Anonymous Identification in *Ad Hoc* Groups

Yevgeniy Dodis[1], Aggelos Kiayias[2], Antonio Nicolosi[1], and Victor Shoup[1]

[1] Courant Institute of Mathematical Sciences, New York University, NY, USA
{dodis,nicolosi,shoup}@cs.nyu.edu
[2] Department of Computer Science and Eng., University of Connecticut, CT, USA
aggelos@cse.uconn.edu

Abstract. We introduce *Ad hoc* Anonymous Identification schemes, a new multi-user cryptographic primitive that allows participants from a user population to form *ad-hoc* groups, and then prove membership anonymously in such groups. Our schemes are based on the notion of *accumulator with one-way domain*, a natural extension of cryptographic accumulators we introduce in this work. We provide a formal model for *Ad hoc* Anonymous Identification schemes and design secure such schemes both generically (based on any accumulator with one-way domain) and for a specific efficient implementation of such an accumulator based on the Strong RSA Assumption. A salient feature of our approach is that all the identification protocols take time independent of the size of the ad-hoc group. All our schemes and notions can be generally and efficiently amended so that they allow the recovery of the signer's identity by an authority, if the latter is desired.

Using the Fiat-Shamir transform, we also obtain *constant-size*, signer-ambiguous group and ring signatures (provably secure in the Random Oracle Model). For ring signatures, this is the first such constant-size scheme, as all the previous proposals had signature size proportional to the size of the ring. For group signatures, we obtain schemes comparable in performance with state-of-the-art schemes, with the additional feature that the role of the group manager during key registration is extremely simple and essentially passive: all it does is accept the public key of the new member (and update the constant-size public key of the group).

1 Introduction

Anonymous identification is an oxymoron with many useful applications. Consider the setting, for a known user population and a known set of resources, where a user wants to gain access to a certain resource. In many cases, accessing the resource is an action that *does not* mandate positive identification of the user. Instead, it would be sufficient for the user to prove that he belongs to the subset of the population that is supposed to have access to the resource. This would allow the user to lawfully access the resource while protect his real identity and thus "anonymously identify" himself.

Given the close relationships between identification schemes and digital signatures, one can easily extend the above reasoning to settings where a user produces a signature that is "signer-ambiguous" i.e., such that the verifier is not capable of distinguishing the actual signer among a subgroup of potential signers. In fact, it was in the digital signature setting that such an anonymous scheme was presented for the first time, with the introduction of the group signature model [19], which additionally mandates the presence of a designated party able to reveal the identity of the signer, were the need to arise.

Subsequent work on group signatures and on anonymous identification in general [20, 24, 13, 18, 16, 23, 3, 1, 11, 14, 6, 2] allowed for more efficient designs and formal modelling of the primitive, with the current state of the art being the scheme by Ateniese et al. [1]. In general, existing group signature schemes are derived from their interactive counterpart (*ID Escrow* schemes [32]) via the Fiat-Shamir transform [28].

A related notion, but of slightly different nature, is that of ring signatures, introduced by Rivest, Shamir and Tauman in [34] and further studied in [12, 33]. Ring signatures differ from group signatures in that they allow group formation to happen in an ad-hoc fashion: group must be formed without the help of a group manager; in fact, a user might not even know that he has been included in a certain group. This is in sharp contrast to the group signature setting where the user must execute a Join protocol with the group manager and obtain a group-membership certificate that cannot be constructed without the help of the group manager. Note that ad-hoc group formation in the context of ring signatures is always understood within the context of a user population and an associated PKI. Based on the PKI, ad-hoc subsets of the user population can be formed without the help of a "subset manager"—but it is assumed that every user has a registered public key.

While ring signatures are attractive because they have simple group formation procedures that can be executed by any user individually, they have the shortcoming that the length of the signature is proportional to the group size. For large groups, the length of a ring signature (growing linearly with the group size) will become impractical. To the contrary, schemes with *constant-size* signatures have been successfully designed in the group signature setting [1]. We remark that in the setting of anonymous identification, the counterpart of "signature size" is the bandwidth consumed by the protocol, which is thus an important complexity measure to minimize.

Based on the above discussion, an important open question in the context of anonymous identification and signature schemes, recently posed by Naor in [33], is the following:

> Is it possible to design secure anonymous identification schemes that enable ad-hoc group formation in the sense of ring signatures and at the same time possess constant-size signature (or proof) length?

Our contribution. In this work we provide an affirmative answer to the above question. Specifically, we introduce a new primitive called *Ad hoc* Anonymous Identification schemes; this is a family of schemes where participants from a user population can form groups in ad-hoc fashion (without the help of a group manager) and then get anonymously identified as members of such groups.

Our main tool in the construction of *Ad hoc* Anonymous Identification schemes is a new cryptographic primitive, *accumulator with one-way domain*, which extends the notion of a collision-resistant accumulator [7, 4, 15]. In simple terms, in an accumulator with one-way domain, the set of values that can be accumulated are associated with a "witness space" such that it is computationally intractable to find witnesses for random values in the accumulator's domain.

First, we demonstrate the relationship between such accumulators and *Ad hoc* Anonymous Identification schemes by presenting a generic construction based on *any* accumulator with one-way domain. Second, we design an efficient implementation of accumulator with a one-way domain based on the Strong RSA Assumption, from which we obtain a more efficient construction of *Ad hoc* Anonymous Identification scheme whose security rests upon the Strong RSA Assumption.

We remark that previous work on anonymous identification that allowed subset queries was done by Boneh and Franklin [8]. They define a more limited security model, and show a protocol which imposes on both parties a computational load proportional to the subset size at each run. Moreover, their scheme is susceptible to collusion attacks (both against the soundness and against the anonymity of the scheme) that do not apply to our setting.

In our Strong-RSA-based *Ad hoc* Anonymous Identification scheme, the computational and communication complexity on both ends is constant in the size of the group. Thus, the signature version of our ad-hoc anonymous identification scheme yields a ring signature with constant size signatures (over a dedicated PKI). Other applications of our scheme include "ad-hoc" group signatures (group signature schemes where the group manager can be offline during the group formation) and identity escrow over ad-hoc groups.

Recently, work by Tsudik and Xu [35], building on the work by Camenisch and Lysyanskaya [15], investigated techniques to obtain more flexible dynamic accumulators, on which to base group signature schemes (which is one of our applications). The specific method used by [35] bears many similarities with our Strong-RSA-based instantiation, with some important differences. Namely, in their solution anonymity revocation takes time proportional to the user population, due to subtle problems concerning the accumulation of composite values inside the accumulator. Our work resolves this technical problem. Moreover, we present a new notion of *Ad hoc* Anonymous Identification scheme, which has more applications than those specific to group signature schemes: for example, they allow us to build the first constant-size ring signature schemes. We present a general construction for our primitives from any accumulator and not just the one of [15]. Last, our formal definitional framework is of independent interest.

2 Preliminaries

2.1 NP-Relations and Σ-Protocols

Throughout the paper, we assume familiarity with the GMR notation [30].

An **NP**-relation R is a relation over bitstrings for which there is an efficient algorithm to decide whether $(x,w) \in R$ in time polynomial in the length of x. The **NP**-language L_R associated to R is defined as $L_R \doteq \{x \mid (\exists w)[(x,w) \in R]\}$

A Σ-protocol [22, 21] for an **NP**-relation R is an efficient 3-round two-party protocol, such that for every input (x,w) to P and x to V, the first P-round yields a *commitment* message, the subsequent V-round replies with a random *challenge* message, and the last P-round concludes by sending a *response* message. At the end of a run, V outputs a 0/1 value, functionally dependent on x and the transcript π only. Additionally, a Σ-protocol satisfies *Special Soundness*, meaning that for any $(x,w) \notin R$ and any commitment message, there is at most one pair of challenge/response messages for which V would output 1; and *Special Honest-Verifier Zero-Knowledge*, meaning that there is an efficient algorithm (called a *Simulator*) that on input $x \in L_R$ and any challenge message, outputs a pair of commitment/response messages for which V would output 1.

