strong copyrighted property of the encryption scheme into a factorization algorithm as shown above). First observe that the choice of $\alpha_0 = \alpha$ can be done independently of $\phi(n)$; indeed we may simple sample α uniformly from $[t]$ and following a similar argument as that in the proof of lemma **??** we can show that the statistical distance between the two probability distributions is negligible. The choice of α_1 during the system key assignment is done by computing $\alpha_1 = \alpha + \rho$ over $\mathbb{Z}$ where α is uniformly selected from $[\rho]$ where $\rho = \frac{p-1}{2}\frac{q-1}{2}$. A simulated key-assignment for α_1 without knowledge of $\phi(n)$ (and thus of ρ) can be as follows $\alpha_1 = \alpha + t$ where $\alpha \in_R [t]$. With a similar argument as in the proof of lemma **??** we can also show that the statistical distance of the simulated choice of α_1 compared to the one performed normally is negligible.

□

We remark that the scheme presented above is strictly 1-copyrighted in the strong sense and not 2-copyrighted in the strong sense since if the two users collude it is immediate that they can construct other keys in $\mathbb{S}_{l,pk}$ as follows: given $\alpha_0, \alpha_1 \in \mathbb{S}_{l,pk}$ we have that $\alpha_1 - \alpha_0$ equals ρ and as a result any integer $\alpha_0 + x(\alpha_1 - \alpha_0)$, where $x \in \mathbb{N}$, is an element of $\mathbb{S}_{lpk}$. Nevertheless, observe that the scheme is still (plain) 2-copyrighted since in the case the analyzer procedure $\mathcal{T}$ is given some key that is not among α_0, α_1 he can accuse correctly either of the two players (i.e., the tracing algorithm will succeed).

5.1.2 Security against active adversaries : black-box traitor tracing

In the public-key encryption scheme we presented it holds that all valid ciphertexts $\langle A, B\rangle$ satisfy $A \in \mathcal{Q}_n$ and $B \in \mathcal{J}_n$. The decryption operates by first computing $V = A^{\alpha_x} \bmod n \in \mathcal{Q}_n$ and subsequently by computing $B/V \bmod n$ and inverting the encoding. The two secret-keys α_0, α_1 have different parity. We take advantage of this fact in designing a special encryption procedure E_0 as follows: the output of E_0 is an *invalid* ciphertext $\langle A, B\rangle$, where $A = -g^r \bmod n$ and $r \in_U [t]$ while B is computed as in E; note that we have that $A \in \mathcal{J}_n - \mathcal{Q}_n$ (recall that due to the choice of n it holds that $-1 \in \mathcal{J}_n - \mathcal{Q}_n$). Now observe that $V = A^{\alpha_x} \bmod n$ will be

in $\mathcal{J}_n$ if α_x is odd (i.e., when $x = 1$ in our key assignment) whereas it will be in $\mathcal{Q}_n$ if α_x is even (i.e., when $x = 0$ in our key assignment).

Based on the above observation the decryption of a special ciphertext of E_0, $\langle -g^r, h^r \cdot enc(m)\rangle$ will be as follows: user 0 who holds the even key α_0 will simply obtain $enc(m)$ and apply to it the inverse encoding $enc^{-1}(\cdot)$ returning correctly the plaintext m. On the other hand, user 1 who holds the odd key will obtain the value $-enc(m) \bmod n$. It is evident by the definition of the encoding that the user 0 will decrypt an E_0 ciphertext hiding $m = m_0 \dots m_{b-1}$ correctly while user 1 will also decrypt the plaintext, but with one bit flipped, in particular it will return $m' = m_0 \dots m_{b-2}\overline{m_{b-1}}$ where $\overline{m_{b-1}}$ denotes the bit m_{b-1} flipped.

The analyzer procedure $\mathcal{N}$ is essentially the one provided in figure **??**. In the remaining of the section we will establish the fact that the 2-key public-key encryption scheme we presented is user-separable and thus theorem **??** will establish black-box traceability.

In the following lemma we establish the property US1 of definition **??**: it is difficult for an adversarial user 0, to distinguish the two functions E and E_0.

Lemma 18 *The ciphertexts that are produced by the encryption function E, i.e., the probability distribution defined by $\langle g^r \bmod n, h^r \cdot enc(m) \bmod n\rangle$ and the special ciphertexts $\langle -g^r \bmod n, h^r \cdot enc(m) \bmod n\rangle$ produced by E_0, are computationally indistinguishable for the user 0 under the QR Assumption.*

Proof. Suppose that g is an element of order ρ in $\mathbb{Z}_n^*$ that generates the quadratic residues and G is a challenge for the QR Assumption modulo n (of unknown factorization). Observe that we can simulate the key-selection α of user 0, by simply selecting a random number α in $[t]$ (recall that $t = \lceil n/4 \rceil$ and that t is an even number) and giving to the adversary the value α_0 which is selected to be the even value among $\{\alpha, \alpha + t - 1\}$. This is statistically indistinguishable from the honest key-assignment. We set the public-key to be $g, h = g^{\alpha_0}$, and consider the ciphertext $\langle G, G^{\alpha_0} \cdot enc(m)\rangle$. Observe that if G is drawn from the quadratic residues modulo n, it holds that the ciphertext is consistent with an encryption of m according to E, whereas if G is drawn from $\mathcal{J}_n - \mathcal{Q}_n$ it holds that the ciphertext is consistent with a special encryption according to E_0. This completes the argument for the proof of the lemma. □

Next we specify the family of probability ensembles Δ that will be applicable to our black-box analyzer $\mathcal{N}$. Δ for any size l, it includes all probability distributions $\mathcal{D}$ for which if $m \hookleftarrow_{\mathcal{D}} \{0,1\}^{l-2}$ it holds that the random variable $\mathsf{Bit}_{b-1}(m)$ is the uniform over $\{0,1\}$ (where $\mathsf{Bit}_{b-1}(m) = m_{b-1}$ whenever $m = m_0 \dots m_{b-1}$). In the next lemma, we establish the property US2 of definition **??** for scheme 1, i.e., that an adversarial user 1, has an upper bound on the successful decoding of any ciphertext produced by E_0.

Lemma 19 *Let $\mathcal{B}$ be a pirate-box that is produced by using the secret-key sk_1 of the public-key encryption scheme and correctly decrypts valid ciphertexts with probability $\sigma(l) \geq 1/2 + \eta(l)$ where $\eta(l)$ is a non-negligible function for plaintext distributions $\mathcal{D} \in \Delta$. Then, under the QR Assumption, $\mathcal{B}$ has success probability at most $1/2$ in decrypting "correctly" any ciphertext prepared by the function E_0 (i.e., decrypting it in the way that the honest user 0 decrypts it).*

Proof. Let g, G, n be a challenge for the QR Assumption, i.e., n is an RSA modulus of the type used by our scheme 1, g is a generator of quadratic residues modulo n, and G is a uniformly distributed element from either $\mathcal{Q}_n$ or $\mathcal{J}_n - \mathcal{Q}_n$. We simulate the key-assignment to user 1,

by selecting $\alpha \in_R [t]$ and providing to the adversary the value α_1 which is the odd number among $\{\alpha, \alpha + t - 1\}$ (recall that in our setting $t = \lceil n/4 \rceil$ is even) and setting the public-key as $h = g^{\alpha_1}$.

Let $m = m_0 m_1 \dots m_{b-1} \hookleftarrow_{\mathcal{D}} \{0,1\}^b$. Consider the ciphertext $\langle -G, G^{\alpha_1} \cdot enc(m) \rangle$. Observe that if G is a uniform element of $\mathcal{Q}_n$ then the ciphertext is an E_0-ciphertext encrypting $m_0 \dots m_{b-1}$. On the other hand if G is a uniform element of $\mathcal{J}_n - \mathcal{Q}_n$ the ciphertext is a valid E ciphertext encrypting $m_0 \dots m_{b-2} \overline{m_{b-1}}$.

Now consider the event Flip that the pirate box $\mathcal{B}$ returns $m_0 \dots m_{b-2} \overline{m_{b-1}}$. In case G is a uniform element of $\mathcal{J}_n - \mathcal{Q}_n$ it follows that the probability of the event Flip should be at least $\sigma(l)$, the probability that the pirate-box correctly decrypts a valid ciphertext. Note here that both $m_0 m_1 \dots m_{b-1}$ and $m_0 m_1 \dots m_{b-2} \overline{m_{b-1}}$ are equally likely as plaintexts of the pirate-box distribution $\mathcal{D} \in \Delta$.

