Efficient Asymmetric Self-Enforcement Scheme with Public Traceability

Hirotaka Komaki, Yuji Watanabe, Goichiro Hanaoka, and Hideki Imai

The Third Department, Institute of Industrial Science, the University of Tokyo
7-22-1 Roppongi, Minato-ku, Tokyo 106-8558, Japan
Phone & Fax: +81-3-3402-7365
{komaki,mue,hanaoka}@imailab.iis.u-tokyo.ac.jp
imai@iis.u-tokyo.ac.jp

Abstract. Traitor tracing schemes deter traitors from giving away their keys to decrypt the contents by enabling the data supplier to identify the source of a redistributed copy. In asymmetric schemes, the supplier can also convince an arbiter of this fact.
Another approach to the same goal was suggested by Dwork, Lotspiech and Naor, so called self-enforcement schemes. In these schemes, traitors have to either divulge their private sensitive information or distribute fairly large amount of data. However, the same private information must be revealed to the data supplier, which invokes the necessity of more discussion about the model underlying this scheme.
In this paper, we present an efficient asymmetric self-enforcement scheme, which also supports the asymmetric traceability without any trusted third parties, assuming the situation where the authenticity of the exponent of each subscriber's sensitive information bound to the subject entity is publicly certified, such as PKI derived from discrete logarithm based cryptosystems. In our scheme, the sensitive information needs not to be revealed to any entities. As far as we know, there has never been any proposal of asymmetric self-enforcement schemes. Furthermore, our scheme is as efficient as the previous most efficient symmetric or asymmetric traitor tracing schemes proposed so far.

1 Introduction

In the contents distribution system over a broadcast channel, such as pay TV, online database, DVD distribution, the system authority gives each authorized subscriber a hardware or software decoder containing a decryption key and broadcasts encrypted digital contents, in order to prevent non-subscribers (*pirates*) from accessing the contents. However, some non-subscribers might obtain a decryption key from one or a set of authorized subscribers (*traitors*) and construct a pirate decoder. In practice today, the so-called *secure* hardware solution is commonly used against such piracy, where cryptanalyzing the secret is assumed to be hard. Unfortunately, such assumption about the security is not always correct. There are several *side-channel attacks* that threat to the assumption,

K. Kim (Ed.): PKC 2001, LNCS 1992, pp. 225–239, 2001.

such as the power analysis [7]. Moreover, secure hardware solutions are often expensive.

In the absence of secure hardware, the cryptographic solution cannot prevent the subscribers from copying their decryption keys themselves. Against the piracy of digital data, such as contents or decryption keys, a lot of cryptographic techniques have been proposed that make redistribution either inconvenient or traceable.

Traitor tracing, which was introduced by Chor, Fiat and Naor [5], is an extension of broadcast encryption, in such a way that it can reveal the traitor on the confiscation of a pirate decoder. To offer traceability, each subscriber is given a different set of decryption keys that identify the subscriber. In a sense, traitor tracing can be seen as one of fingerprinting schemes [15,12] where the decryption keys are fingerprinted. A traitor tracing is called *k-resistant* if the scheme can reveal at least one traitor on the confiscation of a pirate decoder which was constructed by up to k traitors.

The schemes in [5] are *symmetric* in the sense that subscribers share all their secrets with the system authority. In symmetric schemes, the authority itself, or malicious employee or someone with illegal access to the authority could incriminate an honest subscriber as a traitor using the secrets. Thus the result of tracing is no real evidence that could unambiguously convince a third party. Pfitzmann [11] pointed out this problem and introduced *asymmetric* traitor tracing schemes, where using an interactive key distribution protocol the system authority cannot construct a pirate decoder that frames an honest subscriber, but if a pirate decoder which was constructed by malicious subscribers is found the authority is able to trace the source of it.

The problem with relying on the traceability is that the authority or other entities must always monitor all over the world for potentially redistributed copies. To overcome such shortcoming, Dwork, Lotspiech and Naor [6] introduced another approach, so-called *self-enforcement scheme*, where the system authority needs not the confiscation of a pirate decoder nor trace the source of it. Instead, traitors have to either divulge their private sensitive information (e.g., a credit card number) or distribute fairly large amount of data in order to succeed the piracy. A similar approach using sensitive information has been suggested by Sander and Ta-Shma [13] in the context of electronic payment systems, in order to make the coin *non-transferable* as a countermeasure to financial crimes, such as the tax evasion.

The scheme in [6] still remains a serious problem that the same private information must be revealed to the data supplier, which invokes the necessity of more discussion about the model underlying this scheme. In [11], Pfitzmann mentioned that one can try to combine those techniques with the ideas of asymmetric traitor tracing, so that the private information can also remain private from the authority. But it is not so easy to construct such *asymmetric self-enforcement scheme* because the authority must confirm that subscriber's decryption key certainly contains his private sensitive information without the knowledge of it. Asymmetric traitor tracing schemes do not need such a verification. As far

as we know, there has never been any proposal of asymmetric self-enforcement schemes.

