Traceable Signatures

Aggelos Kiayias[1], Yiannis Tsiounis[2], and Moti Yung[3]

[1] Computer Science and Engineering, University of Connecticut
Storrs, CT, USA. aggelos@cse.uconn.edu
[2] Etolian Capital Management, LP.
New York, NY, USA. yiannist@etolian.com.
Research supported in part by NIST, under grant SB1341-02-W-1113
[3] Computer Science, Columbia University
New York, NY, USA. moti@cs.columbia.edu

Abstract. This work presents a new privacy primitive called "Traceable Signatures", together with an efficient provably secure implementation. To this end, we develop the underlying mathematical and protocol tools, present the concepts and the underlying security model, and then realize the scheme and its security proof. Traceable signatures support an extended set of fairness mechanisms (mechanisms for anonymity management and revocation) when compared with the traditional group signature mechanism. The extended functionality of traceable signatures is needed for proper operation and adequate level of privacy in various settings and applications. For example, the new notion allows (distributed) tracing of all signatures of a single (misbehaving) party without opening signatures and revealing identities of any other user in the system. In contrast, if such tracing is implemented by a state of the art group signature system, such wide opening of all signatures of a single user is a (centralized) operation that requires the opening of *all* anonymous signatures and revealing the users associated with them, an act that violates the privacy of all users.
To allow efficient implementation of our scheme we develop a number of basic tools, zero-knowledge proofs, protocols, and primitives that we use extensively throughout. These novel mechanisms work directly over a group of unknown order, contributing to the efficiency and modularity of our design, and may be of independent interest. The interactive version of our signature scheme yields the notion of "traceable (anonymous) identification."

1 Introduction

A number of basic primitives have been suggested in cryptographic research to deal with the issue of privacy. The most flexible private authentication tool to date is "group-signatures," a primitive where each group member is equipped with a signing algorithm that incorporates a proof of group-membership. Group-signatures were introduced by Chaum and Van Heyst in [14] and were further studied and improved in many ways in [15, 13, 7, 12, 4, 2, 25]. Each signature value is anonymous, in the sense that it only reveals that the issuer is a member of the group, without even linking signatures by the same signer.

Privacy comes at a price. Unconditional privacy seems to be an attractive notion from the user's viewpoint, nevertheless it can potentially be a very dangerous tool against public safety (and can even be abused against the user herself). Undoubtedly everybody understands that privacy is a right of law-abiding citizens, while at the same time a community must be capable of revoking such privacy when illegal behavior (performed under the “mask of privacy”) is detected; this balancing act is thus called “fairness”. Group-signatures were designed with one embedded fairness mechanism which, in fact, allows for the “opening” of an atomic signature value, revealing the identity of its signer.

We observe that while group signatures are a very general “private credentials” tool, their opening capability is not a sufficient mechanism to ensure safety and/or privacy in a number of settings. What we need is additional mechanisms for lifting of privacy conditions. It may sound paradoxical that offering more mechanisms for revoking privacy actually contributes to privacy; still, consider the following scenario: a certain member of the group is suspected of illegal activity (potentially, its identity was revealed by opening a signature value). It is then crucial to detect which signatures were issued by this particular member so that his/her transactions are traced. The only solution with the existing group signature schemes is to have the Group Manager (GM) open all signatures, thus violating the privacy of all (including law-abiding) group members. Furthermore, this operation is also scalability impairing, since the GM would have to open all signatures in the system and these signatures may be distributed in various locations. What would be desirable, instead, is to have a mechanism that allows the selective linking of the existing signatures of a misbehaving user without violating the privacy of law-abiding group members; this mechanism should be efficient (e.g. done in parallel by numerous agents when required). This capability, in fact, implements an “oblivious data mining” operation where only signature values of a selected misbehaving user are traced. Such traceability property should be offered in conjunction with the standard opening capability of group signatures.

Another type of traceability, “self-traceability,” is helpful to the user and is important in our setting. It suggests that a user should also be capable of claiming that he is the originator of a certain signature value if he wishes (or when a certain application protocol requires this). In other words, a group-member should be capable of stepping out and *claiming* a certain group-signature value as his own, *without* compromising the privacy of the remaining past or future group-signatures that he/she issues. Adding self-traceability to the existing efficient solutions in group-signatures is also far from ideal: the user will be required to remember her private random coin-tosses for all the signatures she signed, which is an unreasonable user storage overhead in many settings.

Our Notion: Motivated by the above, in this work we introduce a new basic primitive which we call *Traceable Signatures.* It incorporates the following three different types of traceability: (i) user tracing: check whether a signature was issued by a given user; it can be applied to all signatures by agents running in parallel; (ii) signature opening: reveal the signer of a given signature (as in group signature); and (iii) signature claiming: the signer of a signature provably claims a given signature that it has signed (in a stateless fashion). When recovering all transactions by performing user tracing it may be useful to avoid collecting all signatures to a central location and in order to reduce the burden of the GM (which may be a distributed entity), we divide user tracing into two steps: the first is executed by the GM and reveals some secret information about the user; this is given to a set of designated agents (clerks) that scan all signatures in parallel and reveal those signed by the suspected user. Note that the secret information revealed should not allow the agents to impersonate the user or violate the anonymity of law-abiding users.

Modeling: We model our concepts of traceable signatures and their interactive version (as traceable identification) and define their correctness and security.

We introduce a novel general way of modeling privacy systems. The model includes the definition of correctness and of security properties of the system. In a security system, like encryption, it is obvious who is the attacker and who tries to defend the encryption device, so adversary modeling is relatively easy. In a privacy system, on the other hand, a protocol between many parties may involve mutually distrusting, malicious users attacking each other from many sides and in various coalitions: e.g., a server (perhaps collaborating with a subset of some users) trying to violate the user's privacy interacting with a user trying to impersonate a group member. Since in privacy systems we deal with mutually adversarial parties, we develop a model that copes with this situation. The adversaries are described in the spirit that adversaries against a signature scheme or an encryption scheme have been dealt with in the past (i.e., by describing attack capabilities and goals for an adversary), while the model is constructed with simulation-based security proofs in mind.

To this effect, we introduce a set of queries (basic capabilities) by which adversaries can manipulate the system (and the simulator during the security proof). Then we present an "array of security definitions," where each definition is modeled as an adversary with partial access to the queries, representing a capability that the attack captures. This allows us to deal with various notions of simultaneous adversarial behavior within one system, modeling them as an "array of attacks" and proving security against each of them. Specifically in our setting, we classify three general security requirements that cover all perceived adversarial activities: misidentification attacks, anonymity attacks and framing attacks. We note that previous intuitive security notions that have appeared in the group signature literature such as unforgeability, coalition-resistance and exculpability are subsumed by our classification. We also compare our model to other models.

Constructions: Our construction is motivated by the state of the art and in particular by the mathematical assumptions that allow a group of users to generate a multitude of keys modulo a composite number that are private, namely are (partially) unknown even to the group manager who owns a trapdoor (prime factorization of the composite); such an ingenious mathematical setting was presented in [2]. Due to the refined notions of fairness of our model and its extended functionality, we need to introduce a number of new tools as well as employ a number of new cryptographic constructs that enable the various mechanisms that our model and scheme employ. We also note that our scheme is consistent with the present state-of-the-art revocation method for group signatures presented in [9], thus member revocation can be added modularly to our construction. We remark that the user tracing (combined with the GM publishing the user's "tracing trapdoor") can be used to implement a type of "CRL-based revocation" that nullifies all signatures by a private key. This type of revocation has been considered recently in [3] (also [21] has been brought to our attention).

In order to implement the scheme efficiently, we design a number of basic protocols and primitives that we use extensively throughout (as useful subroutines). A pleasing feature of these novel notions and protocols is that they work directly over a group of unknown order. We show useful properties of such groups of quadratic residues that are required for the security proofs. We then introduce the notion of "discrete-log relation sets" which is a generic way of designing zero-knowledge proof systems that allows an entity to prove efficiently the knowledge of a number of witnesses for any such relation set that involves various discrete-logarithms and satisfies a condition that we call "triangularity." Discrete-log relation sets are employed extensively in our protocols but, in fact, they are a useful as an abstraction that can be used elsewhere and are therefore of independent interest. We then define a notion called "discrete-log representations of arbitrary powers," as well as a mechanism we call "drawing random powers" which is a two party protocol wherein one party gets a secret discrete logarithm whose value she does not control, while at the same time the other party gets the public key version, i.e., the exponentiated value.

Based on the above primitives we present traceable signatures and prove their correctness and security. We remark that our traceable signature scheme adds only a constant overhead to the complexity measures of the state of the art group signature scheme of [2].

Applications: One generic application of traceable signatures is transforming an anonymous system to one with "fair privacy" (by combing traceable signature with the original system). Membership revocation of the CRL-type is also an immediate application.

