Extracting Group Signatures from Traitor Tracing Schemes

Aggelos Kiayias[1] and Moti Yung[2]

[1] Department of Computer Science and Engineering,
University of Connecticut, Storrs, CT, USA
aggelos@cse.uconn.edu
[2] Department of Computer Science,
Columbia University, New York, NY, USA
moti@cs.columbia.edu

Abstract. Digital Signatures emerge naturally from Public-Key Encryption based on trapdoor permutations, and the "duality" of the two primitives was noted as early as Diffie-Hellman's seminal work. The present work is centered around the crucial observation that two well known cryptographic primitives whose connection has not been noticed so far in the literature enjoy an analogous "duality." The primitives are Group Signature Schemes and Public-Key Traitor Tracing. Based on the observed "duality," we introduce new design methodologies for group signatures that convert a traitor tracing scheme into its "dual" group signature scheme.

Our first methodology applies to generic public-key traitor tracing schemes. We demonstrate its power by applying it to the Boneh-Franklin scheme, and obtaining its "dual" group signature. This scheme is the first provably secure group signature scheme whose signature size is not proportional to the size of the group and is based only on DDH and a random oracle. The existence of such schemes was open. Our second methodology introduces a generic way of turning any group signature scheme with signature size linear in the group size into a group signature scheme with only logarithmic dependency on the group size. To this end it employs the notion of traceability codes (a central component of combinatorial traitor tracing schemes already used in the first such scheme by Chor, Fiat and Naor). We note that our signatures, obtained by generic transformations, are proportional to a bound on the anticipated maximum malicious coalition size. Without the random oracle assumption our schemes give rise to provably secure and efficient Identity Escrow schemes.

1 Introduction.

Drawing equivalences, relationships, and dualities between different and even seemingly unrelated primitives is at the heart of cryptographic research. Such discoveries typically lead to new understanding and novel constructions of the related primitives. For example, digital signatures is an important primitive that is naturally implied by the existence of trapdoor-permutation-based public-key encryption (by associating decryption with signing operation and encryption with verification). This (by now obvious) "duality" (i.e., analogy which allows translation of one scheme to the other) was noted by Diffie and Hellman in their seminal work [DH76]. In this work we look at two primitives whose relationships have not been yet noted in the literature and they are seemingly unrelated (and perhaps somewhat antagonistic): Group Signatures, which is essentially an anonymity management system, and Traitor Tracing, which is a broadcast encryption system for identifying pirates within digital rights management systems. We make the observation that group signature schemes and public-key traitor tracing schemes are "dual" in similar ways to the above mentioned "duality" between regular digital signatures and public-key encryption. This new outlook leads to new design methodologies for group signatures that allow us to answer an open question in the area and build the first provably secure group signature scheme whose size is independent of the size of the group (it depends though on other parameters of the system) and its security is based only on the Decisional Diffie-Hellman assumption (DDH) and a random oracle.

Group signatures were introduced in [CH91]. In such a scheme, a member of a group of users is able to sign messages so that it is not possible to distinguish which member has actually signed. Nevertheless, in the case of a dispute or other extraordinary occasion, the group manager is capable of "opening" the signature and revealing the identity of the group member who has signed. Group signature schemes constitute a very useful primitive in many settings. Additionally, group signature schemes have applications in the context of identity escrow [KP98]. Numerous works improved several aspects of group signatures [CP95,Pet97,CS97,Cam97,AT99,ACJT00,CL01], with the current state of the art represented by the very elegant scheme of [ACJT00].

Group signatures can be categorized into two main classes: those that have signature size linear in the group size (where size is defined in terms of number of encrypted elements) and are based on traditional or generic assumptions, e.g., the scheme of [Cam97] that is based on the DDH, and those that have constant signature size (as a function of the group size) and are based on "more specialized" assumptions, e.g., the scheme of Ateniese et al. [ACJT00] that is based on the strong-RSA assumption as well as the DDH over a group of unknown order. The specialized assumption (combining strong-RSA and DDH) is elegant and ingenious in the way it naturally allows many independent keys based on the same composite modulus. We remark that all these schemes are proved secure in the random-oracle model (their interactive versions however constitute "identity escrow" schemes without the need of a random oracle assumption). Even though, from a signature-size point of view the scheme of Ateniese et al. is optimal (and our goal is not to improve on that achievement), there are still important questions regarding the understanding and design of group signature schemes, specifically the following:

Question. Is it possible to construct a group signature scheme with signature size smaller than the size of the group whose security is based on the DDH over a prime order group?

We believe that the above question has also significant relevance to practice. This is so, since an approach based only on DDH would permit an efficient Elliptic-Curve based group signature; something not possible using the current state-of-the-art constant-size signature scheme.

A main algebraic hurdle towards achieving a positive answer to the above question, and one of the reasons why the existing research resorted to novel intractability assumptions is that "traditional assumptions" (such as the DDH), unlike the combined Strong RSA DDH assumption as noted above, do not possess an inherent property which allows, based on a common group manager key, the establishment of a compact multi-user keying scheme, where keys of different users are independent in some sense.

In this work, we answer the above question in the affirmative, by employing novel design methodologies for constructing group signatures that are based on the "duality" that we have observed between this primitive and the primitive of traitor tracing. Note that the signature size obtained using our first methodology is $O(w^{1+\epsilon}k)$ where w is a cryptographic security parameter, k is a bound on the anticipated maximum malicious coalition size and ϵ depends on the length of the non-interactive zero-knowledge string. This reduces the signature size to be linearly dependent only on the anticipated malicious coalition size which is typically much smaller than the size of the entire group's size (recall that the scheme of [ACJT00] has a signature of size $\mathcal{O}(w)$ where w is a security parameter related to the Strong-RSA problem; this size is optimal up to a constant multiplicative factor).

Traitor-Tracing, introduced by Chor, Fiat and Naor in [CFN94] is based on a "multicast encryption scheme" where a sender can distribute encrypted messages to a group of users. Each user decrypts a ciphertext to get the same message. Users may sell their keys or share them with pirates. In order to deter this practice, traitor tracing enables an authority to trace the identities of users whose keys were used by a pirate decoder (such users are called traitors). A public-key traitor tracing scheme allows the encryption to be done by any interested third party (e.g., any pay-T.V. station) using a public-key issued by the authority. Public-key traitor tracing schemes were presented in [KD98,BF99,NP00,KY02a]. Other works that further enhanced the understanding of traitor tracing schemes are [SW98a,SW98b,NP98,FT99,GSY99,SW00,CFNP00,GSW00,KY01,NNL01], [DF02,DF03]. The "asymmetric" setting where the authority needs to provide a non-repudiable proof of the involvement of a user in piracy was considered in [Pfi96,WHI01,KY02b]. Essentially, in viewing the schemes in the literature, we can distinguish two families: combinatorial (following that of [CFN94]) and algebraic (following that of [KD98,BF99]).