Our Contribution In this paper, we present an efficient asymmetric self-enforcement scheme, which also offers asymmetric traceability. In our scheme, we assume the situation where each subscriber i has a pair of values (d_i, g^{d_i}) such that d_i is i's private sensitive information (e.g., the secret key in Public Key Infrastructure (PKI), biometrical information, code number of the bank account or driver's license number etc.) and g^{d_i} is the exponent of it, and the authenticity of g^{d_i} bound to the subject entity i is publicly certified. If we take each subscriber's secret key in PKI derived from discrete logarithm based cryptosystems as their private sensitive information d_i, then g^{d_i} corresponds to the public key and the Certification Authority (CA) vouches for the authenticity of the public key. Thus such an assumption underlies every cryptosystem based on PKI. For concreteness we consider the situation that each user's private sensitive information is their secret key in PKI derived from discrete logarithm based cryptosystems hereafter unless otherwise mentioned.

In our scheme, even if at most k malicious subscribers collude to construct a pirate decoder, arbitrary entity can obtain all of their secret keys in PKI on the confiscation of it. Thus the redistribution of the decryption key which contains the secret keys damages the traitors very much. For example, if the secret key is used for the signature, a receiver of it can impersonate the owner, e.g. for signing contracts, receiving loans, etc. Thus the owner of the key may not want to give it away.

Furthermore if a pirate decoder which is constructed by at most k traitors is confiscated by any entities, they can trace and reveal the secret of all the traitors and convince any third party of the validity of the tracing results without the participation of the accused traitors and the system authority in the trial phase, unless the system authority plays the role of the tracer (*direct non-repudiation*). Our scheme does not need any trusted third parties (if we assume PKI) and offer *full frameproof* that arbitrary collusion of entities including the system authority cannot frame an honest subscriber. In our scheme, the encryption key of the broadcasted contents can be open to the public and an arbitrary entity can play a role of the data supplier.

Our scheme is very efficient compared with the previous most efficient symmetric or asymmetric traitor tracing schemes proposed so far. Table 1 shows a comparison about the efficiency among related works and our proposal. [1] $1/\rho, 1/\rho_B$ are defined by

$$1/\rho \stackrel{\triangle}{=} max\left\{\frac{log|\mathcal{U}_i|}{log|\mathcal{S}|} : i \in \Phi\right\}$$

[1] In the original scheme in [8] $1/\rho_B = k + 2$, but it was not *k-resistant* but $\lfloor \frac{k+1}{2} \rfloor$-*resistant* on the *convex combination attack* [14,3]. If we take $k \to 2k - 1$, then the scheme in [8] offers *k-resistant*. See Section4 for more details.

Table 1. A comparison of the decryption key size and the data redundancy.

		$1/\rho$	$1/\rho_B$
[12]	Scheme 1	$O(k \log n)$	$O(k^2 \log n)$
[12]	Scheme 2	$O(\sigma k)$	$O(\sigma^2 k^2)$
[8]		1	$2k+1$
[18]		2	$4k+3$
[16,17]		2	$3k+3$
Proposal		2	$2k+1$

$$1/\rho_B \triangleq max\left\{\frac{log|\mathcal{B}|}{log|\mathcal{S}|}\right\}$$

where $\mathcal{U}_i$ denotes the set of all possible subsets of decryption keys, $\mathcal{B}$ denotes the set of all possible subsets of the data redundancy, $\mathcal{S}$ denotes the set of all possible subsets of the session keys and Φ denotes the set of subscribers of the system [14]. Thus $1/\rho$ is a parameter on the size of each user's decryption key and $1/\rho_B$ is a parameter on the size of data redundancy. n is the number of subscribers and σ is a parameter where the system authority cannot frame an honest subscriber as a traitor with probability more than $1/2^\sigma$. From a brief view of Table 1, one can see that our scheme is one of the most efficient schemes.

Related Works. The asymmetric traitor tracing scheme suggested by Pfitzmann and Waidner in [11,12] used the symmetric scheme in [5] as a building block. The scheme was, however, not efficient because the scheme of [5] on which it is based required large overhead and large decryption key.

Kurosawa and Desmedt [8] proposed a more efficient construction of asymmetric public key traceability scheme where the encryption key could be public. Their scheme is based on the ElGamal type threshold cryptosystem and very efficient. But their approach to the asymmetric traceability was to share the secret information of the system authority among some trusted third parties, which implies that they can still frame an honest subscriber on their conspiracy.

Another constructions of asymmetric public key traceability schemes without any trusted third parties were given in [16,17,18]. In [18], any entities can trace the source of a pirate decoder and can convince any arbiter of the result, but a suspected traitor must participate in trial phase to prove his guilt. The scheme in [16,17] offers the tracer enough information to convince the arbiter and needs not the accused traitor's participation in trial phase. However those schemes do not support self-enforcement as mentioned above.

Organization. In Section 2, we describe the model, the definition and building blocks which include efficient public key traitor tracing schemes and *oblivious polynomial evaluation* protocol. Our construction is described in Section 3. We analyze the security of our proposal in Section 4. Finally, conclusions are given in Section 5.