Due to lack of space proofs and many details are omitted. We refer to [24] for an extended version.

Notations: The notation $S(a, b)$ (called a sphere of radius b centered at a) where $a, b \in \mathbb{Z}$ denotes the set $\{a - b + 1, \ldots, a + b - 1\}$. A function in w will be called negligible if it holds that it is smaller than any fraction of the form $\frac{1}{w^c}$ for any c and sufficiently large w; we use the notation $\mathsf{negl}(w)$ for such functions. The concatenation of two strings a, b will be denoted by $a||b$. If a is a bitstring we denote by $(a)_{l,\ldots,j}$ the substring $(a)_l|| \ldots ||(a)_j$ where $(a)_i$ denotes the i-th bit of a. The cardinality of a set A, will be denoted by $\#A$. If X and Y are parameterized probability distributions with the same support, we will write $X \approx Y$ if the statistical distance between X, Y is a negligible function in the parameter. Furthermore, if f and g are functions over a variable, we will write $f \approx g$ if their absolute distance is a negligible function in the same variable. Finally note that $\log$ denotes the logarithm base 2, PPT stands for "probabilistic polynomial-time," and $=_{\mathsf{df}}$ means "equal by definition."

2 Preliminaries

Throughout the paper we work (unless noted otherwise) in the group of quadratic residues modulo n, denoted by $QR(n)$, with $n = pq$ and $p = 2p' + 1$ and $q = 2q' + 1$. All operations are to be interpreted as modulo n (unless noted otherwise). We will employ various related security parameters (as introduced in the sequel); with respect to $QR(n)$ the relevant security parameter is the number of bits needed to represent the order of the group, denoted by $\nu =_{\mathsf{df}} \lfloor \log p'q' \rfloor + 1$. Next we define the Cryptographic

Intractability Assumptions that will be relevant in proving the security properties of our constructions.

The first assumption is the so called Strong-RSA assumption. It is similar in nature to the assumption of the difficulty of finding e-th roots of arbitrary elements in $\mathbb{Z}_n^*$ with the difference that the exponent e is not fixed (part of the instance).

Definition 1. Strong-RSA. *Given a composite n and $z \in QR(n)$, it is infeasible to find $u \in \mathbb{Z}_n^*$ and $e > 1$ such that $u^e = z(\bmod n)$, in time polynomial in ν.*

The second assumption that we will employ is the Decisional Diffie-Hellman Assumption over the quadratic residues modulo n; in stating this assumption we also take into account the fact that the exponents may belong to pre-specified integer spheres $\mathcal{B} \subseteq \{1, \ldots, p'q'\}$.

Definition 2. Decisional Diffie-Hellman *(over $\mathcal{B}_1, \mathcal{B}_2, \mathcal{B}_3$) Given a generator g of a cyclic group $QR(n)$ where n is as above, a DDH distinguisher $\mathcal{A}$ is a polynomial in ν time* PPT *that distinguishes the family of triples of the form $\langle g^x, g^y, g^z \rangle$ from the family of triples of the form $\langle g^x, g^y, g^{xy} \rangle$, where $x \in_R \mathcal{B}_1$, $y \in_R \mathcal{B}_2$, and $z \in_R \mathcal{B}_3$.*

The maximum distance of these two distributions of triples as quantified over all possible PPT *distinguishers will be denoted by $\mathsf{Adv}^{DDH}_{\mathcal{B}_1,\mathcal{B}_2,\mathcal{B}_3}(\nu)$; if $\mathcal{B}_1 = \mathcal{B}_2 = \mathcal{B}_3 = \{1, \ldots, p'q'\}$ we will write simply $\mathsf{Adv}^{DDH}(\nu)$ instead. The* DDH *assumption suggests that this advantage is a negligible function in ν.*

We remark that when the size of the spheres $\mathcal{B}_1, \mathcal{B}_2, \mathcal{B}_3$ are sufficiently close to the order of $QR(n)$ it will hold that $\mathsf{Adv}^{DDH}_{\mathcal{B}_1,\mathcal{B}_2,\mathcal{B}_3}(\nu) \approx \mathsf{Adv}^{DDH}(\nu)$. Nevertheless we discover that the spheres can be selected to be much smaller than that without any degradation in security (see the remark at the end of section 3).

Finally, we will employ the discrete-logarithm assumption over the quadratic residues modulo n and a pre-specified sphere $\mathcal{B}$, when the factorization of n is known:

Definition 3. Discrete-Logarithm. *Given two values a, b that belong to the set of quadratic residues modulo n with known factorization, so that $\exists x \in \mathcal{B} : a^x = b$, find in time polynomial in ν the integer x so that $a^x = b$. Again $\mathcal{B}$ is an integer sphere into the set $\{1, \ldots, p'q'\}$.*

Conventions. our proofs of knowledge will only be proven to work properly in the honest-verifier setting. On the one hand, the honest-verifier setting is sufficient for producing signatures. On the other hand, even in the general interactive setting the honest-verifier scenario can be enforced by assuming the existence, e.g., of a beacon, or some other mechanism that can produce trusted randomness; alternatively the participants may execute a distributed coin flipping algorithm (which are by now standard tools for converting random coin honest verifier scenario to a general proof). Such protocols where the randomness that is used to select the challenge is trusted will be called "canonical."

3 Sphere Truncations of Quadratic Residues

Let n be a composite so that $n = pq$ and $p = 2p' + 1$ and $q = 2q' + 1$ with p, q, p', q' all prime. Let a be a generator of the cyclic group of quadratic residues modulo n. Recall that the order of $QR(n)$ is $p'q'$. Let $S(2^\ell, 2^\mu) = \{2^\ell - 2^\mu + 1, \ldots, 2^\ell + 2^\mu - 1\}$ be a sphere for two parameters $\ell, \mu \in \mathbb{N}$. Observe that $\#S(2^\ell, 2^\mu) = 2^{\mu+1} - 1$.

In this section we will prove a basic result that will be helpful later in the analysis of our scheme. In particular we will show that, assuming factoring is hard and the fact the sphere $S(2^\ell, 2^\mu)$ is sufficiently large (but still not very large) the random variable a^x with $x \in_R S(2^\ell, 2^\mu)$ is indistinguishable from the uniform distribution over $QR(n)$; note that the result becomes trivial if the size of the sphere is very close to the order of $QR(n)$; we will be interested in cases where the size of the sphere is exponentially smaller (but still sufficiently large). Intuitively, this means that a truncation of the $QR(n)$ as defined by the sphere $S(2^\ell, 2^\mu)$ is indistinguishable to any probabilistic polynomial-time observer.

Consider the function $f_{g,n}(x) = g^x (\mathrm{mod} n)$ defined for all $x < n$. The inverse of this function $f^{-1}_{g,n}$ is defined for any element in $QR(n)$ so that $f^{-1}_{a,n}(y) = x$ where $x \leq p'q'$ and it holds that $a^x = y (\mathrm{mod} n)$. Observe that x can be written as a ν-bitstring. Note that if y is uniformly distributed over $\mathbb{Z}^*_n$ it holds that every bit $(x)_i$ of x with $i = 1, \ldots, \nu$ follows a probability distribution $\mathcal{D}^\nu_i$ with support the set $\{0, 1\}$. Note that for the $\mathcal{O}(\log \nu)$ most significant bits i it holds that the distribution $\mathcal{D}^\nu_i$ is biased towards 0, whereas for the remaining bits the distribution $\mathcal{D}^\nu_i$ is uniform; this bias is due to the distance between 2^ν and $p'q'$. Below we define the simultaneous hardness of the bits of the discrete-logarithm function, (cf. [22]):

Definition 4. *The bits $[l, \ldots, j]$, $l > j$, of $f^{-1}_{g,n}$ are* simultaneously hard *if the following two distributions are* PPT*-indistinguishable:*

- *the $\mathcal{SD}^j_i$ distribution: $\langle (f^{-1}_{g,n}(y))_{i,\ldots,j}, y \rangle$ where $y \in_R QR(n)$.*
- *the $\mathcal{SR}^j_i$ distribution:$\langle r_l || \ldots || r_j, y \rangle$ where $y \in_R QR(n)$ and $r_i \leftarrow \mathcal{D}^\nu_i$ for $i = l, \ldots, j$.*

Håstad et al. [22] studied the simultaneous hardness of of the discrete-logarithm over composite groups and one of their results imply the following theorem:

Theorem 1. *The bits $[\nu, \ldots, j]$ of $f^{-1}_{g,n}$ are simultaneously hard under the assumption that factoring n is hard, provided that $j = \lceil \frac{\nu}{2} \rceil - \mathcal{O}(\log \nu)$.*

