Anonymous Identification for Ad Hoc Group

Xingye Lu
Department of Computing
The Hong Kong Polytechnic University
csxylu@comp.polyu.edu.hk

Man Ho Au
Department of Computing
The Hong Kong Polytechnic University
csallen@comp.polyu.edu.hk

ABSTRACT

An anonymous identification scheme for ad hoc group allows a participant to identify himself as a member of a group of users in a way that his actual identity is not revealed. We propose a highly efficient construction of this cryptographic primitive in the symmetric key setting based on the idea of program obfuscation. The salient feature of our scheme is that only hash evaluations are needed. Consequently, our scheme outperforms all existing constructions for a reasonably large ad hoc group size (of around 50000 users) since no exponentiation nor pairing operation is involved. Technically, the participant only needs to evaluate one hash operation to identify himself. While the time complexity of the verifier is linearly in the size of the ad hoc group, the actual running time is rather insignificant since the constant factor of this linear dependence is the time of a single hash evaluation. To analyse the security of our proposal, we develop a security model to capture the security requirements of this primitive and prove that our construction satisfies these requirements in the random oracle model against unbounded attackers. Similar to other identification schemes secure in the random oracle model, our proposed protocol requires only two message flow.

CCS Concepts

•**Security and privacy → Symmetric cryptography and hash functions; Privacy-preserving protocols;**

Keywords

Cryptography; Anonymous Identification; Obfuscations; Ad Hoc Group

1. INTRODUCTION

An interactive protocol that allows a prover to prove his/her identity to a verifier is commonly referred to as an identification scheme. A typical requirement of such schemes is the resistance against impersonation. Specifically, an identification scheme should forbid anyone from pretending to be another user in the system. However, with the growing awareness of user privacy, an identification providing merely security against impersonation may not be sufficient. People may simply prefer to establish themselves without revealing his/her identity in some situations. For example, due to the fear of retaliation, students participating in an online evaluation for their teachers or professors would prefer logging into the evaluation system without revealing their identity. This has been studied in the ABC4trust project[1], where they launched a pilot to employ privacy-preserving technologies to allow eligible students to conduct online evaluations without revealing their identities. Achieving a mean of 3.373 and a standard deviation of 1.03 on a 5-point Lickert scale, it was concluded that most participants found the system useful for protecting their privacy in the pilot assessment [35]. An identification scheme that hide prover identity, known as an anonymous identification scheme, is desirable in many application scenarios.

More concretely, an anonymous identification scheme allows participants in a user group to prove their membership without revealing any information about the participants' identity. It has been successfully implemented using various approaches. Group signatures, whose concept was first introduced in [11], allow users to anonymously sign a message on behalf of some group. It has been used to implement anonymous identification schemes (for examples [9, 31, 20, 5]). The group formation is performed by a trusted entity known as the group manager who is responsible for registering users into the group. A group signature attests the fact that one of the registered users endorsed the message being signed. In case of a dispute, the group manager is capable of determining the identify of actual signer of a group signature in a process known as "opening of a group signature". In this sense, group signatures offers user anonymity to the outsiders. Based on group signatures, Boneh and Franklin in [6] proposed an anonymous identification scheme allowing identity escrow and subset queries.

A variant of group signatures known as ring signatures was formalized in [29] and further studied in [1, 13, 28, 12, 32, 10, 22]. Similar to group signatures, a ring signature allows a signer to endorse a message on behalf of a group of potential signers. Unlike group signatures, however, the formation of the group of the potential signer in a ring signature is spontaneous, meaning that users could be completely unaware of being conscripted into the group. Furthermore, ring sig-

ASIA CCS '16, May 30-June 03, 2016, Xi'an, China

ISBN 978-1-4503-4233-9/16/05…$15.00

DOI: http://dx.doi.org/10.1145/2897845.2897903

[1]https://abc4trust.eu

natures support full anonymity in the sense that there is no way to revoke the anonymity and reveal the identity of the signer. Ring signatures can be used as an anonymous identification in a typical challenge-response protocol where the verifier challenge the prover to sign a random message. Besides regular ring signature schemes which provide signer full anonymity, there are also ring signature schemes with special functionality. Linkable ring signatures, which were first proposed by Liu et al. [24] and then further studied in [25, 26, 39, 23], provides the function of linking the signatures signed by same signer. They can be applied in scenarios like e-voting [14]. Revocable ring signatures [21] allows any group members to revoke the anonymity of the signer within the same group. Identity-based ring signature, the combination of identity-based signature and ring signature, has been studied in [28, 3, 36, 2]. The difference between a ring signature and an identity-based ring signature is how user identity are presented in the system. For the former, user identities are bind to their public keys via a public key certificate while in the later, a user's identity is used directly as his or her public key. Nguyen [28] proposed a dynamic accumulator scheme for bilinear pairings to construct identity-based ring signatures[2].

To this end, we re-visit the course evaluation scenario discussed above. We observe that several security requirements are desired. Firstly, anonymity is desirable, since the students would be afraid of retribution from the teachers of the course being evaluated. Secondly, security, meaning that only students of the course are eligible to provide feedback, is necessary. Finally, the ability to support ad hoc group identification is needed as students may enroll in or withdraw from different subjects throughout the semesters. Ad hoc anonymous identification based on ring signature schemes described above would fulfill these three requirements. Having said that, we believe that there is no need to use a scheme as powerful as a ring signature scheme, which allows a signer to convince any verifier that he is the owner of a public key listed in a group of public keys associated with the ring signature. In the evaluation scenario mentioned above, we observe that there is only a single verifier who need to anonymously identify enrolled students of the course being evaluated. Requiring all students to have their own public/private key pairs, and to have the evaluation system to verify each of these keys might be too expensive.

Based on this observation, the research problem we plan to address in this paper is to develop a new and more efficient approach to achieving anonymous identification for ad hoc groups that can be applied to scenarios like system login.

1.1 Our Contributions

We solve the research problem mentioned hereinbefore through a modular approach that provides a conceptually simple and efficient solution. Specifically, we made the following contributions.

- We formalize the notion of symmetric-key based anonymous identifications for ad hoc group and develop security models to capture security requirements.
- We introduce a conceptually simple approach based on program obfuscation and a concrete construction of this primitive. We prove that our proposal satisfies the security definitions.
- We conduct empirical analysis on the efficiency of our proposal and show that our system out-perform existing solutions in the setting where there the single verifier in the system shares symmetric keys with the provers.

[2]A flaw of this construction was identified and rectified in [40].

1.2 Overview of Our Approach

We outline the conceptual approach of our design. Assume each user is represented by a unique user identity, $\mathcal{I}$, and that he/she shares a secret key, $\mathsf{sk}_{\mathcal{I}}$, with the verifier. Define function $g_{\mathcal{L}}(\cdot)$ for set $\mathcal{L}$ as follows.

$$g_{\mathcal{L}}(\mathsf{sk}_{\mathcal{I}}) = \begin{cases} 1 & \text{if } \mathsf{sk}_{\mathcal{I}} \in \mathcal{L}, \\ 0 & \text{otherwise} \end{cases}$$

At an abstract level, an identification for ad hoc group in the symmetric key setting is a mechanism that realises a multi-point function. Specifically, the server specifies $\mathcal{L}$ and accept an identification from a user if and only if $g_{\mathcal{L}}(\mathsf{sk}) = 1$.

