Anonymous Identity-Based Identification Scheme in Ad-Hoc Groups without Pairings

Prateek Barapatre and Chandrasekaran Pandu Rangan

Theoretical Computer Science Lab.,
Department of Computer Science and Engineering,
IIT Madras, Chennai, India
{pbarapatre.64,prangan55}@gmail.com

Abstract. Anonymous identification schemes in ad-hoc groups are cryptographic primitives that allow a participant from a set of users to prove her identity in that group, without revealing her actual identity or the group secret key. All the existing ad-hoc anonymous identification schemes in the literature make use of the bilinear pairing operation, resulting in a computational overhead. In this paper we propose a novel anonymous identity-based identification scheme for ad-hoc groups without using bilinear pairings. This scheme, to the best of our knowledge, is the first of its kind which does not use pairing operations. The proof of our scheme is based on the hardness assumption of RSA problem.

Keywords: Identity-based identification, Anonymity, Ad-hoc group, RSA assumption.

1 Introduction

An identification(ID) scheme allows an entity called a *prover*(say Alice) to securely identify herself to another entity called a *verifier*(say Bob). ID schemes enable the *prover* to convince a *verifier* that she is indeed the same entity which she claims to be, by showing that she knows some secret information without revealing her secret information. Secure identification schemes were introduced by Fiat and Shamir [1] followed by many other identification schemes [2,3,4]. Anonymous identification scheme is yet another important primitive which has wide ranging applications in the domains of e-commerce and auctions. Ad-hoc group refers to participants from a user population that can form group in an ad-hoc fashion(without the help of a group manager). An ad-hoc anonymous identification scheme is a multi-user cryptographic primitive that allows participants from a user population to form ad-hoc groups and then prove their membership in such groups anonymously. In an anonymous ad-hoc group identification scheme, a member A of a group $\mathcal{G}$ convinces another entity B outside the group, that she is one amongst those in $\mathcal{G}$ in a secure fashion without revealing any information about her own identity, thus maintaining her privacy. This is a very useful primitive which enables an entity A to control her privacy while enjoying privileges of the groups. There are many applications of such anonymous

B. Gierlichs, S. Guilley, and D. Mukhopadhyay (Eds.): SPACE 2013, LNCS 8204, pp. 130–146, 2013.

identification schemes such as providing access to a resource to only certain privileged group of users without the need for the user to reveal her actual identity, for entry in some restricted building by some group members only, etc. Anonymous authentication for dynamic group is also an indispensable component in online auctions, electronic voting and open procurement, which are becoming very popular business areas in e-commerce. Authenticating membership in a group is an important task because many privileges(such as the right to read a document, access to a hardware or application resources) are often assigned to many individuals. While the permission to exercise a privilege requires that members of the group be distinguished from non-members, members need not be distinguished from one another, just a confirmation of them belonging to the group is sufficient to authorize them to access the resource.

The concept of ad-hoc anonymous identification scheme was first introduced by Dodis et al [5] in Public-Key Infrastructure(PKI) setting and it was extended to identity-based setting by Nguyen [6]. The latter work makes use of the notions of dynamic cryptographic accumulators, which in turn are derived using bilinear pairings in their scheme. Following Nguyen's work [6], other ad-hoc anonymous identity-based identification schemes were proposed, some of which do not make use of cryptographic accumulators, but still use pairings. To the best of our knowledge, the most efficient of such schemes is the one proposed by Chunxiang Gu et al [7], which even though do not use accumulators, still make use of bilinear pairings.

Our Contribution: Many anonymous identity-based identification schemes for ad-hoc groups are available, but they all make use of the bilinear pairing operations. In this work we propose a new ad-hoc anonymous IBI scheme, which preserves the security requirements for an anonymous IBI without using bilinear maps. The proposed scheme is more efficient computationally, as the pairing operation increases the computational cost incurred. Moreover, for implementing our scheme, we do not have to be concerned for choosing appropriate bilinear maps. The security of our scheme is based on the hardness assumption of RSA problem in the composite group of integers.

Paper Organisation: The paper is organised as follows: In Section 2, we explain the preliminaries required and cover the formal definitions of ad-hoc anonymous IBI schemes along with their security requirements. In the third Section, we show the construction of our scheme followed by its security arguments in Section 4. In Section 5, we compare our scheme with various existing schemes in the literature. Finally, we conclude our work in Section 6.

2 Formal Definitions and Security Model

We first describe the hardness assumption used, then proceed to describe the canonical three-move identification protocol, followed by the formal definitions and security model for ad-hoc anonymous IBI scheme.

2.1 Hard Problem Assumption

Definition 2.1. ***RSA Problem*** *[8]- Let $N = pq$ be a composite integer computed from two large prime numbers p and q each k-bit long, where, k is the security parameter. Let e be a random prime number, greater than 2^l for some fixed parameter l, such that $gcd(e, \phi(N)) = 1$. Let y be a random element from $\mathbb{Z}_N^*$.*

We say that an algorithm $\mathcal{I}$ solves the RSA problem, if it receives as input the tuple (N, e, y) and outputs an element z, such that $z^e = y \bmod N$.

2.2 Canonical 3-Move Identification

A three-move protocol of the form depicted in Figure 1 is said to be *canonical* as given by Bellare et al [9]. This protocol is initiated by the *prover*. The *prover*'s first message is called commitment, then the *verifier* selects a challenge uniformly at random from a set, called challenge set $ChSet_v$, associated with its input v. After this step, the *prover* sends a response and upon receiving the response, the *verifier* applies a deterministic procedure DEC_v to arrive at a decision whether to *Accept* or *Reject*. The *prover P* has input q, a random tape R and maintains a state St. The *verifier V* has input v and returns boolean decision d of *Accept* or *Reject*.