Now let us return to the study of the subset of $QR(n)$ defined by the sphere $S(2^\ell, 2^\mu)$. Consider the uniform probability distribution $\mathcal{U}$ over $QR(n)$ and the probability distribution $\mathcal{D}^{S(2^\ell, 2^\mu)}_a$ with support $QR(n)$ that assigns the probability $1/(2^{\mu+1} - 1)$ to all elements a^x with $x \in S(2^\ell, 2^\mu)$ and probability 0 to all remaining elements of the support. The main result of this section is the following theorem:

Theorem 2. *The probability distributions $\mathcal{D}^{S(2^\ell, 2^\mu)}_a$ and $\mathcal{U}$ with support $QR(n)$ are* PPT*-indistinguishable under the assumption that factoring n is hard, provided that $\#S(2^\ell, 2^\mu) = 2^{\lceil \frac{\nu}{2} \rceil - \mathcal{O}(\log \nu)}$.*

Remark. The results of this section suggest that we may truncate the range of a random variable a^x, $x \in_R \{1,\ldots,p'q'\}$, into a subset of $QR(n)$ that is of size approximately $\sqrt{p'q'}$; this truncation will not affect the behavior of any polynomial-time bounded observer. In particular, for the case of the Decisional Diffie Hellman assumption in $QR(n)$ over the spheres $\mathcal{B}_1, \mathcal{B}_2, \mathcal{B}_3$, we may use spheres of size approximately $\sqrt{p'q'}$; under the assumption that factoring is hard, we will still maintain that $\mathsf{Adv}^{DDH}_{\mathcal{B}_1,\mathcal{B}_2,\mathcal{B}_3}(\nu) \approx \mathsf{Adv}^{DDH}(\nu)$. In some few cases we may need to employ the DDH over spheres that are smaller in size than $\sqrt{p'q'}$ (in particular we will employ the sphere $\mathcal{B}_2$ to be of size approximately $\sqrt[4]{p'q'}$). While the DDH over such sphere selection does not appear to be easier it could be possible that this version of DDH is a stronger intractability assumption. Nevertheless we remark that if we assume that factoring remains hard even if $\lceil \nu/4 \rceil$ of bits of the prime factors of n are known[4] then as stated in [22] approximately 3/4 of the bits of $f^{-1}_{g,n}$ are simultaneously hard and thus, using the methodology developed in this section, we can still argue that $\mathsf{Adv}^{DDH}_{\mathcal{B}_1,\mathcal{B}_2,\mathcal{B}_3}(\nu) \approx \mathsf{Adv}^{DDH}(\nu)$, even if $\mathcal{B}_2$ is of size approximately $\sqrt[4]{p'q'}$.

4 Discrete-log Relation Sets

Discrete-log relation sets are quite useful in planning complex proofs of knowledge for protocols operating over groups of unknown order in general. We note that special instances of such proofs have been investigated individually in the literature, see e.g. [12, 11](also, various discrete-log based protocols over known and unknown order subgroups have been utilized extensively in the literature, [16, 19, 17]). Our approach, that builds on this previous work, homogenizes previous instantiations in the context of signatures into a more generic framework. Below, let G be the unknown order group of quadratic residues modulo n, denoted also by $QR(n)$.

Definition 5. *A* discrete-log relation set *R with z relations over r variables and m objects is a set of relations defined over the objects $A_1,\ldots,A_m \in G$ and the free variables $\alpha_1,\ldots,\alpha_r$ with the following specifications: (1) The i-th relation in the set R is specified by a tuple $\langle a^i_1,\ldots,a^i_m\rangle$ so that each a^i_j is selected to be one of the free variables $\{\alpha_1,\ldots,\alpha_r\}$ or an element of $\mathbb{Z}$. The relation is to be interpreted as $\prod_{j=1}^m A_j^{a^i_j} = 1$. (2) Every free variable α_j is assumed to take values in a finite integer range $S(2^{\ell_j}, 2^{\mu_j})$ where $\ell_j, \mu_j \geq 0$.*

We will write $R(\alpha_1,\ldots,\alpha_r)$ to denote the conjunction of all relations $\prod_{j=1}^m A_j^{a^i_j} = 1$ that are included in R.

Below we will design a 3-move honest verifier zero-knowledge proof (see e.g. [16]) that allows to a prover that knows witnesses $x_1,\ldots,x_r$ such that $R(x_1,\ldots,x_r) = 1$ to prove knowledge of these values. We will concentrate on a discrete-log relation sets that have a specific structure that is sufficient for our setting: a discrete-log relation set R is said to be *triangular*, if for each relation i involving the free variables $\alpha_w, \alpha_{w_1},\ldots,\alpha_{w_b}$ it holds that the free-variables $\alpha_{w_1},\ldots,\alpha_{w_b}$ are contained in relations $1,\ldots,i-1$.

[4] Efficient factorization techniques are known when at least $\lceil \nu/3 \rceil$ bits of the prime factors of n are known, [22].

Proof of knowledge for a Discrete-Log Relation Set R

objects $A_1, \ldots, A_m$, r free-variables $\alpha_1, \ldots, \alpha_r$, parameters: $\epsilon > 1, k \in \mathbb{N}$,
Each variable α_j takes values in the range $S(2^{\ell_j}, 2^{\mu_j})$
$\mathcal{P}$ proves knowledge of the witnesses $x_j \in S(2^{\ell_j}, 2^{\epsilon(\mu_j+k)+2})$ s.t. $R(x_1, \ldots, x_r) = 1$

$\mathcal{P}$		$\mathcal{V}$
for $w \in \{1, \ldots, r\}$ select $t_w \in_R \pm\{0,1\}^{\epsilon(\mu_w+k)}$		
for $i \in \{1, \ldots, z\}$ set $B_i = \prod_{j:\exists w, a^i_j = \alpha_w} A_j^{t_w}$	$\xrightarrow{B_1, \ldots, B_z}$	$c \in_R \{0,1\}^k$
	$\xleftarrow{c}$	
for $w \in \{1, \ldots, r\}$ set $s_w = t_w - c \cdot (x_w - 2^{\ell_w})$	$\xrightarrow{s_1, \ldots, s_r}$	Verify:
		for $w \in \{1, \ldots, r\}$
		$s_w \in_? \pm\{0,1\}^{\epsilon(\mu_w+k)+1}$
		for $i \in \{1, \ldots, z\}$
		$\prod_{j:\exists w, a^i_j = \alpha_w} A_j^{s_w} \stackrel{?}{=} B_i (\prod_{j:a^i_j \in \mathbb{Z}} A_j^{a^i_j} \prod_{j:\exists w, a^i_j = \alpha_w} A_j^{2^{\ell_w}})^c$

Fig. 1. *Proof of Knowledge for a Discrete-Log relation set R.*

Theorem 3. *For any triangular discrete-log relation set R the 3-move protocol of figure 1 is a honest verifier zero-knowledge proof that can be used by a party (prover) knowing a witness for R to prove knowledge of the witness to a second party (verifier).*

We remark that the proof assumes that the prover is incapable of solving the Strong-RSA problem; under this assumption the cheating probability of the prover is $1/2^k$. Regarding the length of the proof we note that the proof requires the first communication flow from the prover to the verifier to be of size z $QR(n)$ elements (where z is the number of relations in R) and the second communication flow from the prover to the verifier to be of total bit-length $\sum_{w=1}^{r}(\epsilon(\mu_w + k) + 1)$.

Below, for a sphere $S(2^\ell, 2^\mu)$, the notation $S^k_\epsilon(2^\ell, 2^\mu) =_{\mathsf{df}} S(2^\ell, 2^{\frac{\mu-2}{\epsilon}-k})$ will be called the innersphere of $S(2^\ell, 2^\mu)$ for parameters ϵ, k.

5 Discrete Log Representations of Arbitrary Powers

In this section we introduce and present some basic facts about "discrete log representations of arbitrary powers" inside the set of Quadratic Residues $QR(n)$ where n. We will define three spheres $\Lambda, \Gamma, \mathrm{M}$ inside the set $\{0, \ldots, 2^\nu - 1\}$ so that the following conditions are satisfied:
[S1.] $(\min \Gamma)^2 > \max \Gamma$. [S2.] M has size approximately equal to $2^{\lceil \nu/2 \rceil}$. [S3.] $\min \Gamma > \max \mathrm{M} \max \Lambda + \max \Lambda + \max \mathrm{M}$. This set of conditions is attainable as shown by the following possible selection: for simplicity, we assume that ν is divisible by 4: $\Lambda = S(2^{\frac{\nu}{4}-1}, 2^{\frac{\nu}{4}-1})$, note that $\#\Lambda = 2^{\frac{\nu}{4}} - 1$ and $\max \Lambda = 2^{\frac{\nu}{4}} - 1$. $\mathrm{M} = S(2^{\frac{\nu}{2}-1}, 2^{\frac{\nu}{2}-1})$, note that $\#\mathrm{M} = 2^{\frac{\nu}{2}} - 1$ and $\max \mathrm{M} = 2^{\frac{\nu}{2}} - 1$. $\Gamma = S(2^{\frac{3\nu}{4}} + 2^{\frac{\nu}{4}-1}, 2^{\frac{\nu}{4}-1})$, note that $\#\Gamma = 2^{\frac{\nu}{4}} - 1$, $\min \Gamma = 2^{\frac{3\nu}{4}} + 1 > \max \Lambda \max \mathrm{M} + \max \Lambda + \max \mathrm{M} = 2^{\frac{3\nu}{4}} - 1$.