We further define function $f_{\mathcal{L},\beta}(\cdot)$ as

$$f_{\mathcal{L},\beta}(x) = \begin{cases} \beta & \text{if } g_{\mathcal{L}}(x) = 1, \\ 0 & \text{otherwise} \end{cases}$$

Note that $f_{\mathcal{L},\beta}(\cdot)$ is a multi-input multi-bit output point function. Below we outline our idea in constructing an anonymous identification for ad hoc group in the symmetric key setting based on function $f_{\mathcal{L},\beta}$. As a warm-up, we first give a construction that possesses anonymity but not security against impersonation. The warm-up construction is illustrated in Figure 1. Note that the construction is trivially anonymous, since the output of all legitimate users in list $\mathcal{L}$ is indistinguishable. The same can be said for users not in the list. However, one challenge remains. In the above protocol, there is no mechanism to prevent an attacker from reading the value β from the implementation of function $f_{\mathcal{L},\beta}$ and returns β to authenticate without the need to use a secret key.

To tackle this challenge, we observe that it suffices if $f_{\mathcal{L},\beta}$ can be implemented as a black-box. So, for the security of our scheme, we need to turn the function $f_{\mathcal{L},\beta}$ into a black box which means no one can get any useful information from the implementation of function $f_{\mathcal{L},\beta}$. In our scheme, we use an obfuscation technique to obfuscate function $f_{\mathcal{L},\beta}$ and turn $f_{\mathcal{L},\beta}$ into a black box. Looking ahead, the anonymity of our scheme is unconditional even in the situation when the obfuscation technique is broken while security against impersonation is based on whether or not we can implement function $f_{\mathcal{L},\beta}$ as a black box. This requirement is equivalent to the obfuscation of a multi-input multi-bit output function of which efficient solution exists in the random oracle model.

1.3 Related Work

We discussed briefly the applications of anonymous identification. In the literature, anonymous identification has been applied to various scenarios such as vehicular ad hoc network (VANETs), roaming service in mobility network and cloud data storage. In VANETs, anonymous identification is typically used to protect location privacy of each vehicle in the network. Yao et al. [38] proposed an identification scheme in VANETs that uses biometric encryption along with pseudonymous authentication scheme to achieve

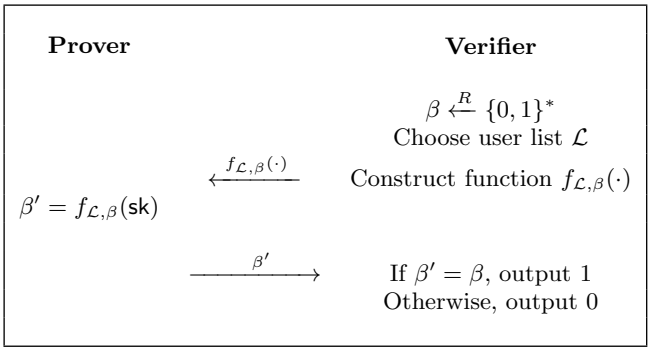

Figure 1: Warm-Up Construction

anonymity. A group signature based anonymous identification scheme supporting threshold authentication, efficient traceability and message linkability in VANETs was presented by Shao et al. [34]. Zheng et al. [18] applied conditionally anonymous ring signature in VANETs to achieve anonymous identification and conditionally tracing. Thanks to the properties of ring signatures, there is no central authority (CA) and roadside units (RSU) which act as group manager as in Shao et al.'s scheme. Another application scenario is roaming services, where users enjoy services provided by home server in the network of a different region, in which anonymous identifications is used to protect user privacy against the foreign network provider. Hafizul et al. [19] proposed an enhanced anonymous identification protocol for roaming service which is devised in the random oracle model using chaotic map and with strong resistance to different attacks and higher efficiency.

As discussed, anonymous identification for ad hoc group is commonly implemented using ring signatures. We briefly reviewed some of the recent advancements. Bresson, Stern and Szydlo presentd extensions to ring signatures and proposed its application to ad hoc groups [7]. The first constant-size ad hoc anonymous identification scheme was introduced by Dodis, Kiayias, Nicolosi and Shoup [13]. The scheme, whose security is rest on the strong RSA assumption, is based on the notion of accumulators with one-way domain. The verification cost is time independent of the size of the ad hoc group while the prover's cost is nearly constant provided that the group does not change rapidly. [23] proposed a linkable ring signature scheme with unconditional anonymity. This work change the common view that linkable ring signatures can provide, at best, computational anonymity. A lightweight ring signature scheme for anonymous identification in ad hoc group was introduced by Yang et al. [37]. While the time and space cost of this scheme is linear in the size of the group, it is suitable for lightweight device due to its special features. Comparing with existing ring signatures, the actual computational cost is lower since there is no exponentiation and pairing operations involved. Instead, the scheme relies on hashing and lightweight modular operations such as squaring and addition.

As mentioned above, we apply obfuscation technique to ensure security against impersonation attack. Informally, an *obfuscator* $\mathcal{O}$ is an efficient and probabilistic "compiler" that transforms a program P into a new program $\mathcal{O}(P)$ which still has the same functionality with P and reveals no secrets that may be used by P. This techniques can be very useful and with wide applications, for example, to prevent tampering or protect copyright a software developer needs obfuscation to hide secrets in the code while maintaining its functionality.

The theoretical investigation of obfuscation was initiated by Barak *et al.* [4] in which they discovered several impossibility results. The first positive results in program obfuscation was presented by Lynn, Prabhakaran and Sahai [27]. Before this, none of the proposed program obfuscation schemes had proven its security properties. In [27], several provably-secure obfuscation techniques were presented in the random oracle model including the obfuscation of multi-point functions. Since the seminal work of [15], research in the applications of the general-purpose indistinguishability obfuscation was fruitful [30]. The main different of our work and this line of results is that our construction only requires obfuscation of a simple function which allows us to take advantage of the efficient obfuscator.

2. PRELIMINARIES

2.1 Notations

Let A be a probabilistic polynomial time (PPT) algorithm. We use $A(x)$ to denote running A on input x with a uniform random tape. For a finite space X, $x \xleftarrow{R} X$ denotes randomly sampling an element x from space X. For a two-party protocol running between a pair of algorithms (P, V), we use $o_x \leftarrow P(x) \overset{z}{\leftrightarrow} V(y) \rightarrow o_y$ to denote the execution of protocol between P and V, with input (z, x) and (z, y) and output o_x and o_y respectively. The set of messages exchanged between P and V is called the transcript of the protocol execution. We use $\pi \leftarrow [P(x) \leftrightarrow V(y)]$ to denote assigning the transcript of the protocol execution to π. We use $\mathsf{negl}(\lambda)$ to denote a function that is negligible in λ.

2.2 Syntax

An ad hoc anonymous identification scheme consists of four efficient algorithms, namely, Setup, Register, Pr, Vf, where:

- $\mathsf{Setup}(1^\lambda)$. On input a security parameter 1^λ, this algorithm generates the system's parameter param. We assume param is an implicit input to all the algorithms listed below.
- $\mathsf{Register}(\mathcal{I})$. This algorithm allows users to register with the system. On input a new user identity, $\mathcal{I}$, this algorithm outputs the corresponding user secret key $\mathsf{sk}_\mathcal{I}$.
- $\mathsf{Pr}(\mathsf{sk}_\mathcal{I}) \overset{\mathcal{L}_\mathcal{I}}{\longleftrightarrow} \mathsf{Vf}(\mathcal{L}_{\mathsf{sk}})$. This is the interactive identification protocol runs between PPT Pr and Vf. The common inputs to the algorithms are a list of user identities $\mathcal{L}_\mathcal{I} := \{\mathcal{I}_1, \mathcal{I}_2, \ldots, \mathcal{I}_\ell\}$ and the list of corresponding secret key $\mathcal{L}_{\mathsf{sk}} := \{\mathsf{sk}_{\mathcal{I}_1}, \mathsf{sk}_{\mathcal{I}_2}, \ldots, \mathsf{sk}_{\mathcal{I}_\ell}\}$. Upon successful completion of the protocol, Vf outputs 0/1 to indicate rejection or acceptance.