On the other hand, if G is a uniform element of $\mathcal{Q}_n$, we have that the probability of the event Flip should be $|\mathbf{Prob}[\mathsf{Flip}] - \sigma(l)| \leq \mathsf{negl}(l)$ under the QR Assumption (otherwise we can turn the event Flip into a distinguisher). It follows that the probability of the event Flip is at least $\sigma(l) - \mathsf{negl}(l)$ which implies that the probability of correct decryption of the E_0 ciphertext (i.e., in the way that user 0 decrypts it) can be at most $1 - \sigma(l) + \mathsf{negl}(l)$ since the Flip event suggests incorrect decryption for the pirate-box $\mathcal{B}$. Since $\sigma(l) \geq 1/2 + \eta(l)$ it follows that $\mathcal{B}$ can correctly decrypt invalid E_0 ciphertexts with probability at most $1/2 - \eta(l) + \mathsf{negl}(l) \leq 1/2$. □

Armed with the two lemmas above we obtain the following corollary:

Corollary 20 *The 2-user public-key encryption scheme 2 is a $[\Delta, \frac{1}{2}]$-user-separable public-key encryption scheme, where Δ is a family of all plaintext distributions as defined above.*

Using now theorem **??** and the above corollary the following theorem follows easily.

Theorem 21 *The 2-key public-key encryption scheme 1 described above is a 2-key public-key encryption scheme satisfying $[\Delta, \frac{1}{2}, 1]$-semantic-black-box-traceability with the analyzer $\mathcal{N}$ of figure **??** using the special encryption E_0.*

In particular, for any $\langle \Delta, \sigma \rangle$-pirate $\mathcal{P}$ such that $\sigma(l) \geq 1/2 + \alpha(l)$ where $\alpha(l)$ is a non-negligible function it holds that (1) $\mathcal{N}$ has success probability $1 - \epsilon$, (2) the running time of $\mathcal{N}$ is polynomial in $\ln(\epsilon^{-1}) + \alpha(l)^{-1} + \sigma(l)^{-1}$.

Proof. The proof follows directly from the proof of theorem **??** using the results of lemmas **??** and **??**. □

We note that the above suggests a rather weak form of black-box traceability. First, it is restricted to plaintext distributions that have high entropy on a certain bit of the input (note that the bit that one relies upon can be changed by modifying the encoding accordingly, i.e., our assumption is that the plaintext distribution contains a truly random bit). Still, the biggest restriction in the above definition is that it allows the existence of pirate-boxes that can decrypt correctly about half of the time and can remain undetected; we note that such pirate-boxes may in fact be quite useful in many settings. In the coming section, our scheme 2, will circumvent these limitations and provide black-box traceability for essentially any arbitrary distribution and pirate decoders.

5.2 Scheme 2

Consider the following public-key encryption function $\langle \mathbb{P}, \mathbb{C}, \mathbb{K}, \mathbb{S}, G, E, D \rangle$: The function G given 1^l simulates desc_{prime} to obtain the two prime numbers p, q as specified in section **??**. Let $\mathcal{G}$ be the subgroup of quadratic residues of $\mathbb{Z}_p^*$ where $p = 2q + 1$ where both p, q, are primes. It follows that the order of $\mathcal{G}$ is q. Let g be a generator of $\mathcal{G}$. The public-key of the scheme is set to $pk =_{\text{df}} \langle p, f, g, h \rangle$ where $f =_{\text{df}} g^\alpha, h =_{\text{df}} g^\beta$ and $\alpha, \beta \in_R [q]$. The two users are given two "representations" of α with respect to the "base" g, h as their secret-keys, i.e. two vectors $\langle d_0, d_0' \rangle, \langle d_1, d_1' \rangle$ over $\mathbb{Z}_q$ so that $d_i + \beta d_i' = \alpha$ for both $i \in \{0, 1\}$. The two vectors are chosen so that they are linearly independent over $\mathbb{Z}_q$. Note that the set of all possible keys is $\mathbb{S}_{l,pk} =_{\text{df}} \{\langle d, d' \rangle \mid d + d'\beta = \alpha (\text{mod} q)\}$. The tracing trapdoor τ of scheme 2 will be set to the tuple $\langle d_0, d_0', d_1, d_1', \beta \rangle$.

Encryption is performed as follows: given the public-key $\langle f, g, h \rangle$ and a plaintext message $m \in \{0,1\}^b$, the encryption of m is $\langle enc(m) \cdot f^r \bmod p, g^r \bmod p, h^r \bmod p \rangle$, where $enc : \{0,1\}^b \to \mathcal{G}$ is an appropriate encoding (to be specified below). Decryption works as follows: given one of the two keys $\langle d_i, d_i' \rangle$ and a ciphertext $\langle A, B, C \rangle$ the receiver computes $enc^{-1}(A(B^{-1})^{d_i}(C^{-1})^{d_i'} \bmod p)$. It is easy to verify that the decryption operation inverts encryption.

Next we specify the appropriate encoding function for the plaintext-space $\{0,1\}^b$. We set $b = |p| - 2 = l - 2$. The invertible encoding function $enc : \{0,1\}^b \to \mathcal{G}$ is as follows: given $m = m_1 \dots m_b \in \{0,1\}^b$ let $m' = m_1 + 2m_2 + \dots 2^b m_b + 1$. It is easy to verify that $m' \in \{1, \dots, q\}$. Then, we set $enc(m) =_{\text{df}} (m')^2 \bmod p$. It is easy to see that $enc(m) \in \mathcal{G}$ for any $m \in \{0,1\}^b$: this is because $\mathcal{G} = \langle g \rangle$ is the subgroup of quadratic residues modulo p. The encoding function enc can be inverted as follows: given $enc(m)$ we compute its two square roots modulo p and let m' be the one that belongs in $\{1, \dots, q\}$. It follows that, the decoding of $enc(m)$ is the binary representation of $m' - 1$. The rates of the parameters of the system are illustrated in the figure **??** (recall that $|p| = b + 2$). Traceability for collusions of size 1 will be established in the remaining of the section.

Plaintext Space	Ciphertext Rate	User-Key Rate	Public-Key Rate	Max Traceable Collusion
$\{0,1\}^b$	$\frac{3(b+2)}{b} \sim 3$	$\frac{2(b+1)}{b} \sim 2$	$\frac{4(b+2)}{b} \sim 4$	1

Figure 3: Efficiency Parameters of Scheme 2 (Two-User Setting)

5.2.1 Semantic security and security against passive adversaries

We proceed to the security analysis of the 2-key public-key encryption scheme 2.

Theorem 22 *The public-key encryption function described above is*
(i) Semantically Secure under the DDH *Assumption over* $\mathcal{G}$.
(ii) 1-copyrighted against passive adversaries (in the strong sense): given the public-key information pk and a key $\langle d, d' \rangle \in \mathbb{S}_{l,pk}$ *it is computationally infeasible to construct another key in* $\mathbb{S}_{l,pk}$ *under the Discrete-Log assumption over* $\mathcal{G}$.

Proof. (i) Recall, from the proof of theorem **??**, the definition of semantic security in the sense of message indistinguishability as a game G_0. The challenge ciphertext has the form $\langle A, B, C \rangle$.

Let g, h, G, H be a challenge for the DDH assumption over $\mathcal{G}$. We use the following public-key : $\langle g, h, f = g^{a_0} h^{a_1} \rangle$ where $a_0, a_1 \in_R [q]$. Next, we modify the challenge ciphertext as follows: $\langle G, H, G^{a_0} H^{a_1} \cdot enc(m_b) \rangle$.

Observe that in this modified game as described above, whenever $\langle g, h, G, H \rangle$ is a valid DDH triple then the modified game induces an identical distribution to that of the original indistinguishability game G_0. On the other hand, if $\langle g, h, G, H \rangle$ is distributed according to the uniform distribution, we have that $G = g^{r_0}, H = h^{r_1}$ and $G^{a_0} H^{a_1} = g^{r_0 a_0} h^{a_1 r_1}$.