The main result we will need about Σ-protocols is the following:

Theorem 1 ([29, 27]). *A Σ-protocol for any* **NP***-relation can be efficiently constructed if one-way functions exist.*

2.2 Accumulators

An *accumulator family* is a pair $(\{F_\lambda\}_{\lambda\in\mathbb{N}}, \{X_\lambda\}_{\lambda\in\mathbb{N}})$, where $\{F_\lambda\}_{\lambda\in\mathbb{N}}$ is a sequence of families of functions such that each $f \in F_\lambda$ is defined as $f : U_f \times X_f^{\mathsf{ext}} \to U_f$ for some $X_f^{\mathsf{ext}} \supseteq X_\lambda$ and additionally the following properties are satisfied:

- (efficient generation) There exists an efficient algorithm G that on input a security parameter 1^λ outputs a random element f of F_λ, possibly together with some auxiliary information a_f.
- (efficient evaluation) Any $f \in F_\lambda$ is computable in time polynomial in λ.
- (quasi-commutativity) For all $\lambda \in \mathbb{N}$, $f \in F_\lambda$, $u \in U_f$, $x_1, x_2 \in X_\lambda$,

$$f(f(u,x_1),x_2) = f(f(u,x_2),x_1)$$

We will refer to $\{X_\lambda\}_{\lambda\in\mathbb{N}}$ as the *value domain* of the accumulator. For any $\lambda \in \mathbb{N}, f \in F_\lambda$ and $X = \{x_1,\ldots,x_s\} \subset X_\lambda$, we will refer to $f(\ldots f(u,x_1)\ldots,x_s)$ as the *accumulated value* of the set X over u: due to quasi-commutativity, such value is independent of the order of the x_i's and will be denoted by $f(u,X)$.

Definition 1. *An accumulator is said to be* collision resistant *if for any $\lambda \in \mathbb{N}$ and any adversary $\mathcal{A}$:*

$$\begin{aligned} Pr[f \stackrel{R}{\leftarrow} F_\lambda; u \stackrel{R}{\leftarrow} U_f; (x,w,X) \stackrel{R}{\leftarrow} \mathcal{A}(f,U_f,u) \quad | \\ (X \subseteq X_\lambda) \wedge (w \in U_f) \wedge (x \in X_f^{\mathsf{ext}} \setminus X) \wedge (f(w,x) = f(u,X))] = \nu(\lambda) \end{aligned}$$

For $\lambda \in \mathbb{N}$ and $f \in F_\lambda$, we say that $w \in U_f$ is a *witness* for the fact that $x \in X_\lambda$ has been accumulated within $v \in U_f$ (or simply that w is a witness for x in v) whenever $f(w,x) = v$. We extend the notion of witness for a set of values $X = \{x_1,\ldots,x_s\}$ in a straightforward manner.

Accumulators with One-Way Domain. An *accumulator with one-way domain* is a quadruple $(\{F_\lambda\}_{\lambda\in\mathbb{N}}, \{X_\lambda\}_{\lambda\in\mathbb{N}}, \{Z_\lambda\}_{\lambda\in\mathbb{N}}, \{R_\lambda\}_{\lambda\in\mathbb{N}})$, such that the pair $(\{F_\lambda\}_{\lambda\in\mathbb{N}}, \{X_\lambda\}_{\lambda\in\mathbb{N}})$ is a collision-resistant accumulator, and each R_λ is a relation over $X_\lambda \times Z_\lambda$ with the following properties:

- (efficient verification) There exists an efficient algorithm D that on input $(x, z) \in X_\lambda \times Z_\lambda$, returns 1 if and only if $(x, z) \in R_\lambda$.
- (efficient sampling) There exists a probabilistic algorithm W that on input 1^λ returns a pair $(x, z) \in X_\lambda \times Z_\lambda$ such that $(x, z) \in R_\lambda$. We refer to z as a *pre-image* of x.
- (one-wayness) It is computationally hard to compute any pre-image z' of an x that was sampled with W. Formally, for any adversary $\mathcal{A}$:

$$Pr[(x,z) \overset{\mathrm{R}}{\leftarrow} W(1^\lambda); z' \overset{\mathrm{R}}{\leftarrow} \mathcal{A}(1^\lambda, x) \mid (x, z') \in R_\lambda] = \nu(\lambda)$$

2.3 The Strong RSA Assumption

We briefly review some definitions [7, 4] regarding the computational assumption underlying our efficient construction in Section 5.

A number n is an *RSA integer* if $n = pq$ for distinct primes p and q such that $|p| = |q|$. For $\lambda \in \mathbf{N}$, let RSA_λ be the set of RSA integers of size λ. A number p is a *safe prime* if $p = 2p' + 1$ and both p and p' are odd primes. A number n is a *rigid integer* if $n = pq$ for distinct safe primes p and q such that $|p| = |q|$. For $\lambda \in \mathbf{N}$, let Rig_λ be the set of λ-bit rigid integers.

Definition 2 (Strong RSA Assumption, [4]).
For any integer λ and for any adversary $\mathcal{A}$:

$$Pr[n \overset{R}{\leftarrow} \mathsf{Rig}_\lambda; z \overset{R}{\leftarrow} \mathbb{Z}_n^*; (x', y') \overset{R}{\leftarrow} \mathcal{A}(1^\lambda, n, z) \mid \big(y' > 1\big) \wedge \big((x')^{y'} \equiv z(n)\big)] < \nu(\lambda)$$

the probability being over the random choice of n and z, and $\mathcal{A}$'s random coins.

3 *Ad hoc* Anonymous Identification scheme

3.1 Syntax

An *Ad hoc* Anonymous Identification scheme is a six-tuple of efficient algorithms (Setup, $\mathsf{Register}$, $\mathsf{Make\text{-}GPK}$, $\mathsf{Make\text{-}GSK}$, $\mathsf{Anon\text{-}ID^P}$, $\mathsf{Anon\text{-}ID^V}$), where:

- Setup initializes the state of the system: on input a security parameter 1^λ, Setup creates a public database DB (that will be used to store information about the users' public keys), and then generates the system's parameters param; its output implicitly defines a domain of possible global parameters.
- $\mathsf{Register}$, the *registration algorithm*, allows users to initially register with the system. On input the system's parameters param and the identity of the new user u (from a suitable universe of users' identity $\mathcal{U}$), $\mathsf{Register}$ returns a secret key/public key pair (sk, pk). To complete the subscription process,

the user then sends his public key to a bulletin board for inclusion in a public database DB.
The $\mathsf{Register}$ algorithm implicitly defines a domain $\mathcal{SK}$ of possible user secret keys and a domain $\mathcal{PK}$ of possible user public keys; its output induces a relation over user secret key/public key pairs, that we will denote by $\rightleftharpoons$. We also require a superset $\mathcal{PK}' \supseteq \mathcal{PK}$ to be specified, such that membership to $\mathcal{PK}'$ can be tested in polynomial time.