Typically, the identification system consists of one server (verifier) and a number of users (provers). Algorithm Setup is usually executed by the server. Register could be initiated by the server or a user, depending on the application scenario. When a user wishes to identify anonymously to the server, the two parties engage in an interactive identification

protocol where Pr and Vf will be executed by the user and the server respectively. We note that the common input $\mathcal{L}_{\mathcal{I}}$ can be chosen by either the server or the user, depending on the application scenario.

Correctness. We required that an honest verifier will always accept the identification from an honest prover. More formally, we require that the quantity

$$\Pr\left[\begin{array}{c|c} \begin{array}{c}\mathsf{param} \leftarrow \mathsf{Setup}(1^\lambda)\\ \mathsf{sk}_{\mathcal{I}_i} \leftarrow \mathsf{Register}(\mathcal{I}_i)\\ \text{for } i = 1 \text{ to } n\\ \mathcal{L}_{\mathcal{I}} := \{\mathcal{I}_1, \ldots, \mathcal{I}_n\}\\ \mathcal{L}_{\mathsf{sk}} := \{\mathsf{sk}_{\mathcal{I}_1}, \ldots, \mathsf{sk}_{\mathcal{I}_n}\}\\ \mathsf{sk}_{\mathcal{I}} \xleftarrow{R} \mathcal{L}_{\mathsf{sk}}\end{array} & \mathsf{Pr}(\mathsf{sk}_{\mathcal{I}}) \xleftrightarrow{\mathcal{L}_{\mathcal{I}}} \mathsf{Vf}(\mathcal{L}_{\mathsf{sk}}) \rightarrow 1 \end{array}\right]$$

is greater than or equal to $1 - \mathsf{negl}(\lambda)$.

2.3 Security Requirements

We consider two security requirements for an ad hoc anonymous identification scheme, namely, anonymity and soundness. In this subsection, we formalize these requirements as games between a challenger and an attacker.

2.3.1 Anonymity

To define the anonymity of an ad hoc anonymous identification scheme, we define the following game, denoted as $\mathsf{Game}_{\mathsf{anon}}$, between a challenger $\mathcal{C}$ and an adversary $\mathcal{A}$.

Setup. Challenger $\mathcal{C}$ runs Setup with security parameter 1^λ and generates system parameter param. Then $\mathcal{C}$ sends param to adversary $\mathcal{A}$.

Challenge. $\mathcal{A}$ sends a list of identity-key pairs $\{(\mathcal{J}_i, \mathsf{sk}_{\mathcal{J}_i})\}_{i=1}^{\ell}$ to $\mathcal{C}$. $\mathcal{A}$ further specifies two identities $\mathcal{I}_0, \mathcal{I}_1$ such that $\mathcal{I}_0 = \mathcal{J}_x$ and $\mathcal{I}_1 = \mathcal{J}_y$ for some $x, y \in \{1, \ldots, \ell\}$. We use $\mathsf{sk}_{\mathcal{I}_0}$ (resp. $\mathsf{sk}_{\mathcal{I}_1}$) to denote $\mathsf{sk}_{\mathcal{J}_x}$ (resp. $\mathsf{sk}_{\mathcal{J}_y}$). Define $\mathcal{L}^*_{\mathcal{I}}$ to be $\{\mathcal{J}_i\}_{j=1}^{\ell}$ and $\mathcal{L}^*_{\mathsf{sk}} := \{\mathsf{sk}_{\mathcal{J}_i}\}_{i=1}^{\ell}$.

Then $\mathcal{C}$ flips a fair coin $b \in \{0, 1\}$ and invokes $\pi^* \leftarrow [\mathsf{Pr}(\mathsf{sk}_{\mathcal{I}_b}) \xleftrightarrow{\mathcal{L}^*_{\mathcal{I}}} \mathsf{Vf}(\mathcal{L}^*_{\mathsf{sk}})]$. The resulting transcript, π^*, is given to $\mathcal{A}$.

Guess. Finally, $\mathcal{A}$ outputs a bit b'. We say that $\mathcal{A}$ wins $\mathsf{Game}_{\mathsf{anon}}$ if $b = b'$.

The advantage of $\mathcal{A}$, $\mathsf{adv}_{\mathcal{A},\mathsf{anon}}$, is defined as the probability that it wins the above game minus $\frac{1}{2}$.

Definition 1. *An ad hoc anonymous identification scheme is said to offer anonymity if for any adversary $\mathcal{A}$, the advantage $\mathsf{adv}_{\mathcal{A},\mathsf{anon}}$ in the above game is negligible.*

In the above definition, $\mathcal{A}$ is not computationally bounded. In other words, a scheme satisfying Definition 1 offers unconditional anonymity. We also note that the above game allows the adversary to present maliciously chosen keys (i.e., keys that do not follow the distribution of algorithm $\mathsf{Register}$). In other words, anonymity has to be preserved even when the keys are not properly chosen.

2.3.2 Soundness

Soundness of ad hoc anonymous identification scheme captures the requirement that a user without a legitimate secret key for an identity in the list of authenticating users should be rejected in an identification protocol by the verifier. We introduce $\mathsf{Game}_{\mathsf{sec}}$ between a challenger $\mathcal{C}$ and an attacker $\mathcal{A}$ to formally capture this intuition. The goal of $\mathcal{A}$ in the game is to prove his identity without valid user ID and secret key pair $(\mathcal{I}, \mathsf{sk})$.

Setup. Challenger $\mathcal{C}$ runs Setup with security parameter 1^λ and generates system parameter param. Then $\mathcal{C}$ sends param to adversary $\mathcal{A}$. $\mathcal{C}$ maintains two lists, namely, the list of honest users ($\mathcal{L}_H$) and corrupted users ($\mathcal{L}_C$).

Query. $\mathcal{A}$ can issue three types of queries in an adaptive manner.

- $\mathsf{Register}(\mathcal{I}, \omega)$. $\mathcal{A}$ can issue $\mathsf{Register}$-queries to $\mathcal{C}$ to introduce users into the system. If $\omega = \perp$, $\mathcal{C}$ invokes $\mathsf{Register}$ on input $\mathcal{I}$ and obtains $\mathsf{sk}_{\mathcal{I}}$. $\mathcal{I}$ is added to $\mathcal{L}_H$. Otherwise, $\mathcal{C}$ sets $\mathsf{sk}_{\mathcal{I}} := \omega$ adds $\mathcal{I}$ to $\mathcal{L}_C$.
- $\mathsf{Corrupt}(\mathcal{I})$. $\mathcal{A}$ submits a user identity $\mathcal{I}$. If $\mathcal{I}$ is included in $\mathcal{L}_H$, $\mathcal{C}$ returns to $\mathcal{A}$ the corresponding $\mathsf{sk}_{\mathcal{I}}$ and moves $\mathcal{I}$ from $\mathcal{L}_H$ to $\mathcal{L}_C$.
- $\mathsf{Trans}(\mathcal{I}, \mathcal{L}_{\mathcal{I}})$. $\mathcal{A}$ chooses a set of users $\mathcal{L}_{\mathcal{I}} \subset \mathcal{L}_C \cup \mathcal{L}_H$ and a user $\mathcal{I} \in \mathcal{L}_{\mathcal{I}}$ to obtain an identification transcript. $\mathcal{C}$ first collects the corresponding user secret key $\mathcal{L}_{\mathsf{sk}} := \{\mathsf{sk}_{\mathcal{I}} | \mathcal{I} \in \mathcal{L}_{\mathcal{I}}\}$. Next, it executes $\pi \leftarrow [\mathsf{Pr}(\mathsf{sk}_{\mathcal{I}}) \xleftrightarrow{\mathcal{L}_{\mathcal{I}}} \mathsf{Vf}(\mathcal{L}_{\mathsf{sk}})]$ and returns π to $\mathcal{A}$.