Now define the value $u = r_0 a_0 + a_1 r_1 \log_g(h)$ in $\mathbb{Z}_q$. We will show that this value in the information theoretic sense can assume with (almost) equally likely probability any value in $[q]$ in the view of the semantic security adversary $\mathcal{A}$. This is the case, since the only information communicated regarding the values a_0, a_1 is through the value $v = a_0 + a_1 \log_g(h)$. It follows that the linear system of the two equations for u and v has a single unique solution (a_0, a_1) for each choice of u as long as $r_0 \log_g(h) \neq r_1$ (i.e., its determinant is non-zero). Given that r_0, r_1 are selected uniformly at random over $[q]$ we conclude that conditioning on the event $r_1 \neq r_0 \log_g(h)$, the value u is uniformly distributed over $[q]$ and thus the probability that the adversary $\mathcal{A}$ guesses b correctly is exactly $1/2$ (since no information about b is conveyed by $g^u \cdot enc(m_b)$). Moreover the event $r_1 = r_0 \log_g(h)$ is a $1/q$ probability event, thus we conclude that any semantic security adversary can have advantage at most $1/2 + \epsilon + 1/q$ where ϵ is the advantage of the best possible polynomial time distinguisher for the DDH assumption over $\mathcal{G}$. This concludes the proof for (i).

(ii) Observe that given any two distinct keys $\langle d_0, d_0' \rangle, \langle d_1, d_1' \rangle \in \mathbb{S}_{l,pk}$, we have that $d_i + d_i' \beta = \alpha$. It cannot be the case that $d_0' = d_1'$. Indeed if this is the case we have that $d_0 = d_1$ as well and thus the two keys are the same. It follows that the above two equations, yield a linear system with two uknowns α, β and determinant $d_0' - d_1'$ that is solvable uniquely revealing the secret exponents α, β. It follows that any adversary that violates the strong copyrighted property can be used to recover the secret exponents α, β.

Consider g, h an instance of the discrete-logarithm problem. We define $f = g^a h^{a'}$ with $a, a' \in_R [q]$ and set the public-key as g, h, f. It follows that a, a' is a valid secret-key that can be given to an adversary against the strong copyrighted property. Note that $\langle a, a' \rangle$ and a secret key as given in the system's key assignment operation follow identical distributions. Now suppose that the adversary returns a pair $\langle b, b \rangle$ that is different from $\langle a, a' \rangle$ and is also a valid key of the system. This in turn reveals $\alpha = \log_g(h)$ which is the solution of the given instance of the discrete-logarithm problem. □

As in the case of scheme 1, the construction presented above is strictly 1-copyrighted in the strong sense and not 2-copyrighted in the strong sense since a collusion of two users is capable of recovering the secret-keys α, β and thus capable of sampling $\mathbb{S}_{l,pk}$ at will. For example, given $\langle d_0, d_0' \rangle$ and $\langle d_1, d_1' \rangle$ it holds that $\langle r d_0 + (1-r) d_1, r d_0' + (1-r) d_1' \rangle \in \mathbb{S}_{l,pk}$ for any $r \in \mathbb{Z}_q$. Still the scheme is 2-copyrighted against passive adversaries since if a key is recovered that belongs to none of the two users it holds that we can accuse correctly any of the users as a traitor.

5.2.2 Security against active adversaries : black-box traitor tracing

In this section we describe the analyzer procedure $\mathcal{N}$ against active adversaries, or black-box traitor tracing. First we establish user separability. We describe a special encryption function E_0 that for a given message distribution $\mathcal{D}$ it encrypts plaintexts so that user 0 finds them indistinguishable from other ciphertexts whereas the other user cannot decrypt correctly. Given

$m \hookleftarrow \mathcal{D}$ and the tracing trapdoor information τ, E_0 selects random $s_0, s_1 \in_R [q]$ and computes r_0, r_1, $r_0 d_0 + r_1 d'_0 \beta = s_0$ and $r_0 d_1 + r_1 d'_1 \beta = s_1$. Note that the determinant of the above system equals $\beta(d_0 d'_1 - d_1 d'_0)$ and is non-zero as long as $d_0 d'_1 \neq d'_0 d_1$ and $\beta \neq 0$. Recall that $\langle d_0, d'_0 \rangle$ and $\langle d_1, d'_1 \rangle$ are both solutions of the equation $x + \beta y = \alpha$ and the equality $d_0 d'_1 = d'_0 d_1$ implies that $(\alpha - \beta d'_0) d'_1 = d'_0(\alpha - \beta d'_1)$. In turn from this we obtain that, as long as $\alpha \neq 0$, it holds that $d'_0 = d'_1$ which is a $1/q$ probability event given that we select these values at random (and we compute the values d_0, d_1 based on them). It follows that with probability at least $1 - 3/q$ over the choices of the public-key of the encryption scheme, the linear system suggested above yields a unique solution for the variables r_0, r_1 using any choice of s_0, s_1. The ciphertext that is prepared by E_0 is of the following form: $\langle g^{s_0} \cdot enc(m), g^{r_0}, h^{r_1} \rangle$.

The analyzer procedure $\mathcal{N}$ will use this special encryption function E_0 to submit ciphertexts to the pirate decoder box $\mathcal{B}(\cdot)$ and obtain the decoder's answer m' following the description of figure **??**. The process is repeated K times. Let $K' \leq K$ be the number of times that the decoder responds correctly (i.e., it holds that $m' = m$). The analyzer $\mathcal{N}$ will return user 0 (that is the user holding the key $\langle d_0, d'_0 \rangle$) if $K'/K > \psi \cdot (\sigma - \epsilon_0)$ where $\psi \in (0, 1)$ is the parameter specified in theorem **??**, otherwise it concludes that user 1 is responsible.

In the remaining of the section we will establish the fact that the above described procedure $\mathcal{N}$ is a black-box traitor tracing algorithm for the 2-key encryption scheme by arguing first, that the 2-key encryption is user-separable and then employing theorem **??**.

First, in the following lemma, we argue that from the point of view of user 0, acting as the advesary, it is impossible to distinguish the ciphertexts that are produced by E_0 compared to regular ciphertexts produced by E, thus property US1 of definition **??** is satisfied.

Lemma 23 *Suppose that $\mathcal{A}$ is a* PPT *that is given d_0, d'_0 and the public-key f, g, h, acts as a distinguisher of the probability distributions $\langle f^r \cdot enc(m), g^r, h^r \rangle$ and $\langle g^{s_0} \cdot enc(m), g^{r_0}, h^{r_1} \rangle$ where f, g, h is a public-key of the public-key encryption scheme as produced by $G(1^l)$, m is any plaintext in $[\mathbb{P}]_l$ and r_0, r_1, s_0 are selected as above. Suppose that the distinguishing probability of $\mathcal{A}$ is ϵ. Given $\mathcal{A}$ we can build a* DDH *distinguisher with the same distinguishing probability.*

Proof. We describe the DDH distinguisher $\mathcal{B}$. Given $\langle g, h, G, H \rangle$, we select d_0, d'_0 at random from $[q]$ and set $f = g^{d_0} h^{d'_0}$. The values $\langle f, g, h \rangle$ form the public-key of the system. $\mathcal{B}$ simulates $\mathcal{A}$ with input $\langle f, g, h \rangle$ and a secret key $\langle d_0, d'_0 \rangle$. Observe that d_0, d'_0 are indistinguishable from the distribution during the honest key generation.

Next, $\mathcal{B}$ prepares the ciphertext $\langle G, H, G^{d_0} H^{d'_0} \cdot enc(m) \rangle$ and simulates $\mathcal{A}$ on that input. It returns the output that $\mathcal{A}$ returns.