– $\mathsf{Make\text{-}GPK}$, the *group public key construction algorithm*, is a deterministic algorithm used to combine a set of user public keys S into a single group public key gpk_S, suitable for use in the $\mathsf{Anon\text{-}ID}$ protocol described below. Syntactically, $\mathsf{Make\text{-}GPK}$ takes as input param and a set $S \subseteq \mathcal{PK}'$; its output implicitly defines a domain $\mathcal{GPK}$ of possible group public keys. We also require a superset $\mathcal{GPK}' \supseteq \mathcal{GPK}$ to be specified, such that membership to $\mathcal{GPK}'$ can be tested in polynomial time.
The $\mathsf{Make\text{-}GPK}$ algorithm shall run in time linear in the number of public keys being aggregated; we also remark here that our definition forces $\mathsf{Make\text{-}GPK}$ to be *order-independent* i.e., the order in which the public keys to be aggregated are provided shall not matter.

– $\mathsf{Make\text{-}GSK}$, the *group secret key construction algorithm*, is a deterministic algorithm used to combine a set of user public keys S', along with a secret key/public key pair (sk_u, pk_u), into a single group secret key gsk_u, suitable for use in the $\mathsf{Anon\text{-}ID}$ protocol described below.
$\mathsf{Make\text{-}GSK}$ takes as input param, a set $S' \subseteq \mathcal{PK}'$ and a key pair (sk_u, pk_u) satisfying $sk_u \rightleftharpoons pk_u$, and it shall run in time proportional to the size of S'. Its output implicitly defines a domain $\mathcal{GSK}$ of possible group secret keys.
The $\mathsf{Make\text{-}GPK}$ and $\mathsf{Make\text{-}GSK}$ algorithms can be used to extend the $\rightleftharpoons$-relation to $\mathcal{GSK} \times \mathcal{GPK}$, as follows: A group secret key $gsk \doteq \mathsf{Make\text{-}GSK}(\mathsf{param}, S', (sk, pk))$ is in $\rightleftharpoons$-relation with a group public key $gpk \doteq \mathsf{Make\text{-}GPK}(\mathsf{param}, S)$ if and only if $S = S' \cup \{pk\}$. Observe that even in the case that the $\rightleftharpoons$-relation is one-to-one over $\mathcal{SK} \times \mathcal{PK}$, it is usually many-to-one over $\mathcal{GSK} \times \mathcal{GPK}$, as more than one group secret key correspond to the same group public key.

– $\mathsf{Anon\text{-}ID} \doteq (\mathsf{Anon\text{-}ID}^{\mathsf{P}}, \mathsf{Anon\text{-}ID}^{\mathsf{V}})$, the *Anonymous Identification Protocol*, is an efficient two-party protocol, in which both $\mathsf{Anon\text{-}ID}^{\mathsf{P}}$ (the *prover*) and $\mathsf{Anon\text{-}ID}^{\mathsf{V}}$ (the *verifier*) get in input the system's parameters param and a group public key gpk (corresponding to some set S of user public keys i.e., $gpk \doteq \mathsf{Make\text{-}GPK}(\mathsf{param}, S)$); $\mathsf{Anon\text{-}ID}^{\mathsf{P}}$ is also given a group secret key gsk as an additional input.
Any execution of the $\mathsf{Anon\text{-}ID}$ protocol shall complete in time independent from the number of public keys that were aggregated when constructing gpk and/or gsk; at the end of each protocol run, $\mathsf{Anon\text{-}ID}^{\mathsf{V}}$ outputs a 0/1-valued answer.

Correctness. For correctness, we require that any execution of the $\mathsf{Anon\text{-}ID}$ protocol in which the additional input to $\mathsf{Anon\text{-}ID}^{\mathsf{P}}$ is a group secret key gsk $\rightleftharpoons$-related to the common input gpk, shall terminate with $\mathsf{Anon\text{-}ID}^{\mathsf{V}}$ outputting a 1 answer, with overwhelming probability.

<table>
<tr><th>Honest user registration oracle $\mathcal{O}_{\mathsf{HReg}}$</th><th>User corruption oracle $\mathcal{O}_{\mathsf{Cor}}$</th></tr>
<tr><td>IN: $u \in \mathcal{U}$
RUN: 1. $(sk_u, pk_u) \xleftarrow{R} \mathsf{Register}(\mathsf{param}, u)$
2. $\mathsf{DB.Store}(sk_u, pk_u)$
OUT: pk_u</td><td>IN: $pk_u \in \mathcal{PK}'$
RUN: 1. $sk_u \leftarrow \mathsf{DB.Lookup}(pk_u)$
/* $sk_u \leftarrow \perp$ if no match found */
OUT: sk_u</td></tr>
</table>

<table>
<tr><th>Transcript oracle $\mathcal{O}_{\mathsf{Scr}}$</th></tr>
<tr><td>IN: $S' \subseteq \mathcal{PK}', pk_u \in \mathcal{PK}'$
RUN: 1. $sk_u \leftarrow \mathsf{DB.Lookup}(pk_u)$
2. <code>if</code> $sk_u = \perp$
3. <code>then</code> $\pi \leftarrow \perp$
4. <code>else</code> $gpk \leftarrow \mathsf{Make\text{-}GPK}(\mathsf{param}, S' \cup \{pk_u\})$
5. $gsk \leftarrow \mathsf{Make\text{-}GSK}(\mathsf{param}, S', (sk_u, pk_u))$
6. $\pi \xleftarrow{R} \mathsf{Anon\text{-}ID}^{\mathsf{P}}(\mathsf{param}, gpk, gsk) \leftrightarrow \mathsf{Anon\text{-}ID}^{\mathsf{V}}(\mathsf{param}, gpk)$
OUT: π</td></tr>
</table>

Fig. 1. Oracles for the soundness attack game. DB denotes a database storing user secret key/public key pairs, indexed by public key.

3.2 Soundness

The Attack Game. We formalize the soundness guarantees that we require from an *Ad hoc* Anonymous Identification scheme in terms of a game being played between an honest dealer and an adversary $\mathcal{A}$. In this game, the adversary is allowed to interact with three oracles $\mathcal{O}_{\mathsf{HReg}}$ (the *honest user registration oracle*), $\mathcal{O}_{\mathsf{Cor}}$ (the *user corruption oracle*), and $\mathcal{O}_{\mathsf{Scr}}$ (the *transcript oracle*) (see Fig. 1).

The game begins with the honest dealer running the Setup algorithm for the security parameter 1^λ, and handing the resulting global parameters param to the adversary. Then, $\mathcal{A}$ arbitrarily interleaves queries to the three oracles, according to any adaptive strategy she wishes: eventually, she outputs a *target group* $S^* \subseteq \mathcal{PK}'$. At this point, $\mathcal{A}$ starts executing, in the role of the prover, a run of the Anon-ID protocol with the honest dealer, on common inputs param and $gpk^* \doteq \mathsf{Make\text{-}GPK}(\mathsf{param}, S^*)$. Notice that during such interaction, the adversary is still allowed to query the three oracles $\mathcal{O}_{\mathsf{Reg}}, \mathcal{O}_{\mathsf{Scr}}$ and $\mathcal{O}_{\mathsf{Cor}}$. Let $\tilde{\pi}$ be the transcript resulting from such run of the Anon-ID protocol. $\mathcal{A}$ wins the game if the following conditions hold:

1. for all $pk^* \in \mathcal{S}^*$, there is an entry indexed by pk^* in the SK-DB Database, **and**
2. $\tilde{\pi}$ is a valid transcript i.e., the run completed with the honest dealer outputting 1, **and**
3. for all $pk^* \in \mathcal{S}^*$, $\mathcal{A}$ never queried $\mathcal{O}_{\mathsf{Cor}}$ on input pk^*;

Define $\mathsf{Succ}^{\mathsf{Imp}}_{\mathcal{A}}(\lambda)$ to be the probability that $\mathcal{A}$ wins the above game.