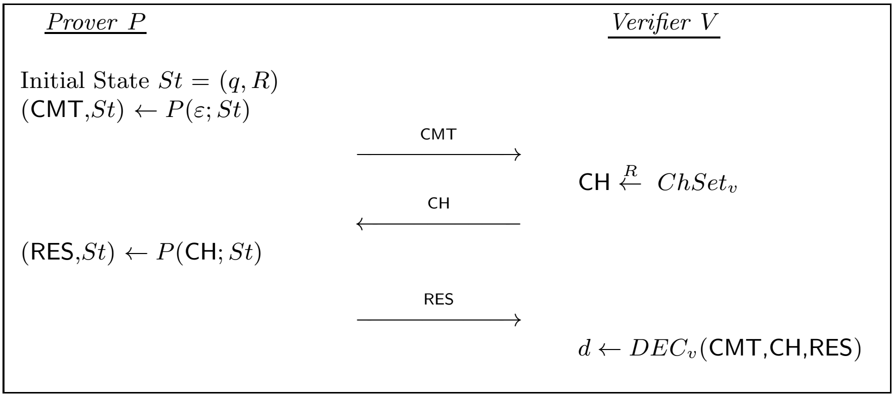

Prover P
Verifier V
Initial State $St = (q, R)$
$(\mathsf{CMT}, St) \leftarrow P(\varepsilon; St)$
CMT
$\mathsf{CH} \overset{R}{\leftarrow} ChSet_v$
CH
$(\mathsf{RES}, St) \leftarrow P(\mathsf{CH}; St)$
RES
$d \leftarrow DEC_v(\mathsf{CMT}, \mathsf{CH}, \mathsf{RES})$

Fig. 1. A canonical three-move identification protocol

2.3 Identity-Based Ad-Hoc Anonymous Identification Schemes

The definition of anonymous identification in ad-hoc groups was originally given in the PKI setting by Dodis et al [5]. Nguyen extended the definition to the ID-based setting [6]. The changes made by Nguyen [6] to reconcile the security notions for the ID-based model are that: the *Register* algorithm is replaced by the *KeyGen* algorithm and the *Setup* algorithm is not required to maintain a database of public key of users.

We follow the same definition as given by Nguyen [6], but we also elaborate further to describe the formal games concerning the security of the ad-hoc anonymous IBI schemes.

An ad-hoc anonymous IBI scheme is defined as a five tuple, $\mathcal{IAID}$ = (**Setup**, **KeyGen**, **MakeGPK**, **MakeGSK**, **IAID**) of Probabilistic Polynomial Time (PPT) algorithms which are described below:

1. **Setup**: The central authority, called the Private Key Generator(PKG) runs the *Setup* algorithm. This algorithm takes as input the security parameter 1^k and outputs the public parameters param and master secret key M_{sk}. The PKG keeps the M_{sk} to itself.
2. **KeyGen**: It takes as input the public parameters param, the master secret key(M_{sk}) and the identity of the user and outputs the private key of the user. The private key of the user is then send to the user by PKG through a secure channel. The identity used is the public key of the user.
3. **MakeGPK**: This PPT algorithm takes as input the public parameters param, the set of identities and deterministically outputs the group public key, which is later used in the identification protocol $IAID$. The algorithm is *order-independent* i.e., the order in which the public keys to be aggregated are provided does not matter. The algorithm runs in linear time in the order of the number of public keys being aggregated.
4. **MakeGSK**: It takes as input param, the set of entities, an entity amongst the set of entities and its corresponding private key, and outputs the group secret key which is used in the group identification protocol. Its cost also varies linearly with the number of entities being aggregated. It should be observed that the group secret key g_{sk} must correspond to a group public key g_{pk}.
5. **IAID** = (IAID$_P$,IAID$_V$) is the two party identification protocol which allows the *prover* IAID$_P$ to anonymously show her membership amongst the group of identities constructed by him. In this protocol both the IAID$_P$ and IAID$_V$ takes as input the system's parameters param and a group public key *gpk*. IAID$_P$ also has group secret key *gsk* as an additional input. At the end of an IAID protocol run, IAID$_V$ outputs a 0/1 signifying either a *Reject*/*Accept*.

2.4 Security Requirements for Ad-Hoc Anonymous IBI Scheme

There are three security requirements for an ad-hoc anonymous identification as proposed by Dodis et al [5]. We describe them below and specify their formal games of security in the next subsection.

1. Correctness : This property requires that during an execution of an IAID protocol, an *honest prover* will always be able to convince a *verifier*. In other words, if IAID$_P$ owns the group secret key corresponding to the common

input group public key, then the IAID_V will *Accept* with an overwhelming probability, i.e. with a probability almost equal to 1.

2. Soundness : This property ensures that any dishonest entity not possessing a private key in the target ad-hoc group, will not be able to convince an *honest verifier*, and even if it does, it will be with a negligibly small probability. This requirement is modelled by a game being played between an honest dealer and an adversary and the adversary can send queries to the *Transcript Oracle*, which takes as input an identity of the user and a set of other entities and outputs a valid transcript of the IAID protocol's execution, where the user anonymously proves her membership of the group formed between him and the group of entities.

   The overall game is played as follows: The honest dealer runs the *Setup* algorithm and sends the resulting public parameters to the adversary. After this the adversary is allowed to adaptively ask for the key extract queries from the *User Secret-Key-Extract Oracle* and also queries the *Transcript Oracle* during the game, or even during the execution of the IAID protocol later as a part of its training. After a certain point, the adversary who now plays the role of a *prover*, returns a target group of identities and then executes the IAID protocol with the honest dealer. Both the adversary and the honest dealer takes as inputs the public parameters and the group public key corresponding to the target group. The adversary wins the game if the honest dealer outputs *Accept* and the adversary does not have a private key corresponding to an identity in the target group. The ID-based anonymous identification scheme provides Soundness if the probability that the adversary wins this game is negligible.

3. Anonymity: This requirement is modelled by a game being played between an honest dealer and an adversary, where the adversary can send only one query to a *Challenge Oracle*. This oracle takes as input two 'identity-private key' pairs and a set of other identities and returns a transcript of the IAID protocol's execution, where the *prover* randomly uses one of the two private keys to prove membership of the group formed by - the set of identities along with the two identities from the pairs. The honest dealer first runs the Setup algorithm and sends the resulting public parameters to the adversary. Then, the adversary can find many pairs of 'identity-private key' during the game, even after receiving the challenge transcript from the *Challenge Oracle* later. At a point, she queries the *Challenge Oracle* and gets a challenge transcript. The adversary then can do experiments with the system before outputting an identity amongst the two identities. The adversary wins the game if the identity she outputs corresponds to the private key the *Challenge Oracle* used to generate the challenge transcript. The ID-based ad-hoc anonymous identification scheme provides anonymity if the probability that the adversary wins the game is negligibly more than a random guess. If this condition holds, even if the adversary has unlimited computing resources, the scheme is said to provide Unconditional Anonymity.