It is easy to see that if $\langle g, h, G, H \rangle$ is a valid DDH triple then the ciphertext calculated by $\mathcal{B}$ is indistinguishable from a valid ciphertext under the public-key $\langle f, g, h \rangle$. On the other hand, if $\langle g, h, G, H \rangle$ is a random DDH triple then the ciphertext $\langle G, H, G^{d_0} H^{d'_0} \cdot enc(m) \rangle$ has the form $\langle g^{r_0}, h^{r_1} g^{r_0 d_0} h^{r_1 d'_0} \cdot enc(m) \rangle$. If we denote by s_0 the value $r_0 d_0 + r_1 d'_0 \log_g(h)$ then we have that the ciphertext is of the form $\langle g^{r_0}, h^{r_1}, g^{s_0} \cdot enc(m) \rangle$, which is indistinguishable from the ciphertexts produced by the E_0 alternative encryption. We conclude that if user 0 has distinguishes valid ciphertexts from invalid ciphertexts prepared by $\mathcal{N}$ then with the same distingushing probability we can solve the DDH assumption. □

In this way we have established that from the point of view of user 0, the ciphertexts produced by the analyzer procedure using E_0 are indistinguishable from valid ciphertexts. This is not the case of course for user 1. Nevertheless, based on our construction, user 1,

despite the fact that he may realize that the ciphertext is invalid he will be unable to deliver the appropriate output to appear as user 0. This case will be argued in the following lemma that will establish property US2 of the definition of user-separability.

Lemma 24 *Suppose that $\mathcal{A}$ is a* PPT *that is given $\langle d_1, d'_1\rangle$ the public-key f, g, h and a special E_0 ciphertext of the form $\langle g^{s_0} \cdot enc(m), g^{r_0}, h^{r_1}\rangle$ as defined above. The plaintext m follows the distribution $\mathcal{D}$ that has min-entropy $\mu(l)$. The probability that $\mathcal{A}$ returns m is bounded from above by $2^{-\mu(l)}$.*

Proof. Recall that the selection of r_0, r_1 is as follows: first random s_0, s_1 are selected; then the system $r_0 d_0 + r_1 d'_0 \beta = s_0$ and $r_0 d_1 + r_1 d'_1 \beta = s_1$ is solved to calculate r_0, r_1. The ciphertext prepared by $\mathcal{N}$ is of the form $\langle g^{s_0} \cdot enc(m), g^{r_0}, h^{r_1}\rangle$.

In the information theoretic sense the following values are bound in the view of the adversary $\mathcal{A}$: $\beta = \log_g(h), \log_g(f), r_0, r_1$. Note that by selection the value s_1 is also bound. On the other hand the only information about the value s_0 given to $\mathcal{A}$ is through the value $F = g^{s_0} \cdot enc(m)$. It follows that conditioning on F, for each $m' \in [\mathbb{P}]_l$ there is exactly one choice of s_0 that "explains" the value F, $s_0 = \log_g(F/enc(m'))$. In our conditional space, s_0 can assume any of these values with the same probability of success. It follows that in the view of $\mathcal{A}$ the value F carries no information about m and as a result the probability of returning m will be bounded by $2^{-\mu(l)}$ where $\mu(l)$ is the min entropy of the distribution $\mathcal{D}$. □

Armed with the two lemmas above we obtain the following corollary:

Corollary 25 *The 2-user public-key encryption scheme 2 is a $[\Delta, 2^{-\mu(l)}]$-user-separable public-key encryption scheme, where Δ is a family of all plaintext distributions that have min-entropy $\mu(l)$.*

Using now theorem **??** and the above corollary the following theorem follows easily.

Theorem 26 *The 2-key public-key encryption scheme 2 described above satisfies $[\Delta, 2^{-\mu(l)}, 1]$-semantic-black-box-traceability with the analyzer $\mathcal{N}$ of figure **??** using the special encryption E_0, where Δ contains any plaintext probability ensemble that has min-entropy $\mu(l)$.*

In particular, for any $\langle \Delta, \sigma\rangle$-pirate $\mathcal{P}$ such that $\sigma(l) \geq 2^{-\mu(l)} + \alpha(l)$ it holds that (1) $\mathcal{N}$ has success probability $1 - \epsilon$, (2) the running time of $\mathcal{N}$ is polynomial in $\ln(\epsilon^{-1}) + \alpha(l)^{-1} + \sigma(l)^{-1}$.

Proof. The proof follows directly from the proof of theorem **??** using the results of lemmas **??** and **??**. □

6 The multi-user case

Let $\langle \mathbb{P}, \mathbb{C}, \mathbb{K}, \mathbb{S}, G, E, D\rangle$ be a 2-key public-key encryption scheme. In this section we will show how is it possible to extend this construction to n-keys using a composition of the underlying construction with a collusion secure code [**?**]. This technique was employed in [**?**] to construct a copyrighted hash function based on RSA by composing such codes to a basic 2-key copyrighted hash function. Here, we show how to obtain an multi-key public-key traitor tracing schemes by a parallel composition of independent instantiations of a 2-key encryptions based on collusion-secure codes. Note that [**?**] employed a nested composition instead, whereas here we employ a

parallel type of composition; our approach is geared towards attaining black-box traceability (something that is not apparent how to achieve in the previous work).

Let $\mathcal{C} =_{\text{df}} \{\omega_1, \ldots, \omega_n\}$ be a $\langle n, v\rangle_2$-collusion-secure code over the alphabet $\{0,1\}$ with v-long codewords, that allows collusions of up to c and has a tracing algorithm that succeeds with probability $1-\epsilon$ in returning a traitor. More specifically, to detail the traceability of collusion secure codes we need the following definition:

Definition 27 *Given a set of codewords $C = \{\omega_{i_1}, \ldots, \omega_{i_t}\} \subseteq \mathcal{C}$ an undetectable position is a location $i \in \{1, \ldots, v\}$, such that $(\omega_{i_1})_i = \ldots = (\omega_{i_t})_i$. The set of undetectable positions is denoted by $U(C)$. The* feasible set *of C denoted by $F(C)$ is defined as:*

$$F(C) = \left\{\omega \in \{0,1,?\}^v \mid (\omega)_{U(C)} = (\omega_{i_1})_{U(C)}\right\}$$

The traceability of collusion secure codes imply that there is a tracing procedure $\mathcal{T}$ such that given any string $\omega^* \in F(C)$ where $C = \{\omega_{i_1}, \ldots, \omega_{i_t}\}$ with $t \leq c$ it holds that $\mathcal{T}$ returns with probability at least $1-\epsilon$ one of the indices $i_1, \ldots, i_t$. Collusion secure codes were introduced in [**?**], and further investigated in [**?**, **?**, **?**, **?**]. Note that collusion secure codes are generated by a probabilistic procedure that also creates a secret tracing trapdoor τ_0 and is employed by the tracing procedure $\mathcal{T}$. The code generation procedure defines a family of code distributions one for each value of $n \in \mathbb{N}$.

We define next our composite multi-user public-key encryption scheme and argue about its traceability properties. The scheme will be built on a collusion secure code family and a 2-key public-key encryption scheme.

Key-Generation. The key-generation procedure, generates a collusion secure code $\mathcal{C} = \{\omega_1, \ldots, \omega_n\}$ with tracing trapdoor τ_0 and then generates v independent key-instantiations of a 2-key public-key encryption scheme:

$$\{\langle pk_i, sk_{0,i}, sk_{1,i}\rangle\}_{i=1}^{v}$$

with tracing trapdoors $\tau_1, \ldots, \tau_v$.

Without loss of generality we assume that the plaintext-space $\mathbb{P}$ over all instantiations is the same ($= \{0,1\}^b$) and that $\text{len}[c \in \mathbb{C}_1] = \ldots = \text{len}[c \in \mathbb{C}_1]$. The i-th decryption key of the n-key system is defined as the following sequence $\vec{sk}_i =_{\text{df}} \langle sk_{i,\omega_{i,1}}, \ldots, sk_{i,\omega_{i,v}}\rangle$ where $\omega_{i,\ell}$ is the ℓ-th bit of the i-th codeword of $\mathcal{C}$. The tuple $\langle pk_1, \ldots, pk_v\rangle$ constitutes the public-key.

Encryption and Decryption. The plaintext space of the n-key system is $\mathbb{P}^v$. A message $\langle M_1, \ldots, M_v\rangle$ is encrypted by the tuple $\langle E(pk_1, M_1), \ldots, E(pk_v, M_v)\rangle$. Because each user has one key that inverts $E(pk_\ell, \cdot)$ (either $sk_{0,\ell}$ or $sk_{1,\ell}$) for all $\ell = 1, \ldots, v$ it is possible for any user to invert a ciphertext and compute $\langle M_1, \ldots, M_v\rangle$.