Definition 3. *An* Ad hoc *Anonymous Identification scheme is sound against passive chosen-group attacks if any adversary $\mathcal{A}$ has negligible advantage to win the above game:*

$$(\forall \lambda \in \mathbf{N})(\forall PPT \mathcal{A})[\mathsf{Succ}^{\mathit{Snd}}_{\mathcal{A}}(\lambda) \leq \nu(\lambda)]$$

Challenge oracle $\mathcal{O}_{\mathsf{Ch}}$	
IN:	$S' \subseteq \mathcal{PK}', (sk_0, pk_0), (sk_1, pk_1)$
RUN:	1. $b^* \stackrel{\text{R}}{\leftarrow} \{0,1\}$
	2. `if` $sk_0 \not\Leftrightarrow pk_0$ `or` $sk_1 \not\Leftrightarrow pk_1$ `then abort`
	3. $gpk \leftarrow \mathsf{Make\text{-}GSK}(\mathsf{param}, S' \cup \{pk_0, pk_1\})$
	4. $gsk^* \leftarrow \mathsf{Make\text{-}GSK}(\mathsf{param}, S' \cup \{pk_{1-b^*}\}, (sk_{b^*}, pk_{b^*}))$
	5. $\pi^* \stackrel{\text{R}}{\leftarrow} \mathsf{Anon\text{-}ID}^{\mathsf{P}}(\mathsf{param}, gpk, gsk^*) \leftrightarrow \mathsf{Anon\text{-}ID}^{\mathsf{V}}(\mathsf{param}, gpk)$
OUT:	π^*

Fig. 2. The oracle for the anonymity attack game.

A Note on Active Security. Our definition of soundness models an adversary that, in her attempt to fool an honest verifier into accepting a "fake" run of the Anon-ID protocol, can actively (and, in fact, adaptively) corrupt users, but can only passively eavesdrop the communication between honest provers and verifiers. One could, of course, define stronger notions of security by considering active, concurrent or even reset attacks, along the lines of previous work on Identification Schemes [26, 5]; however, we refrain from doing so, both to keep the presentation simpler, and because the main application of our *Ad hoc* Anonymous Identification schemes is to obtain new ring and group signatures scheme by means of the Fiat-Shamir Heuristic (see Section 6.3), for which security against a passive adversary suffices.

3.3 Anonymity

The Attack Game. We formalize the anonymity guarantees that we require from an *Ad hoc* Anonymous Identification scheme in terms of a game being played between an honest dealer and an adversary $\mathcal{A}$. In this game, the adversary is allowed to interact only once with a "challenge" oracle $\mathcal{O}_{\mathsf{Ch}}$, described in Fig. 2.

The game begins with the honest dealer running the Setup algorithm for the security parameter 1^λ, and handing the resulting global parameters param to the adversary. Then, the adversary $\mathcal{A}$ creates as many user secret key/public key pairs as she wishes, and experiments with the Make-GPK, Make-GSK, $\mathsf{Anon\text{-}ID}^{\mathsf{P}}$ and $\mathsf{Anon\text{-}ID}^{\mathsf{V}}$ algorithms as long as she deems necessary; eventually, she queries the $\mathcal{O}_{\mathsf{Ch}}$ oracle, getting back a "challenge" transcript π^*. The adversary then continues experimenting with the algorithms of the system, trying to infer the random bit b^* used by the oracle $\mathcal{O}_{\mathsf{Ch}}$ to construct the challenge π^*; finally, $\mathcal{A}$ outputs a single bit $\tilde{b}$, her best guess to the "challenge" bit b^*.

Define $\mathsf{Succ}_{\mathcal{A}}^{\mathsf{Anon}}(\lambda)$ to be the probability that the bit $\tilde{b}$ output by $\mathcal{A}$ at the end of the above game is equal to the random bit b^* used by the $\mathcal{O}_{\mathsf{Ch}}$ oracle.

Definition 4. *An* Ad hoc *Anonymous Identification scheme is* fully anonymizing *if any probabilistic, polynomial-time adversary $\mathcal{A}$ has success probability at most negligibly greater than one half:*

$$(\forall \lambda \in \mathbf{N})(\forall PPT \mathcal{A})\left[\left|\mathsf{Succ}_{\mathcal{A}}^{Anon}(\lambda) - \frac{1}{2}\right| \leq \nu(\lambda)\right]$$

3.4 Extensions

Identity Escrow. In some scenarios, complete anonymity might create more security concerns than what it actually solves. Instead, some degree of "limited anonymity", not hindering user accountability, would be preferable. In our context, this can be achieved with the help of a trusted *Identity Escrow Authority*, or IEA (also called *Anonymity Revocation Manager* elsewhere [15]), endowed with the capability of "reading" the identity of the prover "between the lines" of any transcript produced by a run of the Anon-ID protocol.

To enable such escrow capability, the definition of *Ad hoc* Anonymous Identification scheme from Section 3.1 is modified as follows:

- The Setup algorithm is run by the IEA, and it additionally outputs an identity escrow key sk_{IE} (from some domain $\mathcal{SK}_{\mathsf{IE}}$), which the IEA keeps for himself.
- Register is replaced by an efficient two-party protocol $(\mathsf{Register}^{\mathsf{user}}, \mathsf{Register}^{\mathsf{IEA}})$, meant to be run between the prospective user and the IEA, at the end of which the IEA learns the user's newly generated public key pk_u (possibly along with some other information aux_u about u that the IEA stores in a public registry database DB), but he doesn't learn anything about the corresponding secret key sk_u.
- An additional (deterministic) Extract algorithm is defined, which takes as input a transcript π (for the Anon-ID protocol), along with the Identity Escrow secret key sk_{IE} and the registry database DB, and returns a public key $pk \in \mathcal{PK}'$ or one of the special symbols $\perp$ and ?. Intuitively, the algorithm should be able to recover the identity of the user who participated as the prover in the run of the Anon-ID protocol that produced π as transcript; the symbol $\perp$ should be output when π is ill-formed (e.g., when π comes from a ZK simulator), whereas ? indicates failure to trace the correct identity.

Our definitions of the security properties of the system have to be adjusted, since we now have an additional functionality that the adversary may try to attack; moreover, the presence of the IEA may open new attack possibilities to the adversary.

The security requirements for the new Extract algorithm are formalized by augmenting the attack scenario defining the soundness property (Section 3.2). In this new, combined game, the adversary initially gets the IEA's secret key sk_{IE}, along with the public parameters param of the system. Then, the game proceeds as described in Section 3.2, except that we loosen the conditions under which the adversary is considered to win the game, substituting the last two caveats with the following:

2′. $\tilde{\pi}$ is a valid transcript i.e., $\mathsf{Extract}(\tilde{\pi}, sk_{\mathsf{IE}}, \mathsf{DB}) \neq \perp$ **and**
3′. for all $pk^* \in \mathcal{S}^*$, either $\mathsf{Extract}(\tilde{\pi}, sk_{\mathsf{IE}}, \mathsf{DB}) \neq pk^*$, **or** $\mathcal{A}$ never queried $\mathcal{O}_{\mathsf{Cor}}$ on input pk^*;

As for the anonymity property, the definition from Section 3.3 is changed in that the adversary is now given access to two more oracles (beside the challenge oracle $\mathcal{O}_{\mathsf{Ch}}$): a *corrupted-user registration oracle* $\mathcal{O}_{\mathsf{CReg}}() \doteq$

$\mathsf{Register}^{\mathsf{IEA}}(sk_{\mathsf{IE}}, \mathsf{param}, \mathsf{DB})$, and a *user identity extraction oracle* $\mathcal{O}_{\mathsf{Xtr}}(\cdot) \doteq \mathsf{Extract}(\cdot, sk_{\mathsf{IE}}, \mathsf{DB})$. The adversary wins the game if she successfully guesses the random bit chosen by the challenge oracle $\mathcal{O}_{\mathsf{Ch}}$, without ever submitting the challenge transcript π^* to the extraction oracle $\mathcal{O}_{\mathsf{Xtr}}$.