2.5 Formal Games of Security for Ad-Hoc Anonymous IBI Scheme

We formalize the security requirements in the form of games between an adversary and a challenger. The adversary will be given access to various oracles to aid him in carrying out the impersonation attacks. Let $\mathcal{PK}$ denote the domain of the user's public key and $\mathcal{SK}$ denote the domain of the corresponding user's secret key. We also assume $\mathcal{PK}'$ to be a superset of the possible user public keys i.e. $\mathcal{PK}' \supseteq \mathcal{PK}$.

```
User Secret-Key-Extract Oracle ($\mathcal{O}_{HReg}$)
IN: $u \in \mathcal{U}$
RUN: $d_{ID_i} \leftarrow KeyGEN(ID_i)$
OUT: $d_{ID_i}$
```

```
User Corruption Oracle ($\mathcal{O}_{Cor}$)
IN: $ID_i \in \mathcal{PK}'$
RUN: $d_{ID_i} \leftarrow KeyGEN(ID_i)$
OUT: $d_{ID_i}$
```

```
Transcript Oracle ($\mathcal{O}_{Scr}$)
IN: $S' \subseteq \mathcal{PK}', ID_i \in \mathcal{PK}'$
RUN: 1. $d_{ID_i} \leftarrow KeyGEN(ID_i)$
        $KeyGEN$ is the key generation algorithm
2. If $d_{ID_i} = \perp$
3. then $\pi \leftarrow \perp$
4. Else $gpk \leftarrow$ MakeGPK($\mathsf{param}$,$S' \cup \{ID_i\}$)
5.      $gsk \leftarrow$ MakeGSK($\mathsf{param}, S', (d_{ID_i}, ID_i)$)
6.          $\pi \overset{R}{\leftarrow}$ (IAID$_P$($\mathsf{param}$,$gpk, gsk$) $\leftrightarrow$ IAID$_V$($\mathsf{param}$,$gpk$))
OUT: $\pi$
```

```
Challenge Oracle ($\mathcal{O}_{Ch}$)
INPUT: $S' \subseteq \mathcal{PK}', (sk_0, pk_0), (sk_1, pk_1)$
RUN: 1. $b^* \overset{R}{\longleftarrow} \{0, 1\}$
2. If, $sk_0 \not\leftrightharpoons pk_0$ or $sk_1 \not\leftrightharpoons pk_1$, then $Abort$
   where $\leftrightharpoons$ depicts a correspondence between the public key $pk_i$
   and the associated valid secret key $sk_i$
3. $G_{pk} \leftarrow MakeGSK(\mathsf{param}, S' \cup \{pk_0, pk_1\})$
4. $G^*_{sk} \leftarrow MakeGSK(\mathsf{param}, S' \cup \{pk_{1-b^*}\}, (sk_{b^*}, pk_{b^*}))$
5. $\pi^* \overset{R}{\leftarrow} (IAID_P(\mathsf{param}, gpk, gsk^*) \leftrightarrow IAID_V(\mathsf{param}, gpk)$
OUT: $\pi^*$
```

Fig. 2. Oracles given to adversary attacking ad-hoc IBI scheme

Game for Correctness. For correctness, we require that any honest execution of the IAID protocol shall terminate with the *verifier* outputting an *Accept* or 1, with an overwhelming probability. In this game, the IAID$_P$ is given an additional input of group secret key gsk, related to the common input gpk.

Game for Correctness

$(\forall t \in \mathbb{N})(\forall (u_1, u_2, \ldots\ldots, u_n) \in \mathcal{U})$
$\Pr[\mathsf{param} \overset{R}{\leftarrow} \mathrm{Setup}(1^t);$ (where t is the security parameter)
$(sk_i, pk_i) \overset{R}{\leftarrow} KeyGEN(\mathsf{param},\ u_i),\ i = 1, \ldots\ldots, n$
$gpk \leftarrow MakeGPK(\mathsf{param}, \{pk_1, \ldots\ldots, pk_n\});$
$gsk \leftarrow MakeGSK(\mathsf{param}, \{pk_2, \ldots\ldots, pk_n\}, (sk_1, pk_1));$ such that,
$IAID_V(\mathsf{param}, gpk)_{IAID_P(\mathsf{param}, gpk, gsk)} = 1] \geq 1 - \nu(t)$
(where $\nu(t)$ is a negligible function in t)

Fig. 3. Correctness *imp-atk* security of IBI scheme

Game for Soundness. The soundness guarantee can be expressed in terms of a game being played between an honest challenger and the adversary $\mathcal{A}$. In the attack game for soundness, the adversary is allowed to interact with three oracles O_{Ext}(the *honest User Secret-Key-Extract Oracle*), O_{Cor}(the *User Corruption Oracle*) and O_{Scr}(the *Transcript Oracle*) described in Figure 2.

The game begins with the honest challenger running the *Setup* algorithm with the security parameter 1^t and handing the resulting global parameters param to $\mathcal{A}$. Then, $\mathcal{A}$ arbitrarily interleaves queries to the three oracles, according to any adaptive strategy she wishes and eventually outputs a target group $S^* \subseteq \mathcal{PK}'$. After a point, $\mathcal{A}$(in the role of the *prover*) starts executing a run of the $IAID$ protocol with the challenger on common inputs param and $gpk^* = MakeGPK(\mathsf{param}, S^*)$.

Also during this interaction, the adversary is still allowed to query the three oracles O_{Ext}, O_{Cor}, O_{Scr}. Let $\tilde{\pi}$ be the transcript resulting from such a run of the $IAID$ protocol. $\mathcal{A}$ wins the game if the following conditions hold:

1. $\forall pk^* \in S^*$, there is a valid sk^*(secret key) corresponding to the pk^*.
2. $\tilde{\pi}$ is a valid transcript i.e., the protocol run completed with the challenger outputting 1, and
3. $\forall pk^* \in S^*$, $\mathcal{A}$ never queried O_{Cor} on input pk^*.

We define $\mathsf{Succ}_{\mathcal{A}}^{\mathsf{Snd}}(t)$ to be the probability that $\mathcal{A}$ wins the above game.

Definition 2.2. *An ad-hoc anonymous IBI is sound against active chosen-ring attacks if any adversary $\mathcal{A}$ has negligible advantage to win the above game:*

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

where $\nu(t)$ is a negligible function in security parameter t.

Game for Anonymity. We formalize the anonymity requirements for an ad-hoc anonymous IBI 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* $\mathcal{O}_{Ch}$, described in Figure 2. The game begins with the honest challenger running the *Setup* algorithm for the security parameter 1^t 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, Anon-ID$_P$ and Anon-ID$_V$ algorithms as long as she deems necessary; eventually, she queries the $\mathcal{O}_{Ch}$ 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 $\mathcal{O}_{Ch}$ to construct the challenge π; finally, $\mathcal{A}$ outputs a single bit b$'$, her best guess to the "Challenge" bit b^*. Define $\mathrm{Succ}_{\mathcal{A}}^{Anon}(t)$ to be the probability that the bit b' output by $\mathcal{A}$ at the end of the above game is equal to the random bit b^* used by the $\mathcal{O}_{Ch}$ oracle.