In the exposition below we use some fixed values $a_0, a, b \in QR(n)$.

Definition 6. *A discrete-log representation of an arbitrary power is a tuple* $\langle A, e : x, x' \rangle$ *so that it holds* $A^e = a_0 a^x b^{x'}$ *with* $x, x' \in \Lambda$ *and* $e \in \Gamma$.

In this work we will be interested in the following computational problem:

◇ *The One-more Representation Problem.* Given n, a_0, a, b and K discrete-log representations of arbitrary powers find "one-more" discrete-log representation of an arbitrary power inside $QR(n)$.

The theorem below establishes that solving the One-more representation problem cannot be substantially easier than solving the Strong-RSA problem. We remark that a variant of this problem and of the theorem below has been proposed and proved in a recent work of Camenisch and Lysyanskaya [10] (without the sphere constraints). Note that the sphere constraints that we employ will allow shorter membership certificates later on, thus contributing in the efficiency of the general design.

Theorem 4. *Fix* $a_0, a, b \in QR(n)$ *and spheres* $\Lambda, \mathrm{M}, \Gamma$ *satisfying the above properties. Let* $\mathcal{M}$ *be a* PPT *algorithm that given* K *discrete-log representations of arbitrary powers inside* $QR(n)$ *it outputs a different discrete-log representation of an arbitrary power inside* $QR(n)$ *with non-negligible probability* α. *Then, the Strong-RSA problem can be solved with non-negligible probability at least* $\alpha/2K$.

6 Non-adaptive Drawings of Random Powers

Consider the following game between two players A and B: player A wishes to select a random power a^x so that $x \in_R S(2^\ell, 2^\mu)$ where $a \in QR(n)$. Player B wants to ensure that the value x is selected "non-adaptively" from its respective domain. The output specifications of the game is that player A returns x and that player B returns a^x. Player B is assumed to know the factorization of n. In this section we will carefully model and implement a protocol for achieving this two-player functionality. The reader is referred to [20] for a general discussion of modeling secure two-party computations.

In the ideal world the above game is played by two Interactive TM's (ITM's) A_0, B_0 and the help of a trusted third party ITM T following the specifications below. We note that we use a special symbol $\perp$ to denote failure (or unwillingness to participate); if an ITM terminates with any other output other than $\perp$ we say that it accepts; in the other case we say it rejects. From all the possible ways to implement A_0, B_0 one is considered to be the honest one; this will be marked as A_0^H, B_0^H and is also specified below.

0. The modulus n is available to all parties and its factorization is known to B_0. The sphere $S(2^\ell, 2^\mu)$ is also public and fixed.
1. A_0 sends a message in $\{\mathsf{go}, \perp\}$ to T. A_0^H transmits go.
2. B_0 sends a message in $\{\mathsf{go}, \perp\}$ to T. B_0^H transmits go.
3. If T receives go from both parties, it selects $x \in_R S(2^\ell, 2^\mu)$ and returns x to A_0; otherwise T transmits $\perp$ to both parties.
4. A_0 selects a value $C \in \mathbb{Z}_n^*$ and transmits either C or $\perp$ to T. A_0^H transmits $C = a^x \bmod n$.

5. T verifies that $a^x \equiv C(\bmod n)$ and if this is the case it transmits C to both players. Otherwise, (or in the case A_0 transmitted $\perp$ in step 4), T transmits $\perp$ to both players. B_0^H terminates by returning C or $\perp$ in the case of receiving $\perp$ from T. Similarly A_0^H terminates by returning x, or $\perp$ in the case of receiving $\perp$ from T.

Let $\mathsf{Im}_T =_{\mathsf{df}} \langle A_0, B_0\rangle$ be two ITM's that implement the above protocol with the help of the ITM T. We define by $\mathsf{OUT}_{A_0}^{\mathsf{Im}_T}(\mathsf{init}_A(\nu))$ and $\mathsf{OUT}_{B_0}^{\mathsf{Im}_T}(\mathsf{init}_B(\nu))$ be the output probability distributions of the two players. Note that $\mathsf{init}_A(\nu)$ contains the initialization string of player A which contains the modulus n, and the description of the sphere $S(2^\ell, 2^\mu)$; similarly $\mathsf{init}_B(\nu)$ is defined as $\mathsf{init}_A(\nu)$ with the addition of the factorization of n. Below we will use the notation $\mathsf{IDEAL}^{\mathsf{Im}_T}(in_A, in_B)$ to denote the pair $\langle \mathsf{OUT}_{A_0}^{\mathsf{Im}_T}(in_A), \mathsf{OUT}_{B_0}^{\mathsf{Im}_T}(in_B)\rangle$. Finally, we denote by Im_T^H the pair $\langle A_0^H, B_0^H\rangle$.

The goal of a protocol for non-adaptive drawing of random powers is the simulation of the trusted third party by the two players. Let $\mathsf{Im} = \langle A_1, B_1\rangle$ be a two-player system of interactive TM's that implement the above game without interacting with the trusted third party T. As above we will denote by $\mathsf{OUT}_{A_1}^{\mathsf{Im}}(in_A)$ the output probability distribution A_1, and likewise for $\mathsf{OUT}_{B_1}^{\mathsf{Im}}(in_B)$. Also we denote by $\mathsf{REAL}^{\mathsf{Im}}(in_A, in_B)$ the concatenation of these two distributions.

Definition 7. *(Correctness) An implementation* $\mathsf{Im} = \langle A_1, B_1\rangle$ *for non-adaptive drawings of random powers is* correct *if the following is true:*

$$\mathsf{REAL}^{\mathsf{Im}}(in_A, in_B) \approx \mathsf{IDEAL}^{\mathsf{Im}_T^H}(in_A, in_B)$$

where $in_A \leftarrow \mathsf{init}_A(\nu)$ *and* $in_B \leftarrow \mathsf{init}_B(\nu)$. *Intuitively the above definition means that the implementation* Im *should achieve essentially the same output functionality for the two players as the ideal honest implementation.*

Defining security is naturally a bit trickier as the two players may misbehave arbitrarily when executing the prescribed protocol implementation $\mathsf{Im} = \langle A_1, B_1\rangle$.

Definition 8. *(Security) An implementation* $\mathsf{Im} = \langle A_1, B_1\rangle$ *for non-adaptive drawings of random powers is* secure *if the following is true:*

$$\forall A_1^* \ \exists A_0^* \ \mathsf{REAL}^{\langle A_1^*, B_1\rangle}(in_A, in_B) \approx \mathsf{IDEAL}^{\langle A_0^*, B_0^H\rangle}(in_A, in_B)$$

$$\forall B_1^* \ \exists B_0^* \ \mathsf{REAL}^{\langle A_1, B_1^*\rangle}(in_A, in_B) \approx \mathsf{IDEAL}^{\langle A_0^H, B_0^*\rangle}(in_A, in_B)$$

where $in_A \leftarrow \mathsf{init}_A(\nu)$ *and* $in_B \leftarrow \mathsf{init}_B(\nu)$. *Intuitively the above definition means that no matter what adversarial strategy is followed by either player it holds that it can be transformed to the ideal world setting without affecting the output distribution.*

Having defined the goals, we now take on the task of designing an implementation Im without a trusted third party; below we denote by $\tilde{m} =_{\mathsf{df}} \#S(2^\ell, 2^\mu) = 2^{\mu+1} - 1$.

1. The two players read their inputs and initiate a protocol dialog.
2. Player A selects $\tilde{x} \in_R \mathbb{Z}_{\tilde{m}}, \tilde{r} \in_R \{0, \ldots, n^2 - 1\}$ and transmits to player B the value $C_1 = g^{\tilde{x}} h^{\tilde{r}}(\bmod n)$ and $C_2 = y^{\tilde{r}}(\bmod n)$.