Supporting Multiple Large *Ad Hoc* Groups. In many applications where *Ad hoc* Anonymous Identification schemes could be useful, new *ad hoc* groups are often created as supersets of existing ones: for example, if *ad hoc* groups are used to enforce access control, new users may be added to the group of principals authorized to access a given resource. In such cases, the ability to "augment" a group public key with the a new user's public key can be very handy, especially if coupled with algorithms to efficiently create the corresponding group secret key for the new user, and to update the group secret keys for the existing users. Our model can be easily amended to capture this *incremental* functionality; we refer the reader to the full version of this paper [25] for the details.

4 Generic Construction

In this section, we will establish the fact that the existence of accumulators with one way domain implies the existence of *Ad hoc* Anonymous Identification schemes. Below we describe how the algorithms (Setup, Register, Make-GPK, Make-GSK, $\mathsf{Anon\text{-}ID}^{\mathsf{P}}$, $\mathsf{Anon\text{-}ID}^{\mathsf{V}}$) can be implemented given an accumulator with one-way domain $(\{F_\lambda\}_{\lambda\in\mathbb{N}}, \{X_\lambda\}_{\lambda\in\mathbb{N}}, \{Z_\lambda\}_{\lambda\in\mathbb{N}}, \{R_\lambda\}_{\lambda\in\mathbb{N}},)$.

- Setup executes the accumulator generation algorithm G on 1^λ to obtain $f \in F_\lambda$. Then it samples U_f to obtain $u \in_R U_f$. Setup terminates by setting $\mathsf{param} := (\lambda, u, f, D, W)$, where D and W are polynomial-time algorithms respectively to decide and to sample the relation R_λ.
- Register first samples a pair $(x, z) \in X_\lambda \times Z_\lambda$ such that $(x, z) \in R_\lambda$ using the sampling algorithm W of the relation R_λ on input 1^λ. Then, Register outputs $sk \doteq z$ (the user secret key) and $pk \doteq x$ (the user public key). Observe that $\mathcal{SK}' = \mathcal{SK} \doteq Z_\lambda$, $\mathcal{PK}' = X_f^{\mathsf{ext}}$ and $\mathcal{PK} \doteq X_\lambda$.
- Make-GPK operates as follows: given a set of user public keys $S = \{x_1, \ldots, x_t\}$ and the parameters (λ, u, f, D), it sets the group public key of S to be the (unique) accumulated value of S over u i.e., $gpk_S \doteq f(u, S)$. Note that thanks to the quasi-commutativity property of f, Make-GPK is indeed order-independent.
- Make-GSK operates as follows: given the set of user public keys $S' \doteq \{x_1, \ldots, x_t\}$, a user secret key/public key pair (z, x) and the system parameters $\mathsf{param} = (\lambda, u, f, D, W)$, it first computes the accumulated value $w \doteq f(u, S')$, and then sets the group secret key gsk to be the tuple (x, z, w) (where $S \doteq S' \cup \{x\}$). Observe that w is a witness for x in $f(u, S')$, and that $\mathcal{GSK} \doteq X_\lambda \times Z_\lambda \times U_f$ and $\mathcal{GPK} \doteq U_f$.
- $\mathsf{Anon\text{-}ID}^{\mathsf{P}}$ and $\mathsf{Anon\text{-}ID}^{\mathsf{V}}$ are obtained generically as the Σ-protocol corresponding to the following **NP**-relation $\mathcal{R}_{\mathsf{param}} \subset \mathcal{GPK} \times \mathcal{GSK}$:

$$\mathcal{R}_{\mathsf{param}} \doteq \big\{ \big(v, (x, z, w)\big) \mid \big((x, z) \in R_\lambda\big) \wedge \big(f(w, x) = v\big) \big\}$$

It is easy to see that the above relation is polynomial-time verifiable: indeed, given v and (x, z, w), one can check in time polynomial in $|v|$ whether $(x, z) \in R_\lambda$ (by verifying that $D(x, z) = 1$), and whether w is indeed a witness for x in v (by verifying that $f(w, x) = v$). Thus, by Theorem 1, we can construct a Σ-protocol (P, V) for the **NP**-relation $\mathcal{R}_{\mathsf{param}}$. In the resulting protocol, the common input to the prover and the verifier is the accumulated value v (i.e. a group public key) and the additional input to the prover is a tuple of the form (x, z, w) (i.e., a group secret key). Hence, the protocol (P, V) meets the specification of the Anon-ID protocol.

As for the correctness of the above construction, observe that relation $\mathcal{R}_{\mathsf{param}}$ is essentially equivalent to the $\rightleftharpoons$ relation. Consequently, a prover holding a group secret key $gsk \doteq (x, z, w)$ $\rightleftharpoons$-related to the group public key $gpk \doteq v$ given as input to the verifier, possesses a tuple belonging to the relation $\mathcal{R}_{\mathsf{param}}$, so that the execution of the Anon-ID protocol will terminate with the verifier outputting 1, with overwhelming probability.

Soundness. Intuitively, the soundness of the above generic construction stems from the following considerations. The Special Honest-Verifier Zero-Knowledge property of the Σ-protocol Anon-ID guarantees that the Transcript Oracle doesn't leak any information to the adversary that she could not compute herself. By the Special Soundness property, in order to make the honest dealer accept (with non-negligible probability) a run of the Anon-ID protocol in which the group public key $gpk \doteq v$ consists solely of the aggregation of public keys of non-corrupted users, $\mathcal{A}$ should posses a tuple $gsk \doteq (x, z, w)$ such that $(x, z) \in R_\lambda$ and w is a witness of x in v. Now, the collision resistance of the accumulator implies that the user public key x must indeed have been accumulated within v, which means (by the third caveat of the soundness attack game in Section 3.2) that x belongs to a non-corrupted user. Hence, the adversary didn't obtain the pre-image z via the user corruption oracle, which implies that $\mathcal{A}$ was able to find it by herself, contradicting the one-wayness of the accumulator's domain.

The above intuition could be turned into a rigorous reduction argument: we refer the reader to the full version [25] for a formal proof.

Anonymity. In attacking the anonymity of the proposed scheme, the adversary basically chooses a group public key $gpk \doteq v$ and two group secret keys $gsk_1 \doteq (x_1, z_1, w_1)$ and $gsk_2 \doteq (x_2, z_2, w_2)$, both $\rightleftharpoons$-related to gpk. To subvert anonymity, the adversary should then be able (cfr. Section 3.3) to tell whether gsk_1 or gsk_2 was used in producing the (honest) "challenge" transcript. Since in the generic construction above the Anon-ID protocol is implemented as a Σ-protocol, this would mean that the adversary is able to tell which "witness" (gsk_1 or gsk_2) was used by the prover to show that v belongs to the **NP**-language $\mathcal{L}_{\mathsf{param}}$ associated to the **NP**-relation $\mathcal{R}_{\mathsf{param}}$. In other words, a successful adversary would break the Witness Indistinguishability of the Anon-ID protocol, which contradicts the fact that Anon-ID enjoys Special Honest-Verifier Zero-Knowledge.

The reader is referred to [25] for a formalization of the above argument.

4.1 Adding ID Escrow

The generic construction described above can be extended to deal with Identity Escrow as follows. During the initialization, the Setup algorithm additionally runs the key generation algorithm $\mathcal{K}$ of some $\mathsf{CCA2}$-secure encryption scheme $(\mathcal{K}, \mathcal{E}, \mathcal{D})$. The resulting public key pk_{IE} is included in the system parameters param, and the secret key sk_{IE} is given to the Identity Escrow Authority (IEA).

