Traitor Tracing with Constant Transmission Rate

Aggelos Kiayias[1] and Moti Yung[2]

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

Abstract. An important open problem in the area of Traitor Tracing is designing a scheme with constant expansion of the size of keys (users' keys and the encryption key) and of the size of ciphertexts with respect to the size of the plaintext. This problem is known from the introduction of Traitor Tracing by Chor, Fiat and Naor. We refer to such schemes as traitor tracing with constant transmission rate. Here we present a general methodology and two protocol constructions that result in the first two public-key traitor tracing schemes with constant transmission rate in settings where plaintexts can be calibrated to be sufficiently large. Our starting point is the notion of "copyrighted function" which was presented by Naccache, Shamir and Stern. We first solve the open problem of discrete-log-based and public-key-based "copyrighted function." Then, we observe the simple yet crucial relation between (public-key) copyrighted encryption and (public-key) traitor tracing, which we exploit by introducing a generic design paradigm for designing constant transmission rate traitor tracing schemes based on copyrighted encryption functions. Our first scheme achieves the same expansion efficiency as regular ElGamal encryption. The second scheme introduces only a slightly larger (constant) overhead, however, it additionally achieves efficient black-box traitor tracing (against any pirate construction).

1 Introduction.

Distributing data securely to a set of subscribers is an important problem in cryptography with a variety of practical applications. A direct solution to this problem is to give to each subscriber a common secret-key. Nevertheless, such a solution is not satisfactory: this enables a subscriber to distribute its secret-key to other parties thus enabling illegal data reception. This situation is identified as piracy in the context of digital content distribution. Preventing piracy via tamper–proof devices is uneconomical and not applicable in many scenarios; additionally software obfuscation has not produced cryptographically strong results that would adequately protect the common secret-key. In light of this, the notion of "traitor-tracing" that originated in [CFN94] suggests a solution to piracy when it is assumed that subscribers' decoders are open and therefore the secret-keys are accessible. In a traitor-tracing scheme (TTS) each user possesses a different secret-key that allows the reception of the data in a non-ambiguous fashion. The scheme discourages piracy as follows: given a pirate decoder the scheme allows the distributor to recover the identities of some subscribers that collaborated in its construction (henceforth called traitors).

From the time of the primitive's introduction in [CFN94] a series of works [Pfi96,SW98,NP98,KD98,BF99,FT99][GSY99,NP00,SW00,NNL01,KY01b] proposed more efficient/ robust schemes or schemes with advanced capabilities such as revocation and non-repudiation. Two extremely desirable properties of a traitor tracing scheme are (i) Public-key Traitor Tracing where any third party (e.g., any of a number of pay-T.V. stations) is able to send secure messages to the set of subscribers, (ii) Black-Box Traitor Tracing which suggests that the tracing procedure can be accomplished with merely black-box access to the pirate-decoder (something that allows less costly, even remote access tracing).

Traitor Tracing has also its shortcomings: the size of the ciphertexts and keys used by traitor-tracing schemes depends on quantitites such as the number of users and/or the maximum traitor collusion that is expected. Even though progress has been made from the initial scheme of [CFN94] 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" size, is constant (where the "rate" of a parameter expresses the ratio of the its size over the size of the plaintexts – which is 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 TTS. Minimizing the transmission rate has been, in fact, open since [CFN94] (the issue was reiterated in some stronger form also in [BF99]).

In this work we present the first two constant transmission rate Traitor-Tracing Schemes (for settings where plaintexts can be calibrated to be large enough), thus answering the question that postulated the existence of such schemes in the affirmative. Our results, in comparison to previous schemes, are presented in figure 1 (where the ElGamal scheme which does not provide traitor tracing is given for comparison).

Our methodology starts by investigating the notion of "copyrighted function" proposed by Naccache, Shamir and Stern in [NSS99]. In the copyrighted function setting each member of a set of users possesses a different implementation of a function with the same functionality though, so that an authority is capable of exposing the responsible user(s) if an implementation is used illegally. The techniques presented in [NSS99] applied to one-way (hash) functions and to symmetric encryption (and were implemented based on the RSA function used as a private key function). It was left as an open problem whether it is possible to achieve function copyright based on the Discrete-Logarithm Problem. Here we answer this question in the affirmative. Moreover, we reformulate the setting

	Ciphertext Rate	User-Key Rate	Encryption-Key Rate	Max Traceable Collusion	Black-Box Traitor Tracing
[ElGamal84]	~ 2	~ 1	~ 3	$\times$	$\times$
[CFN94] (Sch.1)	$\mathcal{O}(t^4 \log n)$	$\mathcal{O}(t^2 \log n)$	$\mathcal{O}(t^2 \log n)$	t	$\surd$
[BF99]	$\sim (2t+1)$	$\sim 2t$	$\sim (2t+1)$	t	inefficient (see [KY01a])
TTS Scheme 1	~ 2	~ 1	~ 3	$\Omega(l^c) = t$	?
TTS Scheme 2	~ 3	~ 2	~ 4	$\Omega(l^c) = t$	$\surd$

Fig. 1. A Comparison of our traitor-tracing schemes with previous work. Note that l is a security parameter and c is a constant < 1; the plaintext space in all cases is considered to be $\{0,1\}^l$. The rate of keys/ciphertexts is defined w.r.t. the size of the plaintexts: l.

of [NSS99] in the public-key encryption setting and we then present a simple yet crucial step of our methodology, showing that the goal of copyrighting public-key encryption is equivalent to the construction of a public-key traitor tracing scheme.

The reformulation of the [NSS99]-setting in the public-key context, along with our novel design paradigm for the construction of traitor-tracing schemes based on copyrighted encryption functions and the two concrete copyrighted public-key functions we present, allow us to construct two public-key traitor tracing schemes, which are the first that have constant transmission rate. In our schemes, 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). Our size adjustment method employs collusion-secure codes [BS95], and we further found that in order to support the adjustment while retaining the traitor tracing capability with constant rate, it is crucial to employ the All-or-Nothing-Transform (AONT) [Riv97] prior to encryption (or alternatively employ a threshold assumption similar to the one used in [NP98]). Our first scheme is as efficient, rate-wise, as ElGamal encryption, whereas our second scheme uses slightly extended ciphertexts and keys (by a constant additive factor). However, the second scheme achieves also efficient black-box traitor tracing against any pirate-construction (as in the black-box traitor tracing model of [CFN94]) and not just a single-key traitor as in [BF99]. The scheme relies on ElGamal-like encryption as the [BF99]-scheme and is the first traitor tracing scheme beyond [CFN94] and its variants to achieve black-box traceability.