Definition 2.3. *An ad-hoc anonymous IBI scheme is fully anonymous if for any probabilistic polynomial-time adversary, $\mathcal{A}$ has success probability at most negligibly greater than one half:*

$$(\forall \lambda \in \mathrm{N})(\forall PPT\mathcal{A})|\mathrm{Succ}_{\mathcal{A}}^{Anon}(t)\text{-}\ \tfrac{1}{2}\ |\leq \nu(t)$$

where $\nu(t)$ is a negligible function in t.

2.6 Reset Lemma

The Reset lemma was first proposed by Bellare and Palacio [9]. The Reset lemma upper bounds the probability, that a *cheating prover* $\mathcal{Q}$ can convince the *verifier* to accept as a function of the probability that a certain experiment based on resetting the *prover* yields two accepting conversation transcripts. We recall the definition of the Reset lemma as stated in [9]. Consider again the canonical three-move identification protocol between *prover* and *verifier* [10]. The *prover*'s first message is called commitment. The *verifier* selects a challenge uniformly at random from a set ChSet$_v$, associated with its input v and upon receiving a response from the *prover*, the *verifier* applies a deterministic decision predicate DEC$_v$(Cmt,Ch,Rsp) to compute a boolean decision. The *verifier* is represented by the pair (ChSet$_v$,DEC), which when given the verifier input v, defines the challenge set and decision predicate. Formally describing,

Reset Lemma: Let $\mathcal{Q}$ be a *prover* in a canonical protocol with a *verifier* represented by (ChSet,DEC), and let q and v be inputs for the *prover* and *verifier* respectively. Let $acc(q,v)$ be the probability that the *verifier* outputs *Accept* in its interaction with $\mathcal{Q}$. In other words, the probability that the following experiment returns $d = 1$.

Choose random tape R for $\mathcal{Q}$;
$St \longleftarrow (q,R)$;$(\mathsf{CMT},St) \longleftarrow \mathcal{Q}(\varepsilon,St)$;
$\mathsf{CH} \overset{R}{\longleftarrow} \mathrm{ChSet}_v; (\mathsf{RSP},St) \leftarrow \mathcal{Q}(\mathsf{CH},St); d \leftarrow DEC_v(\mathsf{CMT},\mathsf{CH},\mathsf{RSP})$;
Return d.

Let $\mathsf{res}(q,v)$ be the probability that the following reset experiment returns 1.

Choose random tape R for $\mathcal{Q}$; $St \longleftarrow (q,R)$; $(\mathsf{CMT},St)\longleftarrow \mathcal{Q}(\epsilon,St)$

$\mathsf{CH}_1 \overset{R}{\leftarrow} \mathrm{ChSet}_v$; $(\mathsf{RSP}_1, St_1) \leftarrow \mathcal{Q}(\mathsf{CH}_1; St)$; $d_1 \leftarrow \mathrm{DEC}_v(\mathsf{CMT}, \mathsf{CH}_1, \mathsf{RES}_1)$

$\mathsf{CH}_2 \overset{R}{\leftarrow} \mathrm{ChSet}_v$; $(\mathsf{RSP}_2, St_2) \leftarrow \mathcal{Q}(\mathsf{CH}_2; St)$; $d_2 \leftarrow \mathrm{DEC}_v(\mathsf{CMT}, \mathsf{CH}_2, \mathsf{RES}_2)$
If($d_1 = 1$ AND $d_2 = 1$ AND $\mathsf{CH}_1 \neq \mathsf{CH}_2$) *return* 1, else *return* 0.

Then,

$$\mathsf{acc}(q, v) \leq \frac{1}{|\mathrm{ChSet}_v|} + \sqrt{\mathsf{res}(q, v)}.$$

3 Proposed Scheme

We now present our ad-hoc anonymous IBI scheme. We build on the ideas and constructs from Guilliou-Quisquater identification scheme [11] and Herranz ring signatures scheme [8] to construct our new identification scheme for ad-hoc anonymous group. To the best of our knowledge, this is the first ad-hoc anonymous IBI scheme which does not use pairings. The various protocols involved in the scheme are described below.

- **Setup**: Based on the security parameter t, the PKG generates two random t-bit prime numbers p and q and then computes $N = pq$. For some fixed parameter l, the PKG chooses a prime number e at random, satisfying 2^l <e< 2^{l+1} and $gcd(e,\phi(n))$=1, and computes $d = e^{-1}\ mod\ \phi(n)$. Moreover, the PKG uses two hash functions $H_1 : \{0,1\}^* \longrightarrow \mathbb{Z}_N^*$, $H_2 : \{0,1\}^* \longrightarrow \{0,1\}^l$. The public output of this algorithm are the param=(t, l, N, e, H_1, H_2) and the master secret key (p, q, d).
- **KeyGen**: When a user with identity $id \in$ {0,1}* queries or asks for secret key, the PKG computes $\mathrm{SK}_{id} = H_1(id)^d\ mod\ N$. SK_{id} is then sent to the user through a secure channel. The user can verify whether the received secret key is valid or not by checking if $\mathrm{SK}_{id}^e = H_1(id)\ mod\ N$.
- **MakeGPK and MakeGSK**: From a given set of identities which are selected in an ad-hoc fashion, the ring $\mathcal{U} = \{ID_1, ID_2, ..., ID_n\}$ is formed. A user with identity $ID_s \in \mathcal{U}$ having the secret key SK_s runs the following algorithm:
  1. For all $i \in \{1, 2, ..., n\} \setminus \{s\}$, do:
     (a) Choose $\mathcal{A}_i \in \mathbb{Z}_N^*$.
     (b) Compute $R_i = \mathcal{A}_i^e\ mod\ N$
     and $h_i = H_2(\mathcal{U}, ID_i, R_i)$.
  2. Choose $\mathcal{A} \in_R \mathbb{Z}_N^*$
  3. Compute $R_s = \mathcal{A}^e \prod_{i \neq s} [H_1(ID_i)]^{-h_i}\ mod\ N$.

     If $R_s = 1\ mod\ N$ or $R_s = R_i$ for some $i \neq s$; then GOTO *Step* 2, and *Repeat*.
  4. Compute $h_s = H_2(\mathcal{U}, ID_s, R_s)$.
  5. The Group Public key(GPK) is:
     $GPK = [\ \{R_i\}_{i=1}^n,\ \{\ h_i\ \}_{i=1}^n,\ \mathcal{U}\ ]$

     The Group Secret key(GSK) is:
     $GSK = \mathrm{SK}_S^{h_s}$.