Traceability. To argue about the traceability of the above construction suppose that the 2-key public-key encryption scheme satisfies 1-traceability. Suppose now that $\langle sk_1^*, \ldots, sk_v^*\rangle$ is a key that was constructed by a coalition of t users s.t. $t \leq c$. Given such key the tracer constructs a codeword $\omega^* =_{\text{df}} \omega_1^* || \ldots || \omega_v^*$ as follows

$$\omega_i^* =_{\text{df}} 0 \ \text{(if } \ sk_i^* = sk_{0,i}) \ \text{ OR } \ \omega_i^* =_{\text{df}} 1 \ \text{(if } \ sk_i^* = sk_{1,i}) \ \text{ OR } \ \omega_i^* =_{\text{df}} ? \ \text{(otherwise)}$$

Because of the fact that each instance of the encryption satisfies the traceability property against coalitions of one user it is possible to recover the key used in a coordinate where

Analyzer $\mathcal{N}$ for given plaintext distribution $\mathcal{D} \in \Delta_v$
Input : tracing trapdoor $\tau_0, \tau_1, \ldots, \tau_v$; public-key $\langle pk_1, \ldots, pk_v \rangle$; security parameter 1^l
Oracle access: pirate-box $\mathcal{B}$
Parameters: $K, v \in \mathbb{N}, \psi, \sigma \in (0,1)$
For $\ell = 1, \ldots, v$
    $K' = 0$;
    For $i = 1, \ldots, K$
        sample $m_i \hookleftarrow_{\mathcal{D}} ([\mathbb{P}]_l)^v$;
        parse m_i as $m_{i,1} \ldots m_{i,v}$;
        set $C_{i,\ell} \leftarrow E_0(pk_\ell, \tau_\ell, m_\ell)$;
        set $C_{i,\ell'} = E(pk_{\ell'}, m_{\ell'})$ for all $\ell' \in \{1, \ldots, v\} - \{\ell\}$;
        $m'_{i,1} \ldots m'_{i,v} \leftarrow \mathcal{B}(C_{i,1} \ldots C_{i,v})$;
        if $(m_{i,\ell} = m'_{i,\ell})$ then $K' = K' + 1$;
    if $K'/K \geq \psi \cdot (\sigma - \epsilon_0)$ set $\omega^*_\ell = 0$ else set $\omega^*_\ell = 1$;
Return output of tracing procedure of $\mathcal{C}$ with input τ_0 and $\omega^* = \omega^*_1 \ldots \omega^*_v$;

Figure 4: Black-Box Analyzer Procedure for a n-user composite public-key encryption based on v instances of 2-key $[\Delta, \kappa]$-user-separable public-key encryption scheme and a collusion secure code $\mathcal{C}$.

all traitors share the same key. This in turn suggests that if $C =_{\text{df}} \{\omega_{i_1}, \ldots, \omega_{i_t}\}$ is the set of codewords that corresponds to the keys of the coalition of traitor users that constructed $\langle sk^*_1, \ldots, sk^*_v \rangle$ as returned by the traceability algorithm, it holds that $\omega^* \in F(C)$, where $F(C)$ is the *feasible* set of the codewords C (see [**?**]); it follows that if ω^* is given as input to the tracing algorithm of the collusion-secure-code $\mathcal{C}$, and because $|C| \leq c$, we are guaranteed to obtain the identity of one of the traitors. In order to achieve probability of success $1 - \epsilon$ we will need to employ a public-key encryption scheme with traceability success at least $1 - \epsilon/2v$ and a collusion secure code with probability of success at least $1 - \epsilon/2$. Note that the above argumentation assumes implicitly that a key for all v instantiations will be found inside the pirate decoder, i.e. implemenations of pirate decoders that omit keys are not useful. We deal with how this can be enforced in more details in section **??** where we describe the two public-key traitor tracing schemes based on this construction.

Black-box traceability. Suppose now that the underlying 2-key public-key encryption scheme satisfies $[\Delta, \kappa]$-user-separability. We show that it is possible in the composite scheme to construct the codeword ω^* using merely black-box access to the pirate decoder: the analyzer procedure (cf. figure **??**) performs the black-box tracing analysis procedure for each coordinate independently in a left-to-right sweep of the v-long composite ciphertext. This will have the effect of identifying the key employed by the pirate-box for each coordinate. Thus, we can calculate the symbol ω^*_i as defined above as the output of the analyzer procedure for the i-th coordinate. Based on this argumentation, the proof of the following theorem follows easily.

Theorem 28 *The n-key composite public-key encryption scheme described above that is based on a $\langle n, v \rangle_2$-collusion-secure code for coalitions up to c and v instantiations of a 2-key public-key encryption scheme that is $[\Delta, \kappa]$-user-separable satisfies $[\Delta_v, \kappa, c]$-semantic-black-box-traceability, where Δ_v is a family of plaintext distributions over $\mathbb{P}^v$ that adheres to the following assumption:*

Projection Assumption*: if $\mathcal{D}_v \in \Delta_v$ then the projection of $\mathcal{D}_v$ into its i-th coordinate induces a probability distribution over $\mathbb{P}$ that belongs to Δ.*

In particular for any $\langle \Delta_v, \sigma \rangle$-pirate $\mathcal{P}$, such that $\sigma(l) \geq \kappa(l) + \alpha(l)$ it holds that (1) $\mathcal{N}$ has success probability at least $1-\epsilon$, (2) the running time of $\mathcal{N}$ is polynomial time in $v(\ln(v+1) + \ln(\epsilon^{-1}) + \alpha(l)^{-1} + \sigma(l)^{-1})$ plus the time required to perform tracing on the collusion-secure code $\mathcal{C}$ with success probability $1 - \frac{\epsilon}{2}$.

Proof. Let $\mathcal{P}$ be a $[\Delta_v, \sigma]$-pirate against the composite TTS that for a certain choice of the public-key produces a pirate-box $\mathcal{B}$ based on up to c of the secret-keys and a plaintext distribution $\mathcal{D} \in \Delta_v$. Based on the specifications, the success probability of $\mathcal{B}$ when given a valid ciphertext would be at least $\sigma(l)$.

To establish the black-box traceability of the composite scheme we demonstrate that the analyzer procedure of figure **??** will determine one traitor given access to the pirate box produced by $\mathcal{P}$. Based on the statement of the theorem we have that $\mathcal{D}$ when projected to its v coordinates induces a sequence of distributions $\mathcal{D}_1, \ldots, \mathcal{D}_v$ each one belonging to Δ. Given that $\mathcal{B}$ correctly decrypts $\mathcal{D}$ with success at least $\sigma(l)$ it follows that during the (ℓ, i)-th execution of the body of the main loop of the analyzer of figure **??** it holds that the pirate-box $\mathcal{B}$ will be successful in decrypting correctly the E_0 ciphertext with probability at least $\sigma(l) - \epsilon_0(l)$ if the coalition of traitor keys all agree in their ℓ-th location and possess the key $sk_{0,\ell}$, whereas the pirate-box $\mathcal{B}$ will successfully decrypt the E_0 ciphertext with probability at most $\kappa(l)$ if the coalition of traitor keys all agree in their ℓ-th location and possess the key $sk_{1,\ell}$. Note that if both keys $sk_{0,\ell}, sk_{1,\ell}$ are available to the traitor coalition no guarrantees are given on the output of $\mathcal{B}$.

Next we specify the parameters ψ and γ as follows: $\psi = 1 - \frac{\alpha}{2\sigma}$ and $\gamma = \alpha/2$; then based on lemma **??** we know that we can set K to be a polynomial in $\ln(v) + \ln(\epsilon^{-1}) + \alpha(l) + \sigma(l)^{-1}$ and obtain a correct prediction for ω^*_ℓ with probability at least $1 - \frac{\epsilon}{2v}$.