Challenge. $\mathcal{A}$ chooses a set of user identities $\mathcal{L}^* \subset \mathcal{L}_H$ on which it wishes to be challenged. $\mathcal{C}$ parses $\mathcal{L}^*_{\mathsf{sk}} := \{\mathsf{sk}_{\mathcal{I}} | \mathcal{I} \in \mathcal{L}^*\}$. Next, $\mathcal{C}$ plays the role of the verifier with input $(\mathcal{L}^*, \mathcal{L}^*_{\mathsf{sk}})$ with $\mathcal{A}$ acting as a prover. We says that $\mathcal{A}$ wins the game if and only if

$$\mathcal{A} \xleftrightarrow{\mathcal{L}^*} \mathsf{Vf}(\mathcal{L}^*_{\mathsf{sk}}) \rightarrow 1$$

The advantage of $\mathcal{A}$, $\mathsf{adv}_{\mathcal{A},\mathsf{sec}}$, is defined as the probability that it wins the above game.

Definition 2. *An ad hoc anonymous identification scheme is sound if for any PPT adversary $\mathcal{A}$ the advantage $\mathsf{adv}_{\mathcal{A},\mathsf{sec}}$ is negligible .*

We would like to remark that our definitions (Definition 1 and Definition 2) only allow the attacker to passively eavesdrop the communications. We note that this is a common security requirement for identification protocols as in [13]. One possible reason is that in most cases, the constructions are Σ-protocol that will be converted generically to its non-interactive form in which the generic construction requires the identification protocol to be passively sound. However, we shall see in Section 3 that our protocol is not a Σ-protocol. The implication of the choice this security definition will be discussed after we present our construction.

3. OUR CONSTRUCTION

In this section, we give the details of our scheme.

$\mathsf{Setup}(1^\lambda)$: On input 1^λ, the algorithm chooses a hash function $\mathcal{R} : \{0,1\}^* \rightarrow \{0,1\}^{2\lambda + s(\lambda)}$, where $s(\lambda)$ is a quantity polynomial in λ. Set param as $\mathcal{R}$, which will be modelled as a random oracle.

$\mathsf{Register}(\mathcal{I})$: On input a new user identity, $\mathcal{I}$, the algorithm first compares $\mathcal{I}$ with all the identities stored in its database. If $\mathcal{I}$ exists, it returns false and abort. Otherwise,

it randomly generates random bit-string $\mathsf{sk}_{\mathcal{I}} \in_R \{0,1\}^{\lambda}$. Then, the tuple $(\mathcal{I}, \mathsf{sk}_{\mathcal{I}})$ is stored in its database.

$\mathsf{Pr}(\mathsf{sk}_{\mathcal{I}}) \overleftrightarrow{\mathcal{L}_{\mathcal{I}}} \mathsf{Vf}(\mathcal{L}_{\mathsf{sk}})$: The pair of interactive algorithms are to be executed by the prover and the verifier respectively. The prover and the verifier first agrees on the list of user identities $\mathcal{L}_{\mathcal{I}} := \{\mathcal{I}_1, \mathcal{I}_2, \ldots, \mathcal{I}_\ell\}$. The anonymous identification protocol between the prover and the verifier is a two-move protocol:

1. From $\mathcal{L}_{\mathcal{I}}$, the verifier obtains the list of corresponding secret keys $\mathcal{L}_{\mathsf{sk}} := \{\mathsf{sk}_{\mathcal{I}_1}, \mathsf{sk}_{\mathcal{I}_2}, \ldots, \mathsf{sk}_{\mathcal{I}_\ell}\}$ from the database. Then, it picks a random $\beta \in_R \{0,1\}^{s(\lambda)}$. The verifier computes $\mathcal{O}^{\mathcal{R}}(f_{\mathcal{L}_{\mathsf{sk}},\beta})$, the obfuscation of the function $f_{\mathcal{L}_{\mathsf{sk}},\beta}$ and sends it to the prover. Recall that $f_{\mathcal{L}_{\mathsf{sk}},\beta}$ is a multi input multi-bit output point function with the following specification:

$$f_{\mathcal{L}_{\mathsf{sk}},\beta}(\mathsf{sk}_{\mathcal{I}}) = \begin{cases} \beta & \text{if } \mathsf{sk} \in \mathcal{L}_{\mathsf{sk}} \\ 0 & \text{otherwise} \end{cases}$$

   Details of $\mathcal{O}^{\mathcal{R}}(f_{\mathcal{L}_{\mathsf{sk}},\beta})$ will be discussed in the next paragraph.

2. Upon receiving $\mathcal{O}^{\mathcal{R}}(f_{\mathcal{L}_{\mathsf{sk}},\beta})$, the prover evaluates $\beta' := \mathcal{O}^{\mathcal{R}}(f_{\mathcal{L}_{\mathsf{sk}},\beta})(\mathsf{sk}_{\mathcal{I}})$. The prover returns β' to the verifier if $\beta' \neq 0$.

3. The verifier outputs 1 if and only if $\beta = \beta'$. Otherwise, it outputs 0.

Our construction is shown in Figure 2.

Details of $\mathcal{O}^{\mathcal{R}}(f_{\mathcal{L}_{\mathsf{sk}},\beta})$.

In this paragraph, we discuss an efficient implement of $\mathcal{O}^{\mathcal{R}}(f_{\mathcal{L}_{\mathsf{sk}},\beta})$ using obfuscation of multi-point functions based on the techniques from [27]. The obfuscation of $f_{\mathcal{L}_{\mathsf{sk}},\beta}$ is constructed as follows.

- Denote by $\mathcal{R}_1(\cdot)$ the first 2λ bits output of $\mathcal{R}$ and $\mathcal{R}_2(\cdot)$ the last $s(\lambda)$ bits of $\mathcal{R}$. Choose a random $\delta \in_R \{0,1\}^{\lambda}$.
- Parse $\mathcal{L}_{\mathsf{sk}}$ as $\{\mathsf{sk}_1, \ldots, \mathsf{sk}_\ell\}$, where $\ell = |\mathcal{L}_{\mathsf{sk}}|$. For $i = 1$ to ℓ, compute $a_i = \mathcal{R}_1(\delta, sk_i)$, $b_i = \mathcal{R}_2(\delta, sk_i)$, $c_i = \beta \oplus b_i$. The obfuscated function $\mathcal{O}^{\mathcal{R}}(f_{\mathcal{L}_{\mathsf{sk}},\beta})$ is defined as $(\delta, \{a_i, c_i\}_{i=1}^{\ell})$.
- To evaluate $\mathcal{O}^{\mathcal{R}}(f_{\mathcal{L}_{\mathsf{sk}},\beta})(x)$, locate i such that $a_i = \mathcal{R}_1(\delta, x)$ and outputs $c_i \oplus \mathcal{R}_2(\delta, x)$. If i cannot be found, output 0.