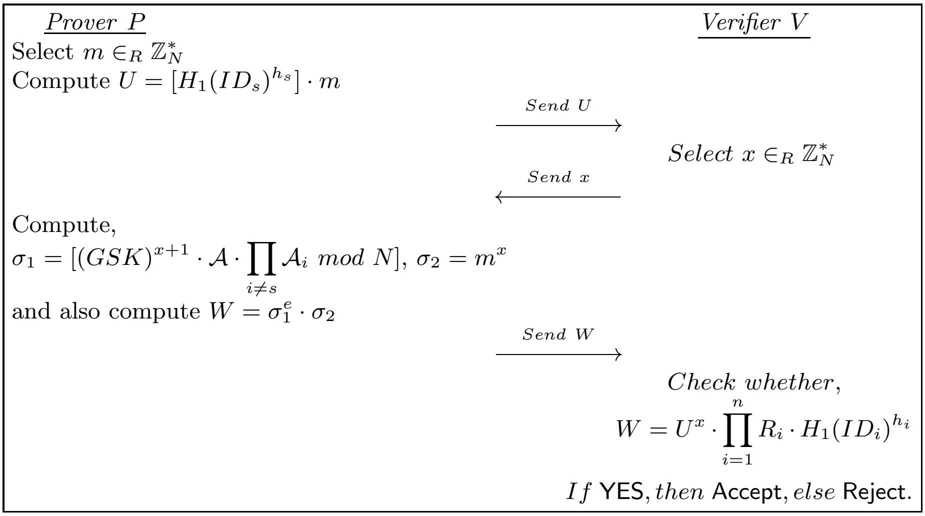
- **IAID**

  The *IAID* protocol is depicted in Figure 4. The various steps performed in the protocol are:

  1. The *prover* P selects a message $m \in_R \mathbb{Z}_N^*$ and then computes $U = H_1(ID_s)^{h_s} \cdot m$
  2. P sends U as commitment to *verifier* V.
  3. V selects a random $x \in_R \mathbb{Z}_N^*$ as the challenge and sends it to P.
  4. P computes $\sigma_1 = \left[(GSK)^{x+1} \cdot \mathcal{A} \cdot \prod_{i \neq s} \mathcal{A}_i \ mod\ N \right]$ and $\sigma_2 = m^x$. It then computes $W = \sigma_1^e \cdot \sigma_2$.
  5. P sends W as the response to V.
  6. V checks for consistency of W as:

     If $W = U^x \cdot \prod_{i=1}^{n} \left[(R_i.H_1(ID_i)^{h_i}) \right] \ mod\ N$,

     Then V Accepts, else it Rejects



Prover P

Verifier V

Select $m \in_R \mathbb{Z}_N^*$

Compute $U = [H_1(ID_s)^{h_s}] \cdot m$

$\xrightarrow{Send\ U}$

$Select\ x \in_R \mathbb{Z}_N^*$

$\xleftarrow{Send\ x}$

Compute,

$\sigma_1 = [(GSK)^{x+1} \cdot \mathcal{A} \cdot \prod_{i \neq s} \mathcal{A}_i \ mod\ N]$, $\sigma_2 = m^x$

and also compute $W = \sigma_1^e \cdot \sigma_2$

$\xrightarrow{Send\ W}$

$Check\ whether,$

$W = U^x \cdot \prod_{i=1}^{n} R_i \cdot H_1(ID_i)^{h_i}$

If YES, $then$ Accept, $else$ Reject.

Fig. 4. The *IAID* protocol

4 Security Proof

We depict the proof of security by showing that the three required properties of CORRECTNESS, SOUNDNESS and ANONYMITY hold true for our scheme.

1. **Correctness**: It can be easily seen that an honest entity of the ring acting as *prover* will always be able to genuinely identify herself.

$$\mathbf{L.H.S.} = W = \sigma_1^e \cdot \sigma_2$$

$$= GSK^{(x+1)\cdot e} \cdot \mathcal{A}^e \cdot \left(\prod_{i\neq s} \mathcal{A}_i \ mod\ N\right)^e \cdot \sigma_2$$

$$= (H_1(ID_s)^{d\cdot h_s})^{(x+1)\cdot e} \cdot \left(\prod_{1\leq i\leq n} \mathcal{A}_i \ mod\ N\right)^e \cdot \sigma_2$$

$$= (H_1(ID_s)^{(x+1)\cdot h_s}) \cdot \left(\prod_{1\leq i\leq n} \mathcal{A}_i \ mod\ N\right)^e \cdot \sigma_2$$

$$= (H_1(ID_s)^{(x+1)\cdot h_s}) \cdot m^x \cdot \left(\prod_{1\leq i\leq n} \mathcal{A}_i \ mod\ N\right)^e$$

$$\mathbf{R.H.S} = U^x \cdot \prod_{1\leq i\leq n} [R_i.H_1(ID_i)^{h_i}] \ mod\ N$$

$$= U^x \cdot R_s \cdot H_1(ID_s)^{h_s} \cdot \prod_{i\neq s}(R_i.H_1(ID_i)^{h_i}) \ mod\ N$$

$$= (H_1(ID_s)^{x\cdot h_s}) \cdot m^x \cdot \mathcal{A}^e \cdot \prod_{i\neq s}[H_1(ID_i)^{-h_i}] \cdot H_1(ID_s)^{h_s} \cdot$$

$$\prod_{i\neq s}(\mathcal{A}_i^e) \cdot H_1(ID_i)^{h_i} \ mod\ N$$

$$= (H_1(ID_s)^{(x+1)\cdot h_s}) \cdot m^x \cdot \left(\prod_{1\leq i\leq n} \mathcal{A}_i \ mod\ N\right)^e$$

L.H.S. = **R.H.S.** Hence, correctness property holds.

2. **Soundness**: We show that, if an impersonator is successfully able to get herself verified as an *honest prover*, then the impersonator should have the knowledge of the GSK under the hardness assumption of RSA problem. We use Reset lemma on H_2 oracle to arrive at this result. We use the technique similar to Herranz [8] to prove the security of our scheme.