It follows that with probability $(1 - \frac{\epsilon}{2v})^v \geq 1 - \frac{\epsilon}{2}$ the analyzer procedure will recover the bitstring ω^* in time polynomial in $v \cdot (\ln(v+1) + \ln(\epsilon^{-1}) + \alpha(l)^{-1} + \sigma(l)^{-1})$, and by applying the tracing of the collusion secure code the analyzer $\mathcal{N}$ of figure **??** will obtain the result in time-polynomial in n and success at least $1 - \frac{\epsilon}{2}$. It follows that the overall success probability of the analyzer will be at least $(1 - \frac{\epsilon}{2})^2 \geq 1 - \epsilon$. □

Efficiency Parameters and Constant Transmission Rate. It is easy to see that the derived scheme has the same ciphertext rate, user-key rate and public-key rate as the underlying 2-key public-key encryption scheme. This is because the v-fold expansion of these parameters is cancelled by the simultaneous v-fold expansion of the plaintext-space.

7 Two multi-user public-key traitor tracing schemes

The application of the construction of the previous section to the 2-key public-key encryption schemes of sections **??** and **??** yields two public-key traitor tracing schemes. In the following we will use the collusion secure code $\mathcal{C} = \{\omega_1, \ldots, \omega_n\}$ of Tardos [**?**] that has code length $v = \mathcal{O}(c^2 \log(n/\epsilon))$ where c is the maximum collusion size and $1 - \epsilon$ is the lower bound on the success probability of the tracing algorithm.

7.1 Public-key traitor tracing scheme 1

In the following ℓ is interpreted as a value in $\{1, \ldots, v\}$.
Key Generation. Select $N_1, \ldots, N_v$ composite numbers so that $N_\ell = p_\ell q_\ell$ and p_ℓ, q_ℓ satisfy the properties described in section **??**. Also recall that $\rho_\ell = \frac{p_\ell - 1}{2}\frac{q_\ell - 1}{2}$. Without loss of generality we assume that $l =_{\text{df}} |N_1| = \ldots = |N_v|$. The public-key of the system is the set to

$$\langle N_1, g_1, y_1 =_{\text{df}} g_1^{\alpha_1} \bmod N_1\rangle, \ldots, \langle N_v, g_v, y_v =_{\text{df}} g_v^{\alpha_v} \bmod N_v\rangle$$

where each $\langle g_\ell\rangle = \mathcal{Q}_{N_\ell}$ and $\alpha_\ell \in_U [\rho_\ell]$. User i is given as its personal decryption key the tuple $\langle sk_{1,\omega_{i,1}}, \ldots, sk_{v,\omega_{i,v}}\rangle$, where $\{sk_{\ell,0}, sk_{\ell,1}\} = \{\alpha_\ell, \alpha_\ell + \rho_\ell\}$ where each $sk_{\ell,0}$ is selected to be the evennumber of the pair.
Encryption. Any third party can encrypt a message $\langle m_1, \ldots, m_v\rangle \in \{0,1\}^{v\cdot b}$ where $b = l - 2$ by employing the encoding defined in section **??** in the following way: $\langle g_1^{r_1} \bmod N_1, y_1^{r_1} \cdot enc(m_1) \bmod N_1, \ldots, g_v^{r_v} \bmod N_v, y_v^{r_v} \cdot enc(m_v) \bmod N_v\rangle$ where $r_\ell \in_U [N_\ell]$.
Decryption. Given a ciphertext $\langle A_1, B_1, \ldots, A_v, B_v\rangle$ and a user-key $\langle sk_1, \ldots, sk_v\rangle$ the decryption is $enc^{-1}(B_1(A_1^{-1})^{sk_1} \bmod N_1)|| \ldots ||enc^{-1}(B_v(A_v^{-1})^{sk_v} \bmod N_v)$.
Traceability. It is easy to see that the construction satisfies the traceability property; in particular as we have shown in theorem **??**(ii) the underlying 2-key public-key encryption scheme is 1-copyrighted in the strong sense under the factoring assumption, thus if a pirate has to construct a pirate-key of the form $\langle sk_1, \ldots, sk_v\rangle \in \mathbb{S}_{l,pk}$ and it is the case that in the ℓ-th coordinate all traitor-keys agree, the pirate will be forced to use that exact key assuming factoring is hard. Based on this, it follows easily :

Proposition 29 *The n-key public-key encryption scheme presented above satisfies c-traceability under the factoring Assumption where c is a* $(\mathbb{N} \to \mathbb{N})$ *function such that* $c(n)$ *is the maximum traitor collusion size of the underlying* $\langle n, v\rangle_2$ *collusion secure code employed in the construction.*

Proof. Follows directly from the traceability of the underlying collusion secure codes and theorem **??**(ii). □

Black-Box Traceability. The analyzer procedure is directly derived from theorem **??** and corollary **??**. In particular, the analyzer procedure will be successful in tracing any pirate-box that has success probability in returning the correct plaintext with probability at least $1/2 + \alpha$ where α is a non-negligible function in the security parameter and moreover the plaintext distribution is restricted so that the plaintext distribution Δ_v contains at least one truly random bit in each of its v coordinates (and the location of this bit is known and incorporated into the encoding).

Theorem 30 *The n-key public-key encryption scheme presented above satisfies* $[\Delta, \frac{1}{2}, c]$*-semantic-black-box-traceability under the QR Assumption where* Δ *is the family of plaintext distributions as defined above and c is a* $(\mathbb{N} \to \mathbb{N})$ *function such that* $c(n)$ *is the maximum traitor collusion size of the underlying* $\langle n, v\rangle_2$ *collusion secure code employed in the construction.*

Proof. Follows directly from theorem **??** and corollary **??**. □

The efficiency parameters of the scheme are presented in figure **??**.

7.2 Public-key traitor tracing scheme 2

Key Generation. The primes $p_1, \ldots, p_v$ are selected so that $p_\ell = 2q_\ell + 1$ with q_ℓ also prime. Without loss of generality we assume that $l = |p_1| = \ldots = |p_v|$. The public-key of the system is the set to $\langle p_1, f_1, g_1, h_1\rangle, \ldots, \langle p_v, f_v, g_v, h_v\rangle$ where f_ℓ, g_ℓ, h_ℓ are generators of the q_ℓ-order subgroup $\mathcal{G}_\ell$ of $\mathbb{Z}^*_{p_\ell}$, with known relative discrete-logs for the authority.

Let $\vec{d}_{0,\ell} = \langle d_{0,\ell}, d'_{0,\ell}\rangle$ and $\vec{d}_{1,\ell} = \langle d_{1,\ell}, d'_{1,\ell}\rangle$ be two random, linearly independent representations of f_ℓ w.r.t. g_ℓ, h_ℓ, i.e. $f_\ell = g_\ell^{d_{u,\ell}} h_\ell^{d'_{u,\ell}}$ for $u \in \{0,1\}$. User i is given as the decryption key the tuple $\langle \vec{d}_{1,\omega_{i,1}}, \ldots, \vec{d}_{v,\omega_{i,v}}\rangle$,

Encryption. Any third party can encrypt a message $\langle M_1, \ldots, M_v\rangle \in \{0,1\}^{b\cdot v}$ where $b = l-2$, in the following way: $\langle enc(M_1) \cdot f_1^{r_1} \bmod p_1, g_1^{r_1} \bmod p_1, h_1^{r_1} \bmod p_1, \ldots, enc(M_v) \cdot f_v^{r_v} \bmod p_v, g_v^{r_v} \bmod p_v, h_v^{r_v} \bmod p_v\rangle$ where $r_\ell \in_U [q_\ell]$ and $enc(\cdot)$ is the encoding function defined in section **??**.

Decryption. Given a ciphertext $\langle A_1, B_1, C_1, \ldots, A_v, B_v, C_v\rangle$ and a user-key $\langle s\vec{k}_1, \ldots, s\vec{k}_v\rangle$ the decryption is computed as follows

$$\langle enc^{-1}(A_1\langle B_1^{-1}, C_1^{-1}\rangle^{s\vec{k}_1} \bmod p_1), \ldots, enc^{-1}(A_v\langle B_v^{-1}, C_v^{-1}\rangle^{s\vec{k}_v} \bmod p_v)\rangle$$

where $\langle a, b\rangle^{\langle c,d\rangle} =_{\text{df}} a^c b^d$.