The "duality" (i.e., analogy) we observed between public-key traitor tracing and group signatures is present in several levels (a graphical demonstration of the analogy is given in figure 1):

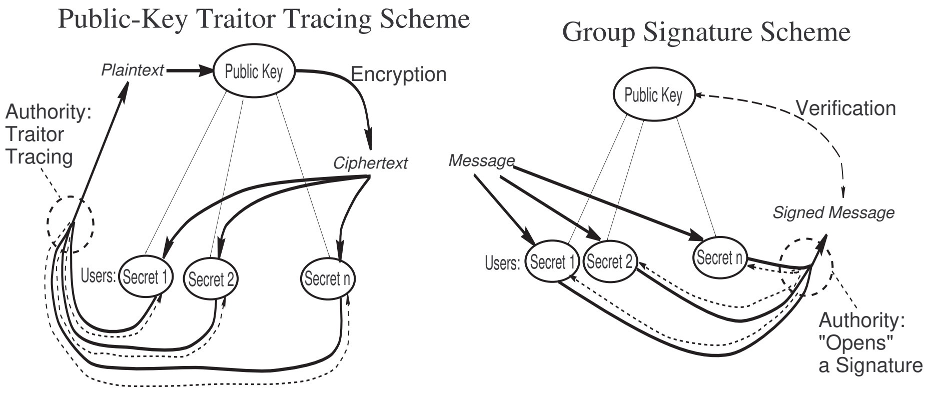

Fig. 1. Schematic Representation of the "Duality" of the two Primitives.

– In both primitives there is an underlying structure of a single public-key (used for encryption or verification, respectively) that corresponds to many different secret-keys (used for decryption or signing, respectively).
– In both primitives an authority should be able to reveal the identity of the decrypting/signing entity (traitor tracing or "opening the signature," respectively). Note that in traitor tracing schemes, a bound on the number of collaborating traitors is imposed, thus we deal with group signatures with similarly bounded coalitions.
– In both primitives it is desirable that the authority (when not trusted) should not be able to frame a user of being a traitor or of producing a signature that it did not sign, respectively.
– Both primitives share common efficiency measures: how large are the public-key size, the secret-key size and the ciphertext/signature-size as a function of the group size.

We note that the individual primitives possess additional specialized (enhanced) properties which are not directly comparable. For example, "black-box tracing" in traitor tracing schemes (which is a stronger notion than the notion of "tracing" which by itself is analogous to "opening" in group signature schemes).

Our results, which exploit the observed "duality," are as follows:

1. First we introduce the new design methodology (Methodology 1) for extracting a group signature from a given public-key traitor tracing scheme.
2. Second, we provide a formal security model for group signatures based on an adversary model, and we apply methodology 1 to the scheme of Boneh-Franklin [BF99] to obtain the corresponding "dual" group signature (or identity escrow) scheme whose properties and assumptions have been discussed above. We discuss the generality of the approach and how to use the methodology with other schemes to get group signatures with enhanced sets of properties.

3. Finally, we present Methodology 2, which is a generic transformation of a group signature scheme with signature/key size linear in the size of the group to one with merely polylog dependency assuming polylog bound on collaborating traitors. (Again, using this generic method, our goal is not to compete with the most efficient schemes available, in terms of signature size). The methodology employs Traceability Codes, the fundamental combinatorial construct that is central in the construction of many traitor tracing schemes. Again, opening a signature is achieved by employing the traceability property.

2 The Group Signature Model

A group signature scheme is made of five procedures $\langle \mathsf{Setup}, \mathsf{Join}, \mathsf{Sign}, \mathsf{Verify}, \mathsf{Open} \rangle$ that are executed by the active participants of the system, which are: (1) the Group Manager (GM), and (2) the users.

Setup (invoked by the GM), is the initialization of the group signature system. For a given security parameter w, this p.p.t. TM (probabilistic polynomial-time Turing Machine) produces a publicly-known string pk (the "public-key" of the system) and some private string sk to be used for user key generation.
Join (a protocol between the GM and the user that introduces the user to the system). The GM employs the private key sk in the course of the protocol, and the user obtains the private key $\mathsf{sk}_{\mathcal{U}}$ that is needed in order to issue valid digital signatures.
Sign (invoked by the user). A p.p.t. TM that given the private key $\mathsf{sk}_{\mathcal{U}}$ of a user and a message, generates a signature on the given message.
Verify (invoked by any recipient). A deterministic polynomial-time TM that verifies the validity of a signature generated by Sign, given the public-key information pk.
Open (invoked by the GM) A p.p.t. TM that given a signature, and the internal key-generation string sk and all transcripts of the Join protocols, outputs the identity of the signer $\mathcal{U}$.

Definition 1. (Correctness for group signature schemes) *A group signature scheme with security parameter w is* **correct** *if the following two conditions are satisfied (with overwhelming probability in w):*

(1) **Sign-Correctness:** *Suppose $\mathcal{U}$ is a user of the system that obtained $\mathsf{sk}_{\mathcal{U}}$ from the* Join *protocol. Then it should hold that for all M,* $\mathsf{Verify}(M, \mathsf{pk}, \mathsf{Sign}_{\mathcal{U}}(M)) = \mathsf{true}$.

(2) **Open-Correctness:** *Suppose $\mathcal{U}$ is a user of the system that obtained $\mathsf{sk}_{\mathcal{U}}$ from the* Join *protocol. Then it should that for all M, if* $\mathsf{Verify}(M, \mathsf{pk}, \mathsf{Sign}_{\mathcal{U}}(M)) = \mathsf{true}$ *then* $\mathsf{Open}(\mathsf{sk}, \mathsf{Sign}_{\mathcal{U}}(M)) = \mathcal{U}$.

2.1 Security

In this section we give the formal specifications for a secure group signature. This has some interest in its own right since in most of the previous work on group signatures such formal security model is omitted. An exception is [CL01], where an "ideal world" security model is introduced and employed. In the present work we opt for a direct model where an adversary is postulated by its capabilities and goals.

In particular, we will employ a signature adversarial model that is stronger than the one typically employed in the context of digital signatures, which we call an "adaptive chosen message attacker with user-selection capability": i.e., an adversary prior to launching an attack against a certain security property is allowed to obtain signatures of messages of his choice from group members *he also selects* (note that for the purpose of the security definition, we assume that the set of group members is publicly known and identifiable). Further, we allow this to be done in an *adaptive* manner. User selection capability is more appropriate in the formulation of security in the context of group signatures since, in reality, an adversary might be capable of actually obtaining some signatures from group members he knows. An interesting feature of this formulation is the following: against an adversary with user selection capability the notions of anonymity (hardness of extracting the identity of a signer) and unlinkability (hardness of linking signatures of the same signer) collapse.