   We show that the advantage of any imp-ca attacker against our scheme can be upper bounded by the advantage of a related RSA adversary and a function of the challenge length l. We first assume an instance of the RSA problem which the challenger will try to solve. Fix $t \in N$ and let (N, e, y) be an output of $\mathcal{K}_{rsa}$ running on input t, where t is the input parameter. We assume that V never repeats a request. Let (N, e, y) be the instance of RSA problem. We are going to construct a probabilistic polynomial time algorithm $\mathcal{A}$ that satisfies the condition of the Reset Lemma. This algorithm will use the impersonator $\mathcal{I}$ as a sub-routine to solve the RSA problem. Thus, the goal of $\mathcal{A}$ is to compute a value z such that $z^e = y \ mod\ N$. $\mathcal{A}$ will try to simulate the game, required oracles and identification transcripts to $\mathcal{I}$ perfectly.

   We initialize the challenger machine $\mathcal{A}$ giving the data (N, e, y) as input to it. The machine then runs the impersonator $\mathcal{I}$ against our ad-hoc anonymous

IBI scheme. The two hash functions H_1, H_2 are modelled as random oracles, so their values will be computed and stored by $\mathcal{A}$. The RSA public key of the master entity is defined to be (N, e) and is also known to the impersonator $\mathcal{I}$. Without loss of generality, we can assume that the impersonator queries the H_1 random oracle for the value $H_1(ID)$ before asking the corresponding secret key of ID. $\mathcal{A}$ replies to the various hash oracle queries, key extract queries and impersonation queries of $\mathcal{I}$ in a manner mentioned below:

H₁ queries: The machine $\mathcal{A}$ constructs a table TAB_{H_1} to simulate the random oracle H_1. For this we use the technique as proposed by Coron [12]. Every time an identity ID_i is queried by $\mathcal{A}$ to the oracle H_1, the machine $\mathcal{A}$ responds as:

First, $\mathcal{A}$ checks if this input is already in TAB_{H_1}. If this is the case, then $\mathcal{A}$ sends to $\mathcal{I}$ the corresponding relation $H_1(ID_i) = PK_i$. Otherwise, $\mathcal{A}$ chooses a bit $\beta \in \{0,1\}$, which will be β_i=0 with probability μ, and β_i =1 with probability 1-μ, where we define $\mu = (5/6)^{1/\mathcal{Q}_e}$, here, $\mathcal{Q}_e$ is the number of extraction queries. Then, $\mathcal{A}$ chooses a random element $x_i \in \mathbb{Z}_N^*$ and defines $PK_i = y^{\beta_i} \cdot x_i^e \ mod\ N$. The entry $(ID_i, PK_i, x_i, \beta_i)$ is stored in the table TAB_{H_1}. The relation $H_1(ID_i) = PK_i$ is sent to $\mathcal{I}$. The condition $PK_i \neq PK_j$ must be satisfied for all the different entries in TAB_{H_1}. If this is not the case, the process is repeated for one of the user.

Since we are assuming that H_1 behaves as a random function and the values PK_i are randomly chosen, the information that $\mathcal{I}$ receives is consistent.

H₂ queries: When $\mathcal{I}$ queries the random oracle H_2, the challenger $\mathcal{A}$ asks its own oracle for the output values of this hash function and then returns it to $\mathcal{I}$.

Key Extract Queries: Every time $\mathcal{I}$ asks for the secret key corresponding to an identity ID_i, the machine $\mathcal{A}$ looks for ID_i in the table TAB_{H_1}. If $\beta_i = 0$, then $\mathcal{A}$ sends SK_i =x_i to $\mathcal{I}$ since $SK_i^e = PK_i \ mod\ N$ as is required. If $\beta_i = 1$, then the machine $\mathcal{A}$ cannot answer the query and halts. Note that the probability that $\mathcal{A}$ halts in this process is less than $1-\mu^{\mathcal{Q}_e} = 1/6$. So with probability greater than 5/6, $\mathcal{A}$ will reply to the key extract queries of $\mathcal{I}$.

Identification Queries: The impersonator can ask for polynomial number of identification transcripts for the ring of identities $\mathcal{U}'$. We assume that $\mathcal{I}$ has not asked for the secret key of any of the identities in $\mathcal{U}'$, because, if this is the case, then $\mathcal{I}$ can obtain a valid identification transcript by itself. We also assume that $\mathcal{I}$ has asked for the public key of all the identities $PK_i = H_1(ID_i)$ in the ring $\mathcal{U}'$ to the random oracle H_1. To answer an identification query, the machine $\mathcal{A}$ responds as follows:

- If $\beta_i = 0$ for some $i \in \{1, \ldots, n'\}$, then $PK_i = H_1(ID_1) = (x_i)^e \ mod\ N$, so $\mathcal{I}$ knows the secret key for this identity and can easily compute a valid *honest prover* transcript by following the $IAID$ protocol.

- If $\beta_i = 1$ for all $i = 1, \ldots, n'$, then $\mathcal{A}$ does the following:
  (a) For all $i = \{1, \ldots, n'\}, i \neq s$, choose pairwise different $\mathcal{A}_i \in \mathbb{Z}_N^*$ uniformly at random and compute $R_i = \mathcal{A}_i^e \ mod \ N$.
  (b) By querying the random oracle H_2, compute $h_i = H_2(\mathcal{U}', ID_i, R_i)$, for all $i \neq s$. We can assume that $\mathcal{I}$ will later query the random oracle H_2 with these inputs.
  (c) Choose a random $h_s \in \{0, 1\}^l$.
  (d) Choose at random $\sigma' \in \mathbb{Z}_N^*$.
  (e) Compute $R_s = (\sigma')^e.H_1(ID_s)^{-h_s}.\prod_{i \neq s}(R_i^{-1}.H_1(ID_i)^{-h_i}) \ mod \ N$. If $R_s = 1 \ mod \ N$ or $R_s = R_i$ for some $i \neq s$, then go back to the previous step and repeat.
  (f) At this point, the machine $\mathcal{A}$ *falsifies* the random oracle H_2 by imposing the relation $h_s = H_2(\mathcal{U}', ID_s, R_s)$. Later, when $\mathcal{I}$ asks the random oracle H_2 for this input, then $\mathcal{A}$ will answer with the same h_s.
  (g) Return the response $\theta = (\mathcal{U}', U, R_1, \ldots, R_{n'}, h_1, \ldots, h_{n'}, \sigma)$, where $U = H_1(ID_s)^{h_s} \cdot m$ is the commitment.

Since h' is a random oracle and we are considering H_2 to be a random oracle, so the information provided to $\mathcal{I}$ is indistinguishable from real execution of the identification protocol. However, some collisions may occur because of the values falsified by $\mathcal{A}$.