Traceability. It is also easy to see that the construction above satisfies the traceability property; in particular as we have shown in theorem **??**(ii) the underlying 2-key public-key encryption scheme is 1-copyrighted in the strong sense under the Discrete-log assumption, thus if a pirate has to construct a pirate-key of the form $\langle sk_1, \ldots, sk_v\rangle \in \mathbb{S}_{l,pk}$ and it is the case that in the ℓ-th coordinate all traitor-keys agree, the pirate will be forced to use that exact key. Based on this it follows easily :

Proposition 31 *The n-key public-key encryption scheme presented above satisfies c-traceability under the Discrete-log Assumption where c is a* $(\mathbb{N} \to \mathbb{N})$ *function such that* $c(n)$ *is the maximum traitor collusion size of the underlying* $\langle n, v\rangle_2$ *collusion secure code employed in the construction.*

Proof. Follows directly from the traceability of the underlying collusion secure codes and theorem **??**(ii). □

Black-Box Traceability. The analyzer procedure is directly derived from theorem **??** and corollary **??**. In particular, the analyzer procedure will be successful in tracing any pirate-box that has success probability in returning the correct plaintext at least $2^{-\mu(l)/v} + \alpha(l)$ where α is a non-negligible function in the security parameter and moreover the plaintext distribution $\mathcal{D} \in \Delta$ is such that it has min-entropy $\mu(l)$ that is evenly spread across the v components of plaintexts (i.e., each of the v components has min-entropy $\mu(l)/v$).

Theorem 32 *The n-key public-key encryption scheme presented above satisfies* $[\Delta, 2^{-\mu(l)/v}, c]$*-semantic-black-box-traceability under the* DDH *Assumption, where* Δ *is the family of plaintext distributions with min-entropy* $\mu(l)$ *as specified above and c is a* $(\mathbb{N} \to \mathbb{N})$ *function such that* $c(n)$ *is the maximum traitor collusion size of the underlying* $\langle n, v\rangle_2$ *collusion secure code employed in the construction.*

	Plaintext Space	Ciphertext Expansion Factor	User-Key Expansion Factor	Public-Key Expansion Factor Factor	Max Traceable Collusion with $(1-\epsilon)$-success
TTS 1	$\{0,1\}^{bv}$	$\frac{2v(b+3)}{bv} \sim 2$	$\frac{v(b+4)}{bv} \sim 1$	$\frac{3v(b+3)}{bv} \sim 3$	$\Omega(\sqrt{\frac{v}{\log(n/\epsilon)}})$
TTS 2	$\{0,1\}^{bv}$	$\frac{3v(b+2)}{bv} \sim 3$	$\frac{2v(b+1)}{bv} \sim 2$	$\frac{4v(b+2)}{bv} \sim 4$	$\Omega(\sqrt{\frac{v}{\log(n/\epsilon)}})$

Figure 5: Efficiency Parameters of the two Traitor Tracing Schemes assuming plaintext calibration, over a $\langle n, v\rangle_2$-collusion secure code of codeword length $v = \mathcal{O}(c^2 \log(n/\epsilon))$, where ϵ denotes the error probability of the tracer and c the maximum traitor collusion size.

Proof. Follows directly from theorem **??** and corollary **??**. The projection assumption is ensured by the fact that the $\mu(l)$ min-entropy of the pirate distribution is evenly spread across the v components that comprise the plaintext random variable. □

The efficiency parameters of the scheme are presented in figure **??**.

7.3 Remarks on Traceability

We note that the projection assumption essentially enforces the pirate-decoder to include at least one key for each of the v-components. For example in the public-key traitor tracing scheme #2, we require that each pirate decoder successfully decodes a coordinate of a ciphertext with probability at least $2^{-\mu/v} + \alpha$ where μ/v is the min-entropy the plaintext distribution in one of the v coordinates (and μ is the overall min-entropy).

Dealing with pirates that violate the projection assumption (for example pirates whose success probability on some of the coordinates drops much sharper than what is required above) can be still made possible by employing collusion secure codes that are resistant against "erasures" or even "shortening" of codewords, i.e., essentially considering codes that can handle more general "marking conditions" (cf. [**?**]).

To deal with the similar issue of why the pirate should include all keys within the pirate decoder it was suggested in [**?**] to employ an all-or-nothing transform (AONT) [**?**]. The employment of the AONT will force the pirate to include one key from each component in order to reach an acceptable success ratio and this will enable the non-black-box traceability argument to be made more compelling. On the other hand, in the black-box setting, the AONT will not prevent the cropping or otherwise tampering of the plaintext once it is decrypted and thus it can be seen not to offer a significant advantage in the black-box setting.

References

[BGI+01] Boaz Barak, Oded Goldreich, Russell Impagliazzo, Steven Rudich, Amit Sahai, Salil P. Vadhan, Ke Yang: On the (Im)possibility of Obfuscating Programs. in Joe Kilian (Ed.): Advances in Cryptology - CRYPTO 2001, 21st Annual International Cryptology Conference, Santa Barbara, California, USA, August 19-23, 2001, Proceedings. Lecture Notes in Computer Science 2139 Springer 2001 pp. 1-18

[Bon98] Dan Boneh, The Decision Diffie-Hellman Problem, In Proceedings of the Third Algorithmic Number Theory Symposium, Lecture Notes in Computer Science, Vol. 1423, Springer-Verlag, pp. 48–63, 1998.

[BF99] Dan Boneh and Matthew Franklin, An Efficient Public-Key Traitor Tracing Scheme, in Michael J. Wiener (Ed.): Advances in Cryptology - CRYPTO '99, 19th Annual International Cryptology Conference, Santa Barbara, California, USA, August 15-19, 1999, Proceedings. Lecture Notes in Computer Science 1666 Springer 1999. pp. 338-353.

[BSW06] Dan Boneh, Amit Sahai and Brent Waters, Fully Collusion Resistant Traitor Tracing with Short Ciphertexts and Private Keys. Serge Vaudenay (Ed.): Advances in Cryptology - EUROCRYPT 2006, 25th Annual International Conference on the Theory and Applications of Cryptographic Techniques, St. Petersburg, Russia, May 28 - June 1, 2006, Proceedings. Lecture Notes in Computer Science 4004, pp. 573-592.

[BS98] Dan Boneh and James Shaw, Collusion-Secure Fingerprinting for Digital Data, IEEE Transactions on Information Theory, Vol. 44(5) pp. 1897-1905, 1998.

[CG00] Dario Catalano and Rosario Gennaro, New Efficient and Secure Protocols for Verifiable Signature Sharing and Other Applications, Journal of Computer and System Sciences Vol. 61(1), pp. 51-80, 2000.

[CPP05] Herv Chabanne, Duong Hieu Phan and David Pointcheval, Public Traceability in Traitor Tracing Schemes, in Ronald Cramer (Ed.): Advances in Cryptology - EUROCRYPT 2005, 24th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Aarhus, Denmark, May 22-26, 2005, Proceedings. Lecture Notes in Computer Science 3494 Springer 2005, pp. 542-558.

[CFN94] Benny Chor, Amos Fiat, and Moni Naor, Tracing Traitors, in Yvo Desmedt (Ed.): Advances in Cryptology - CRYPTO '94, 14th Annual International Cryptology Conference, Santa Barbara, California, USA, August 21-25, 1994, Proceedings. Lecture Notes in Computer Science 839 Springer 1994, pp. 257-270.

[CFNP00] Benny Chor, Amos Fiat, Moni Naor, and Benny Pinkas, Tracing Traitors, IEEE Transactions on Information Theory, Vol. 46, 3, 893-910, 2000.

[DGHKR04] Yevgeniy Dodis, Rosario Gennaro, Johan Hstad, Hugo Krawczyk, Tal Rabin: Randomness Extraction and Key Derivation Using the CBC, Cascade and HMAC Modes. in Matthew K. Franklin (Ed.): Advances in Cryptology - CRYPTO 2004, 24th Annual International Cryptology Conference, Santa Barbara, California, USA, August 15-19, 2004, Proceedings. Lecture Notes in Computer Science 3152 Springer 2004, pp. 494-510.

[ElG85] T. Elgamal, A public key cryptosystem and a signature scheme based on discrete logarithms. IEEE Transactions on Information Theory, Vol. 31(4): pp. 469-472, 1985.

[FT01] Amos Fiat and Tamir Tassa, Dynamic Traitor Tracing. Journal of Cryptology Vol. 4(3), pp. 211-223, 2001.