We note that, interestingly, our schemes employ and combine in a unique way results from all the major contributions in the area: traceability codes introduced in the context of traitor tracing by Chor, Fiat and Naor [CFN94], collusion secure codes defined by Boneh and Shaw [BS95] (in the context of fingerprinting) and the public-key traitor tracing concepts of Kurosawa and Desmedt [KD98], and Boneh and Franklin [BF99].

The intractability assumptions used for the first scheme are the DDH Assumption over the quadratic-residues group modulo a composite and the Quadratic Residuosity assumption, whereas the security of the second scheme is based on DDH-Assumption over a prime order subgroup.

The scope of the design paradigm we propose for the construction of traitor tracing schemes based on copyrighted encryption functions goes beyond the two public-key traitor tracing schemes we propose. It can be readily applied to any basic 2-user copyrighted encryption mechanism yielding a traitor tracing scheme with the same transmission rate as the underlying basic copyrighted encryption function.

2 Preliminaries

Notations. 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 l will denote a security parameter (typically, the plaintext size); all sets of objects that we consider are exponential in size w.r.t. l and contain objects of size polynomial in l. All procedures are polynomial-time in l. If K is a set of objects and f is a procedure that samples an element of K, denote by $k \hookleftarrow_f K$ such element; note that we may occasionally omit f or K from this notation if this is allowed in the context. If K is a set of objects of the same size, let $\mathrm{len}[k \in K]$ denote the size of the objects in K. As stated above $\mathrm{len}[k \in K]$ is polynomial in the security parameter and perhaps it may depend on other factors as well. We use $|x|$ to denote the size of an object x, e.g. $|x| = \lceil \log_2 x \rceil$ 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 if $\lim_{b,v\to\infty} f(b, v) = c$. The notation $a \in_U R$ stands for "a is sampled from R following the uniform distribution."

Next we define the notion of public-key encryption scheme (note that the definition is tailored to our setting).

Definition 1. *A public-key encryption scheme is a tuple* $\langle \mathbb{P}, \mathbb{C}, \mathcal{P}, \cup_{pk\in\mathcal{P}}\mathcal{K}_{pk}, G, E, D\rangle$ *so that*
1. $\mathbb{P}$ *and* $\mathbb{C}$ *are the plaintext-space and ciphertext-space respectively. Without loss of generality we assume that the objects in these sets are of the same size.*
2. Key Generation. It holds that: $\langle pk, \kappa\rangle \hookleftarrow_G (\mathcal{P} \times \cup_{pk\in\mathcal{P}}\mathcal{K}_{pk})$ *so that* $\kappa \in \mathcal{K}_{pk}$.
3. Encryption. $E : (\mathcal{P} \times \mathbb{P}) \to \mathbb{C}$ *is a probabilistic poly-time procedure.*
4. Decryption. $D : (\cup_{pk\in\mathcal{P}}\mathcal{K}_{pk} \times \mathbb{C}) \to \mathbb{P}$ *is a deterministic procedure so that* $D(\kappa, E(pk, m)) = m$ *for all* $m \in \mathbb{P}$ *and* $\langle pk, \kappa\rangle \hookleftarrow_G$.
5. Semantic Security (i.e. polynomial indistinguishability): for some $\langle pk, \kappa\rangle \hookleftarrow_G$, *any adversary that given* pk *generates* $m_1, m_2 \in \mathbb{P}$, *when given* $E(m_x)$ *with* $x \in_U \{1, 2\}$ *it can predict* x *with negligible advantage. Note that the definition can be extended to include stronger notions of security such as chosen-ciphertext security or non-malleability.*

2.1 Intractability Assumptions

The security of our schemes is based on the hardness of the Decisional Diffie Hellman (DDH) Problem over a multiplicative cyclic group $\mathbb{G} = \langle g \rangle$:

Definition 2. Decision Diffie Hellman Assumption. *Let $\mathcal{V}$ be the distribution $\{\langle g, g^x, g^y, g^{xy} \rangle \mid x, y < |\mathbb{G}|\}$, and $\mathcal{R}$ be the distribution $\{\langle g, g^x, g^y, g^z \rangle \mid x, y, z < |\mathbb{G}|\}$. The DDH assumption over $\mathbb{G} = \langle g \rangle$ states that any poly-time distinguisher D for the two distributions $\mathcal{V}, \mathcal{R}$ has negligible success probability, i.e. $|\mathbf{Prob}_{X \in \mathcal{V}}[D(X) = 1] - \mathbf{Prob}_{X \in \mathcal{R}}[D(X) = 1]|$ is negligible in $\log |\mathbb{G}|$.*

The DDH assumption has been used in a variety of settings and over many different groups; for an overview and applications the reader is referred to [NR97] [Bon98]. The DDH assumption over a group of prime order is known to be equivalent to the security of ElGamal encryption, see [TY99]. We note here that ElGamal-like encryption with composite modulus has also been used extensively, e.g. [FH96,CG98].

Here we use the DDH assumption (i) over the cyclic subgroup $\mathcal{G}$ of quadratic residues of $\mathbb{Z}_p^*$ of order q, where $p = 2q + 1$ and both p, q are primes; (ii) over the cyclic subgroup $\mathbb{Q}_N$ of quadratic residues of $\mathbb{Z}_N^*$ where $N = pq$ and $p = 2p' + 1, q = 2q' + 1$ with p, q, p', q' all primes. It is believed that DDH over the subgroup of quadratic residues modulo p or modulo N is hard (see [Bon98]). We also utilize the Quadratic Residuosity (QR) Assumption [GM84]:

Definition 3. Quadratic Residuosity Assumption. *If $N = pq$, so that $p = 2p' + 1, q = 2q' + 1$ with p, q, p', q' all primes, any probabilistic algorithm that given $x \in \mathbb{J}_N$ (Jacobi +1 elements) it decides whether $x \in \mathbb{Q}_N$ or $x \in \mathbb{J}_N - \mathbb{Q}_N$, has success probability $1/2 + \epsilon$ where ϵ is negligible in $\log N$.*