3. Player A engages player B in a proof of knowledge for the discrete-log relation set $\langle -1, 0, \tilde{x}, \tilde{r}, 0\rangle$ and $\langle 0, -1, 0, 0, \tilde{r}\rangle$ over the objects C_1, C_2, g, h, y. Observe that the relation set is triangular.
4. Player B selects $\tilde{y} \in_R \mathbb{Z}_{\tilde{m}}$ and transmits $\tilde{y}$ to A.
5. Player A computes $x' = \tilde{x} + \tilde{y}(\mathrm{mod}\tilde{m})$ and transmits to player B the value $C_3 = a^{x'}$.
6. Player A engages player B in a proof of knowledge for the discrete-log relation set $\langle -1, 0, \alpha, \beta, \gamma, 0, 0\rangle$, $\langle 0, -1, 0, 0, 0, 0, 0, \gamma\rangle$, $\langle 0, 0, -1, 0, 0, 0, \alpha, 0\rangle$ over the objects $C_1 g^{\tilde{y}}, C_2, C_3, g, g^{\tilde{m}}, h, a, y$ (observe again, that the relation set is triangular).
7. Player A engages with player B to a tight interval proof for C_3 ensuring that $\log_a C_3 \in \mathbb{Z}_{\tilde{m}}$ (treating $\mathbb{Z}_{\tilde{m}}$ as an integer range); this is done as described in [6].
8. Player A outputs $x := x' + 2^{\ell} - 2^{\mu} + 1$ and Player B outputs $C := C_3 a^{2^{\ell} - 2^{\mu} + 1}$.

Theorem 5. *The above protocol implementation for non-adaptive drawing of random powers is correct and secure (as in definitions 7 and 8) under the Strong-RSA and DDH assumptions.*

7 Traceable Signatures and Identification

In this section we describe the traceable signature syntax and model, focusing first on the interactive version, called a traceable identification scheme. Traceable identification employs seven sub-protocols Setup, Join, Identify, Open, Reveal, Trace, Claim that are executed by the active participants of the system, which are identified by the Group Manager (GM), a set of users and other non-trusted third parties called tracers.

Setup (executed by the GM). For a given security parameter ν, the GM produces a publicly-known string $\mathsf{pk}_{\mathcal{GM}}$ and some private string $\mathsf{sk}_{\mathcal{GM}}$ to be used for user key generation.

Join (a protocol between a new user and the GM). In the course of the protocol the GM employs the secret-key string $\mathsf{sk}_{\mathcal{GM}}$. The outcome of the protocol results in a membership certificate cert_i that becomes known to the new user. The entire Join protocol transcript is stored by the GM in a database that will be denoted by Jtrans. This is a private database and each Join transcript contains also all the coin tosses that were used by the GM during the execution.

Identify (traceable identification) It is a proof system between a prover and a verifier with the user playing the role of the prover and the verifier played by any non-trusted third party. The Identify protocol is a proof of knowledge of a membership certificate cert_i. In our setting, we will restrict the protocol to operate in 3 rounds, with the verifier selecting honestly a random challenge of appropriate length in the second round.

Open (invoked by the Trustee) A PPT TM which, given an Identify protocol transcript, the secret-key $\mathsf{sk}_{\mathcal{GM}}$ and access to the database Jtrans it outputs the identity of the signer.

Reveal (invoked by the GM) A PPT TM which, given the Join transcript for a user i, it outputs the "tracing trapdoor" for the user i denoted by trace_i.

Trace (invoked by designated parties, called tracers). A PPT TM which, given an $\mathsf{Identify}$ protocol transcript π and the tracing trapdoor of a certain user trace_i, checks if π was produced by user i.

Claim. It is a proof system between a prover and a verifier where the role of the prover is played by the user and the role of the verifier is played by any claim recipient. In our setting, the Claim protocol is a proof of knowledge that binds to a given $\mathsf{Identify}$ protocol transcript and employs the membership certificate cert_i of the user. As in the case of $\mathsf{Identify}$ protocol we restrict Claim to be a 3-round protocol so that in round 2 the verifier selects honestly a random challenge of appropriate length.

Definition 9. (Correctness for traceable identification) *A traceable identification scheme with security parameter ν is* **correct** *if the following four conditions are satisfied (with overwhelming probability in ν). Let* $\mathsf{Identify}_{\mathcal{U}}(\mathsf{pk}_{\mathcal{GM}})$ *be the distribution of* $\mathsf{Identify}$ *protocol transcripts generated by user $\mathcal{U}$ and* $\mathsf{Claim}_{\mathcal{U}}(\pi)$ *the distribution of* Claim *protocol transcripts generated by user $\mathcal{U}$ for an* $\mathsf{Identify}$ *protocol transcript π.*

(1) **Identify-Correctness:** *The* $\mathsf{Identify}$ *protocol is a proof of knowledge of a membership certificate for the public-key* $\mathsf{pk}_{\mathcal{U}}$ *that satisfies completeness.*

(2) **Open-Correctness:** $\mathsf{Open}(\mathsf{sk}_{\mathcal{GM}}, \mathsf{Jtrans}, \mathsf{Identify}_{\mathcal{U}}) = \mathcal{U}$.

(3) **Trace-Correctness:** $\mathsf{Trace}(\mathsf{Reveal}(\mathcal{U}, \mathsf{Jtrans}), \mathsf{Identify}_{\mathcal{U}}) = \mathsf{true}$ *and for any* $\mathcal{U}' \neq \mathcal{U}$, $\mathsf{Trace}(\mathsf{Reveal}\ (\mathcal{U}, \mathsf{Jtrans}), \mathsf{Identify}_{\mathcal{U}'}) = \mathsf{false}$.

(4) **Claim-Correctness:** *The* Claim *protocol over the* $\mathsf{Identify}$ *transcript π, is a proof of knowledge of the membership certificate embedded into π that satisfies completeness.*

Given an traceable identification scheme as described above, we will derive a traceable signature by employing the Fiat-Shamir transformation [18].

7.1 Security Model for Traceable Schemes

In this section we formalize the security model for traceable schemes. To claim security we will define the notion of an interface $\mathcal{I}$ for a traceable scheme which is a PTM that simulates the operation of the system. The purpose behind the definition of $\mathcal{I}$ is to capture all possible adversarial activities against a traceable scheme in an intuitive way. As in the previous section, we will focus first on traceable identification. We model the security of a traceable identification scheme as an interaction between the adversary $\mathcal{A}$ and an entity called the *interface*. The interface maintains a (private) state denoted by $\mathsf{state}_{\mathcal{I}}$ (or simply state) and communicates with the adversary over a handful of pre-specified *query actions* that allow the adversary to learn information about $\mathsf{state}_{\mathcal{I}}$; these queries are specified below. The initial state of the interface is set to $\mathsf{state}_{\mathcal{I}} = \langle \mathsf{sk}_{\mathcal{GM}}, \mathsf{pk}_{\mathcal{GM}} \rangle$. The interface also employs an "internal user counter" denoted by n which is initialized to 0. Moreover three sets are initialized U^p, U^a, U^b, U^r to $\emptyset$. Note that $\mathsf{state}_{\mathcal{I}}$ is also assumed to contain U^p, U^a, U^b, U^r and n. Finally the interface employs two other strings denoted and initialized as follows: $\mathsf{Jtrans} = \epsilon$ and $\mathsf{Itrans} = \epsilon$. The various query action specifications are listed below:

- $\langle \mathcal{Q}_{\mathsf{pub}} \rangle$. The interface returns the string $\langle \mathsf{n}, \mathsf{pk}_{\mathcal{GM}} \rangle$. This allows to an adversary to learn the public-information of the system, i.e., the number of users and the public-key information.
- $\langle \mathcal{Q}_{\mathsf{key}} \rangle$. The interface returns $\mathsf{sk}_{\mathcal{GM}}$; this query action allows to the adversary to corrupt the group-manager.
- $\langle \mathcal{Q}_{\mathsf{p-join}} \rangle$. The interface simulates the Join protocol in *private*, increases the user count n by 1, and sets $\mathsf{state} := \mathsf{state}_{\mathcal{I}} || \langle \mathsf{n}, \mathsf{transcript}_{\mathsf{n}}, \mathsf{cert}_{\mathsf{n}} \rangle$. It also adds n into U^p and sets $\mathsf{Jtrans} := \mathsf{Jtrans} || \langle \mathsf{n}, \mathsf{transcript}_{\mathsf{n}} \rangle$.
  This query action allows to the adversary to introduce a new user to the system (that is not adversarially controlled).
- $\langle \mathcal{Q}_{\mathsf{a-join}} \rangle$. The interface initiates an active Join dialog with the adversary; the interface increases the user count n by 1, and assumes the role of the GM where the adversary assumes the role of the prospective user. If the dialog terminates successfully, the interface sets $\mathsf{state}_{\mathcal{I}} := \mathsf{state}_{\mathcal{I}} || \langle \mathsf{n}, \mathsf{transcript}_{\mathsf{n}}, \perp \rangle$. It finally adds n into the set U^a and $\mathsf{Jtrans} := \mathsf{Jtrans} || \langle \mathsf{n}, \mathsf{transcript}_{\mathsf{n}} \rangle$.
  This query action allows to the adversary to introduce an adversarially controlled user to the system. The adversary has the chance to interact with the GM through the Join dialog.
- $\langle \mathcal{Q}_{\mathsf{b-join}} \rangle$. The interface initiates an active Join dialog with the adversary; the interface increases the user count n by 1 and assumes the role of the prospective user and the adversary assumes the role of the GM. If the dialog terminates successfully the interface sets $\mathsf{state}_{\mathcal{I}} := \mathsf{state}_{\mathcal{I}} || \langle \mathsf{n}, \perp, \mathsf{cert}_{\mathsf{n}} \rangle$. It also adds n into U^b.
  This query allows the adversary to introduce users to the system acting as a GM.
- $\langle \mathcal{Q}_{\mathsf{id}}, i \rangle$. The interface parses $\mathsf{state}_{\mathcal{I}}$ and to recover an entry of the form $\langle i, \cdot, \mathsf{cert}_i \rangle$; then it produces an Identify protocol transcript using the certificate cert_i and selecting the verifier challenge at random; if no such entry is discovered or if $i \in U^a$ the interface returns $\perp$. Finally, if π is the protocol transcript the interface sets $\mathsf{Itrans} = \mathsf{Itrans} || \langle i, \pi \rangle$.
- $\langle \mathcal{Q}_{\mathsf{reveal}}, i \rangle$. The interface returns the output of $\mathsf{Reveal}(i, \mathsf{Jtrans})$ and places $i \in U^r$. Sometimes we will write $\mathcal{Q}_{\mathsf{reveal}}^{\neg A}$ to restrict the interface from revealing users in A. Note that this query returns $\perp$ in case user i does not exist or $i \in U^b$.