Note that in particular, no R_i can appear with probability greater than $1/2^k$ in the output produced. The collisions can occur in two ways:

- A tuple $(\mathcal{U}', ID_i, R_i)$ that $\mathcal{I}$ outputs inside a simulated ring identification, has been asked before to the random oracle H_2 by $\mathcal{A}$. The probability of such a collision is, however, less than $\mathcal{Q}_2 \cdot \mathcal{Q}_i \cdot (1/2^k) \leq (1/6)$, where $\mathcal{Q}_2$, $\mathcal{Q}_i$ are the number of H_2 and identification queries, respectively.
- The same tuple $(\mathcal{U}', ID_i, R_i)$ is output by $\mathcal{I}$ in two different simulated ring identification. The probability of this collision happening is less than 1/6 (by birthday paradox).

Combining the above two cases, we get the probability of collision to be $\leq \frac{1}{3}$.

Summing up we have a PPT turing machine $\mathcal{A}$ that simulates the game to $\mathcal{I}$ which is trying to impersonate our scheme. Let's say the probability with which $\mathcal{I}$ can successfully impersonate is ε.

Now we use the oracle replay technique to machine $\mathcal{A}$, with respect to the hash function H_2. This means that by executing twice the machine $\mathcal{A}$ with different instantiations of the hash function H_2 we will obtain two valid transcripts $(\mathcal{U}, U, R_1, \ldots, R_n, h_1, \ldots, h_n, x, W_1)$ and $(\mathcal{U}, U, R_1, \ldots, R_n, h_1', \ldots, h_n', x, W_2)$ with the same commitment U, same challenge x and the same ring $\mathcal{U}$, such that $h_j \neq h_j'$ for some $j \in \{1, \ldots, n\}$ and $h_i = h_i'$ for all $i = 1, \ldots, n$ such that $i \neq j$. This is because the values $(\mathcal{U}, U, R_1, \ldots, R_n)$

have been chosen before the random oracles H_2 and H_2' differ(the oracle replay technique). We have the two transcripts as shown below:

$$W_1 = \sigma_1^e \cdot \sigma_2 = U^x \cdot \prod_{1 \leq i \leq n} \left[(R_i.H_1(ID_i)^{h_i})\right] \ mod\ N$$

$$\text{And, } W_2 = (\sigma_1')^e \cdot \sigma_2 = U^x \cdot \prod_{1 \leq i \leq n} \left[(R_i.H_1(ID_i)^{h_i'})\right] \ mod\ N$$

Dividing the above two equations we get,

$$W_1/W_2 = (\sigma_1/\sigma_1')^e = H_1(ID_j)^{h_j - h_j'}$$

We now proceed further to solve the hard problem. From above equation we have $(\sigma_1/\sigma_1')^e = H_1(ID_j)^{h_j - h_j'}$. Now we look into the table TAB_{H_1} and look for the entry $(ID_j, PK_j, x_j, \beta_j)$ corresponding to identity ID_j, since the impersonation of $\mathcal{I}$ is valid means that the secret key of user ID_j has not been queried and so, with probability (1-μ), we have $\beta_j = 1$ and $PK_j = H_1(ID_j) = y \cdot x_j^e \ mod\ N$.

The relation now becomes $(\sigma_1/\sigma_1')^e.\ x_j^{(h_j' - h_j)e} = y^{(h_j - h_j')} \ mod\ N$. Since h_j and h_j' are outputs of the hash function $H_2 : \{0,1\}^* \rightarrow \{0,1\}^l$, we have that $\mid h_j - h_j' \mid < 2^l < e$. Furthermore, the element e is a prime number, so it holds $gcd(e, h_j - h_j') = 1$. This means that there exists two integers a and b such that $ae + b(h_j - h_j')$ =1 (by using BEZOUT's Identity). Finally we have the value,

$$z = \left((\sigma_1/\sigma_1') \cdot x_j^{(h_j' - h_j)}\right)^b \cdot y^a \ mod\ N$$

Calculating further to check the value of z^e we have,

$$z^e = \left((\sigma_1/\sigma_1') \cdot x_j^{(h_j' - h_j)}\right)^{b \cdot e} \cdot y^{a \cdot e} \ mod\ N$$

$$z^e = y^{(h_j - h_j') \cdot b} \cdot y^{a \cdot e} \ mod\ N$$

$$z^e = y^{(h_j - h_j') \cdot b + a \cdot e} \quad = y \ mod\ N$$

and thus, we arrive at the solution of the given RSA problem.

We are now left to analyse the probability of solving the hard problem. We compute the probability with which the impersonator $\mathcal{I}$ will indeed succeed i.e. $\mathbf{Adv}_{IBI,\mathcal{I}}^{imp-ca}(t)$ as:

$$\begin{aligned}\mathbf{Adv}_{IBI,\mathcal{I}}^{imp-ca}(t) &= \Pr[\mathcal{I} \textit{ succeeds in impersonation AND } \mathcal{A} \textit{ does not halt AND no collisions occur}] \\ &= \Pr[\mathcal{I} \textit{ suceeds in impersonation} \mid \mathcal{A} \textit{ does not halt AND no collisions occur}] \cdot (1 - Pr[\mathcal{A} \textit{ halts OR collisions occur}]) \\ &\geq \varepsilon\left(1 - \tfrac{1}{6} - \tfrac{1}{3}\right) = \tfrac{\varepsilon}{2}\end{aligned}$$

Now, using the reset lemma, we calculate the probability of solving the hard problem,

$$\mathsf{acc}(St, pk) \leq 2^{-l(t)} + \sqrt{\mathsf{res}(St, pk)}$$

$$\mathbf{Adv}_{IBI,\mathcal{I}}^{imp-ca}(t) \leq 2^{-l(t)} + \sqrt{\mathbf{Adv}_{\mathcal{K}_{rsa},\mathcal{A}}^{rsa}(t)}$$

$$\mathbf{Adv}_{\mathcal{K}_{rsa},\mathcal{A}}^{rsa}(t) \geq (\mathbf{Adv}_{IBI,\mathcal{I}}^{imp-ca}(t) - 2^{-l(t)})^2$$

$$\mathbf{Adv}_{\mathcal{K}_{rsa},\mathcal{A}}^{rsa}(t) \geq \left(\tfrac{\varepsilon}{2} - 2^{-l(t)}\right)^2$$

which is a non-negligible quantity.

3. **Anonymity**: With respect to a given identity and a given valid transcript generated by an identity $ID_j \in \mathcal{U}$ in the ring $\mathcal{U}$, the probability that ID_j generated the response this is exactly $1/|\ \mathcal{U}\ |$. It can be easily seen that the IAID protocol transcript are uniform and independent of the user. By considering any two users i and j, the distribution of transcripts for both of them are computationally indistinguishable.