Discussions.

Note that the downlink is of complexity $O(\ell)$ while the uplink is of constant complexity. The verifier's computation is $O(\ell)$, while that of the prover can be reduced to $\mathcal{O}(1)$ if the list of identities is used to label the values a_i's so that the prover knows exactly which i should he based his computation on.

4. ANALYSIS

In this section we are going to prove the security of our scheme and give the efficiency analysis.

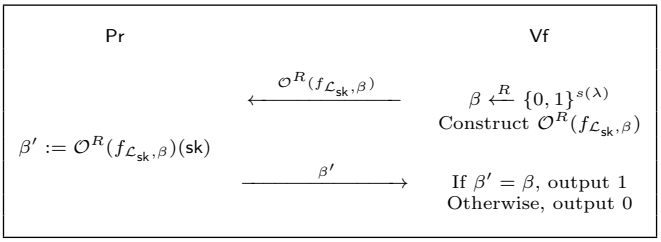

Figure 2: Our anonymous identification protocol based on obfuscations

4.1 Security Proof

THEOREM 1. *Our ad hoc anonymous identification scheme is unconditionally anonymous according to Definition 1.*

PROOF. Assume there exists an adversary trying to attack the anonymity of proposed scheme, the adversary chooses two valid user identity and secret key pairs $(\mathcal{I}_0, \mathsf{sk}_{\mathcal{I}_0})$,$(\mathcal{I}_1, \mathsf{sk}_{\mathcal{I}_1})$. To break the anonymity, the adversary is given the transcript between Pr and Vf generated from one of the two pairs. The goal of the adversary is to guess whether $(\mathcal{I}_0, \mathsf{sk}_{\mathcal{I}_0})$ or $(\mathcal{I}_1, \mathsf{sk}_{\mathcal{I}_1})$ is used to produce this transcript.

In our scheme, the transcript π between Pr and Vf consists of the obfuscation of function $f_{\mathcal{L}_{\mathsf{sk}},\beta}$ and a string β' which is the output of the obfuscation. If the input secret key of obfuscated $f_{\mathcal{L}_{\mathsf{sk}},\beta}$ is a valid one, the string β' should be equal to the string β which is preselected by Vf and obfuscated in the obfuscation of $f_{\mathcal{L}_{\mathsf{sk}},\beta}$ no matter the valid secret key belongs to which user. So in the view of adversary, transcripts generated from $(\mathcal{I}_b, \mathsf{sk}_{\mathcal{I}_b})$, for $b = 0$ or 1, is identical. In other words, the view of the adversary is completely independent to b. Thus, the probability for an adversary winning the game is the same as random guessing. So the advantage for the adversary in game $\mathsf{Game}_{\mathsf{anon}}$ is always negligible. In other words, our scheme is secure according to Definition 1. □

THEOREM 2. *Our ad hoc anonymous identification scheme is sound according to Definition 2.*

PROOF. We describe the proof using the game-hoping technique with two games, where the first game is the original soundness game defined in Section 2.3.2. We prove that for a polynomial time adversary $\mathcal{A}$, the probability that the advantage of $\mathcal{A}$ in the first game is close to that in the second game. Next, we show that $\mathcal{A}$ wins the second game with negligible probability. The two games are defined below:

Game 1: The first game is identical to the original soundness game described in 2.3.2.

Game 2: The second game has the following steps:

Setup. This is the same as in **Game 1**. System parameter param is generated by algorithm Setup and passed to adversary $\mathcal{A}$.

Query. $\mathcal{A}$ will issue three types of query:

- $\mathsf{Register}(\mathcal{I}, \omega)$. When $\mathcal{A}$ issues $\mathsf{Register}$-queries, if $\omega = \perp$, a string will be randomly sampled from secret key space as $\mathsf{sk}_{\mathcal{I}}$. $\mathcal{I}$ is added to $\mathcal{L}_H$. Otherwise, $\mathcal{C}$ sets $\mathsf{sk}_{\mathcal{I}} := \omega$ adds $\mathcal{I}$ to $\mathcal{L}_C$.
- $\mathsf{Corrupt}(\mathcal{I})$. $\mathcal{A}$ submits user identity $\mathcal{I}$. If $\mathcal{I}$ is included in $\mathcal{L}_H$, $\mathcal{C}$ returns to $\mathcal{A}$ the corresponding

$\mathsf{sk}_{\mathcal{I}}$ and moves $\mathcal{I}$ from $\mathcal{L}_H$ to $\mathcal{L}_C$. And meanwhile, $\mathcal{R}(\delta, \mathsf{sk}_{\mathcal{I}})$ will be programmed as $a_{\mathcal{I}}||(c_{\mathcal{I}} \oplus \beta)$ for $(\delta, a_{\mathcal{I}}, c_{\mathcal{I}})$ from all queries of π in $\mathsf{Trans}(\mathcal{I}, \mathcal{L}_{\mathcal{I}})$ which $\mathcal{L}_{\mathcal{I}}$ includes $\mathcal{I}$.

- $\mathsf{Trans}(\mathcal{I}, \mathcal{L}_{\mathcal{I}})$. $\mathcal{A}$ chooses a set of users $\mathcal{L}_{\mathcal{I}} \subset \mathcal{L}_C \cup \mathcal{L}_H$ and user $\mathcal{I} \in \mathcal{L}_{\mathcal{I}}$ to obtain an identification transcript. Parse $\mathcal{L}_{\mathcal{I}} := \{\mathcal{I}_1, \mathcal{I}_2, \ldots, \mathcal{I}_\ell\}$. When $\mathcal{A}$ queries for the transcription, β is randomly chosen from $\{0,1\}^{s(\lambda)}$ and obfuscation of $f_{\mathcal{L}_{\mathsf{sk}},\beta}$ is constructed by randomly choosing $a_{\mathcal{I}_1}, a_{\mathcal{I}_2} ..., a_{\mathcal{I}_\ell}$ from $\{0,1\}^{2\lambda}$, $c_{\mathcal{I}_1}, c_{\mathcal{I}_2} ..., c_{\mathcal{I}_\ell}$ from $\{0,1\}^{s(\lambda)}$ and δ from $\{0,1\}^{\lambda}$. Transcript π is constructed by β and the obfuscation of $f_{\mathcal{L}_{\mathsf{sk}},\beta}$. π then will be sent to $\mathcal{A}$. $\mathcal{R}(\delta, \mathsf{sk}_{\mathcal{I}_i})$ will be programmed as $a_{\mathcal{I}_i}||(c_{\mathcal{I}_i} \oplus \beta)$ for $\{\mathsf{sk}_{\mathcal{I}_i} | i = 1...\ell, \mathcal{I}_i \in \mathcal{L}_C\}$ and $\mathcal{R}(\delta, \mathsf{sk}_{\mathcal{I}_i})$ will remain unprogrammed for the rest $\mathsf{sk}_{\mathcal{I}_i}$.

Challenge. $\mathcal{A}$ chooses a set of user identities $\mathcal{L}^* \subset \mathcal{L}_H$, $\mathcal{L}^* := \{\mathcal{I}_1, \mathcal{I}_2, \ldots, \mathcal{I}_\ell\}$ on which it wishes to be challenged. $\mathcal{A}$ will receive the obfuscation of function $f_{\mathcal{L}_{\mathsf{sk}},\beta}$ in which δ, $a_{\mathcal{I}_i}$ and $c_{\mathcal{I}_i}$ are randomly chosen from $\{0,1\}^{\lambda}$, $\{0,1\}^{2\lambda}$ and $\{0,1\}^{s(\lambda)}$ $(i = 1...\ell)$. Next, $\mathcal{A}$ players the role of prover when receiving the obfuscation. We says that $\mathcal{A}$ wins the game if and only if the output string from $\mathcal{A}$ equals to the randomly chosen β in $\{0,1\}^{s(\lambda)}$ by $\mathcal{C}$.