Definition 2. (Secure group signature scheme) *A group signature scheme with security parameter w is* **secure***, if it satisfies the following conditions:*

(1) **Unforgeability:** *Let $\mathcal{M}$ be a p.p.t. TM that given the public-key* pk *of the system is allowed to adaptively select tuples of the form $\langle \mathcal{U}, M\rangle$ and obtain* $\mathsf{sign}_{\mathcal{U}}(M)$*. We say that a group signature scheme is unforgeable (under adaptive chosen message attacks with user selection) if the probability that $\mathcal{M}$ outputs a tuple $\langle M, s\rangle$ such that* $\mathsf{Verify}(M, \mathsf{pk}, s) = \mathsf{true}$ *is negligible in w.*

(2) **Anonymity/Unlinkability:** *Let $\mathcal{M}$ be a p.p.t. TM that given the public-key* pk *of the system is allowed to adaptively select tuples of the form $\langle \mathcal{U}, M\rangle$ and obtain the corresponding signature* $\mathsf{sign}_{\mathcal{U}_t}(M)$*; additionally, $\mathcal{M}$ is allowed to invoke the Join procedure a certain number of times k to introduce new users in the system; let $\mathcal{U}_1, \mathcal{U}_2$ be two users of the system which were not introduced by the adversary; finally, $\mathcal{M}$ is required to submit a final message M and receive* $\mathsf{sign}_{\mathcal{U}_b}(M)$ *where $b \in_U \{1, 2\}$. We say that a group- signature scheme satisfies anonymity/unlinkability (under adaptive chosen message attacks with user selection) if the success probability of $\mathcal{M}$ in predicting b differs from $1/2$ by a fraction that is at most negligible in w.*

(3) **Coalition-Resistance/Traceability:** *Let $\mathcal{M}$ be a p.p.t. TM that given the public-key* pk *of the system it is allowed to invoke the* Join *procedure a number of times k to introduce the users $\mathcal{U}_1, \ldots, \mathcal{U}_k$ in the system. Additionally $\mathcal{M}$ is allowed to submit tuples of the form $\langle \mathcal{U}, M\rangle$, and obtain the corresponding* $\mathsf{sign}_{\mathcal{U}}(M)$*. We say that a group signature scheme satisfies coalition-resistance/traceability if the probability that $\mathcal{M}$ outputs a tuple $\langle M, s\rangle$ with the property* $\mathsf{Verify}(M, \mathsf{pk}, s) = \mathsf{true}$ *and* $\mathsf{Open}(\mathsf{sk}, s) \notin \{\mathcal{U}_1, \ldots, \mathcal{U}_k\}$ *is negligible in w.*

We remark that in the above formulation the GM is trusted. When this is not the case, an additional property needs to be satisfied that is called "exculpability", see section 4.6. In the exculpability setting it is of interest to strengthen the Open procedure to also provide a *proof* of correct opening.

3 Methodology #1: Group Signatures from Public-Key Traitor-Tracing Schemes

3.1 Description of a Public-Key Traitor Tracing Scheme

A PK-traitor-tracing scheme involves three types of participants: the authority, the users of the system, and the senders that utilize the public encryption function to transmit encrypted messages to the users. The authority is responsible for initializing the system and distributing the decryption keys, and then the users of the system are capable of inverting ciphertexts that are encrypted under the public-key of the group. The scheme is comprised of five basic procedures. To serve the purpose of our work, we formalize them below so that they are as close as possible to the group signature definition.

Setup (invoked by the authority) is the initialization of the system. For a given security parameter w, this p.p.t. TM produces a publicly-known string pk (the "public-key" of the system) and some internal string sk to be used for user key generation.
Join (a protocol between the authority and the user that introduces the user to the system). The authority employs the private key sk in the course of the protocol, and the user obtains the private key $\mathsf{sk}_{\mathcal{U}}$ that will be employed for decryption.
Encrypt (invoked by the sender). A p.p.t. TM that given a message M and the public-key pk, produces an encryption of M.
Decrypt (invoked by the user). A deterministic TM that given the private key $\mathsf{sk}_{\mathcal{U}}$ of a user and some ciphertext, returns the corresponding plaintext.
Tracing. An algorithm that given the contents of a pirate-decoder and secret-key information sk, returns a set of identities. (It is used by the authority to reveal identities of users that participated in the construction of the decoder by leaking their keys.)

It is also desirable to support *Revocation/Suspension* where the authority is able to toggle a user's capability to decrypt messages in an efficient way, and *Black-Box Traitor Tracing* which suggests that the traitor tracing procedure can be executed with merely black-box access to the pirate-decoder.

3.2 Design Methodologies for Group Signature Schemes

In the literature we can identify two basic design methodologies for group signature schemes. In the first one, introduced by Chaum and van Heyst in [CH91], each user's signature is essentially a proof of knowledge of a public commitment. Schemes in this category, typically designed using OR-proofs (e.g., [Cam97]) produce signatures whose size is linear in the number of participants.

The second design methodology, put forth by Camenisch and Stadler [CS97], is based on two "layers" of regular signature schemes. It has the potential of allowing constant key-sizes (that are independent of the size of the group) and constant signature sizes. In this approach each signature is essentially a semantically secure encryption of a valid certificate that is issued to the user by the GM. The shortcoming of the approach is that one has to devise an efficient way of proving that the ciphertext actually contains a valid certificate. It is possible of course to employ generic zero-knowledge proofs but these are not suitable for a practical implementation (in this case we merely have a plausibility argument). Camenisch and Stadler [CS97] discussed this issue and described a specific efficient instantiation of the above model that employed several intractability assumptions (some customized to their design). Subsequent work on group signatures [CM99,ACJT00] was based on the same design principle, and employed the Strong-RSA assumption as well as the DDH over a group of unknown oder to prove security. At present it is not known, and in fact seems quite unlikely, that it is possible to employ this design methodology to produce an efficient group-signature with security based on a cryptographic assumption such as the DDH (or the regular RSA assumption).

3.3 The New Design Methodology based on PK-Traitor Tracing

Our new design methodology can be summarized by the following crucial observation: Given a public-key traitor tracing scheme comprised of the procedures $\langle \mathsf{Setup}, \mathsf{Join}, \mathsf{Encrypt}, \mathsf{Decrypt}, \mathsf{Tracing} \rangle$ we design a group signature scheme $\langle \mathsf{Setup}, \mathsf{Join}, \mathsf{Sign}, \mathsf{Verify}, \mathsf{Open} \rangle$ as follows: we keep the procedures Setup and Join identical to the case of the pk-traitor tracing scheme. Let pk be the public-key information of the scheme as generated by the Setup procedure. Now we need to translate the property of "the ability to trace a box" into an "ability to open a signature" and in some sense lift the key from the user's box into the signature. To this end, assume that we can construct a variant of *non-interactive proof of knowledge* PK that given a secret key of a user $\mathsf{sk}_{\mathcal{U}}$ shows: (i) the prover is capable of inverting the public-key pk; (ii) the key $\mathsf{sk}_{\mathcal{U}}$ that is used in the proof is recoverable by the GM from the transcript.