As for the user registration phase, each new user, after choosing his user secret key/public key pair $(sk, pk) \doteq (z, x)$, registers his public key with the IEA, which simply stores his identity and public key in a publicly-available database DB.

The $\mathsf{Anon\text{-}ID}$ protocol is also changed to be the Σ-protocol corresponding to the following $\mathbf{NP}$-relation $\mathcal{R}^{\mathsf{IE}}_{\mathsf{param}}$:

$$\mathcal{R}^{\mathsf{IE}}_{\mathsf{param}} \doteq \big\{\big((v, \psi), (x, z, w)\big) \mid \big((x, z) \in R_\lambda\big) \wedge \big(f(w, x) = v\big) \wedge \big(\psi \text{ decrypts to } x\big)\big\}$$

In other words, the prover now additionally encrypts his public key x under the IEA's public key pk_{IE}, and proves to the verifier that he did so correctly.

Finally, the $\mathsf{Extract}$ algorithm, on input a transcript π, recovers the ciphertext ψ from π and decrypts ψ, thus obtaining the identity of the user that played the role of the prover.

It is not hard to check that the above changes do not affect the soundness and anonymity properties of the generic construction: in particular, the $\mathsf{CCA2}$-security of the encryption scheme (which is needed since a malicious party could trick the IEA into acting as a decryption oracle) guarantees that honest transcripts cannot be modified so as to alter the prover identity hidden under the ciphertext ψ. See [25] for a security analysis of the extended scheme.

5 Efficient Implementation

5.1 Construction of an Accumulator with One-way Domain

An efficient construction of a collision-resistant accumulator was presented in [15], based on earlier work by [4] and [7]. Based on this construction, we present an efficient implementation of an accumulator with one-way domain.

For $\lambda \in \mathbb{N}$, the family F_λ consists of the exponentiation functions modulo λ-bit rigid integers:

$$f : (\mathbb{Z}_n^*)^2 \times \mathbb{Z}_{n/4} \to (\mathbb{Z}_n^*)^2$$
$$f : (u, x) \mapsto u^x \bmod n$$

where $n \in \mathsf{Rig}_\lambda$ and $(\mathbb{Z}_n^*)^2$ denotes the set of quadratic residues modulo n.

The accumulator domain $\{X_\lambda\}_{\lambda \in \mathbb{N}}$ is defined by:

$$X_\lambda \doteq \left\{e \text{ prime } \mid \left(\frac{e-1}{2} \in \mathsf{RSA}_\ell\right) \wedge \left(e \in S(2^\ell, 2^\mu)\right)\right\}$$

where $S(2^\ell, 2^\mu)$ is the integer range $(2^\ell - 2^\mu, 2^\ell + 2^\mu)$ that is embedded within $(0, 2^\lambda)$ with $\lambda - 2 > \ell$ and $\ell/2 > \mu + 1$. The pre-image domain $\{Z_\lambda\}_{\lambda\in\mathbb{N}}$ and the one-way relation $\{R_\lambda\}_{\lambda\in\mathbb{N}}$ are defined as follows:

$$Z_\lambda \doteq \{(e_1, e_2) \mid e_1, e_2 \text{ are distinct } \ell/2\text{-bit primes and } e_2 \in S(2^{\ell/2}, 2^\mu)\}$$
$$R_\lambda \doteq \{\big(x, (e_1, e_2)\big) \in X_\lambda \times Z_\lambda \mid \big(x = 2e_1e_2 + 1\big)\}$$

The collision resistance of the above construction can be based on the Strong RSA Assumption, as showed in [15]. Regarding the added one-wayness of the domain, assuming the hardness of factoring RSA integers, it is easy to see that the **NP**-relation R_λ satisfies our one-wayness requirement (cfr. Section 2.2): hence, the above construction yields a secure accumulator with one-way domain.

5.2 Efficient proof of witnesses for the Accumulator

The generic construction described in Section 4 derives algorithms $\mathsf{Anon\text{-}ID}^{\mathsf{P}}$ and $\mathsf{Anon\text{-}ID}^{\mathsf{V}}$ from the Σ-protocol corresponding to some **NP**-relation $\mathcal{R}_{\mathsf{param}}$: for our RSA-based accumulator with one-way domain, the relation is defined as:

$$\mathcal{R}^{RSA}_{\mathsf{param}} \doteq \Big\{\big(v, (x, (e_1, e_2), w)\big) \mid \big(w^x \equiv v \bmod n\big) \wedge \big(x \in S(2^\ell, 2^\mu)\big) \\ \wedge \big(x - 1 = 2e_1e_2\big) \wedge \big(e_2 \in S(2^{\ell/2}, 2^\mu)\big)\Big\}$$

However, the protocol generically obtained in virtue of Theorem 1, though polynomial time, is not efficient enough to be useful in practice; thus, below we describe how a practical Σ-protocol for relation $\mathcal{R}^{RSA}_{\mathsf{param}}$ could be constructed, exploiting the framework of discrete-log relation sets [31], which provides a simple method to construct complex proofs of knowledge over groups of unknown order. A discrete-log relation set R is a set of vectors of length m defined over $\mathbb{Z} \cup \{\alpha_1, \dots, \alpha_r\}$ (where the α_j's are called the free variables of the relation) and involves a sequence of base elements $A_1, \dots, A_m \in (\mathbb{Z}_n^*)^2$. For any vector $\langle a_1^i, \dots, a_m^i\rangle$ the corresponding relation is defined as $\prod_{j=1}^m A_i^{a_j^i} = 1$. The conjunction of all the relations is denoted as $R(\alpha_1, \dots, \alpha_r)$. In [31], an efficient Σ-protocol is presented for any discrete-log relation set R, by which the prover can prove of knowledge of a sequence of witnesses $x_1, \dots, x_r$, with $x_i \in S(2^{\ell_i}, 2^{\mu_i})$ that satisfy $R(x_1, \dots, x_r) \wedge \Big(\wedge_{i=1}^r (x_i \in S(2^{\ell_i}, 2^{\epsilon(\mu_i+k)+2})\Big)$, where $\epsilon > 1, k \in \mathbb{N}$ are security parameters. Note that the tightness of the integer ranges can be increased by employing the range proofs of [10], nevertheless the tightness achieved above is sufficient for our purposes, and incurs a lower overhead.

In order to prove the relation $\mathcal{R}^{RSA}_{\mathsf{param}}$, we assume that the public parameters param include the elements $g, h, y, t, s \in (\mathbb{Z}_n^*)^2$ with unknown relative discrete-logarithms. In order to construct the proof, the prover provides a sequence of public values T_1, T_2, T_3, T_4, T_5 such that $T_1 = g^r, T_2 = h^r g^x, T_3 = s^r g^{e_2}, T_4 = wy^r, T_5 = t^r g^{2e_1}$, where $r \stackrel{\mathrm{R}}{\leftarrow} [0, \lfloor n/4 \rfloor - 1]$.