2.2 Public-Key Traitor Tracing Schemes

A public-key traitor tracing scheme involves a key-generation (setup) algorithm G, and the corresponding encryption/decryption function as in the public-key encryption setting: an authority uses G to generate $\langle pk, d_1, \ldots, d_n \rangle$ so that each of the d_i "inverts" the public-key pk. Subsequently it publishes pk and privately communicates the key d_i to each user i. From then on users are capable of decrypting messages encrypted using the public-key pk. If a pirate uses t keys given by some users (the traitors) to construct another key for the purpose of implementing an illegal receiver, the authority is able to recover the identity of one of the traitor users given the pirate-key (a procedure called traitor-tracing). Formally,

Definition 4. *An n-user (public-key) traitor-tracing scheme (TTS) is a tuple $\langle \mathbb{P}, \mathbb{C}, \mathcal{P}, \cup_{pk \in \mathcal{P}} (\mathcal{K}_{pk})^n, G, E, D \rangle$ that*
1. Satisfies properties 1,2,3,5 of definition 1.
2. If $\langle pk, d_1, \ldots, d_n \rangle \hookleftarrow_G$ then $D : \cup_{pk \in \mathcal{P}} \mathcal{K}_{pk} \times \mathbb{C} \to \mathbb{P}$ is a deterministic procedure so that $D(d, E(pk, m)) = m$ for all $m \in \mathbb{P}$ and $d \in \{d_1, \ldots, d_n\}$.

3. Tracing. Let $\langle pk, d_1, \ldots, d_n\rangle \hookleftarrow_G$. There is a procedure $\mathcal{T}$ so that: for any adversary $\mathcal{A}$ that given pk and $\{d_{i_1}, \ldots, d_{i_t}\}$ with $t \leq c$, $\mathcal{A}$ generates some $d \in \mathcal{K}_{pk}$ so that $D(d, E(pk, m)) = m$ for most $m \in \mathbb{P}$, $\mathcal{T}$ given d is capable of recovering at least one of the indices i_ℓ.
The parameter c is maximum collusion size allowed by the traitor tracing scheme. We will call such a scheme: an n-user,c-TTS.

Of course, the pirate may not use directly a certain decryption key d, but instead construct a simulator for the decryption operation that is hard to reverse-engineer and extract its contents. Therefore, it is important for a TTS to allow *black-box traitor tracing*:

3′. Black-Box Tracing. Let $\langle pk, d_1, \ldots, d_n\rangle \hookleftarrow_G$. There is a procedure $\mathcal{T}$ so that for any adversary that given pk and $\{d_{i_1}, \ldots, d_{i_t}\}$ with $t \leq c$ it generates a decryption simulator S so that $S(E(pk, m)) = m$ for almost all $m \in \mathbb{P}$, then $\mathcal{T}$ given oracle access to S is capable of recovering at least one of the indices i_ℓ.

Definition 5. *A n-user,c-TTS with black-box traceability is defined as in definition 4, with item* $3'$ *substituting item* 3.

Definition 6. Efficiency Parameters. *The three basic efficiency parameters of traitor tracing schemes are (i) the ciphertext rate* $\frac{\mathrm{len}[c\in\mathbb{C}]}{\mathrm{len}[m\in\mathbb{P}]}$, *(ii) the user-key rate* $\frac{\mathrm{len}[d\in\mathcal{K}_{pk}]}{\mathrm{len}[m\in\mathbb{P}]}$, *and (iii) the encryption-key rate* $\frac{\mathrm{len}[pk\in\mathcal{P}]}{\mathrm{len}[m\in\mathbb{P}]}$. *The transmission rate of the scheme is defined as the sum of the three rates.*

3 Copyrighting a Function

Nacacche, Shamir and Stern [NSS99] 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 $\mathcal{G}_\mathcal{K}$. The following definition is from [NSS99], slightly amended:

Definition 7. *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 $\mathcal{K}$.*
(ii) 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 κ_0 of $\mathcal{K}$; then, $\mathcal{T}$ given κ_0 is able to reconstruct at least one of the c elements that were given to the adversary.*
(iii) 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 $\mathcal{S}$ is capable of recovering at least one of the c elements that were given to the adversary.*

In [NSS99] a method was presented that allowed copyrighting a hash function based on RSA-encryption. The basic design paradigm of [NSS99] solved the two-user case first and then the multi-user case was addressed by employing collusion-secure codes [BS95]. Although copyrighting a hash function allows a variety of applications, much greater flexibility is allowed by a method for copyrighting a public-key encryption function. (Note that in [NSS99] a method to copyright the RSA-encryption function was given, but only as a symmetric-encryption function, since no public-components were allowed). In [NSS99] it was left as an open question whether it is possible to achieve a copyright mechanism based on the Discrete-Logarithm Problem. Here we answer this question in the affirmative. Another important question that arises from the work of [NSS99] (who show how to copyright symmetric encryption) is whether it is possible to copyright a public-key encryption function. Next we formalize this notion.

3.1 Copyrighting a Public-Key Function

Definition 8. *A n-key,c-copyrighted Public-Key Encryption Scheme against passive (resp. active) adversaries is a tuple* $\langle \mathbb{P}, \mathbb{C}, \mathcal{P}, \cup_{pk \in \mathcal{P}} \mathcal{K}_{pk}, G_n, E, D \rangle$ *so that*
(i) $\langle pk, d_1, \ldots, d_n \rangle \hookleftarrow_{G_n} \mathcal{P} \times \mathcal{K}_{pk}$ *where* $d_1, \ldots, d_n \in \mathcal{K}_{pk}$.
(ii) $\langle \mathbb{P}, \mathbb{C}, \mathcal{P}, \cup_{pk \in \mathcal{P}} \mathcal{K}_{pk}, G_1, E, D \rangle$ *is a public-key encryption scheme.*
(iii) for any $pk \in \mathcal{P}$, $\{D(\kappa, \cdot) : \mathbb{C} \to \mathbb{P}\}_{\kappa \in \mathcal{K}_{pk}}$ *is c-copyrighted against passive (resp. active) adversaries.*

In the following simple but crucial Lemma we establish the relationship between the above generalization of the [NSS99]-setting and (public-key) traitor tracing:

Lemma 1. *An n-key,c-copyrighted public-key encryption scheme against passive (resp. active) adversaries is equivalent to an n-user,c-TTS (resp. n-user,c-TTS with black-box traceability).*

The Lemma provides a construction methodology for public-key traitor tracing schemes: given an n-key,c-copyrighted public-key encryption scheme the corresponding public-key traitor tracing scheme is the following: the authority uses G to generate a tuple $\langle pk, d_1, \ldots, d_n \rangle$. The key pk is published as the public-key and the decryption-key d_ℓ is given to user ℓ. Any third party can use the encryption algorithm E in combination to pk and send encrypted data to the users that possess the decryption-keys. Traitor tracing is taken care of by the copyright properties of the decryption function: if the security is against passive adversaries, the authority can perform non-black-box traitor tracing. If the copyright security of the decryption function is against active adversaries, the authority can use the analyzer to perform black-box traitor tracing.