Given the above definition of an interface we proceed to characterize the various security properties that a traceable scheme should satisfy. We will use the notation $\mathcal{I}[a, \mathcal{Q}_1, \ldots, \mathcal{Q}_r]$ to denote the operation of the interface with (initial) state a that responds to the query actions $\mathcal{Q}_1, \ldots, \mathcal{Q}_r$ (a subset of the query actions defined above). In general we assume that the interface serves one query at a time: this applies to the queries $\mathcal{Q}_{\mathsf{a-join}}$ and $\mathcal{Q}_{\mathsf{b-join}}$ that require interaction with the adversary (i.e., the interface does not allow the adversary to cascade such queries). For a traceable identification scheme we will denote by iV the verifier algorithm for the canonical Identify 3-move protocol as well as by cV the verifier algorithm of the canonical Claim 3-move protocol.

Our definition of security, stated below, is based on the definitions of the three named security properties in the coming subsections.

Definition 10. *A traceable scheme is said to be* **secure** *provided that it satisfies security against misidentification, anonymity and framing attacks.*

Regarding traceable signatures, we note that we model security using canonical 3-move proofs of knowledge and passive impersonation-type of attacks; we remark that identification security in this type of model facilitates the employment of the Fiat-Shamir transform for proving signature security; thus, proving security for the interactive version will be sufficient for ensuring security of the traceable signature in the random oracle model following the proof techniques of [1].

Misidentification Attacks. In a misidentification attack against a traceable scheme, the adversary is allowed to control a number of users of the system (in an adaptive fashion). The adversary is also allowed to observe and control the operation of the system in the way that users are added and produce identification transcripts. In addition, the adversary is allowed to invoke $\mathcal{Q}_{\mathsf{reveal}}$, i.e., participate in the system as a tracer. The objective of the adversary can take either of the following forms: (i) produce an identification transcript that satisfies either one of the following properties: (ia): the adversarial identification transcript does not open to any of the users controlled by the adversary, or (ib): the adversarial identification transcript does not trace to any of the users controlled by the adversary. Alternatively, (ii) produce a claim for an Identify transcript of one of the users that he does not control (in the set U^p). We will formalize this attack using the experiment presented in figure 2.

$$
\mathsf{Exp}^{\mathcal{A}}_{\mathsf{mis}}(\nu) : \left| \begin{array}{l}
\mathsf{state}_{\mathcal{I}} = \langle \mathsf{pk}_{\mathcal{GM}}, \mathsf{sk}_{\mathcal{GM}} \rangle \leftarrow \mathsf{Setup}(1^{\nu});\\
\langle \mathsf{s}, d, \rho_1 \rangle \leftarrow \mathcal{A}^{\mathcal{I}[\mathsf{state}_{\mathcal{I}}, \mathcal{Q}_{\mathsf{pub}}, \mathcal{Q}_{\mathsf{p-join}}, \mathcal{Q}_{\mathsf{a-join}}, \mathcal{Q}_{\mathsf{id}}, \mathcal{Q}_{\mathsf{reveal}}]}(\mathsf{first}, 1^{\nu});\\
c \xleftarrow{r} \{0,1\}^k;\\
\rho_2 \leftarrow \mathcal{A}(\mathsf{second}, d, \rho_1, c);\\
\texttt{if } \mathsf{iV}(\mathsf{pk}_{\mathcal{GM}}, \rho_1, c, \rho_2) = \texttt{true and}\\
\qquad \texttt{if } \mathsf{Open}(\mathsf{sk}_{\mathcal{GM}}, \mathsf{Jtrans}, \rho_1) \notin U^a\\
\qquad\quad \texttt{or } \wedge_{i \in U^a} \mathsf{Trace}(\mathsf{Reveal}(i, \mathsf{Jtrans}), \rho_1, c, \rho_2) = \texttt{false}\\
\qquad \texttt{then output 1}\\
\texttt{else if } \mathsf{s} \texttt{ is such that } \langle i, \mathsf{s} \rangle \in \mathsf{Itrans} \texttt{ and } i \in U^p \cup U^r\\
\qquad \texttt{and } \mathsf{cV}(\mathsf{s}, \rho_1, c, \rho_2) = \mathsf{true} \texttt{ then output 1}\\
\texttt{else output 0}
\end{array} \right.
$$

Fig. 2. The misidentification experiment

We will say that a traceable identification scheme satisfies security against misidentification if for any PPT $\mathcal{A}$, it holds that $\mathbf{Prob}[\mathsf{Exp}^{\mathcal{A}}_{\mathsf{mis}}(\nu) = 1] = \mathsf{negl}(\nu)$.

Anonymity Attacks An anonymity attack is best understood in terms of the following experiment that is played with the adversary $\mathcal{A}$ who is assumed to operate in two phases called play and guess. In the play phase, the adversary interacts with the interface, introduces users in the system, and selects two target users he does not control; then receives an identification transcript that corresponds to one of the two at random; in the guess stage the adversary tries to guess which of the two produced the identification transcript (while accessing the system but without revealing the challenge transcripts). We remark that we allow the adversary to participate in the system also as a tracer (i.e., one of the agents that assist in the tracing functionality). The experiment is presented in figure 3. A traceability scheme is said to satisfy anonymity if for any attacker $\mathcal{A}$ it holds that $|\mathbf{Prob}[\mathsf{Exp}^{\mathcal{A}}_{\mathsf{anon}}(\nu) = 1] - \frac{1}{2}| = \mathsf{negl}(\nu)$.

$$
\mathsf{Exp}^{\mathcal{A}}_{\mathsf{anon}}(\nu) : \left|
\begin{array}{l}
\mathsf{state}_{\mathcal{I}} = \langle \mathsf{pk}_{\mathcal{GM}}, \mathsf{sk}_{\mathcal{GM}} \rangle \leftarrow \mathsf{Setup}(1^{\nu});\\
\langle d, i_0, i_1 \rangle \leftarrow \mathcal{A}^{\mathcal{I}[\mathsf{state}_{\mathcal{I}}, \mathcal{Q}_{\mathsf{pub}}, \mathcal{Q}_{\mathsf{p-join}}, \mathcal{Q}_{\mathsf{a-join}}, \mathcal{Q}_{\mathsf{id}}, \mathcal{Q}_{\mathsf{reveal}}]}(\mathsf{play}, 1^{\nu});\\
\texttt{if } i_0 \texttt{ or } i_1 \texttt{ belong to } U^a \cup U^r \texttt{ output } \perp.\\
b \xleftarrow{r} \{0,1\}.\\
\texttt{parse } \mathsf{state}_{\mathcal{I}} \texttt{ and find the entry } \langle i_b, \mathsf{transcript}_{i_b}, \mathsf{cert}_{i_b} \rangle.\\
\texttt{execute the } \textbf{Identify} \texttt{ protocol for } \mathsf{cert}_{i_b} \texttt{ to obtain } \langle \rho_1, c, \rho_2 \rangle.\\
b_* \leftarrow \mathcal{A}^{\mathcal{I}[\mathsf{state}_{\mathcal{I}}, \mathcal{Q}_{\mathsf{pub}}, \mathcal{Q}_{\mathsf{p-join}}, \mathcal{Q}_{\mathsf{a-join}}, \mathcal{Q}_{\mathsf{id}}, \mathcal{Q}^{\neg(i_0,i_1)}_{\mathsf{reveal}}]}(\mathsf{guess}, 1^{\nu}, d, \langle \rho_1, c, \rho_2 \rangle);\\
\texttt{if } b = b_* \texttt{ then output } 1 \texttt{ else output } 0.
\end{array}\right.
$$