2 Preliminaries

In this section we describe the model underlying our protocol and define the security requirement. Then we describe an efficient public key traitor tracing scheme and oblivious polynomial evaluation protocol, which are building blocks of our proposal.

2.1 Model

The entities who participate in the proposed protocol are as follows. (Considering the practical case, we describe the model where the data supplier plays a role of the tracer. As mentioned in Section 1, arbitrary entites *can* do it.)

System Authority: The system Authority $\mathcal{SA}$ generates one public key and sells the decoder containing the decryption key (*personal key*) to each user, in a complicated way such that the personal key is constructed by each user's secret key of the PKI and $\mathcal{SA}$ cannot know the value of the key itself.

Data Supplier: The data supplier $\mathcal{T}$ distributes encrypted contents using a public key generated by $\mathcal{SA}$, and if he finds the pirate decoder which was illegally distributed, he traces the source of it. $\mathcal{SA}$ may play the role of $\mathcal{T}$.

Users: Users are authorized subscribers of the system and the set of users is denoted by $\Phi = \{1, \cdots, n\}$. Each user decrypts the encrypted contents with the personal key, which was given by $\mathcal{SA}$ when he subscribed to the system.

Traitors: Traitors are person who redistribute their own personal keys, or colluders to construct the pirate decoder, and the set of traitors is denoted by $\Omega = \{u_1, \cdots, u_t\}$. Then, $\Omega \subset \Phi$.

Arbiter: An arbiter $\mathcal{A}$ is a party that verifies the tracing results. In practice, this can be a court of law or an entity publicly agreed on.

Certification Authority: A certification authority $\mathcal{CA}$ vouches for the authenticity of the public key of PKI bound to the subject user. We consider the case where PKI is derived from discrete logarithm based cryptosystems, such as ElGamal encryption.

Before executing the protocol, $\mathcal{CA}$ generates his signatures on $text_i$ for each user i ($i \in \Phi$), where $text_i$ includes the public key $v_i \stackrel{\triangle}{=} g^{d_i}$ (d_i is user i's secret key), the identity of subject entity and other information such as serial numbers or algorithm etc. $\mathcal{CA}$ publicizes $text_i$ and the signature denoted by $sig_{CA}(text_i)$. Then, our scheme proceeds as follows:

System initialization – $\mathcal{SA}$ generates some information that he will need with $\mathcal{T}$ and all users, and he makes the encryption key $e_{\mathcal{SA}}$ public.

User initialization – $\mathcal{SA}$ sells the decoder containing the personal key D_i to user i, in a way that D_i is constructed by d_i, and $\mathcal{SA}$ verifies this with zero-knowledge of the value of d_i nor D_i.

Session sending – The content is divided into smaller parts called sessions denoted by M. $\mathcal{T}$ chooses a random session key s, and broadcasts ($e_{\mathcal{SA}}(s)$,

$ENC_s(M)$), where $e_{\mathcal{SA}}(s)$ is called a *header* and ENC is a symmetric encryption function. Each user i can decrypt $e_{\mathcal{SA}}(s)$ to obtain s with D_i, and thus can decrypt M.

Tracing – If a pirate decoder is confiscated, $\mathcal{T}$ analyzes the personal keys D_{u_1}, $\cdots$, D_{u_t} in it and specifies a set of the suspected traitors Ω.

Trial – $\mathcal{T}$ accuses Ω to $\mathcal{A}$ with $(D_{u_j}, text_{u_j}, sig_{\mathcal{CA}}(text_{u_j}))$, $(j \in \Omega)$.

In this paper, the confiscation of a pirate decoder implies the possibility to obtain the value of the decryption key in it. If a non-subscriber obtains a pirate decoder which was constructed by at most k traitors, he can reveal all the secret keys of the traitors. Thus, in the model of our scheme traitors do not willingly give away their personal keys which include their secret keys, and such a situation is not considered in the model of traitor tracing.

2.2 Definition

Security requirements for $\mathcal{SA}$ on our schemes are as follows.

Definition 1 *We say that secrecy is established, if it is hard for non-subscribers to cryptanalyze a new session key from a new header, the public key, the old headers, and the old session keys.*

Definition 2 *When a pirate decoder is confiscated, which was constructed by at most k traitors, and if $\mathcal{T}$ can identify at least one traitor, then we say that traceability is established.*

Definition 3 *If no collusion among at mosk k malicious users can construct a pirate decoder without divulging at least one of the traitors' sensitive information to the entity who obtained the decoder, then we say that self-enforcement is established.*

Definition 4 *We say that direct non-repudiation is established, if $\mathcal{T}$ can convince $\mathcal{A}$ of the validity of the tracing results without the participation of the suspected traitors and $\mathcal{SA}$ in the trial phase, if not $\mathcal{T} = \mathcal{SA}$.*

Next, we define security for users.

Definition 5 *We say full frameproof is established, if no collusion of arbitrary entities including $\mathcal{SA}$ can frame an honest user*

Now we give our security parameters. Let q be a prime such that $q|p-1$, $q \geq n+1$, where p is a prime power, and let g be a qth root of unity over $GF(p)$. Unless we stated otherwise, all calculation as below is done over $GF(p)$, and all entities agree on p, q and g.