4 The Basic Building Block: The Two-User Case

We will consider two alternative settings for copyrighting a public-key encryption function. Following the [NSS99] design paradigm we will consider the 2-

key,1-copyrighted case first. In the following sections we present two 2-key,1-copyrighted public-key encryption schemes. Scheme 1 is more efficient, however scheme 2 allows security against active adversaries.

4.1 Scheme 1

Let $N = pq$ where p, q are two primes so that $p = 2p' + 1, q = 2q' + 1$ with p', q' also prime. The factorization of N is kept secret by the authority. Let $h \in \mathbb{Z}_N^*$ with maximal order, i.e. $\mathrm{ord}(h) = \lambda(N) = 2p'q'$ where $\lambda(N)$ is the Carmichael function, so that $\langle h \rangle = \mathbb{J}_N$ where $\mathbb{J}_N$ is the subgroup of $\mathbb{Z}_N^*$ that contains all elements with Jacobi Symbol $+1$. The element h can be computed easily given the factorization of N as follows: select h_1, h_2 to be generators of the multiplicative groups $\mathbb{Z}_p^*$ and $\mathbb{Z}_q^*$ respectively and compute h by solving the system $h = h_1 \bmod p$ and $h = h_2 \bmod q$ (solvable by the Chinese Remainder Theorem). It follows easily that h_1, h_2 are both quadratic non-residues modulo p, q respectively and as a result $\langle h \rangle = \mathbb{J}_N$. Now let $g =_{\text{df}} h^2 \bmod N$. It is easy to verify that $\langle g \rangle = \mathbb{Q}_N$ (the group of quadratic residues modulo N). Note that $|\mathbb{Q}_n| = p'q'$.

The tuple $\langle N, g, y =_{\text{df}} g^\alpha \bmod N \rangle$ is the public-key of the system, where $\alpha \in_U [p'q']$ with $(\alpha, p'q') = 1$. The set of possible decryption keys is $\mathcal{K}_{pk} =_{\text{df}} \{x \in \mathbb{N} \mid x = \alpha (\bmod \phi(N))\}$. The two users are assigned the two (shorter) secret keys of $\mathcal{K}_{pk}$, $\alpha_0 =_{\text{df}} \alpha$ and $\alpha_1 =_{\text{df}} \alpha + \phi(N)$ of $\mathcal{K}_{pk}$. It is immediate that: $g^{\alpha_0} \bmod N = g^{\alpha_1} \bmod N = y$.

Encryption is performed following the ElGamal paradigm ([ElG84]): given a message $M \in \mathbb{J}_N$, the sender computes the tuple $\langle g^k \bmod N, y^k \cdot M \bmod N \rangle$ where $k \in_U [N]$. Note that $\langle g \rangle = \mathbb{Q}_N$ is a group of unknown order for the sender. The decryption procedure is as follows: given $\kappa \in \mathcal{K}_{pk}$, and a ciphertext $\langle A, B \rangle$, the receiver computes the plaintext as follows: $B \cdot (A^{-1})^\kappa \bmod N$. It is easy to verify that the decryption operation inverts encryption. The following lemma shows that the choice of the encryption exponent k from $[N]$ is appropriate:

Lemma 2. *The uniform distribution over $\langle g \rangle$ is statistically indistinguishable from the distribution $\mathcal{D}$ induced over $\langle g \rangle$ by the mapping $k \to g^k \bmod N$ where $k \in_U [N]$.*

Theorem 1. *The public-key encryption function described above is*
(i) Semantically Secure under the DDH Assumption over $\mathbb{Q}_N$ and the QR Assumption in $\mathbb{Q}_N$.
(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 $\mathcal{K}_{pk}$ under the assumption that factoring N is hard.

Note that the scheme is strictly 1-copyrighted and not 2-copyrighted since if the two users collude it is immediate that they can construct keys in $\mathcal{K}_{pk}$ as follows: given $\alpha_0, \alpha_1 \in \mathcal{K}_{pk}$ it follows that $\alpha_1 - \alpha_0$ equals $\phi(N)$. Subsequently any $\alpha_0 + x(\alpha_1 - \alpha_0)$, where $x \in \mathbb{N}$, is an element of $\mathcal{K}_{pk}$.

Plaintext-Space and Efficiency Parameters. In order to measure efficiency, first we have to specify the plaintext-space: let the plaintext-space for the encryption operation be $\{0,1\}^b$ with $b = |N| - 3$. We have to determine an encoding function $enc : \{0,1\}^b \rightarrow \mathbb{J}_N$ that is easily invertible. Given a message $M =_{\mathrm{df}} m_1 m_2 \ldots m_b \in \{0,1\}^b$ let $M' =_{\mathrm{df}} m_1 + 2m_2 + \ldots 2^{b-1} m_b + \frac{N}{4} + 1$. It is easy to see that $\frac{N}{4} < M' < \frac{N}{2}$. Suppose now that $p' = 1(\mathrm{mod} 4)$ and $q' = 3(\mathrm{mod} 4)$. Then, it holds that $(\frac{2}{N}) = -1$ (recall that $N = (2p'+1)(2q'+1)$). Now if $(\frac{M'}{N}) = 1$ the encoding of M is M', else if $(\frac{M'}{N}) = -1$ then the encoding of M is defined as $2 \cdot M'$. This completes the description of *enc*.

The encoding function can be inverted as follows: given $enc(M)$ we compute M' so that $M' =_{\mathrm{df}} enc(M)$ if $enc(M) < N/2$, or $M' =_{\mathrm{df}} enc(M)/2$ if $enc(M) > N/2$. The decoding of $enc(M)$ is the binary representation of $M' - \frac{N}{4} - 1$. The rates of the parameters of the system are illustrated in the figure 2 (recall that $|N| = b + 3$).