Observe that property (i) is the standard method that is used to transform public-key encryption into a digital signature. Property (ii) is intrinsic to the setting of group signatures: it convinces the verifier that the prover has embedded sufficient information into the signature so that the GM will be able to recover the key used by the signer. Now observe that the Open procedure for the derived group signature scheme is implemented by employing the traitor tracing algorithm Tracing on the recovered key from a signature.

We will call the derived group signature, the "dual" of the public-key traitor tracing scheme. We remark that the derived group signature will inherit the same collusion-resistance/traceability of the parent public-key traitor tracing scheme, and additionally it will inherit directly properties such as revocation and suspension.

Generality of our Approach. Note that a proof of knowledge achieving properties (i)-(ii) above can be achieved using generic zero-knowledge proofs. As a result it is possible to obtain a correct and secure group signature according to definitions 1, 2 for any given pk-traitor-tracing scheme. Due to lack of space we omit a formal description of this result which constitutes only a plausibility result. Instead, we will focus on specific pk-traitor-tracing schemes where such proofs can be achieved efficiently.

4 The Group Signature "Dual" of the Boneh-Franklin Scheme

4.1 Preliminaries

Assume a large multiplicative cyclic group $\mathcal{G}$ of prime order over which DDH is assumed to be hard. For example $\mathcal{G}$ can be the subgroup of order q of $\mathbf{Z}_p^*$, where $q \mid p-1$ and p, q are large primes. In the following g will denote a generator of $\mathcal{G}$. Note that arithmetic in the exponents is performed in the finite field $\mathbf{Z}_q$.

4.2 Discrete-Log Representations

Let $h_0, h_1, \ldots, h_v$ be random elements of $\mathcal{G}$ so that $h_j := g^{r_j}$ for $j = 0, \ldots, v$. For a certain element $y := g^b$ of $\mathcal{G}$ a representation of y with respect to the base $h_0, \ldots, h_v$ is a $(v+1)$-vector $\boldsymbol{\delta} := \langle \delta_0, \ldots, \delta_v \rangle$ such that $y = h_0^{r_0} \ldots h_v^{r_v}$, or equivalently $\boldsymbol{\delta} \cdot \boldsymbol{r} = b$ where $\cdot$ denotes the inner product between two vectors. It is easy to see that obtaining representations of a given y w.r.t. some base $h_0, \ldots, h_v$ is as hard as the discrete-log problem over $\mathcal{G}$. Furthermore, it was shown in [BF99] that if some adversary is given m representations of some y with respect to some base, with $m < v$ then any additional representation that can be obtained has to be a "convex combination" of the given representations (a convex combination of the vectors $\boldsymbol{\delta_1}, \ldots \boldsymbol{\delta_m}$ is a vector $\sum_{\ell=1}^{m} \mu_\ell \boldsymbol{\delta_\ell}$ with $\sum_{\ell=1}^{m} \mu_\ell = 1$):

Proposition 1. *[BF99] if there is an algorithm that given $y, h_0, \ldots, h_v$ and $m < v$ representations of y denoted by $\boldsymbol{\delta_1}, \ldots \boldsymbol{\delta_m}$, it computes a representation of y that is not a convex combination of $\boldsymbol{\delta_1}, \ldots \boldsymbol{\delta_m}$ then the discrete-log problem over $\mathcal{G}$ is solvable.*

4.3 Description of the PK-Traitor Tracing Scheme of [BF99]

In the [BF99] public-key traitor tracing scheme each user obtains a carefully designed discrete-log representation that can be used for decryption. Any bounded coalition of malicious users (traitors) can only produce alternative keys (discrete-log representations) that have a specific structure (according to proposition 1); further, given the initial design of user keys it is actually possible to recover the identities of the traitors, as long as they form coalitions of size at most $v/2$.

4.4 Proof of Knowledge of a Recoverable Discrete-Log Representation

At the heart of the transformation of a pk-traitor tracing scheme into its "dual" group signature is a proof of knowledge that allows a user to show in zero-knowledge that he is capable of inverting the public-key of the traitor tracing scheme, and at the same time this proof includes sufficient information to reveal the user-key during the opening procedure. In the case of the Boneh-Franklin scheme the required tool is a *proof of knowledge of a recoverable discrete-log representation*. A proof of knowledge of a recoverable representation convinces the verifier that the prover possesses a discrete-log representation, but in addition it shows that the GM can *recover* such representation if necessary.

We remark that the opening of a signature can be separated from the GM if desired and can even become a distributed task to allow for robustness and security against some malicious authorities. In our exposition below let $enc : \mathcal{R} \times \mathcal{P} \to \mathcal{D}$ be a public probabilistic encryption function, with $\mathcal{R}$ the randomness space, and $\mathcal{P}$ the plaintext space; note that we assume that $\mathbf{Z}_q$ can be embedded into $\mathcal{P}$; *enc* can be initialized by the GM or a designated set of "audit" authorities that will be responsible for opening signatures. Let us denote the decryption function by *dec*; note that decryption should be robust (i.e., done with a proof of correctness). The encryption function can be any semantically secure scheme, e.g., ElGamal encryption (which is semantically secure under the DDH assumption).

The proof of knowledge of a recoverable discrete-log representation is presented in figure 2. This proof of knowledge is a generalization of proof techniques employed by [YY99,CD00,YY01].

Theorem 1. *The 3-round proof of knowledge presented in figure 2 satisfies (i) completeness, (ii) soundness with cheating probability 2^{-l}, (iii) honest verifier zero knowledge, provided that the encryption function enc is semantically secure.*

Using the Fiat-Shamir Heuristics [FS87], we can turn the proof of knowledge of figure 2 into a signature: the challenge c computed as $\mathcal{H}(m||a_1||\ldots||a_l||C^{(0)}_{1,0}|| C^{(1)}_{1,0}||\ldots C^{(0)}_{l,v}||C^{(1)}_{l,v})$ where m is the message. When the challenge is defined as above, the tuple $\langle \{a_i\}_i, \{C^{(b)}_{i,j}\}_{b,i,j}, c, \{\langle s_{i,j}, \rho_{i,j}\rangle\}_{i,j}\rangle$ will be denoted as $\mathsf{SIG}^{enc}(\delta_0, \ldots, \delta_v : h_0^{\delta_0} \ldots h_v^{\delta_v} = y)(m)$.