[FH96] Matthew K. Franklin and Stuart Haber, Joint Encryption and Message-Efficient Secure Computation, Journal of Cryptology 9(4), pp. 217-232, 1996.

[GSY99] Eli Gafni, Jessica Staddon and Yiqun Lisa Yin, Efficient Methods for Integrating Traceability and Broadcast Encryption, in Michael J. Wiener (Ed.): Advances in Cryptology - CRYPTO '99, 19th Annual International Cryptology Conference, Santa Barbara, California, USA, August 15-19, 1999, Proceedings. Lecture Notes in Computer Science 1666 Springer 1999, pp. 372-387.

[GM84] Shafi Goldwasser and Silvio Micali, Probabilistic Encryption, Journal of Computer and System Sciences 28(2): pp. 270-299, 1984.

[KY01a] Aggelos Kiayias and Moti Yung, Self Protecting Pirates and Black-Box Traitor Tracing, Joe Kilian (Ed.): Advances in Cryptology - CRYPTO 2001, 21st Annual International Cryptology Conference, Santa Barbara, California, USA, August 19-23, 2001, Proceedings. Lecture Notes in Computer Science 2139 Springer 2001, pp. 63-79.

[KY01b] Aggelos Kiayias and Moti Yung, On Crafty Pirates and Foxy Tracers, in Tomas Sander (Ed.): Security and Privacy in Digital Rights Management, ACM CCS-8 Workshop DRM 2001, Philadelphia, PA, USA, November 5, 2001, Revised Papers. Lecture Notes in Computer Science 2320 Springer 2002, pp. 22-39.

[KY02] Aggelos Kiayias and Moti Yung, Traitor Tracing with Constant Transmission Rate. In Lars R. Knudsen (Ed.): Advances in Cryptology - EUROCRYPT 2002, International Conference on the Theory and Applications of Cryptographic Techniques, Amsterdam, The Netherlands, April 28 - May 2, 2002, Proceedings. Lecture Notes in Computer Science 2332 Springer 2002, pp. 450-465.

[KY06] Aggelos Kiayias, Moti Yung Secure scalable group signature with dynamic joins and separable authorities International Journal of Security and Networks, Vol. 1(1/2) pp.24 - 45, 2006.

[KD98] K. Kurosawa and Y. Desmedt, Optimum Traitor Tracing and Asymmetric Schemes, in Kaisa Nyberg (Ed.): Advances in Cryptology - EUROCRYPT '98, International Conference on the Theory and Application of Cryptographic Techniques, Espoo, Finland, May 31 - June 4, 1998, Proceeding. Lecture Notes in Computer Science 1403, Springer 1998, pp. 145-157.

[Mil76] G. Miller, Riemann's Hypothesis and Tests for Primality, Journal of Computer and System Sciences, vol. 13, 300–317, 1976.

[NSS99] David Naccache, Adi Shamir, and Julien P. Stern, How to Copyright a Function?, in Hideki Imai, Yuliang Zheng (Eds.): Public Key Cryptography, Second International Workshop on Practice and Theory in Public Key Cryptography, PKC '99, Kamakura, Japan, March 1-3, 1999, Proceedings. Lecture Notes in Computer Science 1560 Springer 1999, pp. 188–196.

[NNL01] Dalit Naor, Moni Naor, and Jeffrey B. Lotspiech Revocation and Tracing Schemes for Stateless Receivers, in Joe Kilian (Ed.): Advances in Cryptology - CRYPTO 2001, 21st Annual International Cryptology Conference, Santa Barbara, California, USA, August 19-23, 2001, Proceedings. Lecture Notes in Computer Science 2139 Springer 2001, pp. 41–62.

[NP98] Moni Naor and Benny Pinkas, Threshold Traitor Tracing, in Hugo Krawczyk (Ed.): Advances in Cryptology - CRYPTO '98, 18th Annual International Cryptology Conference, Santa Barbara, California, USA, August 23-27, 1998, Proceedings. Lecture Notes in Computer Science 1462 Springer 1998, pp. 502-517.

[NP00] Moni Naor and Benny Pinkas, Efficient Trace and Revoke Schemes, in Yair Frankel (Ed.): Financial Cryptography, 4th International Conference, FC 2000 Anguilla, British West Indies, February 20-24, 2000, Proceedings. Lecture Notes in Computer Science 1962 Springer 2001, pp. 1–20.

[NR97] Moni Naor and Omer Reingold, Number-Theoretic Constructions of Efficient Pseudo-Random Functions, 38th Annual Symposium on Foundations of Computer Science, FOCS '97, Miami Beach, Florida, USA, October 19-22, 1997. IEEE Computer Society, pp. 458-467.

[Pfi96] Birgit Pfitzmann, Trials of Traced Traitors, in Ross J. Anderson (Ed.): Information Hiding, First International Workshop, Cambridge, U.K., May 30 - June 1, 1996, Proceedings. Lecture Notes in Computer Science 1174 Springer 1996, pp. 49–63.

[PST06] Duong Phan, Reihaneh Safavi-Naini and Dongvu Tonien, Generic Construction of Hybrid Public Key Traitor Tracing with Full-Public-Traceability, in Michele Bugliesi, Bart Preneel, Vladimiro Sassone, Ingo Wegener (Eds.): Automata, Languages and Programming, 33rd International Colloquium, ICALP 2006, Venice, Italy, July 10-14, 2006, Proceedings, Part II. Lecture Notes in Computer Science 4052, pp. 264-275.

[Riv97] Ron Rivest, All-or-nothing Encryption and the Package Transform, Eli Biham (Ed.): Fast Software Encryption, 4th International Workshop, FSE '97, Haifa, Israel, January 20-22, 1997, Proceedings. Lecture Notes in Computer Science 1267 Springer 1997, pp. 210–218.

[SW00] Reihaneh Safavi-Naini and Yejing Wang, Sequential Traitor Tracing, in Mihir Bellare (Ed.): Advances in Cryptology - CRYPTO 2000, 20th Annual International Cryptology Conference, Santa Barbara, California, USA, August 20-24, 2000, Proceedings. Lecture Notes in Computer Science 1880 Springer 2000. pp. 316-332.

[SW01a] Reihaneh Safavi-Naini and Yejing Wang, Collusion Secure q-ary Fingerprinting for Perceptual Content, in Tomas Sander (Ed.): Security and Privacy in Digital Rights Management, ACM CCS-8 Workshop DRM 2001, Philadelphia, PA, USA, November 5, 2001, Revised Papers. Lecture Notes in Computer Science 2320 Springer 2002, pp. 57–75.

[SW01b] Reihaneh Safavi-Naini and Yejing Wang, New Results on Frameproof Codes and Traceability Schemes, IEEE Transactions on Information Theory, Vol. 47(7), pp. 3029-3033, 2001.

[SW02] Reihaneh Safavi-Naini and Yejing Wang, Traitor Tracing for Shortened and Corrupted Fingerprints. In Joan Feigenbaum (Ed.): Security and Privacy in Digital Rights Management, ACM CCS-9 Workshop, DRM 2002, Washington, DC, USA, November 18, 2002, Revised Papers. Lecture Notes in Computer Science 2696 Springer 2003, pp. 81-100.

[SSW00] Jessica N. Staddon, Douglas R. Stinson and Ruizhong Wei, Combinatorial Properties of Frameproof and Traceability Codes, IEEE Transactions on Information Theory, Vol. 47(3), pp. 1042-1049, 2001.

[SW98] Douglas R. Stinson and Ruizhong Wei, Combinatorial Properties and Constructions of Traceability Schemes and Frameproof Codes, SIAM Journal on Discrete Math, Vol. 11(1), pp. 41–53, 1998.

[Tar03] Gábor Tardos, Optimal probabilistic fingerprint codes, in Proceedings of the 35th Annual ACM Symposium on Theory of Computing, June 9-11, 2003, San Diego, CA, USA. ACM 2003, pp. 116-125.

[TY99] Yiannis Tsiounis and Moti Yung, On the Security of ElGamal Based Encryption, in Hideki Imai, Yuliang Zheng (Eds.): Public Key Cryptography, First International Workshop on Practice and Theory in Public Key Cryptography, PKC '98, Pacifico Yokohama, Japan, February 5-6, 1998, Proceedings. Lecture Notes in Computer Science 1431, pp. 117-134.