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

Fig. 2. Efficiency Parameters of Scheme 1 (Two-User Setting)

4.2 Scheme 2

Let $\mathcal{G}$ be the group of quadratic residues modulo $p = 2q + 1$ where both p, q, are large 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 $pk =_{\mathrm{df}} \langle p, f, g, h \rangle$ where $f =_{\mathrm{df}} g^{\alpha}, h =_{\mathrm{df}} g^{\beta}$ and $\alpha, \beta \in_R [q]$. The two users are given two "representations" of α with respect to the "base" g, h, i.e. the authority selects 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 $\mathcal{K}_{pk} =_{\mathrm{df}} \{\langle d, d' \rangle \mid d + d'\beta = \alpha(\mathrm{mod} q)\}$.

Encryption is performed as follows: given the public-key $\langle f, g, h \rangle$ and a message $M \in \mathcal{G}$, the encryption of M is $\langle M \cdot f^r \bmod p, g^r \bmod p, h^r \bmod p \rangle$. 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 $A(B^{-1})^{d_i}(C^{-1})^{d_i'} \bmod p$. It is easy to verify that the decryption operation inverts encryption.

Theorem 2. *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 $\langle d, d' \rangle \in \mathcal{K}_{pk}$ it is computationally infeasible to construct another key in $\mathcal{K}_{pk}$ under the Discrete-Log assumption over $\mathcal{G}$.

Note that the scheme is strictly 1-copyrighted and not 2-copyrighted since if the two users collude, they can construct keys in $\mathcal{K}_{pk}$ as follows: given $\langle d_0, d_0' \rangle$ and $\langle d_1, d_1' \rangle$ it holds that $\langle rd_0 + (1-r)d_1, rd_0' + (1-r)d_1' \rangle \in \mathcal{K}_{pk}$ for any $r \in \mathbb{Z}_q$.

Plaintext-Space and Efficiency Parameters. First we specify the plaintext-space: let the plaintext-space for the encryption-operation be $\{0,1\}^b$ with $b = |p| - 2$. We have to determine an easily invertible encoding function $enc : \{0,1\}^b \rightarrow \mathcal{G}$. Given $M = m_1 \dots m_b \in \{0,1\}^b$ let $M' = m_1 + 2m_2 + \dots 2^{b-1}m_b + 1$. It is easy to verify that $M' \in \{1, \dots, q\}$. Then, $enc(M) =_{\mathrm{df}} (M')^2 \bmod p$. 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 $\{1, \dots, q\}$. 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 3 (recall that $|p| = b + 2$).

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

Fig. 3. Efficiency Parameters of Scheme 2 (Two-User Setting)

4.3 Scheme 2: Security Against Active Adversaries

In this section we establish that scheme 2 is secure against active adversaries.

Theorem 3. *Suppose that there is an adversary $\mathcal{A}$ that:*
(i) Given the public-key information, $\mathcal{A}$ produces a decryption simulator $\mathcal{S}$ that decrypts valid ciphertexts with probability ϵ. Then the Diffie-Hellman Problem is solvable with probability ϵ.
(ii) Given the public-key information pk and a key $\langle d, d' \rangle \in \mathcal{K}_{pk}$, $\mathcal{A}$ produces a simulator $\mathcal{S}$ that decrypts all valid ciphertexts but when given a "randomized" ciphertext of the form $\langle A, g^{r_0}, g^{\alpha r_1} \rangle$ with $r_0, r_1 \in_U [q]$, it outputs a value different than $A/g^{r_0 d + \alpha r_1 d'}$ with probability ϵ. Then the Decision-Diffie-Hellman Problem is decidable with probability ϵ.

Let us now present an analyzer $\mathcal{N}$ that given black-box access to a decryption simulator $\mathcal{S}$ constructed by one of the two users it decides which of the two constructed it:
Description of the Analyzer $\mathcal{N}$: given black-box access to a decryption simulator $\mathcal{S}$, $\mathcal{N}$ selects $a_0, a_1 \in_U \mathbb{Z}_q$ and solves the system $d_0 x + \alpha d_0' y = a_0$ and $d_0 x + \alpha d_1' y = a_1$ (note that the system is solvable because of the choice of $\langle d_0, d_0' \rangle, \langle d_1, d_1' \rangle$). Then, $\mathcal{N}$ submits to $\mathcal{S}$ the "randomized" ciphertext $\langle A, g^x,$

$(g^{\alpha})^y\rangle$. If the output of $\mathcal{S}$ is A/g^{a_0} then $\mathcal{N}$ outputs 0, otherwise, if the simulator's output is A/g^{a_1}, $\mathcal{N}$ outputs 1; finally $\mathcal{N}$ outputs ? if the output of the simulator is not contained in $\{A/g^{a_0}, A/g^{a_1}\}$.

The correctness of $\mathcal{N}$ is guaranteed by theorems 2 and 3. In particular theorem 2(ii) suggests that user 1 cannot incriminate user 2 or use some other key in $\mathcal{K}_{pk}$; additionally theorem 3(i) suggests that at least one representation should be used by the simulator $\mathcal{S}$; finally theorem 3(ii) suggests that the "randomized" ciphertext used by the analyzer $\mathcal{N}$ cannot be distinguished from regular ciphertexts. Due to theorem 3(ii) the simulator $\mathcal{N}$ will output ? only in the case that both users colluded in the construction of $\mathcal{S}$. This leads to the corollary:

Corollary 1. *Scheme 2 is 1-copyrighted against active adversaries.*

Remark. Scheme 2 can be viewed as a special case of the public-key traitor tracing scheme of [BF99] (for two users). However, the approach we take in extending scheme 2 to capture the multi-user case is different from [BF99].

5 The Multi-User Case

Let $\langle \mathbb{P}, \mathbb{C}, \mathcal{P}, \cup_{pk\in\mathcal{P}}\mathcal{K}_{pk}, G_2, E, D\rangle$ be a 2-key,1-copyrighted (in the strong sense) public-key encryption scheme. In this section, following the [NSS99] design paradigm we compose the two-user case with collusion secure codes. Specifically, we show how to obtain an n-key,c-copyrighted public-key encryption scheme (and thus, by Lemma 1, a public-key traitor tracing scheme) by a parallel combination of independent instantiations of a 2-key,1-copyrighted public-key encryption scheme based on collusion-secure codes. Note that for designing one-way (hash) functions, [NSS99] used nested composition rather than parallel. The parallel approach we choose is crucial for maintaining constant transmission rate.