Recovering a Representation. It is easy to see that a signature $\mathsf{SIG}^{enc}(\delta_0, \ldots, \delta_v : h_0^{\delta_0} \ldots h_v^{\delta_v} = y)(m)$ yields a representation $\boldsymbol{\delta} := \langle \delta_0, \ldots, \delta_v\rangle$ to the entity who is capable of inverting *enc*. Indeed, recovering can be done by finding an $i \in \{1, \ldots, l\}$, such that the vector

$$\left\langle dec(C^{(0)}_{i,0}) - dec(C^{(1)}_{i,0})(\mathrm{mod} q), \ldots, dec(C^{(0)}_{i,v}) - dec(C^{(1)}_{i,v})(\mathrm{mod} q)\right\rangle$$

is a representation of y w.r.t. the base $h_0, \ldots, h_v$. For a valid proof such an i will exist with probability $1 - 2^{-l}$. Finally it is easy to see that,

Prover		Verifier
$\boldsymbol{\delta} := \langle \delta_0, \ldots, \delta_v \rangle$		
for $i = 1, \ldots, l$		
$r_{i,0}, r_{i,1}, \ldots, r_{i,v} \in_R \mathbf{Z}_q$		
$a_i := h_0^{r_{i,0}} \ldots h_v^{r_{i,v}}$		
$\langle \rho_{i,0}^{(0)} \rho_{i,0}^{(1)} \rangle, \ldots, \langle \rho_{i,v}^{(0)} \rho_{i,v}^{(1)} \rangle \in_R \mathcal{R}^2$		
$C_{i,j}^{(0)} := enc(\rho_{i,j}^{(0)}, r_{i,j})$ for $j \in \{0, \ldots, v\}$		
$C_{i,j}^{(1)} := enc(\rho_{i,j}^{(1)}, r_{i,j} - \delta_j)$ for $j \in \{0, \ldots, v\}$	$\xrightarrow{\{a_i\}_i, \{C_{i,j}^{(b)}\}_{b,i,j}}$	
let $c[i]$ be the i-th bit of c	$\xleftarrow{c}$	$c \in_R \{0,1\}^l$
for $i = 1, \ldots, l$, $j = 0, \ldots, v$		
$s_{i,j} := r_{i,j} - c[i]\delta_j (\bmod q)$		
$\rho_{i,j} := \rho_{i,j}^{(c[i])}$	$\xrightarrow{\{\langle s_{i,j}, \rho_{i,j} \rangle\}_{i,j}}$	Verify for $i = 1, \ldots, l$, $j = 0, \ldots, v$
		$enc(\rho_{i,j}, s_{i,j}) \stackrel{?}{=} C_{i,j}^{(c[i])}$
		$h_0^{s_{i,0}} \ldots h_v^{s_{i,v}} \stackrel{?}{=} a_i / y^{c[i]}$

Fig. 2. Proving the knowledge of a representation $\boldsymbol{\delta}$ of y w.r.t. $h_0, \ldots, h_v$ and at the same time the fact that $\boldsymbol{\delta}$ is recoverable by anyone who can invert *enc*.

Proposition 2. *The signature* $\mathsf{SIG}^{enc}(\delta_0, \ldots, \delta_v : h_0^{\delta_0} \ldots h_v^{\delta_v} = y)(m)$*, has length* $\mathcal{O}(lv)$ *ciphertexts of enc* ($l = w^{\epsilon}$*, where* $w := \log \#\mathcal{G}$*).*

Using the signature SIG^{enc}, as we will see, it is possible to design a group signature scheme (and then the security will be based on the random oracle model). If the proof of knowledge is treated as an interactive protocol then using the same methodology we present for group signatures one can design a secure identity escrow scheme [KP98] (and then security does not require the employment of a random oracle); we omit details.

4.5 The "Dual" Group Signature Scheme

The Setup and Join procedures are identical to the ones used in the Boneh-Franklin scheme [BF99]. Additionally the GM (or the set of designated authorities) publish the encryption function *enc*, as specified in section 4.4.

- Sign. Given a message M, a user that possesses a representation $\langle \delta_1, \ldots, \delta_{2k} \rangle$ of y w.r.t. $h_1, \ldots, h_{2k}$, publishes the signature $\mathsf{SIG}^{enc}(\delta_1, \ldots, \delta_{2k} : h_1^{\delta_1} \ldots h_{2k}^{\delta_{2k}} = y)(M)$.
- Verify. Given a signature $\mathsf{SIG}^{enc}(\delta_1, \ldots, \delta_{2k} : h_1^{\delta_1} \ldots h_{2k}^{\delta_{2k}} = y)(M)$ it can be verified as described in figure 2 using the public-key $y, h_1, \ldots, h_{2k}$ and the public encryption function *enc*.
- Open. GM recovers the discrete-log representation of a signature $\mathsf{SIG}^{enc}(\delta_1, \ldots, \delta_{2k} : h_1^{\delta_1} \ldots h_{2k}^{\delta_{2k}} = y)(M)$ as described in section 4.4. Subsequently, it employs the traitor tracing algorithm of [BF99] to recover the identities of the users that collaborated in the construction of the signature.

The correctness of the group signature scheme (as in definition 1) follows easily from the description above.

Parameters. The group signature scheme uses the following parameters: the parameter k which is the maximum traitor collusion size, the parameter l which is a security parameter that relates to the probability of producing a fraudulent group signature that passes the verification step (the cheating probability is 2^{-l}), and finally the parameter n which is the number of users. It is immediate from proposition 2 that the size of a signature is $\mathcal{O}(kl)$ ciphertexts/ group elements (there is no direct dependency on n).

Efficiency. Observe that the size of the public-key of the system is $\mathcal{O}(k)$, the size of a signature is $\mathcal{O}(kl)$, and the size of each user key is $\mathcal{O}(1)$. As noted above this size depends on the size of the maximal coalition (and is related to the same property in traitor tracing schemes); the scheme in [ACJT00] has constant size $\mathcal{O}(1)$ independent of the size of the coalition.

Security Based on the properties of the proof of knowledge of figure 2 one can show the following:

Theorem 2. *Assuming the DDH, and that the underlying encryption function enc is semantically secure, the "dual" group signature of the [BF99]-scheme is unforgeable and satisfies anonymity/ unlinkability (as in definition 2) in the random oracle model.*

Coalition-Resistance/Traceability. The coalition-resistance and traceability of our group signature relies on on proposition 1 and the recoverability properties of the signature proof of knowledge of figure 2.

First, proposition 1 suggests that under the assumption that the discrete-logarithm problem is hard, the discrete-log representations that can be obtained by an adversary controlling a number of users up to k, is only vector convex combinations of the users' discrete-log representations.

Second, the recoverability properties of the signature proof of knowledge of figure 2 suggest that every signature will reveal to the GM some representation that was used in the generation of the signature.

By combining these two facts we argue that any set of up to k malicious users that attempt to conceal their identity collectively by avoiding the use of their assigned representation(s), can only use convex combinations of the representations that are available to them. But in this case it is possible to recover *all* their identities by running the traitor tracing procedure of Boneh and Franklin [BF99].