Fig. 3. The anonymity attack experiment

Framing Attacks A user may be framed by the system in two different ways: the GM may construct a signature that opens or trace to an innocent user, or it may claim a signature that was generated by the user. We capture these two framing notions with the experiment described in figure 4 (we remark that "exculpability" of group signatures [2] is integrated in this experiment).

$$
\mathsf{Exp}^{\mathcal{A}}_{\mathsf{fra}}(\nu) : \left|
\begin{array}{l}
\mathsf{state}_{\mathcal{I}} = \langle \mathsf{pk}_{\mathcal{GM}}, \mathsf{sk}_{\mathcal{GM}} \rangle \leftarrow \mathsf{Setup}(1^{\nu});\\
\langle \mathsf{s}, d, \rho_1 \rangle \leftarrow \mathcal{A}^{\mathcal{I}[\mathsf{state}_{\mathcal{I}}, \mathcal{Q}_{\mathsf{pub}}, \mathcal{Q}_{\mathsf{key}}, \mathcal{Q}_{\mathsf{b-join}}, \mathcal{Q}_{\mathsf{id}}]}(\mathsf{first}, 1^{\nu});\\
c \xleftarrow{r} \{0,1\}^k;\\
\rho_2 \leftarrow \mathcal{A}(\mathsf{second}, d, \rho_1, c);\\
\texttt{if } \mathsf{iV}(\mathsf{pk}_{\mathcal{GM}}, \rho_1, c, \rho_2) = \texttt{true and}\\
\qquad \texttt{if } \mathsf{Open}(\mathsf{sk}_{\mathcal{GM}}, \mathsf{Jtrans}, \rho_1) \in U^b\\
\qquad\quad \texttt{or } \exists i \in U^b : \mathsf{Trace}(\mathsf{Reveal}(i, \mathsf{Jtrans}), \rho_1, c, \rho_2) = \texttt{true}\\
\qquad \texttt{then output 1}\\
\texttt{else if } \mathsf{s} \texttt{ is such that } \langle i, \mathsf{s} \rangle \in \mathsf{Itrans} \texttt{ and } i \in U^b\\
\qquad \texttt{and } \mathsf{cV}(\mathsf{s}, \rho_1, c, \rho_2) = \mathsf{true} \texttt{ then output 1}\\
\texttt{else output 0}
\end{array}\right.
$$

Fig. 4. The framing attack experiment

A traceable scheme satisfies security against framing provided that for any probabilistic polynomial-time $\mathcal{A}$ it holds that $\mathbf{Prob}[\mathsf{Exp}^{\mathcal{A}}_{\mathsf{fra}}(\nu) = 1] = \mathsf{negl}(\nu)$.

Comments (i) In modeling misidentification and anonymity attacks we do not allow the adversary to submit "open signature" queries to the interface. This models the fact that opening a signature is an internal operation performed by the GM. On the contrary, this is not assumed for the tracing operation, since we model it as a distributed

operation whose results are made available to distributed agents (and thus the $\mathcal{Q}_{\mathsf{reveal}}$ oracle query is available to the adversary). Allowing opening oracles to be part of the adversarial control is possible, but will require our encryptions and commitments to be of the chosen ciphertext secure type.
(ii) Misidentification and Framing in traceable schemes capture two perspectives of adversarial behavior: in the first case the adversary does not corrupt the GM (and thus does not have at its disposal the GM's keys) and attempts to subvert the system. In the second case, the adversary is essentially the system itself (controls the GM) and attempts to frame innocent users. We find that the distinction of these two perspectives is important in the terms of our modeling of traceable signatures and as we see they rely on different intractability assumptions.
(iii) It is worth noting here the comparison of our model to previous approaches to formal modeling of primitives related to traceable signatures, in particular identity escrow and group signatures. Camenisch and Lysyanskaya [8] formalize security in identity escrow schemes based on a real vs. ideal model formulation, whereas our approach is more along the lines of security against adversaries of signature schemes with adversarial system access capabilities and adversarial goals in mind. Bellare et al. [5] provide a formal model for a relaxed group signature scenario where a dealer is supposed to run the user key-generation mechanism (rather than the user itself interactively with the group manager via the Join protocol). Our approach, employing active interaction between the adversary and the interface that represents the system (and simulates it in a security proof), is more suitable for the traceable schemes setting, which, in turn, follows the setting and attacks considered in [2] (where the group manager enters users into the system and, at the same time, he lacks full knowledge of the joining users' keys).

8 Design of a Traceable Scheme

Parameters. The parameters of the scheme are $\epsilon \in \mathbb{R}$ with $\epsilon > 1$, $k \in \mathbb{N}$ as well as three spheres $\Lambda, \mathrm{M}, \Gamma$ satisfying the properties presented in 5; Below we will denote by Λ^k_ϵ, and Γ^k_ϵ the inner spheres of Λ, M and Γ w.r.t. the parameters ϵ, k .
Setup The GM generates two primes p', q' with $p = 2p' + 1$, $q = 2q' + 1$ also primes. The modulus is set to $n = pq$. The spheres $\Lambda, \mathrm{M}, \Gamma$ are embedded into $\{0, \ldots, p'q'-1\}$. Also the GM selects $a, a_0, b, g, h \in_R QR(n)$ of order $p'q'$. The secret-key $\mathsf{sk}_{\mathcal{GM}}$ of the GM is set to p, q. The public-key of the system is subsequently set to $\mathsf{pk}_{\mathcal{GM}} := \langle n, a, a_0, b, y, g, h\rangle$.
Join (a protocol executed by a new user and the GM). The prospective user and the GM execute the protocol for non-adaptive drawing a random power $x' \in \Lambda^k_\epsilon$ over b (see section 6) with the user playing the role of player A and the GM playing the role of player B; upon successful completion of the protocol the user obtains x'_i and the GM obtains the value $C_i = b^{x'_i}$.

Subsequently the GM selects a random prime $e_i \in \Gamma^k_\epsilon$ and $x_i \in \Lambda^k_\epsilon$ and then computes $A_i = (C_i a^{x_i} a_0)^{e_i^{-1}} (\mathrm{mod} n)$ and sends to the user the values $\langle A_i, e_i, x_i\rangle$. The user forms the membership certificate as $\mathsf{cert}_i := \langle A_i, e_i, x_i, x'_i\rangle$. Observe that $\langle A_i, e_i : x_i, x'_i\rangle$ is a discrete-log representation of an arbitrary power in $QR(n)$ (see section 5); furthermore observe that the portion of the certificate x_i is known to the GM and will be used as the user's tracing trapdoor.

Identify. To identify herself a user first computes the values,

$$T_1 = A_i y^r,\ T_2 = g^r,\ T_3 = g^{e_i} h^r,\ T_4 = g^{x_i k},\ T_5 = g^k,\ T_6 = g^{x'_i k'},\ T_7 = g^{k'}$$

where $r, k, k' \in_R$ M. Subsequently the user proceeds to execute the proof of knowledge of the following triangular discrete-log relation set defined over the objects g, h, y, a_0, a, $b, T_1^{-1}, T_2^{-1}, T_3, T_4, T_5, T_6, T_7$ and the free variables are $x, x' \in \Lambda_\epsilon^k, e \in \Gamma_\epsilon^k, r, h'$.

$$\left[\begin{array}{rccccccccccccc}
 & g & h & (T_2)^{-1} & T_5 & T_7 & y & (T_1)^{-1} & a & b & a_0 & T_3 & T_4 & T_6 \\
T_2 = g^r : & r & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
T_3 = g^e h^r : & e & r & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\
T_2^e = g^{h'} : & h' & 0 & e & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
T_5^x = T_4 : & 0 & 0 & 0 & x & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \\
T_7^{x'} = T_6 : & 0 & 0 & 0 & 0 & x' & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \\
a_0 a^x b^{x'} y^{h'} = T_1^e : & 0 & 0 & 0 & 0 & 0 & h' & e & x & x' & 1 & 0 & 0 & 0
\end{array}\right]$$

Observe that the above proof of knowledge ensures that the values $T_1, \ldots, T_7$ are properly formed and "contain" a valid certificate. In particular the above proof not only enforces the certificate condition $A_i^{e_i} = a_0 a^{x_i} b^{x'_i}$ but also the fact that $e_i \in \Gamma$ and $x_i, x'_i \in \Lambda$.