2.3 Building Blocks

Our protocol is based on the efficient public key traitor tracing [8], and Oblivious Polynomial Evaluation (OPE) protocol [10]. Now, we briefly describe them.

Asymmetric Traitor Tracing with Agents [8]. Kurosawa and Desmedt suggested an efficient public key asymmetric traitor tracing scheme. Their scheme offers asymmetric traceability by dividing $\mathcal{SA}$'s secret into some pieces and separately distributing these to multiple trusted third parties, called *agents*, denoted by $\mathcal{A}_1, \cdots, \mathcal{A}_c$. Their scheme is as follows.

- *System initialization:* Each agent $\mathcal{A}_i$ chooses a random polynomial $f_i(x) = a_{i,0} + a_{i,1}x + \cdots + a_{i,k}x^k$ over Z_q and publicizes

$$y_{i,0} = g^{a_{i,0}}, y_{i,1} = g^{a_{i,1}}, \cdots, y_{i,k} = g^{a_{i,k}}$$

  Let

$$y_j \stackrel{\triangle}{=} \prod_{i=1}^{c} y_{i,j} \quad (j = 0, \cdots, k)$$

  Then, the public encryption key is $(p, g, y_0, \cdots, y_k)$.
- *User initialization:* Each $\mathcal{A}_i$ secretly gives $f_i(j)$ to user j. Let

$$f(x) \stackrel{\triangle}{=} \sum_{i=1}^{c} f_i(x) = a_0 + a_1 x + \cdots + a_k x^k$$

  Then, $y_j = g^{a_j}$ for $j = 1, \cdots, k$. User j computes $f(j) = \sum_{i=1}^{c} f_i(j)$ and now $D_j = (j, f(j))$ is his personal key.
- *Session sending:* For a session key s, $\mathcal{T}$ computes a header as $h(s, r) = (g^r, b_0, b_1, \cdots, b_k)$, where $b_0 = s{y_0}^r$, $b_1 = {y_1}^r$, $\cdots$, $b_k = {y_k}^r$ and r is a random number. Then $\mathcal{T}$ broadcasts $h(s, r)$. Each user i computes s from $h(s, r)$ and his personal key as follows.

$$\prod_{j=0}^{k} b_j^{i^j} \Big/ (g^r)^{f(i)} = s$$

- *Tracing:* If the pirate decoder contains $(u, f(u))$, and if

$$g^{f(u)} = \prod_{i=0}^{k} {y_i}^{u^i} \tag{1}$$

  then $\mathcal{T}$ decides u is a traitor.
- *Trial:* $\mathcal{A}$ is convinced if (1) holds.

In the above protocol, the evidence of the piracy of u is $(u, f(u))$ which satisfies (1). However, the collusion of all agents can generate such evidence and thus frame an honest user. Therefore, their scheme does not offer full frameproof in the sense of our definition.

Oblivious Polynomial Evaluation. The Oblivious Polynomial Evaluation (OPE) protocol, which was introduced by Naor and Pinkas [10], is a novel and useful primitive for two party computation. One party Bob knows a polynomial P and he would like to let Alice compute the value $P(\alpha)$ for an input α known to her in such a way that Bob does not learn α and Alice does not gain any additional information about P (except $P(\alpha)$).

We briefly review the protocol. Bob chooses a random bivariate polynomial $Q(x, y)$, such that $Q(0, y) = P(y)$. Alice chooses a random univariate polynomial $S(x)$, such that $S(0) = \alpha$. Alice's plan is to interpolate the univariate polynomial $R(x) \stackrel{\triangle}{=} Q(x, S(x))$ without revealing $S(x)$, then she knows $R(0) = Q(0, S(0)) = P(S(0)) = P(\alpha)$. This is done by sending n ($\stackrel{\triangle}{=} \deg(R(x)) + 1$) x_i and a list of random values $y_{i,j}$ except one value $S(x_i)$ $(i = 1, \cdots, n)$. Bob computes $Q(x_i, y_{i,j})$ for all these values and Alice retrieves $Q(x_i, S(x_i))$ $(i = 1, \cdots, n)$ using 1-out-of-m oblivious transfer. Then, she can interpolate $R(x)$.

In the above protocol, the secrecy of Alice's input depends on the *noisy polynomial interpolation problem* [10,2]. Recently it was proved that the noisy polynomial interpolation problem can be transformed into the lattice shortest vector problem with high probability, when the parameters satisfy a certain condition, and some other attacks to the problem were suggested [2]. However, a little modification on the protocol makes the secrecy of Alice's input dependent on the *polynomial reconstruction problem*, which is one of the well known most intractable problems [2].

The OPE protocol uses 1-out-of-N oblivious transfer (OT) protocol as the building block. Naor and Pinkas also suggested an efficient construction of 1-out-of-N OT protocol in [10], which invokes 1-out-of-2 OT $\log N$ times and evaluates a pseudo-random function $N \log N$ times. If we use non-interactive 1-out-of-2 OT in [1], the number of communication paths in 1-out-of-N OT and OPE protocol is two.