4.6 Adding Exculpability

Exculpability (against the group manager) suggests that the GM cannot frame an innocent user, blaming him of signing a message he did not sign. In this case, during the Join protocol the GM does not get to know the whole secret-key $\mathsf{sk}_{\mathcal{U}}$ of the user which is generated in a joint manner. Instead, it obtains some commitment to this key $\mathsf{com}_{\mathcal{U}}$ which is signed by the user with some independent digital signature mechanism. Observe that in this case the Opening procedure produces $\mathsf{com}_{\mathcal{U}}$ as an output (which allows the GM to implicate the signing user presenting a non-repudiable evidence). In the context of the formal security model, we have:

Definition 3. Exculpability: *Let $\mathcal{M}$ be a p.p.t. TM that is allowed to initialize the group signature scheme by running the* Setup *procedure (possibly modified), and then execute the* Join *protocol for a user $\mathcal{U}$ (also possibly modified). Then, $\mathcal{M}$ is allowed to submit messages of the form M and obtain the corresponding signature $\mathsf{Sign}_{\mathcal{U}}(M)$ in an adaptive manner. We say that the group signature scheme satisfies "online" exculpability if the distribution of signatures $\langle M, s\rangle \leftarrow \mathcal{M}$ and the distribution of signatures for any valid user are* distinguishable *given a witness that is in the possession of the user. On the other hand, a group signature scheme satisfies "offline" exculpability if the probability that $\mathcal{M}$ outputs a tuple $\langle M, s\rangle$ with the property* $\mathsf{Verify}(M, \mathsf{pk}, s) = \mathsf{true}$ *and* $\mathsf{Open}(\mathsf{sk}, s) = \mathsf{com}_{\mathcal{U}}$ *is negligible in w.*

Interestingly, constructing group signatures from "asymmetric" pk-traitor-tracing schemes does not preserve exculpability ("asymmetry" is the "dual" property of expulpability in the context of traitor tracing schemes, see e.g., [Pfi96]). Indeed, as dictated by the design methodology, opening a signature in the derived scheme will reveal the key (which is essential for employing the tracing algorithm to satisfy coalition-resistance/traceability), and therefore will allow the GM to frame a user if it is malicious (even if the traitor tracing scheme used as a basis is asymmetric).

Nevertheless, it is possible to achieve exculpability by using the modular strategy explained below. Note that the [BF99]-scheme is not asymmetric and as a result it cannot be employed for exculpability. However, there exist asymmetric public-key traitor tracing schemes based on discrete-log representations which can be turned into group signatures in the exact way as the [BF99]-scheme, using proofs of recoverable representations. The only presently known such scheme is the scheme of [KY02a] which we will now employ. Exculpability can be achieved as follows: the GM will initialize two instantiations of the traitor tracing scheme above, and each user will join both. Signing will be performed by issuing two signatures, the first one using the first scheme is based on a proof of a recoverable representation (as in figure 2), and the second one using the second scheme is based on a standard proof of knowledge of a discrete-log representation (that is non-recoverable). Now observe that GM never learns the key of the second scheme. Thus, any signature that a malicious GM can generate is distinguishable from true valid signatures of a certain user given the secret-key of the user for the second instantiation. This corresponds to online exculpability: the user can always deny his participation in a certain signature based on the second component. We remark that previous schemes, e.g., [ACJT00], achieved offline exculpability.

5 Methodology # 2: Group Signatures based on Traceability Codes

In this section we present a generic transformation from any group signature scheme with linear (in the user population) signature and public-key size to a group signature scheme with only logarithmic such dependencies on the user population size (the scheme will depend on other parameters as well). The construction relies on traceability codes, the fundamental combinatorial construct which was invented by Chor, Fiat and Naor [CFN94] to introduce traitor tracing schemes and was later formalized by Staddon et al. [SSW00].

5.1 Traceability Codes

We start with some notational conventions about strings over some alphabet Σ. The j-th symbol of a string s over Σ will be denoted by $s[j]$; it follows that if $|s| = l$, $s = s[1]||\ldots||s[l]$ and that $s[j] \in \Sigma$ for all $j = 1,\ldots,l$. If $I = \{i_1,\ldots,i_k\} \subseteq \{1,\ldots,l\}$ then given a string s of length l, $s|_I := s[i_1]||s[i_2]||\ldots||s[i_k]$. Given $s, s' \in \Sigma^l$ define $EQ(s,s') := \{i \mid s[i] = s'[i]\}$. An $\langle r,l\rangle_q$-code over an alphabet Σ with $|\Sigma| = q$, is a set of strings $\mathcal{C} = \{s_1,\ldots,s_r\} \subseteq \Sigma^l$. Given a code $\mathcal{C}$, and some subset $C \subseteq \mathcal{C}$, the descendant set $\mathsf{desc}(C)$ of the subset $C := \{s_{i_1},\ldots,s_{i_k}\}$ is the set of strings $\{s \in \Sigma^l \mid s[j] \in \{s_{i_1}[j],\ldots,s_{i_k}[j]\} \quad j = 1,\ldots,l\}$. A traceability (TA) $\langle n,L\rangle_q$-code $\mathcal{C}$ for collusions up to k satisfies the following "traceability" property: for all $x \in \mathsf{desc}(C)$ where $C \subseteq \mathcal{C}$ with $|C| \le k$ it holds that $\exists y \in C \;\; \forall z \in (\mathcal{C} - C) \quad |EQ(x,y)| > |EQ(x,z)|$. Observe that due to the traceability property, given $x \in \mathsf{desc}(C)$ one can recover an element of C as follows (as long as $|C| \le k$): for all $y \in \mathcal{C}$ one computes $|EQ(x,y)|$ and pronounces the string y with maximum $|EQ(x,y)|$ to be an element of C.

Traceability codes can be constructed using a probabilistic argument, as in [CFN94], or using a concrete construction based on error-correcting codes with large minimum distance, [SSW00].

The probabilistic construction of [CFN94] yields a traceability code over an alphabet size of $2k^2$ that has length $\mathcal{O}(k^2 \log n)$. The concrete constructions of [SSW00,SSW01] are slightly worse in terms of efficiency, but offer superior traceability both in terms of efficiency (the tracing algorithm is not required to take time proportional to $|\mathcal{C}|$), and effectiveness (given an $x \in \mathsf{desc}(C)$ more than one members of C can be identified).