Open. (invoked by the GM) Given a $\mathsf{Identify}$ transcript $\langle \rho_1, c, \rho_2 \rangle$ and all Join transcripts the GM does the following: it parses ρ_1 for the sequence $\langle T_1, \ldots, T_7 \rangle$ and computes the value $A = (T_2)^{-x} T_1$. Then it searches the membership certificates $\langle A_i, e_i \rangle$ (available from the Join transcripts) to discover the index i such that $A = A_i$; the index i identifies the signer of the message.

Reveal. (invoked by the GM) Given the Join transcript of the i-th user the GM parses the Join transcript to recover the tracing trapdoor $\mathsf{trace}_i := x_i$.

Trace. (invoked by any agent/clerk) Given the value trace_i and an $\mathsf{Identify}$ protocol transcript $\langle \rho_1, c, \rho_2 \rangle$ the agent parses the sequence $\langle T_1, T_2, T_3, T_4, T_5, T_6, T_7 \rangle$ from ρ_1; subsequently it checks whether $T_5^{x_i} = T_4$; if this is the case the agent concludes that user i is the originator of the given $\mathsf{Identify}$ protocol transcript.

Claim. (invoked by the user) Given an $\mathsf{Identify}$ protocol transcript that was generated by user i and contains the sequence $\langle T_1, T_2, T_3, T_4, T_5, T_6, T_7 \rangle$, the user i can claim that he is the originator as follows: he initiates a proof of knowledge of the discrete-log of T_6 base T_7 (which is a discrete-log relation set, see section 4). As a side-note, we remark here that if the proof is directed to a specific entity the proof can be targeted to the receiver using a designated verifier proof, see [23]; such proofs can be easily coupled with our proofs of knowledge for discrete-log relation sets.

Theorem 6. *The traceable identification scheme above is correct according to definition 9 and secure according to definition 10. In particular it satisfies (i) security against misidentification attacks based on the Strong-RSA and the DDH assumptions; (ii) security against anonymity attacks based on the DDH assumption; (iii) security against framing attacks based on the discrete-logarithm problem over $QR(n)$ when the factorization of n is known.*

9 Applications

One immediate application of traceable signatures is membership revocation of the CRL-type. Another motivation for traceable signatures is the development of a generic way to transform any system $\mathcal{S}$ that provides anonymity into a system that provides "fair" or conditional anonymity taking advantage of the various traceability procedures we developed. An anonymity system is comprised of a population of units which, depending on the system's function, exchange messages using anonymous channels. An anonymity system with *fairness* allows the identification of the origin of messages, as well as the tracing of all messages of a suspect unit, if this is mandated by the authorities. A sketch of the idea of using traceable signatures to transform any such an anonymous system into a system with fair anonymity is as follows: each unit of the anonymous system becomes a member of a traceable signature system; any message that is sent by a unit must be signed using the traceable signature mechanism. Messages that are not accompanied by a valid traceable signature are rejected by the recipients. This simple transformation is powerful and generic enough to add "fair" anonymity to a large class of anonymous systems (for example mix-networks).

References

1. M. Abdalla, J. H. An, M. Bellare, and C. Namprempre. From identification to signatures via the Fiat-Shamir transform: Minimizing assumptions for security and forward-security. In L. Knudsen, editor, *Advances in Cryptology – EUROCRYPT 2002*, volume 2332 of *LNCS*, pages 418–433. Springer-Verlag, 2002.
2. G. Ateniese, J. Camenisch, M. Joye, and G. Tsudik. A practical and provably secure coalition-resistant group signature scheme. In M. Bellare, editor, *Advances in Cryptology - CRYPTO 2000*, volume 1880 of *LNCS*, pages 255–270, 2000.
3. G. Ateniese, G. Song, and G. Tsudik. Quasi-efficient revocation of group signatures. In M. Blaze, editor, *Financial Cryptography 2002*, volume 2357 of *LNCS*, pages 183–197, 2002.
4. G. Ateniese and G. Tsudik. Some open issues and new directions in group signatures. In M. Franklin, editor, *Financial Cryptography 1999*, volume 1648 of *LNCS*, pages 196–211. Springer-Verlag, 1999.
5. M. Bellare, D. Micciancio, and B. Warinschi. Foundations of group signatures: Formal definitions, simplified requirements, and a construction based on general assumptions. In E. Biham, editor, *Advances in Cryptology – EUROCRYPT 2003*, volume 2656 of *LNCS*, pages 614–629. Springer, 2003.
6. F. Boudot. Efficient proofs that a committed number lies in an interval. In B. Preneel, editor, *Advances in Cryptology – EUROCRYPT 2000*, volume 1807 of *LNCS*, pages 431–444. Springer-Verlag, 2000.
7. J. Camenisch. Efficient and generalized group signatures. In W. Fumy, editor, *Advances in Cryptology - EUROCRYPT 1997*, volume 1233 of *LNCS*, pages 465–479. Springer, 1997.
8. J. Camenisch and A. Lysyanskaya. An identity escrow scheme with appointed verifiers. In J. Kilian, editor, *Advances in Cryptology - CRYPTO 2001*, volume 2139 of *LNCS*, pages 388–407. Springer, 2001.
9. J. Camenisch and A. Lysyanskaya. Dynamic accumulators and application to efficient revocation of anonymous credentials. In M. Yung, editor, *Advances in Cryptology - CRYPTO 2002*, volume 2442 of *LNCS*, pages 61–76. Springer, 2002.

10. J. Camenisch and A. Lysyanskaya. A signature scheme with efficient protocols. In S. Cimato, C. Galdi, and G. Persiano, editors, *International Conference on Security in Communication Networks – SCN 2002*, volume 2576 of *LNCS*, pages 268–289. Springer Verlag, 2002.
11. J. Camenisch and M. Michels. A group signature scheme based on an RSA-variant. *BRICS Technical Report*, RS98-27, 1998.
12. J. Camenisch and M. Michels. A group signature scheme with improved efficiency. In K. Ohta and D. Pei, editors, *Advances in Cryptology - ASIACRYPT 1998*, volume 1514 of *LNCS*, pages 160–174. Springer, 1998.
13. J. Camenisch and M. Stadler. Efficient group signature schemes for large groups. In B. Kaliski, editor, *Advances in Cryptology — CRYPTO 1997*, LNCS, pages 410–424. Springer-Verlag, 1997.
14. D. Chaum and E. van Heyst. Group signatures. In D. W. Davies, editor, *Advances in Cryptology – EUROCRYPT 1991*, volume 547 of *LNCS*, pages 257–265. Springer-Verlag, 1991.
15. L. Chen and T. P. Pedersen. New group signature schemes (extended abstract). In A. De Santis, editor, *Advances in Cryptology—EUROCRYPT 1994*, volume 950 of *LNCS*, pages 171–181. Springer-Verlag, 1994.
16. R. Cramer, I. Damgård, and B. Schoenmakers. Proofs of partial knowledge and simplified design of witness hiding protocols. In Y. G. Desmedt, editor, *Advances in Cryptology—CRYPTO 1994*, volume 839 of *LNCS*, pages 174–187. Springer-Verlag, 1994.
17. I. Damgård and E. Fujisaki. A statistically-hiding integer commitment scheme based on groups with hidden order. In Y. Zheng, editor, *Advances in Cryptology – ASIACRYPT 2002*, volume 2501 of *LNCS*, pages 125–142. Springer-Verlag, 2002.
18. A. Fiat and A. Shamir. How to prove yourself: Practical solutions to identification and signature problems. In A. Odlyzko, editor, *Advances in Cryptology — CRYPTO 1986*, volume 263 of *LNCS*, pages 186–194. Springer-Verlag, 1987.
19. E. Fujisaki and T. Okamoto. Statistical zero knowledge protocols to prove modular polynomial relations. In B. S. Kaliski, editor, *Advances in Cryptology - CRYPTO 1997*, volume 1294 of *LNCS*, pages 16–30, 1997.
20. O. Goldreich. Secure multi-party computation, manuscript available from the web. http://www.wisdom.weizmann.ac.il/˜oded/ , 1998.
21. J. Groth. Group signatures: revisiting definitions, assumptions and revocation, 2004. manuscript.
22. J. Håstad, A. W. Schrift, and A. Shamir. The discrete logarithm modulo a composite hides O(n) bits. *JCSS: Journal of Computer and System Sciences*, 47(3):376–404, 1993.
23. M. Jakobsson, K. Sako, and R. Impagliazzo. Designated verifier proofs and their applications. In U. M. Maurer, editor, *Advances in Cryptology - EUROCRYPT 1996*, volume 1070 of *LNCS*, pages 143–154, 1996.
24. A. Kiayias, Y. Tsiounis, and M. Yung. Traceable signatures. E-Print Cryptology Archive, Report 2004/007, 2004. http://eprint.iacr.org/.
25. A. Kiayias and M. Yung. Extracting group signatures from traitor tracing schemes. In E. Biham, editor, *Advances in Cryptology – EUROCRYPT 2003*, volume 2656 of *LNCS*, pages 630–648. Springer, 2003.