3 Construction

The central idea of our proposal is to reduce the traitor tracing scheme in [8] to the scheme where user i's personal key is represented by $(d_i, f(d_i))$ (d_i is user i's secret key in the PKI) and $\mathcal{SA}$ can verify it using the public information $text_i, sig_{\mathcal{CA}}(text_i)$. To accomplish this, we divide $f(x)$ into two polynomials $f_1(x), f_2(x)$, such that $(f_1(x) + f_2(x) = f(x))$. $\mathcal{SA}$ distributes $f_1(d_i)$ to user i using OPE protocol and then $\mathcal{SA}$ verifies that the user i's input to OPE was really d_i, without the knowledge of d_i itself. If the verification is passed, $\mathcal{SA}$ distributes $f_2(d_i)$ to user i using OPE. In the second invocation of OPE, even if the user inputs the value which is different from d_i adaptively, he cannot gain any useful information. (See Section 4)

Now, we present the construction of our proposal.

Table 2. System and User Initialization Protocol

$\mathcal{SA}$		User $i \in \Phi$
$f(x) = \sum_{l=0}^{2k-1} a_l x^l \bmod q$		
$f_{i,1}(x) = \sum_{l=0}^{2k-1} b_{i,l} x^l \bmod q$		
$f_{i,2}(x) = f(x) - f_{i,1}(x)$ $= \sum_{l=0}^{2k-1} c_{i,l} x^l \bmod q$		
		d_i: User i 's secret key
	$f_{i,1}(d_i)$ (by OPE) $\longrightarrow$	
		$(w_{i,2}, \cdots, w_{i,2k-1}, w_i) \leftarrow$ $(g^{{d_i}^2}, \cdots, g^{{d_i}^{2k-1}}, g^{f_{i,1}(d_i)})$
	$\longleftarrow$ $(w_{i,2}, \cdots, w_{i,2k-1}, w_i)$	
check $g^{b_{i,0}} {v_i}^{b_{i,1}} \prod_{l=2}^{2k-1} {w_{i,l}}^{b_{i,l}}$ $\stackrel{?}{=} w_i$		
	$f_{i,2}(d_i)$ (by OPE) $\longrightarrow$	
		personal key $D_i = (d_i, f(d_i))$ $= (d_i, f_{i,1}(d_i) + f_{i,2}(d_i))$

System Initialization. $\mathcal{SA}$ chooses a random polynomial $f(x) = \sum_{l=0}^{2k-1} a_l x^l$ and computes

$$y_l = g^{a_l} \quad (l = 0, 1, \cdots, 2k-1)$$

Then, the public encryption key is $(p, g, y_0, \cdots, y_{2k-1})$.

User Initialization. $\mathcal{SA}$ chooses a random polynomial $f_{i,1}(x) = \sum_{l=0}^{2k-1} b_{i,l} x^l$ for each user i and computes

$$f_{i,2}(x) = f(x) - f_{i,1}(x) = \sum_{l=0}^{2k-1} c_{i,l} x^l$$

$$c_{i,l} = a_{i,l} - b_{i,l} \quad (l = 0, 1, \cdots, 2k-1)$$

$\mathcal{SA}$ distributes $f_{i,1}(d_i)$ to user i using the OPE protocol, where d_i is the user i's secret key in PKI. Then $\mathcal{SA}$ verifies that the user i's input was really d_i as follows.

User i computes

$$(w_{i,2}, w_{i,3}, \cdots, w_{i,2k-1}, w_i) \stackrel{\triangle}{=} (g^{{d_i}^2}, g^{{d_i}^3}, \cdots, g^{{d_i}^{2k-1}}, g^{f_{i,1}(d_i)})$$

and sends it to $\mathcal{SA}$. $\mathcal{SA}$ verifies whether

$$g^{b_{i,0}} {v_i}^{b_{i,1}} {w_{i,2}}^{b_{i,2}} {w_{i,3}}^{b_{i,3}} \cdots {w_{i,2k-1}}^{b_{i,2k-1}} = w_i.$$

holds. If the verification is passed, then $\mathcal{SA}$ distributes $f_{i,2}(d_i)$ to user i using the OPE protocol. Now, $D_i = (d_i, f(d_i)) = (d_i, f_{i,1}(d_i) + f_{i,2}(d_i))$ is the user i's personal key.

Session Sending. For a session key s, $\mathcal{T}$ computes a header as $h(s,r) = (g^r, b_0, b_1, \cdots, b_{2k-1})$, where $b_0 = s{y_0}^r$, $b_1 = {y_1}^r$, $\cdots$, $b_{2k-1} = {y_{2k-1}}^r$ and r is a random number. Then $\mathcal{T}$ broadcasts $h(s,r)$. Each user i computes s from $h(s,r)$ and his personal key as follows.

$$\prod_{j=0}^{2k-1} {b_j}^{{d_i}^j} \Big/ (g^r)^{f(d_i)} = s$$

Tracing. If the pirate decoder contains $D_{u_j} = (d_{u_j}, f(d_{u_j}))$, $(j \in \Omega)$, and if

$$g^{f(d_{u_j})} = \prod_{l=0}^{2k-1} {y_l}^{{d_{u_j}}^l} \quad (j \in \Omega) \tag{2}$$

$$g^{d_{u_j}} = v_{u_j} \quad (j \in \Omega) \tag{3}$$

holds, then $\mathcal{T}$ identifies Ω as the set of traitors.