5.2 The Generic Transformation

Let $G := \langle \mathsf{Setup}, \mathsf{Join}, \mathsf{Sign}, \mathsf{Verify}, \mathsf{Open}\rangle$ be a group signature with signature and public-key size linear in the size of the group and user-key constant (e.g., [Cam97]) that satisfies the security properties of definition 2. Below we describe the derived group signature scheme:

Setup. A TA $\langle n,v\rangle_q$-code $\mathcal{C} := \{s_1,\ldots,s_n\}$ for collusions up to k is constructed (as described in section 5.1). The GM runs v independent instantiations of Setup

to produce the public-keys $\mathsf{pk}_1, \ldots, \mathsf{pk}_v$ and the corresponding secret-key information $\mathsf{sk}_1, \ldots, \mathsf{sk}_v$. Then the GM executes the Join protocol of the ℓ-th instantiation of G, q times and records all keys (denoted by $\mathsf{sk}_{\ell,j}$, $j = 1, \ldots, q$). Observe that due to the properties of G, a signature for the ℓ-th instantiation of G will have length q.

Join. The i-th user of the system is assigned the s_i codeword from the TA code (publicly), he privately obtains from the GM the secret-keys $\mathsf{sk}_{\ell,s_i[\ell]}$ for $\ell = 1, \ldots, v$.

Sign. The i-th user signs a message by employing the signing algorithm Sign for each one of the v instantiations of G onto its sequence of signing keys $\langle \mathsf{sk}_{i,s_i[1]}, \ldots, \mathsf{sk}_{i,s_i[v]} \rangle$.

Verify. The verification step requires the verification of each one of the underlying signatures (v executions of the Verify procedure of G, using the corresponding public-keys $\mathsf{pk}_1, \ldots, \mathsf{pk}_v$).

Open. Given a signature $\langle \sigma_1, \ldots, \sigma_v \rangle$, the GM opens each signature using Open of G and reveals the key used: $\mathsf{sk}_{\ell,a_\ell}$, where $a_\ell \in \{1, \ldots, q\}$, $\ell = 1, \ldots, v$. Comparing the recovered keys to the ones that were generated during Setup, the string $a := a_1 \ldots a_v \in \{1, \ldots, q\}^v$ is formed. Observe now that due to the security properties of G, it follows easily that $a \in \mathsf{desc}(C)$ where C is the set of codewords assigned to the group of users that collaborated in forming the signature $\langle \sigma_1, \ldots, \sigma_v \rangle$ (C is a singleton when users do not form malicious coalitions). Using the traceability property of $\mathcal{C}$ it follows that at least one member of C can be identified.

Theorem 3. *The derived group signature scheme based on the secure group signature G and a TA $\langle n, v \rangle_q$-code $\mathcal{C}$ for collusions up to k, satisfies the following properties:*

1. *it satisfies unforgeability.*
2. *it satisfies anonymity/unlinkability.*
3. *it satisfies coalition-resistance/traceability provided that at most k members cooperate to form a signature.*
4. *it has signature size $\mathcal{O}(vq)$, public-key size $\mathcal{O}(vq)$, and user-key size $\mathcal{O}(v)$.*

Example. Using the probabilistic construction of [CFN94] in conjunction with the group signature of [Cam97], one obtains a group signature with public-key and signature size $\mathcal{O}(k^4 \log n)$, and user-key size $\mathcal{O}(k^2 \log n)$, with security based on the DDH over a group of prime order.

Adding Exculpability. Suppose that the underlying group signature scheme also satisfies exculpability (defined in section 4.6). Even though the resulting scheme cannot satisfy the same form of exculpability, we can achieve a threshold variant that depends on a set of servers $S_1, \ldots, S_v$.

The scheme is modified as follows: the GM executes the Join procedure of the ℓ-instantiation of G with server S_ℓ, q times. This step results in server S_ℓ obtaining the signing keys $\mathsf{sk}_{\ell,1}, \ldots, \mathsf{sk}_{\ell,q}$, and the GM obtaining the commitments $\mathsf{com}_{\ell,1}, \ldots, \mathsf{com}_{\ell,q}$, which are signed by the GM and published. When a user wants to join the system he contacts all servers $S_1, \ldots, S_v$ and obtains from server S_ℓ the secret-key $\mathsf{sk}_{\ell,s_i[\ell]}$ for $\ell = 1, \ldots, v$ (in a private manner). The scheme is otherwise as above. One can show that exculpability will be satisfied provided that at least one of the servers $S_1, \ldots, S_v$ does not conspire to frame a user (since one server does not conspire it is impossible for the remaining malicious entities to generate any valid signature in the scheme above).

A subtlety of the above modification for achieving exculpability is that it assumes that $S_1, \ldots, S_v$ honestly keep the correct correspondence between the published traceability code and the keys they provide to each user. This can be relaxed in a generic fashion by assuming that each server is replicated e times and each set of servers $S_1^j, \ldots, S_v^j$, for $j = 1, \ldots, e$ executes an independent instantiation of the derived scheme (for the same traceability code). This method will increase the efficiency measures of the scheme by a factor e. In this setting we only need to assume that the dishonest sets of servers are below $e/2$. More details will be given in the full version.

References

[ACJT00] Giuseppe Ateniese, Jan Camenisch, Marc Joye and Gene Tsudik, *A Practical and Provably Secure Coalition-Resistant Group Signature Scheme*, In Mihir Bellare (Ed.): Advances in Cryptology - CRYPTO 2000, 20th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 20-24, 2000, LNCS 1880, Springer 2000, pp. 255–270.

[AT99] Giuseppe Ateniese and Gene Tsudik, *Some Open Issues and New Directions in Group Signatures*, In Matthew K. Franklin (Ed.): Financial Cryptography, Third International Conference, Springer LNCS Vol. 1648, pp. 196–211.

[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, CA, USA, August 15-19, 1999, LNCS 1666, Springer 1999, pp. 338–353.

[Cam97] Jan Camenisch, *Efficient and Generalized Group Signatures*, In Walter Fumy (Ed.): Advances in Cryptology - EUROCRYPT '97, Konstanz, Germany, May 11-15, 1997, LNCS 1233, Springer 1997, pp. 465–479.

[CD00] Jan Camenisch and Ivan Dåmgard, *Verifiable Encryption, Group Encryption and their Applications to Group Signatures, and Signature Sharing Schemes*, In Tatsuaki Okamoto (Ed.): Advances in Cryptology - ASIACRYPT 2000, Kyoto, Japan, December 3-7, 2000, LNCS 1976, Springer 2000, pp. 331-345.

[CL01] Jan Camenisch and Anna Lysyanskaya, *An Identity Escrow Scheme with Appointed Verifiers*, In Joe Kilian (Ed.): Advances in Cryptology - CRYPTO 2001, 21st Annual International Cryptology Conference, Santa Barbara, CA, USA, August 19-23, 2001, LNCS 2139, Springer 2001, pp. 388-407.

[CM99] Jan Camenisch and Markus Michels, *Separability and Efficiency for Generic Group Signature Schemes*, In Michael J. Wiener (Ed.): Advances in Cryptology - CRYPTO '99, 19th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 15-19, 1999, LNCS 1666, Springer 1999, pp. 413–430.