Key-Generation. Let $\mathcal{C} =_{\rm df} \{\omega_1, \ldots, \omega_n\}$ 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 $1-\epsilon$; collusion secure codes were introduced in [BS95], and further investigated in [SSW00,SW01a,SW01b]. The key-generation procedure, first generates v independent key-instantiations of a 2-user,1-copyrighted scheme:

$$\{\langle pk_i, \kappa_{0,i}, \kappa_{1,i}, E_i, D_i\rangle\}_{i=1}^{v}$$

Without loss of generality we assume that the plaintext-space $\mathbb{P}$ over all instantiations is the same ($=\{0,1\}^b$) and that $\mathrm{len}[c \in \mathbb{C}_1] = \ldots = \mathrm{len}[c \in \mathbb{C}_1]$. The i-th decryption key of the n-key system is defined as the following sequence $\boldsymbol{\kappa_i} =_{\rm df} \langle \kappa_{i,\omega_{i,1}}, \ldots, \kappa_{i,\omega_{i,v}}\rangle$ 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 $\langle E_1(pk_1, M_1), \ldots, E_v(pk_v, M_v)\rangle$. Because each user has one key that inverts $E_i(pk_i, \cdot)$ (either $\kappa_{0,i}$ or $\kappa_{1,i}$) for all $i = 1, \ldots, v$ it is possible for any user to invert a ciphertext and compute $\langle M_1, \ldots, M_v\rangle$.

Security Against Passive Adversaries. Suppose $\langle \kappa_1^*, \ldots, \kappa_v^* \rangle$ is a key that was constructed by a coalition of t users s.t. $t \leq c$. Subsequently the tracer constructs a codeword $\omega^* =_{\rm df} \omega_1^* || \ldots || \omega_v^*$ as follows

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

Because of the fact that each key-instantiation is 1-copyrighted against passive adversaries in the strong sense, if $C =_{\rm df} \{\omega_{i_1}, \ldots, \omega_{i_t}\}$ is the set of codewords that corresponds to the keys of the coalition of traitor users that constructed $\langle \kappa_1^*, \ldots, \kappa_v^* \rangle$, it holds that $\omega^* \in F(C)$, where $F(C)$ is the *feasible* set of the codewords C (see [BS95]); 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 with probability $1 - \epsilon$. Note that we assume that a key for all v instantiations is necessary, i.e. partial decryptions of a ciphertext are not useful. We deal with how this can be enforced in more details in section 6 where we describe the two public-key traitor tracing schemes based on this construction.

Security against Active Adversaries. If the underlying 2-key,1-copyrighted public-key encryption scheme is secure against active adversaries then the tracer can construct the codeword ω^* using merely black-box access to the pirate decoder: the tracer constructs a "randomized" ciphertext $\langle C_1, \ldots, C_v \rangle$ where C_i is constructed as dictated by the analyzer procedure $\mathcal{N}$ in the i-th instantiation of the 1-copyrighted public-key scheme. The value ω_i^* is set to be the output of the analyzer for the i-th coordinate (recall that the output of $\mathcal{N}$ is in $\{0, 1, ?\}$). Note that black-box traitor tracing is achieved with merely a single query to the pirate-decoder (plus the time needed for the collusion-secure code's tracer algorithm).

Theorem 4. *Given v-instantiations of a 2-key,1-copyrighted public-key encryption scheme secure against passive (resp. active) adversaries and a $\langle n, v \rangle_2$-collusion secure code secure that allows collusions of up to c, the scheme described above is a n-key,c-copyrighted public-key encryption scheme, and as a result due to Lemma 1 an n-user,c-TTS (resp. n-user,c-TTS of black-box traceability), can be directly obtained.*

Efficiency Parameters. 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,1-copyrighted 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.

We remark that the methodology we describe in this section can be used to yield traitor tracing schemes over any type of 2-user 1-copyrighted encryption function (not necessarily public-key).

6 The New Public-Key Traitor Tracing Schemes

The application of the construction of the previous section to the 2-key,1-copyrighted schemes of sections 4.1 and 4.2 together with Lemma 1 yields two public-key traitor tracing schemes. We summarize these results in this self-contained section in the context of traitor tracing. In the following let $\mathcal{C} = \{\omega_1, \ldots, \omega_n\}$ be a collusion secure $\langle n, v\rangle_2$-code over $\{0, 1\}$ with tracing success probability $1 - \epsilon$ against collusions of up to c users.

For convenience we will describe our schemes under the following plausible "threshold" assumption (introduced in [NP98]). The assumption is applicable to many plaintext-space settings. However, by employing All-or-Nothing Transform this assumption is not necessary as illustrated in section 6.3.

Definition 9. Threshold Assumption. *A pirate-decoder that always returns correctly a percentage C of a plaintext of length b where $1 - C$ is a non-negligible function in b, is useless.*

6.1 Traitor Tracing Scheme 1

In the following ℓ is interpreted as a value in $\{1, \ldots, v\}$.

Key Generation. The authority selects $N_1, \ldots, N_v$ composites so that $N_\ell = p_\ell q_\ell$ and $p_\ell = 2p'_\ell + 1$, $q_\ell = 2q'_\ell + 1$ with $p_\ell, p'_\ell, q_\ell, q'_\ell$ all prime. Without loss of generality we assume that $\nu =_{\mathrm{df}} |N_1| = \ldots = |N_v|$. The public-key of the system is the set to

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

where each $\langle g_\ell\rangle = \mathbb{Q}_{N_\ell}$ and $\alpha_\ell \in_U [p'_\ell q'_\ell]$. User i is given as its personal decryption key the tuple $\langle \kappa_{1,\omega_{i,1}}, \ldots, \kappa_{v,\omega_{i,v}}\rangle$, where $\kappa_{\ell,x} = \alpha_\ell + x\phi(N_\ell)$ for $x \in \{0, 1\}$.