The proof is constructed as a discrete-log relation set that corresponds to the equations $T_1 = g^r$, $T_2 = h^r g^x$, $(T_1)^x = g^{a_1}$, $(T_1)^{e_2} = g^{a_2}$, $T_3 = s^r g^{e_2}$ $(T_4)^x = vy^{a_1}$, $(T_5)^{e_2} g = t^{a_2} g^x$, for the free variables r, x, e_2, a_1, a_2 such that $x \in S(2^\ell, 2^\mu), e_2 \in S(2^{\ell/2}, 2^\mu)$, $a_1 = rx$ and $a_2 = re_2$. The matrix of the discrete-log relation set is shown below:

$$\begin{array}{rc}
 & \begin{array}{cccccccccccc} g & h & y & t & s & v & T_1^{-1} & T_2^{-1} & T_3^{-1} & T_4^{-1} & T_5^{-1} & g^{-1} \end{array} \\
\begin{array}{r} T_1 = g^r : \\ T_2 = h^r g^x : \\ (T_1)^x = g^{a_1} : \\ T_3 = s^r g^{e_2} : \\ (T_1)^{e_2} = g^{a_2} : \\ (T_4)^x = vy^{a_1} : \\ (T_5)^{e_2} g = t^{a_2} g^x : \end{array} &
\left[\begin{array}{cccccccccccc}
r & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
x & r & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
a_1 & 0 & 0 & 0 & 0 & 0 & x & 0 & 0 & 0 & 0 & 0 \\
e_2 & 0 & 0 & 0 & r & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
a_2 & 0 & 0 & 0 & 0 & 0 & e_2 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & a_1 & 0 & 0 & 1 & 0 & 0 & 0 & x & 0 & 0 \\
x & 0 & 0 & a_2 & 0 & 0 & 0 & 0 & 0 & 0 & e_2 & 1
\end{array}\right]
\end{array}$$

Observe that a proof of the above discrete-log relation set ensures that (i) the prover knows a witness w for some value x in the ad-hoc group accumulated value v, and (ii) for the same x, the value $x - 1$ can be split by the prover into two integers one of which belongs to $S(2^{\ell/2}, 2^\mu)$. This latter range-property guarantees the non-triviality of the splitting i.e., that the prover knows a non-trivial factor of $x - 1$ (i.e., different than $-1, 1, 2$). Note that this will require that the parameters ℓ, μ, ϵ, k should be selected such that $\ell/2 > \epsilon(\mu + k) + 2$.

5.3 ID Escrow

In Section 4.1, we discussed a generic transformation to add Identity Escrow to an *Ad hoc* Anonymous Identification scheme. Most of the required changes do not affect the system's efficiency, except for the need to resort to a generic derivation of the Anon-ID protocol.

This performance penalty is not unavoidable, however: in fact, escrow capabilities can be directly supported by the Σ-protocol for Anonymous Identification described in Section 5.2. using protocols for verifiable encryption and decryption of discrete logarithms from [17].

With notation as in Section 5.2, the Anon-ID protocol is augmented as follows: after sending the commitment T_2 to the verifier, the prover verifiably encrypts an opening of T_2 (namely, x and r) under the IEA public key. By checking that the encryption was carried out correctly, the verifier can be assured that, should the need arise, the IEA would be able to identify the prover by decrypting such opening, which would yield the prover's public key x. Moreover, by using verifiable decryption in the Extract algorithm, we can prevent the IEA from untruthfully opening the identity of the prover for a given transcript, or falsely claiming that the prover's identity cannot be properly recovered.

Alternatively, if only honest users are assumed to have access to the Escrow functionality (so that malicious parties cannot exploit the IEA as a "decryption oracle"), then a more efficient solution is possible, by having the IEA knowing the value $\log_g(h)$ in the proof of knowledge from Section 5. Then, given a transcript of the protocol (which includes the values T_1, T_2, T_3, T_4, T_5) the IEA can recover the value $g^x = T_2 T_1^{-\log_g(h)}$, from which the prover's identity can be recovered by comparing g^x to the public keys published in the public DB database.

6 Applications

6.1 *Ad Hoc* Identification Schemes

This is the most direct application. Imagine a large universe of users, where each user has a public certificate, but otherwise there is no central authority in the system. Now, some party "from the street" has a resources which he is willing to share with some subset of the users. For example, an Internet provider P may want to enable internet access to all its subscribers. However, privacy considerations may lead a user to refuse to positively identify himself; in fact, this is not strictly necessary, as long as he proves he belongs to the group of current subscribers. Our ad-hoc identification schemes are ideally suited for this application, bringing several very convenient feautures. First, P can simply take all the public keys of the users (call this set S) and combine them into one short group public key gpk_S. Notice, this initial setup is the only operation P performs which requires time proportional to the group size. As for each user $u \in S$, once again he will use his secret key and the public keys of other user to prepare one short group secret key gsk_u. After that, all identifications that u makes to P require computation and communication independent of the size of the group. Now, another provider P' can do the same for a totally different subgroup, and so on, allowing truly ad-hoc groups with no trusted authority needed in the system. Additionally, with incremental *Ad hoc* Anonymous Identification schemes (defined in the full version of this paper [25]), one can preserve efficiency even when the ad-hoc group is built gradually, as each new member addition only requires constant computation by P and by every pre-existing user in the system.

6.2 Constant Size Ring Signatures

This is one of our main applications, since it dramatically improves the efficiency of all known ring signature schemes (e.g. [34, 12, 9]). Recall, in a ring signature scheme there again is a universe of registered users, but no trusted authority. Any user u can then form a ring S, and sign a message m in such a way that any verifier (who knows S) can confidently conclude that "the message m was signed by some member u of S", but gets no information about u beyond $u \in S$. Previous papers on the subject suggested that linear dependence of the ring signature size on the size of the ring S was inevitable, since the group is ad-hoc, so the verifier needs to know at least the description of the ring. While the latter is true, in practical situations the ring often stays the same for a long period of time (in fact, there could be many "popular" rings that are used very often by various members of the ring), or has an implicit short decryption (e.g., the ring of public keys of all members of the President's Cabinet). Thus, we feel that the right measure of "signature size" in this situation is that of an "actual signature"—the string one needs in addition to the group description. Indeed, when the ring stays the same for a long time or has a short description, this actual signature is all that the verifier needs in order to verify its correctness. With this in mind, there is no reason why the signature size must be linear in the size of the ring.

In fact, our result shows that it does not have to be. Specifically, by applying the Fiat-Shamir heuristics to our ad-hoc identification scheme, we immediately get ring signatures of constant size. Moreover, our ring signatures enjoy several desirable features not generally required by ring signatures (even those of constant size). For example, both the signer and the verifier need to perform a one-time computation proportional to the size of the ring, and get some constant-size information (gsk_S and gpk_S, respectively) which allows them to produce/verify many subsequent signatures in constant time.

6.3 *Ad Hoc* ID Escrow and Group Signatures

As mentioned in Section 3.4, in some situations complete anonymity might not be desirable. In this case, one wishes to introduce a trusted *Identity Escrow Authority* (IEA), who can reveal the true identity of the user given the transcript of the identification procedure (presumably, when some "anonymity abuse" happens). Such schemes are called ID Escrow schemes [32] and have traditionally been considered for fixed groups. ID Escrow schemes are duals of group signature schemes [19, 1], which again maintain a single group of signers, and where a similar concern is an issue when signing a document anonymously. As argued in Section 4.1 and Section 5.3, our *Ad hoc* Anonymous Identification schemes and the corresponding signer-ambiguous signature schemes can efficiently support identity escrow capabilities. As a result, we get an ID Escrow and a group signature scheme with the following nice features. (For concreteness, we concentrate on group signatures below.) First, just like in current state-of-the-art group signature schemes, the signature is of constant size. Second, a user can join any group by simply telling the group manager about its public key: no expensive interactive protocols, where the user will "get a special certificate" have to be run. Thus, the group manager only needs to decide if the user can join the group, and periodically certify the "current" public key of the group. In other words, we can imagine a simple bulletin board, where the group manager periodically publishes the (certified) group public key the group, the description of the group, and potentially the history of how the public key evolved (which is very handy for incremental *Ad hoc* Anonymous Identification schemes; see [25]). From this information, each member of the group can figure out its group secret key and sign arbitrary many messages efficiently. (Of course, when signing a message the signer should also include the certified version of the current group key, so that "old" signatures do not get invalidated when the group key changes.)