[CS97] Jan Camenisch and Markus Stadler, *Efficient Group Signature Schemes for Large Groups (Extended Abstract)*, In Burton S. Kaliski Jr. (Ed.): Advances in Cryptology - CRYPTO '97, 17th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 17-21, 1997, LNCS 1294, Springer 1997, pp. 410–424.

[CH91] David Chaum and Eugene van Heyst, *Group Signatures*, In Donald W. Davies (Ed.): Advances in Cryptology - EUROCRYPT '91, Brighton, UK, April 8-11, 1991, LNCS 547, Springer 1991,pp. 257-265.

[CP95] Lidong Chen and Torben P. Pedersen, *On the Efficiency of Group Signatures Providing Information-Theoretic Anonymity*, In Louis C. Guillou, Jean-Jacques Quisquater (Eds.): Advances in Cryptology - EUROCRYPT '95, Saint-Malo, France, May 21-25, 1995, LNCS 921, Springer 1995, pp. 39-49.

[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, CA, USA, August 21-25, 1994, LNCS 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, no. 3, pp. 893-910, 2000.

[DH76] Whitfield Diffie and Martin Hellman, *New Directions in Cryptography*, IEEE Transactions on Information Theory IT-22 (6): pp. 644–654, 1976.

[DF02] Yevgeniy Dodis and Nelly Fazio, *Public Key Broadcast Encryption for Stateless Receivers*, 2002 ACM Workshop on Security and Privacy in Digital Rights Management, to appear in Springer LNCS, 2003.

[DF03] Yevgeniy Dodis and Nelly Fazio, *Public Key Trace and Revoke Scheme Secure against Adaptive Chosen Ciphertext Attack*, In Yvo Desmedt (Ed.): Public Key Cryptography - PKC 2003, 6th International Workshop on Theory and Practice in Public Key Cryptography, Miami, FL, USA, January 6-8, 2003, LNCS 2567, Springer 2002, pp. 100–115.

[FS87] Amos Fiat and Adi Shamir, *How to Prove Yourself: Practical Solutions to Identification and Signature Problems*, In Andrew M. Odlyzko (Ed.): Advances in Cryptology - CRYPTO '86, Santa Barbara, CA, USA, 1986, LNCS 263, Springer 1987, pp. 186-194.

[FT99] Amos Fiat and T. Tassa, *Dynamic Traitor Tracing*, In Michael J. Wiener (Ed.): Advances in Cryptology - CRYPTO '99, 19th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 15-19, 1999, LNCS 1666, Springer 1999, pp. 354–371.

[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, CA, USA, August 15-19, 1999, LNCS 1666, Springer 1999, pp. 372-387.

[GSW00] Juan A. Garay, Jessica Staddon, and Avishai Wool, *Long-Lived Broadcast Encryption*, In Mihir Bellare (Ed.): Advances in Cryptology - CRYPTO 2000, 20th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 20-24, 2000, LNCS 1880, Springer 2000, pp. 333–352.

[KY01] Aggelos Kiayias and Moti Yung, *Self Protecting Pirates and Black-Box Traitor Tracing*, In Joe Kilian (Ed.): Advances in Cryptology - CRYPTO 2001, 21st Annual International Cryptology Conference, Santa Barbara, CA, USA, August 19-23, 2001, LNCS 2139 Springer 2001, pp. 63–79.

[KY02a] Aggelos Kiayias and Moti Yung, *Traitor Tracing with Constant Transmission Rate*, In Lars R. Knudsen (Ed.): Advances in Cryptology - EUROCRYPT 2002, Amsterdam, The Netherlands, April 28 - May 2, 2002, LNCS 2332, Springer 2002, pp. 450–465.

[KY02b] Aggelos Kiayias and Moti Yung, *Breaking and Repairing Asymmetric Public-Key Traitor Tracing*, 2002 ACM Workshop on Digital Rights Management, to appear in Springer LNCS, 2003.

[KP98] Joe Kilian and Erez Petrank, *Identity Escrow*, In Hugo Krawczyk (Ed.): Advances in Cryptology - CRYPTO '98, 18th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 23-27, 1998, LNCS 1462, Springer 1998, pp. 169–185.

[KD98] K. Kurosawa and Y. Desmedt, *Optimum Traitor Tracing and Asymmetric Schemes*, In Kaisa Nyberg (Ed.): Advances in Cryptology - EUROCRYPT '98, Espoo, Finland, May 31 - June 4, 1998, LNCS 1403, Springer 1998, pp. 145–157.

[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, CA, USA, August 19-23, 2001, LNCS 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, CA, USA, August 23-27, 1998, LNCS 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, Anguilla, British West Indies, February 20-24, 2000, LNCS 1962, Springer 2001, pp. 1–20.

[Pet97] Holger Petersen, *How to Convert any Digital Signature Scheme into a Group Signature Scheme*, In Bruce Christianson, Bruno Crispo, T. Mark A. Lomas, Michael Roe (Eds.): Security Protocols, 5th International Workshop, Paris, France, April 7-9, 1997, LNCS 1361, Springer 1998, pp. 177–190.

[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, LNCS 1174, Springer 1996, pp. 49-64.

[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, CA, USA, August 20-24, 2000, LNCS 1880, Springer 2000, pp. 316–332.

[SSW01] Alice Silverberg, Jessica Staddon and Judy L. Walker, *Efficient Traitor Tracing Algorithms Using List Decoding*, In Colin Boyd (Ed.): Advances in Cryptology - ASIACRYPT 2001, Springer LNCS 2248, pp. 175-192.

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

[SW98a] Douglas R. Stinson and Ruizhong Wei, *Key preassigned traceability schemes for broadcast encryption*, In Stafford E. Tavares, Henk Meijer (Eds.): Selected Areas in Cryptography 1998, Springer LNCS Vol. 1556, pp. 144–156.

[SW98b] 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.

[WHI01] Yuji Watanabe, Goichiro Hanaoka and Hideki Imai, *Efficient Asymmetric Public-Key Traitor Tracing without Trusted Agents*, In David Naccache (Ed.): Topics in Cryptology - CT-RSA 2001 — The Cryptographer's Track, Springer LNCS Vol. 2020, pp. 392–407.

[YY99] Adam Young and Moti Yung, *Auto-recoverable Cryptosystems with Faster Initialization and the Escrow Hierarchy*, In Hideki Imai, Yuliang Zheng (Eds.): Public-Key Cryptography 1999, Springer LNCS Vol. 1560, pp. 306–314.

[YY01] Adam Young and Moti Yung, *A PVSS as Hard as Discrete Log and Shareholder Separability*, In Kwangjo Kim (Ed.): Public Key Cryptography 2001, Springer LNCS Vol. 1992, pp. 287–299.