Trial, If $\mathcal{T}$ decides that Ω is the set of traitors, then $\mathcal{T}$ gives $\mathcal{A}$ $(D_{u_j} = (d_{u_j}, f(d_{u_j}))$, $text_{u_j}$, $sig_{\mathcal{CA}}(text_{u_j}))$, $(j \in \Omega)$ as the evidence of piracy. $\mathcal{A}$ is convinced if (2)(3) holds.

4 Security

In this section we analyze the security of our scheme, which consists of the security analysis for the system authority and for users.

4.1 Security for the System Authority

Secrecy In [8], it is proved (Theorem 14) that the computational complexity for an eavesdropper to cryptanalyze a new session key s , after having received previous session keys s_j, $(j = 0, 1, \cdots, l)$, the public key $e_{\mathcal{SA}} = (p, g, g^{a_0}, g^{a_1}, \cdots, g^{a_k})$, the old headers $h(s_j, r_j) = (g^{r_j},\ s_j g^{a_0 r_j},\ g^{a_1 r_j},\ \cdots,\ g^{a_k r_j})$, $(j = 0, 1, \cdots, l)$ and the new header $h(s, r) = (g^r,\ sg^{a_0 r},\ g^{a_1 r},\ \cdots,\ g^{a_k r})$, is as hard as to cryptanalyze a plaintext in the ElGamal scheme when the order of g is a prime. In our scheme, the information that a passive eavesdropper could get is the same as in the scheme in [8] except for $k \to 2k-1$, thus secrecy of the proposal also depends on the difficulty of breaking underlying the ElGamal encryption scheme.

Traceability and Self-Enforcement The strategy for malicious users to construct a pirate decoder which is not traceable or which does not divulge any sensitive information of malicious users is classified into two types.

1. to input invalid values in the user initialization phase adaptively
2. to construct a pirate decoder by colluding among up to k traitors after having received legitimate personal keys in the user initialization phase

We demonstrate that such strategies are not effective in our scheme. First, we give some lemmas about the former strategy.

Lemma 1 *In the user initialization phase, a malicious user u cannot pass the verification after the first invocation of OPE protocol if he inputs an invalid value d' different from his secret key d_u assuming the intractability of breaking the OPE protocol.*

Proof (Sketch)

To pass the verification, u must send $(w_{u,2},\ w_{u,3},\ \cdots,\ w_{u,2k-1},\ w_u)$ for which

$$g^{b_{u,0}}(g^{d_u})^{b_{u,1}} {w_{u,2}}^{b_{u,2}} {w_{u,3}}^{b_{u,3}} \cdots {w_{u,2k-1}}^{b_{u,2k-1}} = w_u$$

holds after having $d_u, d', f_{u,1}(d')$. It is obviously information theoretically intractable. For example, if we assume u knows $b_{u,2}, b_{u,3}, \cdots, b_{u,2k-1}$, then he can obtain $b_{u,0} + b_{u,1}d'$ by $f_{u,1}(d') - (b_{u,2}d'^2 + \cdots + b_{u,k}d'^{2k-1})$. However he cannot compute $g^{b_{u,0}}(g^{d_u})^{b_{u,1}}$ using these values without knowledge of $b_{u,0}$ or $b_{u,1}$.$\diamond$

Thus, if the malicious user u selects the former strategy, all the measures he can take is to input an invalid value d' in the second invocation of the OPE protocol. However it is obviously information theoretically intractable to compute $(d'', f(d''))$ using $(d_u, f_{u,1}(d_u))$ and $(d', f_{u,2}(d'))$ where $d_u \neq d'$, $f = f_{u,1} + f_{u,2}$. If he can divide the header as follows

$$(g^r, sg^{a_0 r}, g^{a_1 r}, \cdots, g^{a_{2k-1} r}) = (g^r, sg^{b_{u,0} r} g^{c_{u,0} r}, g^{b_{u,1} r} g^{c_{u,1} r}, \cdots, g^{b_{u,2k-1} r} g^{c_{u,2k-1} r})$$

then he can compute s by

$$\frac{sg^{b_{u,0} r} g^{c_{u,0} r} (g^{b_{u,1} r})^{d_u} (g^{c_{u,1} r})^{d'} \cdots (g^{b_{u,2k-1} r})^{{d_u}^{2k-1}} (g^{c_{u,2k-1} r})^{{d'}^{2k-1}}}{(g^r)^{f_{u,1}(d_u) + f_{u,2}(d')}} = s$$

But such a division is obviously information theoretically intractable. Moreover any collusion attacks on the former strategy is meaningless because $(b_{u,0}, b_{u,1}, \cdots, b_{u,2k-1})$ is unique to user u. Therefore the former strategy is ineffective.

Now we discuss the latter strategy. Due to the similar argument to that of [8], the next lemma holds.