   We show that the transcripts of identification for any two users is similar and is computationally indistinguishable as depicted :

   Interaction Transcript by a user ID_i of the ad-hoc ring:

$$U_i = H_1(ID_i)^{h_i} \cdot m_1 \text{ and}$$

$$W_i = \sigma_{1_i}^e \cdot \sigma_{2_i} = \left[(GSK)^{x+1} \cdot \mathcal{A} \cdot \prod_{i \neq s} \mathcal{A}_i \ mod\ N \right] \cdot m_1^x$$

   Interaction Transcript by a user ID_j of the ad-hoc ring:

$$U_j = H_1(ID_j)^{h_j} \cdot m_2 \text{ and}$$

$$W_j = \sigma_{1_j}^e \cdot \sigma_{2_j} = \left[(GSK)^{x+1} \cdot \mathcal{A} \cdot \prod_{i \neq s} \mathcal{A}_i \ mod\ N \right] \cdot m_2^x$$

   It can be easily seen that the values of U_i and U_j are indistinguishable, similarly W_i and W_j are also indistinguishable. Thus, the communication transcript gives no information of the actual prover, as to who amongst the n users of the ad-hoc ring is actually involved in the identification protocol.

5 Comparison with Existing Schemes

We compare the efficiency and the communication bandwidth consumed by our scheme with the two previous schemes by Gu et al [7] and Nguyen [6]. Let C_E, C_P, C_M be the computational costs of - group exponential operation, bilinear group pairing operation and bilinear group multiplicative operation respectively. In the situation when the value $\prod_{i \neq s} \mathcal{A}_i\ mod\ N$ by the *prover* can be pre-computed ahead of the IAID protocol execution and the value

$\prod_{i=1}^{n} R_i \cdot H_1(ID_i)^{h_i}$ can be pre-computed by the *verifier* after it comes to know the *gpk*, our scheme requires $4C_E$+$4C_M$ computations on the *prover*'s side, and $1C_E$+$1C_M$ computations on the *verifier*'s side. On the other hand the existing schemes of Gu et al [7] and Nguyen [6] require $1C_E$+$2C_M$ and $6C_P$+ $6C_E$ + $12C_M$ for the IAID$_P$ algorithm respectively, and $2C_P$+ $3C_E$ + $1C_M$ and $10C_P$+ $10C_E$ + $8C_M$ for the IAID$_V$ algorithm respectively. Moreover, the scheme prosed by Gu et al [7] assumes a maximum threshold for the ring size which is not the case with our scheme. Our scheme also has a low communication complexity in the identification protocol($IAID$). Table 1 shows the comparison with the most efficient existing schemes and our new ad-hoc anonymous IBI scheme. In this table, q represents a large prime number.

Table 1. Comparison between various ad-hoc anonymous IBI schemes in IAID protocol

Scheme	Our Scheme	Gu et al [7]	Nguyen [6]
Prover Computation	$4C_E$+$4C_M$	$1C_E$+$2C_M$	$6C_P$+ $6C_E$ + $12C_M$
Verifier Computation	$1C_E$+$1C_M$	$2C_P$+ $3C_E$ + $1C_M$	$10C_P$+ $10C_E$ + $8C_M$
Communication	$3\mathbb{Z}_N^*$	$3\mid \mathbb{G} \mid +\mathbb{Z}_q^*$	$7\mid \mathbb{G} \mid +8\mathbb{Z}_q^*$

6 Conclusion

In this paper we present the first IBI scheme for ad-hoc groups without pairings. Anonymous IBI in ad-hoc groups are important cryptographic primitives for access control and resource authorization services among a group of users and hence our scheme can be widely and efficiently used for such purposes. It still remains an open problem to reduce the computation overhead on *prover*'s side and provide an ad-hoc anonymous IBI scheme in standard model and also to propose a novel scheme where the group public key is independent of the ring size involved in the protocol.

References

1. Fiat, A., Shamir, A.: How to prove yourself: Practical solutions to identification and signature problems. In: Odlyzko, A.M. (ed.) CRYPTO 1986. LNCS, vol. 263, pp. 186–194. Springer, Heidelberg (1987)
2. Fiege, U., Fiat, A., Shamir, A.: Zero knowledge proofs of identity. In: Proceedings of the Nineteenth Annual ACM Symposium on Theory of Computing, pp. 210–217. ACM (1987)
3. Beth, T.: Efficient zero-knowledged identification scheme for smart cards. In: Günther, C.G. (ed.) EUROCRYPT 1988. LNCS, vol. 330, pp. 77–84. Springer, Heidelberg (1988)
4. Schnorr, C.-P.: Efficient Signature Generation by Smart Cards. J. Cryptology 4(3), 161–174 (1991)
5. Dodis, Y., Kiayias, A., Nicolosi, A., Shoup, V.: Anonymous Identification in Ad Hoc Groups. In: Cachin, C., Camenisch, J.L. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 609–626. Springer, Heidelberg (2004)

6. Nguyen, L.: Accumulators from bilinear pairings and applications. In: Menezes, A. (ed.) CT-RSA 2005. LNCS, vol. 3376, pp. 275–292. Springer, Heidelberg (2005)
7. Gu, C., Zhu, Y., Ma, C.: An Efficient Identity Based Anonymous Identification Scheme for Ad-Hoc Groups from Pairings. In: 4th International Conference on Wireless Communications, Networking and Mobile Computing, WiCOM 2008, pp. 1–3 (October 2008)
8. Herranz, J.: Identity-based ring signatures from RSA. Theoretical Computer Science 389(1–2), 100–117 (2007)
9. Bellare, M., Palacio, A.: GQ and schnorr identification schemes: Proofs of security against impersonation under active and concurrent attacks. In: Yung, M. (ed.) CRYPTO 2002. LNCS, vol. 2442, pp. 162–177. Springer, Heidelberg (2002)
10. Bellare, M., Namprempre, C., Neven, G.: Security Proofs for Identity-Based Identification and Signature Schemes. J. Cryptol. 22(1), 1–61 (2008)
11. Guillou, L.C., Quisquater, J.-J.: A Practical Zero-Knowledge Protocol Fitted to Security Microprocessor Minimizing Both Transmission and Memory. In: Barstow, D., et al. (eds.) EUROCRYPT 1988. LNCS, vol. 330, pp. 123–128. Springer, Heidelberg (1988)
12. Coron, J.S.: On the Exact Security of Full Domain Hash (2000)