To adversary $\mathcal{A}$, **Game 2** is identical to the original game. In original obfuscation of $f_{\mathcal{L}_{\mathsf{sk}},\beta}$, $a_{\mathcal{I}_i}$, $b_{\mathcal{I}_i}$ are all generated from random oracle and $c_{\mathcal{I}_i} = b_{\mathcal{I}_i} \oplus \beta$, of which the distribution to adversary $\mathcal{A}$ is the same as randomly choosing $a_{\mathcal{I}_i}$, $c_{\mathcal{I}_i}$ from $\{0,1\}^{2\lambda}$, $\{0,1\}^{s(\lambda)}$. Since in the original obfuscation, $a_{\mathcal{I}_i}$ is computed by $\mathcal{R}_1(\delta, \mathsf{sk}_{\mathcal{I}_i})$ and $c_{\mathcal{I}_i}$ is computed by $\mathcal{R}_2(\delta, \mathsf{sk}_{\mathcal{I}_i}) \oplus \beta$. After programming the output of $\mathcal{R}(\delta, \mathcal{I}_i)$ to $a_{\mathcal{I}_i}||(c_{\mathcal{I}_i} \oplus \beta)$, adversary $\mathcal{A}$ will not notice the secret key $\mathsf{sk}_{\mathcal{I}_i}$ he/she owing is not used to construct obfuscation.

The only possible way for $\mathcal{A}$ to behalf differently in Game 1 and Game 2 is that $\mathcal{A}$ queries the random oracle $\mathcal{R}(.)$ on any secret key in $\mathcal{L}_H$. Assume $\mathcal{A}$ queries the random oracle q times, the probability for this happening is:

$$\Pr \leqslant \frac{q\ell}{2^{\lambda}},$$

which is negligible (ℓ is the number of user identity in $\mathcal{L}_H$).

Since in **Game 2**, adversary $\mathcal{A}$ gains no knowledge related to secret keys of users he/she chose to attack during queries for transcript π and user secret keys. Besides, $a_{\mathcal{I}_i}$ and $c_{\mathcal{I}_i}$ in the challenge obfuscation are chose at random and has no relation with either $\mathsf{sk}_{\mathcal{I}_i}$ or β. Thus, the probability for $\mathcal{A}$ to successfully win Game 2 is:

$$\Pr = \frac{1}{2^{s(\lambda)}},$$

which is negligible.

To sum up, the probability for adversary $\mathcal{A}$ to win Game 1 is no more than $\frac{1}{2^{s(\lambda)}} + \frac{ql}{2^{\lambda}}$ which is also negligible. In other words, our scheme is secure according to Definition 2. □

Discussions.

In our construction, the set of eligible users, $\mathcal{L}_{\mathcal{I}}$, has to be agreed by both the prover and the verifier. For maximum user privacy protection, $\mathcal{L}_{\mathcal{I}}$ can be chosen as the set of all eligible users. We would like to remark that our protocol is not a 3-move Σ-protocol. Consequently, we cannot employ the classical results that turns an honest verifier zero-knowledge protocol into a full zero-knowledge protocol [16]. In particular, a malicious server could obfuscate a "wrong" program so that each user secret key will output a different β. In other words, an active malicious verifier could break the anonymity of the scheme. We outline a solution to mitigate this attack. Specifically, the server also publishes $H(\beta)$ for each authentication request. User can abort if the hash of the output from the obfuscated program is not equivalent to the published hash value. This, however, still do not prevent the selective-failure attack as the server can just put one valid secret key inside the program to be obfuscated. This appears to be an inherent limitation in the symmetric key setting, since an honest user has no way to ensure other user identities included in the group are legitimate. Our protocol is, thus, suitable for applications where the verifier is "honest-but-curious".

4.2 Efficiency Analysis

We provide an empirical analysis of the efficiency of our proposal and compare it with existing anonymous identification schemes. We first provide a breakdown of the time cost of the schemes based on the number of exponentiation operations (EXP) and pairing operations (PAIR). We remark that PAIR is usually regarded as an expensive operation in comparison with EXP, which in turn takes significantly more time compared with the evaluation of a hash function[3]. Assuming there are ℓ members in the ad hoc group, the breakdown of the major operations for various schemes is summarised in Table 1. We do not consider hash operations as a part of the time complexity since compared with exponentiation and pairing operations, the time of evaluating a hash function on a short input is rather insignificant. Comparing with [37], which also do not require exponentiation or pairing operations, we are still more efficient. In [37], there are $n-1$ modular squaring, addition and hash operations, plus 1 square-root operation on the prover side, n modular squaring and hash operations on the verifier side. In our scheme there are n hash operations on the prover side and n hash and addition operations on the verifier side. Since modular squaring is an expensive operation comparing with hash evaluation with short input. In other words, our scheme is still more efficient compared with the state-of-the-art.

To give an estimate of the actual running time of these schemes, we instantiate the numbers based on the benchmark results from [17]. The numbers are obtained, as per [17], from a PIV 3-GHZ processor with 512-MB memory and a Windows XP operation system. Running time for these operations is obtained by using a standard cryptographic library MIRACL [33]. The benchmark results of each major operation is shown in Table 2.

Based on these numbers, we can calculate the approximate time for each scheme. The results are presented in Table 1.

It is obvious that, as long as the ad hoc group size of our scheme is not large (say, less than 50000), our scheme is more efficient than all the other schemes in the literature. The space complexity of our scheme is linearly in the group

[3]The exception is to hash an arbitrary string into a point on an elliptic curve group, which could be expensive or impossible to achieve depending on the underlying group structure.

size too, since the obfuscation of $f_{\mathcal{L}_{\mathsf{sk},\beta}}$ should contain all the obfuscated user secret keys in the ad hoc group.

We argue that in practice, however, our scheme is more space-efficient than most existing schemes. For a list of ℓ user identities, the total number of bits to be transmitted is $\ell * (4\lambda) + \lambda$, assuming we use $\mathcal{I}$ to label each a_i, c_i in the obfuscated point function[4]. This number is, in fact, less than the public-key based scheme if we take into account the list of public keys to be transmitted. Specifically, for a ring signature of ℓ ring members, we need to transmit ℓ public keys and ℓ certificates. This is almost certainly larger than that of $2\ell\lambda$. Concrete numbers are illustrated in Table 3.

We would like to remark again that our construction is applicable to a less general scenario. Specifically, anonymous identification scheme based on ring signatures allows a signer to convince all verifiers that the former belongs to a group while in our system, we consider a single verifier. Having said that, the efficiency of our scheme allows it to be used on lightweight devices and applied in system login scenario we mentioned above in which a user only needs to confirm his/her identity with a single server. Lastly, we would like to remark that similar to the other systems in the random oracle model, our protocol is also of two rounds.