Lemma 2 *The computational complexity for* $2k-1$ *traitors of finding* $(d_u, f(d_u))$, *where* $d_u \notin \{d_{u_1}, \cdots, d_{u_{2k-1}}\}$, *when given the public key and their personal keys* $(f(u_1), \cdots, f(u_{2k-1}))$ *is as hard as the discrete logarithm problem (DLP) when the order* g *is prime.*

Proof Sketch

In [8] Theorem 15 proves the same lemma where $2k-1 \to k$. $\diamond$

The above lemma shows that it is hard to construct another legitimate pirate key $(d, f(d))$ by at most k traitors $\Omega = \{u_1, \cdots, u_t\}$, such that $(d, f(d)) \notin \{(d_{u_1}, f(d_{u_1})), \cdots, (d_{u_t}, f(d_{u_t}))\}$, assuming DLP is computationally hard to solve. However the pirate may construct a key which is not a legitimate personal key but can be used to decrypt a session key in such a way that none of the traitors is identified from it. A successful pirate strategy is given in [14,3] which defeats the scheme of [8] by using a convex combination of traitors' personal keys. Fortunately, it is known that by increasing the degree of f from k to $2k-1$, anyone who confiscated a pirate decoder which was constructed on this strategy can compute all the traitors' personal keys if the number of traitors is at most k [4,3]. This means that in our protocol arbitrary entities can recover all of the traitor's secret keys in the PKI, on the confiscation of a pirate decoder which was constructed by at most k traitors.

We review their convex combination attack and how it is solved when the degree of f is $2k-1$. Let $\boldsymbol{\beta} = (\alpha, \beta_0, \cdots, \beta_{2k-1})^T$ such that

$$\boldsymbol{\beta} = t_1 \boldsymbol{u}_1 + \cdots + t_k \boldsymbol{u}_k$$

where

$$\boldsymbol{u}_j = (f(d_{u_j}), 1, {d_{u_j}}, \cdots, {d_{u_j}}^{2k-1})^T \quad (j = 1, \cdots, k)$$
$$t_1 + \cdots + t_k = 1$$

Then $\boldsymbol{\beta}$ is not a legitimate personal key, but it can be used to decrypt any header $h(s, r) = (g^r, b_0, b_1, \cdots, b_k) = (g^r, s g^{a_0 r}, g^{a_1 r}, \cdots, g^{a_{2k-1} r})$ by

$$\prod_{l=0}^{2k-1} {b_l}^{\beta_l} \Big/ (g^r)^{\alpha} = \prod_{m=1}^{k} \left(\prod_{l=0}^{2k-1} {b_l}^{{d_{u_m}}^l} \Big/ (g^r)^{f(d_{u_m})} \right)^{t_m} = \prod_{m=1}^{k} s^{t_m} = s$$

Those who obtained $\boldsymbol{\beta}$ can construct the following equation with $2k$ unknowns $u_1, \cdots, u_k, t_1, \cdots, t_k$.

$$\begin{pmatrix} 1 \\ \beta_1 \\ \vdots \\ \beta_{2k-1} \end{pmatrix} = \begin{pmatrix} 1 & \cdots & 1 \\ d_{u_1} & \cdots & d_{u_k} \\ \vdots & \ddots & \vdots \\ {d_{u_1}}^{2k-1} & \cdots & {d_{u_k}}^{2k-1} \end{pmatrix} \begin{pmatrix} t_1 \\ t_2 \\ \vdots \\ t_k \end{pmatrix}$$

Some efficient solving algorithms of the above equation are already known, such as the Berlekamp-Massey algorithm [9] or the Peterson-Gorenstein-Zierler decoding algorithm etc. (see [19] for example) and everyone can recover $d_{u_1}, \cdots, d_{u_k}$ which are traitors secret keys in PKI. Therefore the latter strategy is also ineffective.

For the discussions stated above we can conclude that our proposal offers traceability and self-enforcement.

Direct Non-repudiation. Let assume that user i is honest, namely, he keeps his personal key $(d_i, f(d_i))$ secret. Then all the information about d_i that the data supplier, malicious users, and the certification authority could obtain is the corresponding public key g^{d_i}. Thus the difficulty of framing a user i by them is the same as DLP. All the information about d_i that the system authority could obtain is $g^{d_i}, g^{{d_i}^2}, \cdots, g^{{d_i}^{2k-1}}$ if we assume that breaking the OPE protocol is intractable. The difficulty of deriving d_i from these values is believed to be as hard as DLP. For the discussion stated above, it is computationally hard for the malicious entities to obtain honest user i's secret key d_i *unless* he divulges it.

Thus, $(d_{u_j}, f(d_{u_j}))(u_j \in \Omega)$ could be the enough evidence to prove the redistribution of the personal key by u_j. (2)(3) in Section 3 means d_{u_j} and $f(d_{u_j})$ matching, and user u_j's secret key and public key matching respectively. The verification process only needs pirate key and public information, and does not need the participation of the pirate user and the system authority.

For the discussion stated above, our proposal offers direct non-repudiation.