Encryption. Any third party can encrypt a message $\langle M_1, \ldots, M_v\rangle \in \mathbb{Q}_{N_1} \times \ldots \times \mathbb{Q}_{N_v}$ in the following way: $\langle g_1^{r_1} \bmod N_1, y_1^{r_1} \cdot M_1 \bmod N_1, \ldots, g_v^{r_v} \bmod N_v, y_v^{r_v} \cdot M_v \bmod N_v\rangle$ 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 $\langle \kappa_1, \ldots, \kappa_v\rangle$ the decryption is $\langle B_1(A_1^{-1})^{\kappa_1} \bmod N_1, \ldots, B_v(A_v^{-1})^{\kappa_v} \bmod N_v\rangle$.

Traitor Tracing. Suppose that a key $\langle \kappa_1^*, \ldots, \kappa_v^*\rangle$ is constructed by a coalition of $t \le c$ traitors. If all t traitors have a the same key $\kappa_{\ell,x}$ for some $\ell \in \{1, \ldots, v\}$ then because of the fact that the underlying scheme is 1-copyrighted (theorem 1) it is infeasible for them to construct another key to be used for decryption in the ℓ-th coordinate. As a result they have to set $\kappa_\ell^* =_{\mathrm{df}} \kappa_{\ell,x}$ (because of the Threshold Assumption: if $\kappa_{\ell,b}$ is missing from the set of keys available to the pirate decoder then it will fail to decrypt a substantial portion of the plaintext). On the other hand if the t traitors have both keys of the ℓ-th coordinate then they may set κ_ℓ^* to either one, or as described in section 4.1 set κ_ℓ^* to some randomized combination of their keys. Now the tracer computes the string $\omega^* =_{\mathrm{df}} \omega_1^* \ldots \omega_v^*$, in the following way: $\omega_\ell^* =_{\mathrm{df}} x$ if $\kappa_\ell^* = \kappa_{\ell,x}$ where $x \in \{0, 1\}$, or $\omega_\ell^* =_{\mathrm{df}} ?$ if $\kappa_\ell^* \notin \{\kappa_{\ell,0}, \kappa_{\ell,1}\}$. If $C =_{\mathrm{df}} \{\omega_{i_1}, \ldots, \omega_{i_t}\}$ is the set of codewords that correspond to the traitor keys it is easy to verify that $\omega^* \in F(C)$ (where $F(C)$ is the feasible set of the set of codewords C). The tracer runs the tracing procedure of $\mathcal{C}$ on input ω^*. This will yield with probability $1 - \epsilon$ one of the traitors.

The efficiency parameters of the scheme are presented in figure 4.

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

Fig. 4. Efficiency Parameters of the two Traitor Tracing Schemes, over a $\langle n, v\rangle_2$-collusion secure code of codeword length $v = \mathcal{O}(t^4 \log(n/\epsilon)\log(1/\epsilon))$, where ϵ denotes the error probability of the tracer and t the maximum traitor collusion size ([BS95]). In order to simplify the table we can select $b = v = l^{1/2}$, where l is a security parameter. Note that in order to allow tracing with negligible in n probability of failure the security parameter l (size of plaintexts) should be polylogarithmic in the number of users. This is a plausible condition, satisfied in many settings.

6.2 Traitor Tracing Scheme 2

Key Generation. The authority selects $p_1, \ldots, p_v$ primes so that $p_\ell = 2q_\ell + 1$ with q_ℓ also prime. Without loss of generality we assume that $\nu =_{\mathrm{df}} |p_1| = \ldots = |p_v|$. 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 $\boldsymbol{d_{\ell,0}}$ and $\boldsymbol{d_{\ell,1}}$ be two random, linearly independent representations of f_ℓ w.r.t. g_ℓ, h_ℓ. User i is given as the decryption key the tuple $\langle \boldsymbol{d}_{1,\omega_{i,1}}, \ldots, \boldsymbol{d}_{v,\omega_{i,v}}\rangle$,
Encryption. Any third party can encrypt a message $\langle M_1, \ldots, M_v\rangle \in \mathcal{G}_1 \times \ldots \times \mathcal{G}_v$ in the following way: $\langle M_1 \cdot f_1^{r_1} \bmod p_1, g_1^{r_1} \bmod p_1, h_1^{r_1} \bmod p_1, \ldots, M_v \cdot f_v^{r_v} \bmod p_v, g_v^{r_v} \bmod p_v, h_v^{r_v} \bmod p_v\rangle$ where $r_\ell \in_U [q_\ell]$.
Decryption. Given a ciphertext $\langle A_1, B_1, C_1, \ldots, A_v, B_v, C_v\rangle$ and a user-key $\langle \boldsymbol{d_1}, \ldots, \boldsymbol{d_v}\rangle$ the decryption is computed as follows $\langle A_1\langle B_1^{-1}, C_1^{-1}\rangle^{\boldsymbol{d_1}} \bmod p_1, \ldots, A_v\langle B_v^{-1}, C_v^{-1}\rangle^{\boldsymbol{d_v}} \bmod p_v\rangle$, where $\langle a, b\rangle^{\langle c,d\rangle} =_{\mathrm{df}} a^c b^d$.
Black-Box Traitor Tracing. Let $\mathcal{S}$ be a pirate-decoder. The tracer prepares the vectors $\langle x_\ell, y_\ell\rangle$ so that they satisfy the system of equations $\langle x, y \log_{g_\ell} h_\ell\rangle \cdot \boldsymbol{d_{\ell,0}} = a_\ell$ and $\langle x, y \log_{g_\ell} h_\ell\rangle \cdot \boldsymbol{d_{\ell,1}} = b_\ell$ where a_ℓ, b_ℓ are random values of $[q_\ell]$. Subsequently it forms the ciphertext $\langle A_1, B_1, C_1, \ldots, A_v, B_v, C_v\rangle$ with A_ℓ chosen randomly over $\mathbb{Z}^*_p$ and $B_\ell =_{\mathrm{df}} g_\ell^{x_\ell} \bmod p_\ell$, $C_\ell =_{\mathrm{df}} h_\ell^{y_\ell} \bmod p_\ell$. The tracer submits this ciphertext to the tracer and observes the decoder's reply $\langle r_1, \ldots, r_v\rangle$. Then it constructs a codeword $\omega^* = \omega_1^* \ldots \omega_v^*$: as follows: If $r_\ell = A_\ell/g_\ell^{a_\ell}(\mathrm{mod} p_\ell)$ then ω_ℓ^* is set to 0; else if $r_\ell = A_\ell/g_\ell^{b_\ell}(\bmod p_\ell)$ then ω_ℓ^* is set to 1. Finally if $r_\ell \notin \{A_\ell/g_i^{a_\ell} \bmod p_\ell, A_\ell/g_\ell^{b_\ell} \bmod p_\ell\}$, ω_ℓ^* is set to ?. It follows from theorem 3 and the threshold assumption that ω^* belongs to the $F(C)$ where $C = \{\omega_{i_1}, \ldots, \omega_{i_t}\}$ is the set of codewords that correspond to the secret-keys assigned to the traitors. Now provided that $t \le c$ the tracer can recover the identity of one of the traitors by using the tracing algorithm of the code $\mathcal{C}$ with probability $1-\epsilon$. We note here that a *single* query to the pirate decoder is sufficient for our black-box traitor tracing method.