References

1. G. Ateniese, J. Camenisch, M. Joye, and G. Tsudik. A practical and provably secure coalition-resistant group signature scheme. In *Advances in Cryptology—CRYPTO '00*, volume 1880 of *LNCS*, pages 255–270. Springer, 2000.
2. G. Ateniese and B. de Medeiros. Efficient group signatures without trapdoors. In *Advances in Cryptology—ASIACRYPT '03*, volume 2894 of *LNCS*, pages 246–268. Springer, 2002.
3. G. Ateniese and G. Tsudik. Some open issues and new directions in group signatures. In *Financial Cryptography (FC '99)*, volume 1648 of *LNCS*, pages 196–211. Springer, 1999.
4. N. Barić and B. Pfitzmann. Collision-free accumulators and fail-stop signature schemes without trees. In *Advances in Cryptology—EUROCRYPT '97*, volume 1233 of *LNCS*, pages 480–494. Springer, 1997.
5. M. Bellare, M. Fischlin, S. Goldwasser, and S. Micali. Identification protocols secure against reset attacks. In *Advances in Cryptology—EUROCRYPT '01*, volume 2045 of *LNCS*, pages 495–511. Springer, 2001.
6. M. Bellare, D. Micciancio, and B. Warinschi. Foundations of group signatures: Formal definitions, simplified requirements, and a construction based on general assumptions. In *Advances in Cryptology—EUROCRYPT '03*, volume 2656, pages 630–648. Springer, 2003.
7. J. Benaloh and M. de Mare. One-way accumulators: a decentralized alternative to digital signatures. In *Advances in Cryptology—EUROCRYPT '93*, volume 765 of *LNCS*, pages 274–285. Springer, 1993.
8. D. Boneh and M. Franklin. Anonymous authentication with subset queries. In *ACM Conference on Computer and Communications Security—CCS '99*, pages 113–119. ACM Press, 1999.
9. D. Boneh, C. Gentry, B. Lynn, and H. Shacham. Aggregate and verifiably encrypted signatures from bilinear maps. In *Advances in Cryptology—EUROCRYPT '03*, volume 2656 of *LNCS*, pages 416–432. Springer, 2003.
10. F. Boudot. Efficient proofs that a commited number lies in an interval. In *Advances in Cryptology—EUROCRYPT '00*, volume 1807 of *LNCS*, pages 431–444. Springer, 2000.
11. E. Bresson and J. Stern. Efficient revocation in group signatures. In *Public Key Cryptography (PKC '01)*, volume 1992 of *LNCS*, pages 190–206. Springer, 2001.
12. E. Bresson, J. Stern, and M. Szydlo. Threshold ring signatures and applications to ad-hoc groups. In *CRYPTO 2002*, volume 2442 of *LNCS*, pages 465–480. Springer, 2002.
13. J. Camenisch. Efficient and generalized group signatures. In *Advances in Cryptology—EUROCRYPT '97*, volume 1233 of *LNCS*, pages 465–479. Springer, 1997.
14. J. Camenisch and A. Lysyanskaya. An identity escrow scheme with appointed verifiers. In *Advances in Cryptology—CRYPTO '01*, volume 2139 of *LNCS*, pages 388–407. Springer, 2001.
15. J. Camenisch and A. Lysyanskaya. Dynamic accumulators and applications to efficient revocation of anonymous credentials. In *Advances in Cryptology—CRYPTO '02*, volume 2442 of *LNCS*, pages 61–76. Springer, 2002.
16. J. Camenisch and M. Michels. A group signature scheme with improved efficiency. In *Advances in Cryptology—ASIACRYPT '98*, volume 1514 of *LNCS*, pages 160–174. Springer, 1998.

17. J. Camenisch and V. Shoup. Practical verifiable encryption and decryption of discrete logarithms. Full length version of extended abstract in CRYPTO'03, available at: `http://shoup.net/papers/`, 2003.
18. J. Camenisch and M. Stadler. Efficient group signature schemes for large groups. In *Advances in Cryptology—CRYPTO '97*, volume 1294 of *LNCS*, pages 410–424. Springer, 1997.
19. D. Chaum and E. van Heyst. Group signatures. In *Advances in Cryptology—EUROCRYPT '91*, volume 547 of *LNCS*, pages 257–265. Springer, 1991.
20. L. Chen and T. P. Pedersen. New group signature schemes (extended abstract). In *Advances in Cryptology—EUROCRYPT 94*, volume 950 of *LNCS*, pages 171–181. Springer, 1994.
21. R. Cramer. *Modular Design of Secure yet Practical Cryptographic Protocols*. PhD thesis, CWI and University of Amsterdam, November 1996.
22. R. Cramer, I. B. Damgård, and B. Schoenmakers. Proofs of partial knowledge and simplified design of witness hiding protocols. In *Advances in Cryptology—CRYPTO '94*, volume 839 of *LNCS*, pages 174–187. Springer, 1994.
23. A. De Santis, G. Di Crescenzo, and G. Persiano. Communication-efficient anonymous group identification. In *5th ACM Conference on Computer and Communications Security*, pages 73–82. ACM Press, 1998.
24. A. De Santis, G. Di Crescenzo, G. Persiano, and M. Yung. On monotone formula closure of SZK. In *35th Annual Symposium on Foundations of Computer Science*, pages 454–465. IEEE Computer Society Press, 1994.
25. Y. Dodis, A. Kiayias, A. Nicolosi, and V. Shoup. Anonymous identification in *Ad Hoc* groups. Full version: `http://www.cs.nyu.edu/~nicolosi/papers/`, 2004.
26. U. Feige, A. Fiat, and A. Shamir. Zero-knowledge proof of identity. *Journal of Cryptology*, 1(2):77–94, 1988.
27. U. Feige and A. Shamir. Zero knowledge proofs of knowledge in two rounds. In *Advances in Cryptology—CRYPTO '89*, volume 435 of *LNCS*, pages 526–544. Springer, 1989.
28. A. Fiat and A. Shamir. How to Prove Yourself: Practical Solutions to Identification and Signature Problems. In *Advances in Cryptology—CRYPTO '86*, volume 263 of *LNCS*, pages 186–194. Springer, 1986.
29. O. Goldreich, S. Micali, and A. Wigderson. Proof that yield nothing bur their validity or all languages in np have zero-knowledge proof systems. *Journal of the ACM*, 38(3):691–729, 1991.
30. S. Goldwasser, S. Micali, and C. Rackoff. The knowledge complexity of interactive proof systems. *SIAM Journal on computing*, 18(1):186–208, 1989.
31. A. Kiayias, Y. Tsiounis, and M. Yung. Traceable signatures. In *Advances in Cryptology—EUROCRYPT 04*, LNCS. Springer, 2004.
32. J. Kilian and E. Petrank. Identity escrow. In *Advances in Cryptology—CRYPTO '98*, volume 1462 of *LNCS*, pages 169–185. Springer, 1998.
33. M. Naor. Deniable ring authentication. In *Advances in Cryptology—CRYPTO '02*, volume 2442 of *LNCS*, pages 481–498. Springer, 2002.
34. R. Rivest, A. Shamir, and Y. Tauman. How to leak a secret. In *Advances in Cryptology—ASIACRYPT '01*, volume 2248 of *LNCS*, pages 552–565. Springer, 2001.
35. G. Tsudik and S. Xu. Accumulating composites and improved group signing. In *Advances in Cryptology—ASIACRYPT '03*, volume 2894 of *LNCS*, pages 269–286. Springer, 2003.