4.2 Security for Users

Full Frameproof. In order to win the trial, the tracer must submit the suspected traitor's secret key to the arbiter. Due to the same discussion as the one in the proof of direct non-repudiation, if an honest user i keeps his personal key $D_i = (d_i, f(d_i))$ secret, it is computationally hard for the other malicious entities to obtain user i's secret key d_i, which is a part of the proof of piracy.

Thus our proposal offers full frameproof.

5 Conclusion

In this paper we present a concrete construction of an asymmetric self-enforcement scheme. Our proposal also offers asymmetric traceability without trusted third parties, and is very efficient compared to previous traitor tracing schemes. In the model of our scheme, malicious users do not willingly give away their personal keys because our scheme guarantees that each user's personal key includes the user's sensitive information e.g. secret key in PKI, and such situation is not considered in the model of traitor tracing schemes. Furthermore, our protocol offers direct non-repudiation and full frameproof.

Acknowledgments

Part of this work was supported by Research for the Future Program (RFTF), Japan Society for the Promotion of Science (JSPS), under contract number JSPS-RETF 96P00604.

The authors would like to thank Tatsuyuki Matsushita for very useful discussion in preparing this paper. We are grateful to anonymous referees for their valuable comments that improved the presentation of this paper.

References

1. M.Bellare and S.Micali, "Non-interactive oblivious transfer and applications," *Advances in Cryptology - CRYPTO '89*, LNCS 435, Springer-Verlag, pp.547-557, 1990.
2. D.Bleichenbacher and P.Q.Nguyen, "Noisy Polynomial Interpolation and Noisy Chinese Remaindering," *Advances in Cryptology - EUROCRYPT 2000*, LNCS 1807, Springer-Verlag, pp.53-69, 2000.
3. D.Boneh and M.Franklin, "An Efficient Public Key Traitor Tracing Scheme," *Advances in Cryptology - CRYPTO '99*, LNCS 1666, Springer-Verlag, pp.338-353, 1999.
4. Burmester, Desmedt, Kurosawa, Ogata and Okada, *manuscript.*
5. B.Chor, A.Fiat and M.Naor, "Tracing Traitors," *Advances in Cryptology - CRYPTO '94*, LNCS 839, Springer-Verlag, pp.257-270, 1994.
6. C.Dwork, J.Lotspiech and M.Naor, "Digital Signets: Self-Enforcing Protection of Digital Information," *Proc. of 28th ACM Symposium on Theory of Computing(STOC)*, pp.489-498, 1996.
7. P.Kocher, J.Jaffe and B.Jun, "Differential Power Analysis," *Advances in Cryptology - CRYPTO '99*, LNCS 1666, Springer-Verlag, pp.388-397, 1999.
8. K.Kurosawa and Y.Desmedt, "Optimum Traitor Tracing and Asymmetric Schemes," *Advances in Cryptology - EUROCRYPT '98*, LNCS 1403, Springer-Verlag, pp.145-157, 1998.
9. J.L.Massey, "Shift Register Synthesis and BCH Decoding," *IEEE Transactions on Information Theory*, vol. IT-15, No.1, pp.122-127, January 1969.
10. M.Naor and B.Pinkas, "Oblivious Transfer and Polynomial Evaluation," *Proc. of 31th ACM Symposium on Theory of Computing(STOC)*, pp.245-254, 1999.
11. B.Pfitzmann, "Trials of Traced Traitors," *Proc. of Information Hiding, First International Workshop*, LNCS 1174, Springer-Verlag, pp.49-64,1996.
12. B.Pfitzmann and M.Waidner, "Asymmetric Fingerprinting for Lager Collusions," *Proc. of ACM Conference on Computer and Communication Security*, pp.151-160, 1997.
13. T.Sander and A.Ta-Shma, "Flow Control: A New Approach for Anonymity Control in Electronic Cash Systems," *Proc. of Financial Cryptography: Third International Conference, FC'99*, LNCS 1648, Springer-Verlag, pp.46-61, 1999.
14. D.R.Stinson and R.Wei, "Key Preassigned Traceability Schemes for Broadcast Encryption," *Proc. of SAC '98*, LNCS 1556, Springer-Verlag, pp.144-156, 1998.
15. N.R.Wagner, "Fingerprinting," *Proc. of IEEE 1983 Symposium on Security and Privacy*, April, pp.18-22, 1983.
16. Y.Watanabe, G.Hanaoka and H.Imai, "Asymmetric Public-Key Traitor Tracing without Trusted Agents," *Proc. of the Symposium on Information Theory and Its Application (SITA 2000)*, October, 2000.

17. Y. Watanabe, G. Hanaoka and H. Imai, "Efficient Asymmetric Public-Key Traitor Tracing without Trusted Agents," *to appear in Proc. of the RSA Conference Cryptographer's Track*, April, 2001 (to be published in LNCS).
18. Y.Watanabe, H.Komaki, G.Hanaoka and H.Imai, "Asymmetric Traitor Tracing based on Oblivious Polynomial Evaluation," (in Japanese) *IEICE Technical Report, ISEC*, September, 2000.
19. S.B.Wicker, "Error Control Systems for Digital Communication and Storage," Prentice-Hall, Inc., 1995.