Scheme	$\mathbf{EXP}_p$	$\mathbf{EXP}_v$	$\mathbf{PAIR}_p$	$\mathbf{PAIR}_v$
Rivest-Shamir-Tauman [29]	ℓ	ℓ	0	0
Abe-Ohkubo-Suzuki [1]	ℓ	ℓ	0	0
Dodis-Kiayias-Nicolosi-Shoup [13]	8	14	0	0
Nguyen [28]	13	10	0	2
Chow-Liu-Wei-Yuen [12]	ℓ	ℓ	0	0
Shacham-Waters [32]	$4\ell + 3$	0	0	$2\ell + 3$
Chandran-Groth-Sahai [10]	$\frac{\ell+1}{3} + 6\sqrt{\ell}$ +5	1	0	$l + 7\sqrt{\ell}$ +5
Liu-Au-Susilo-Zhou [22]	offline $\ell - 1$ online 1	0 0	0 0	ℓ ℓ
Xiu-Wu-Liu-Chen[37]	0	0	0	0
Our scheme	0	0	0	0

Table 1: $\mathbf{EXP}_p$ and $\mathbf{EXP}_v$ represents the number of exponentation operation done by Pr and Vf. $\mathbf{PAIR}_p$ and $\mathbf{PAIR}_v$ represents the number of pairing operation done by Pr and Vf. ℓ is the group size

operation	pairing	exponentiation	hash
time	20.04	5.31	< 0.001

Table 2: Running time for each operation (in milisecond) [17]

[4]This will allow a prover to identify directly which index should he use, and thus reduce the evaluation of the obfuscated function to the computation of a single hash value.

scheme	Prover	Verifier
[29]	5.31ℓ	5.31ℓ
[1]	5.31ℓ	5.31ℓ
[13]	42.48	74.34
[28]	69.03	93.18
[12]	5.31ℓ	5.31ℓ
[32]	21.24ℓ+15.93	40.08ℓ+60.12
[10]	$1.77\ell + 31.86\sqrt{\ell} + 28.32$	$20.04\ell + 140.28\sqrt{\ell} + 105.51$
[22]	offline:$5.31(\ell$-1) online:5.31	20.04ℓ 20.04ℓ
ours	$<0.001\ell$ <0.001 (optimized version)	$<0.001\ell$ $<0.001\ell$

Table 3: Operation time for each scheme (in millisecond)

5. CONCLUSIONS

In this paper, we presented a new approach to support ad hoc anonymous identification in the symmetric key setting. We presented a concrete construction of our proposed approach and proved that it satisfies the security requirements. We analysed the efficiency of our proposal and showed that it compares favourably with the existing constructions in the setting that features a single server and a set of users.

6. REFERENCES

[1] M. Abe, M. Ohkubo, and K. Suzuki. 1-out-of-n signatures from a variety of keys. In Y. Zheng, editor, *Advances in Cryptology - ASIACRYPT 2002, 8th International Conference on the Theory and Application of Cryptology and Information Security, Queenstown, New Zealand, December 1-5, 2002, Proceedings*, volume 2501 of *Lecture Notes in Computer Science*, pages 415–432. Springer, 2002.

[2] M. H. Au, J. K. Liu, W. Susilo, and T. H. Yuen. Secure id-based linkable and revocable-iff-linked ring signature with constant-size construction. *Theor. Comput. Sci.*, 469:1–14, 2013.

[3] M. H. Au, J. K. Liu, T. H. Yuen, and D. S. Wong. Id-based ring signature scheme secure in the standard model. In H. Yoshiura, K. Sakurai, K. Rannenberg, Y. Murayama, and S. Kawamura, editors, *Advances in Information and Computer Security, First International Workshop on Security, IWSEC 2006, Kyoto, Japan, October 23-24, 2006, Proceedings*, volume 4266 of *Lecture Notes in Computer Science*, pages 1–16. Springer, 2006.

[4] B. Barak, O. Goldreich, R. Impagliazzo, S. Rudich, A. Sahai, S. P. Vadhan, and K. Yang. On the (im)possibility of obfuscating programs. *J. ACM*, 59(2):6, 2012.

[5] M. Bellare, D. Micciancio, and B. Warinschi. Foundations of group signatures: Formal definitions, simplified requirements, and a construction based on general assumptions. In E. Biham, editor, *Advances in Cryptology - EUROCRYPT 2003, International Conference on the Theory and Applications of Cryptographic Techniques, Warsaw, Poland, May 4-8, 2003, Proceedings*, volume 2656 of *Lecture Notes in Computer Science*, pages 614–629. Springer, 2003.

[6] D. Boneh and M. K. Franklin. Anonymous authentication with subset queries (extended abstract). In J. Motiwalla and G. Tsudik, editors, *CCS '99, Proceedings of the 6th ACM Conference on*

Computer and Communications Security, Singapore, November 1-4, 1999., pages 113–119. ACM, 1999.

[7] E. Bresson, J. Stern, and M. Szydlo. Threshold ring signatures and applications to ad-hoc groups. In M. Yung, editor, *Advances in Cryptology - CRYPTO 2002, 22nd Annual International Cryptology Conference, Santa Barbara, California, USA, August 18-22, 2002, Proceedings*, volume 2442 of *Lecture Notes in Computer Science*, pages 465–480. Springer, 2002.

[8] C. Cachin and J. Camenisch, editors. *Advances in Cryptology - EUROCRYPT 2004, International Conference on the Theory and Applications of Cryptographic Techniques, Interlaken, Switzerland, May 2-6, 2004, Proceedings*, volume 3027 of *Lecture Notes in Computer Science*. Springer, 2004.

[9] J. Camenisch and M. Stadler. Efficient group signature schemes for large groups (extended abstract). In B. S. K. Jr., editor, *Advances in Cryptology - CRYPTO '97, 17th Annual International Cryptology Conference, Santa Barbara, California, USA, August 17-21, 1997, Proceedings*, volume 1294 of *Lecture Notes in Computer Science*, pages 410–424. Springer, 1997.

[10] N. Chandran, J. Groth, and A. Sahai. Ring signatures of sub-linear size without random oracles. In L. Arge, C. Cachin, T. Jurdzinski, and A. Tarlecki, editors, *Automata, Languages and Programming, 34th International Colloquium, ICALP 2007, Wroclaw, Poland, July 9-13, 2007, Proceedings*, volume 4596 of *Lecture Notes in Computer Science*, pages 423–434. Springer, 2007.

[11] D. Chaum and E. van Heyst. Group signatures. In D. W. Davies, editor, *Advances in Cryptology - EUROCRYPT '91, Workshop on the Theory and Application of of Cryptographic Techniques, Brighton, UK, April 8-11, 1991, Proceedings*, volume 547 of *Lecture Notes in Computer Science*, pages 257–265. Springer, 1991.

[12] S. S. M. Chow, J. K. Liu, V. K. Wei, and T. H. Yuen. Ring signatures without random oracles. *IACR Cryptology ePrint Archive*, 2005:317, 2005.

[13] Y. Dodis, A. Kiayias, A. Nicolosi, and V. Shoup. Anonymous Identification in Ad Hoc Groups. In Cachin and Camenisch [8], pages 609–626.

[14] Q. Feng, Y. L. Sun, L. Liu, Y. Yang, and Y. Dai. Voting systems with trust mechanisms in cyberspace: Vulnerabilities and defenses. *IEEE Trans. Knowl. Data Eng.*, 22(12):1766–1780, 2010.

[15] S. Garg, C. Gentry, S. Halevi, M. Raykova, A. Sahai, and B. Waters. Candidate indistinguishability obfuscation and functional encryption for all circuits. *IACR Cryptology ePrint Archive*, 2013:451, 2013.

[16] O. Goldreich, A. Sahai, and S. P. Vadhan. Honest-verifier statistical zero-knowledge equals general statistical zero-knowledge. In J. S. Vitter, editor, *Proceedings of the Thirtieth Annual ACM Symposium on the Theory of Computing, Dallas, Texas, USA, May 23-26, 1998*, pages 399–408. ACM, 1998.