The efficiency parameters of the scheme are presented in figure 4.

6.3 Obviating the Threshold Assumption

The Threshold assumption was instrumental in the traitor tracing methods of our two schemes since it made it necessary for the pirate decoder to include a key for each of the v components. Nevertheless this can also be enforced by employing an all-or-nothing transform (AONT) [Riv97] (alternatively one can use collusion secure codes under weaker marking assumptions, e.g. [SW01a]).

The two public-key traitor tracing schemes described in sections 6.1 and 6.2 have as plaintext space set of strings $\{0,1\}^{bv}$. Let us formulate the construction of [Riv97] in our setting:

All-or-Nothing Transform. ([Riv97]) Let f be a block-cipher and let K_0 be a publicly known key for f. Given a message $\langle m_1, \ldots, m_{v-1} \rangle \in \{0,1\}^{b(v-1)}$ the sender selects a random key K for f and computes $m'_1, \ldots, m'_v \in \{0,1\}^b$ as follows: $m'_i =_{\mathrm{df}} m_i \oplus f(K, i)$ for $i = 1, \ldots, v-1$; note that $\oplus$ stands for the xor operation. The last block m'_v is computed as $m'_v =_{\mathrm{df}} K \oplus h_1 \oplus \ldots \oplus h_{v-1}$, with $h_i =_{\mathrm{df}} f(K_0, m'_i \oplus i)$ for $i = 1, \ldots, v-1$. The output of the transform is the bitstring $m'_1 \ldots m'_v \in \{0,1\}^{bv}$. It is easy to see that the transform can be inverted by anyone that holds *all* blocks $m'_1, \ldots, m'_v$ as follows: $K = m'_v \oplus f(K_0, m'_1 \oplus 1) \oplus \ldots \oplus f(K_0, m'_{v-1} \oplus (v-1))$ and $m_i = m'_i \oplus f(K, i)$ for $i = 1, \ldots, v-1$.

The concept of AONT has been investigated formally in [CDHKS00]. By employing the above AONT in the encryption and decryption operation of our public-key traitor tracing schemes we enforce the pirate to include a secret-key for each one of the v components and therefore there is no need for the Threshold Assumption. The efficiency loss introduced by the use of the AONT is marginal, and it does not affect the stated expansion factors. Finally, note that another plaintext preprocessing which is possible without much ciphertext expansion is employing one of the preprocessing methods (based on random oracle hash and added randomness) in the case where chosen ciphertext security is required.

References

[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, CRYPTO 1999.

[BS95] Dan Boneh and James Shaw, Collusion-Secure Fingerprinting for Digital Data (Extended Abstract), CRYPTO 1995, Springer, pp. 452-465.

[CDHKS00] Ran Canetti, Yevgeniy Dodis, Shai Halevi, Eyal Kushilevitz, and Amit Sahai, Exposure-Resilient Functions and All-or-Nothing Transforms, EUROCRYPT 2000.

[CG98] Dario Catalano, and Rosario Gennaro, New Efficient and Secure Protocols for Verifiable Signature Sharing and Other Applications, CRYPTO 1998.

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

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

[ElG84] Taher El Gamal, A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms, CRYPTO 1984.

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

[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, CRYPTO 1999.

[GM84] Shafi Goldwasser and Silvio Micali, Probabilistic Encryption, JCSS 28(2): pp. 270-299, 1984.

[KY01a] Aggelos Kiayias and Moti Yung, Self Protecting Pirates and Black-Box Traitor Tracing, CRYPTO 2001.

[KY01b] Aggelos Kiayias and Moti Yung, On Crafty Pirates and Foxy Tracers, Proceedings of the 1st Workshop on Security and Privacy in Digital Rights Management, 2001.

[KD98] K. Kurosawa and Y. Desmedt, Optimum Traitor Tracing and Asymmetric Schemes, Eurocrypt 1998.

[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 the Proceedings of Public Key Cryptography 1999, Springer, 188–196.

[NNL01] Dalit Naor, Moni Naor, and Jeffrey B. Lotspiech Revocation and Tracing Schemes for Stateless Receivers, CRYPTO 2001.

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

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

[NR97] Moni Naor and Omer Reingold, Number-Theoretic Constructions of Efficient Pseudo-Random Functions, FOCS 1997.

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

[Riv97] Ron Rivest, All-or-nothing Encryption and the Package Transform, Fast Software Encryption 1997.

[SW00] Reihaneh Safavi-Naini and Yejing Wang, Sequential Traitor Tracing, CRYPTO 2000.

[SW01a] Reihaneh Safavi-Naini and Yejing Wang, Collusion Secure q-ary Fingerprinting for Perceptual Content, Proceedings of the 1st Workshop on Security and Privacy in Digital Rights Management, 2001.

[SW01b] Reihaneh Safavi-Naini and Yejing Wang, New Results on Frameproof Codes and Traceability Schemes, "IEEE Transactions on Information Theory, Vol. 47, No. 7, pp. 3029-3033, 2001.

[SSW00] Jessica N. Staddon, Douglas R. Stinson and Ruizhong Wei, Combinatorial Properties of Frameproof and Traceability Codes, Cryptology ePrint Archive, report 2000/004.

[SW98] Douglas R. Stinson and Ruizhong Wei, Combinatorial Properties and Constructions of Traceability Schemes and Frameproof Codes, SIAM J. on Discrete Math, Vol. 11, no. 1, 1998.

[TY99] Yiannis Tsiounis and Moti Yung, On the Security of ElGamal Based Encryption, Public Key Cryptography 1998.