[17] D. He, J. Chen, and J. Hu. An id-based proxy signature schemes without bilinear pairings. *Annales des Télécommunications*, 66(11-12):657–662, 2011.

[18] Y. Huang, S. Zeng, and X. Liu. Privacy-preserving communication for vanets with conditionally anonymous ring signature. *I. J. Network Security*, 17(2):135–141, 2015.

[19] S. H. Islam, M. K. Khan, M. S. Obaidat, and F. T. B. Muhaya. Provably secure and anonymous password authentication protocol for roaming service in global mobility networks using extended chaotic maps. *Wireless Personal Communications*, 84(3):2013–2034, 2015.

[20] C. H. Lee, X. Deng, and H. Zhu. Design and security analysis of anonymous group identification protocols. In D. Naccache and P. Paillier, editors, *Public Key Cryptography, 5th International Workshop on Practice and Theory in Public Key Cryptosystems, PKC 2002, Paris, France, February 12-14, 2002, Proceedings*, volume 2274 of *Lecture Notes in Computer Science*, pages 188–198. Springer, 2002.

[21] D. Y. W. Liu, J. K. Liu, Y. Mu, W. Susilo, and D. S. Wong. Revocable ring signature. *J. Comput. Sci. Technol.*, 22(6):785–794, 2007.

[22] J. K. Liu, M. H. Au, W. Susilo, and J. Zhou. Online/offline ring signature scheme. In S. Qing, C. J. Mitchell, and G. Wang, editors, *Information and Communications Security, 11th International Conference, ICICS 2009, Beijing, China, December 14-17, 2009. Proceedings*, volume 5927 of *Lecture Notes in Computer Science*, pages 80–90. Springer, 2009.

[23] J. K. Liu, M. H. Au, W. Susilo, and J. Zhou. Linkable ring signature with unconditional anonymity. *IEEE Trans. Knowl. Data Eng.*, 26(1):157–165, 2014.

[24] J. K. Liu, V. K. Wei, and D. S. Wong. Linkable spontaneous anonymous group signature for ad hoc groups (extended abstract). In H. Wang, J. Pieprzyk, and V. Varadharajan, editors, *Information Security and Privacy: 9th Australasian Conference, ACISP 2004, Sydney, Australia, July 13-15, 2004. Proceedings*, volume 3108 of *Lecture Notes in Computer Science*, pages 325–335. Springer, 2004.

[25] J. K. Liu and D. S. Wong. Linkable ring signatures: Security models and new schemes. In O. Gervasi, M. L. Gavrilova, V. Kumar, A. Laganà, H. P. Lee, Y. Mun, D. Taniar, and C. J. K. Tan, editors, *Computational Science and Its Applications - ICCSA 2005, International Conference, Singapore, May 9-12, 2005, Proceedings, Part II*, volume 3481 of *Lecture Notes in Computer Science*, pages 614–623. Springer, 2005.

[26] J. K. Liu and D. S. Wong. Enhanced security models and a generic construction approach for linkable ring signature. *Int. J. Found. Comput. Sci.*, 17(6):1403–1422, 2006.

[27] B. Lynn, M. Prabhakaran, and A. Sahai. Positive results and techniques for obfuscation. In Cachin and Camenisch [8], pages 20–39.

[28] L. Nguyen. Accumulators from bilinear pairings and applications. In A. Menezes, editor, *Topics in Cryptology - CT-RSA 2005, The Cryptographers'*

Track at the RSA Conference 2005, San Francisco, CA, USA, February 14-18, 2005, Proceedings, volume 3376 of *Lecture Notes in Computer Science*, pages 275–292. Springer, 2005.

[29] R. L. Rivest, A. Shamir, and Y. Tauman. How to leak a secret. In C. Boyd, editor, *Advances in Cryptology - ASIACRYPT 2001, 7th International Conference on the Theory and Application of Cryptology and Information Security, Gold Coast, Australia, December 9-13, 2001, Proceedings*, volume 2248 of *Lecture Notes in Computer Science*, pages 552–565. Springer, 2001.

[30] A. Sahai and B. Waters. How to use indistinguishability obfuscation: deniable encryption, and more. In D. B. Shmoys, editor, *Symposium on Theory of Computing, STOC 2014, New York, NY, USA, May 31 - June 03, 2014*, pages 475–484. ACM, 2014.

[31] A. D. Santis, G. D. Crescenzo, and G. Persiano. Communication-efficient anonymous group identification. In L. Gong and M. K. Reiter, editors, *CCS '98, Proceedings of the 5th ACM Conference on Computer and Communications Security, San Francisco, CA, USA, November 3-5, 1998.*, pages 73–82. ACM, 1998.

[32] H. Shacham and B. Waters. Efficient ring signatures without random oracles. In T. Okamoto and X. Wang, editors, *Public Key Cryptography - PKC 2007, 10th International Conference on Practice and Theory in Public-Key Cryptography, Beijing, China, April 16-20, 2007, Proceedings*, volume 4450 of *Lecture Notes in Computer Science*, pages 166–180. Springer, 2007.

[33] Shamus Software Ltd. Miracl library. http://www.shamus.ie/index.php?page=home.

[34] J. Shao, X. Lin, R. Lu, and C. Zuo. A threshold anonymous authentication protocol for vanets. *Vehicular Technology, IEEE Transactions on*, PP(99):1–1, 2015.

[35] Souheil Bcheri, Erik Bjork, Daniel Deibler, Goran Hanell, Jimm Lerch, Maksym Moneta, Monika Orski, Eva Schlehahn, Welderufael Tesfay. D6.3 evaluation of the school pilot. https://abc4trust.eu/download/Deliverable%20D6.3.pdf.

[36] P. P. Tsang, M. H. Au, J. K. Liu, W. Susilo, and D. S. Wong. A suite of non-pairing id-based threshold ring signature schemes with different levels of anonymity (extended abstract). In S. Heng and K. Kurosawa, editors, *Provable Security - 4th International Conference, ProvSec 2010, Malacca, Malaysia, October 13-15, 2010. Proceedings*, volume 6402 of *Lecture Notes in Computer Science*, pages 166–183. Springer, 2010.

[37] X. Yang, W. Wu, J. K. Liu, and X. Chen. Lightweight anonymous authentication for ad hoc group: A ring signature approach. In M. H. Au and A. Miyaji, editors, *Provable Security - 9th International Conference, ProvSec 2015, Kanazawa, Japan, November 24-26, 2015, Proceedings*, volume 9451 of *Lecture Notes in Computer Science*, pages 215–226. Springer, 2015.

[38] L. Yao, C. Lin, J. Deng, F. Deng, J. Miao, K. Yim, and G. Wu. Biometrics-based data link layer anonymous authentication in vanets. In L. Barolli, I. You, F. Xhafa, F. Leu, and H. Chen, editors, *Seventh International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing, IMIS 2013, Taichung, Taiwan, July 3-5, 2013*, pages 182–187. IEEE Computer Society, 2013.

[39] T. H. Yuen, J. K. Liu, M. H. Au, W. Susilo, and J. Zhou. Efficient linkable and/or threshold ring signature without random oracles. *Comput. J.*, 56(4):407–421, 2013.

[40] F. Zhang and X. Chen. Cryptanalysis and improvement of an id-based ad-hoc anonymous identification scheme at ct-rsa 05. *Information Processing Letters*, 109(15):846 – 849, 2009.