k-Anonymous Secret Handshakes with Reusable Credentials

Shouhuai Xu
Department of Computer Science
University of Texas at San Antonio
shxu@cs.utsa.edu

Moti Yung
Department of Computer Science
Columbia University
moti@cs.columbia.edu

ABSTRACT

The problem of privacy-preserving authentication has been extensively investigated in a set of diverse system settings. However, a full-fledged such mechanism called *secret handshake*, whereby two users (e.g., CIA agents) authenticate each other in a way that no one reveals its own membership (or credential) unless the peer's legitimacy was already ensured of, remains to be elusive because simultaneity of authentication must be guaranteed even in the presence of an active adversary that may act as a handshake initiator or responder. The state-of-the-art secret handshake scheme is very efficient, but imposes on the users the following restriction: either they have to use *one-time credentials*, or they have to suffer from the privacy degradation that all the sessions involving a same user (or credential) are trivially *linkable*. In this paper, we present the *first* secret handshake schemes that achieve *unlinkability* while allowing the users to *reuse* their credentials (i.e., unlinkability is not achieved by means of one-time credentials). Specifically, we introduce the concept of k-anonymous secret handshakes where k is an adjustable parameter indicating the desired anonymity assurance. We present a detailed construction based on public key cryptosystems, and sketch another based on symmetric key cryptosystems. Both schemes are efficient, and their security analysis does not resort to any random oracle.

Categories and Subject Descriptors

C.2.4 [**Computer-Communication Networks**]: Distributed Systems; D.4.6 [**Security and Protection**]: Authentication

General Terms

Security

Keywords

privacy-preserving authentication, anonymity, unlinkability, secret handshake, credential systems, reusable credentials

CCS'04, October 25-29, 2004, Washington, DC, USA.

1. INTRODUCTION

By "privacy-preserving authentication" we mean a class of authentications that can, besides achieving the traditional goals in enforcing a certain security policy, protect the anonymity of an authenticator, a verifier, or both. The importance of privacy-preserving authentication can be justified by the extensive investigations in a set of diverse system settings: digital cash [10] whereby a merchant can verify a buyer's capability in paying a certain amount of money, group signatures [11] whereby anyone can verify a user's membership w.r.t. an existing group without even requesting the user's pseudonym, ring signatures and ring authentication [12, 27, 24] whereby an authenticater can hide itself among an ad hoc group of users, authenticated key exchange while protecting the participants' anonymity [19, 22, 8, 2], to name just a few.

However, a full-fledged privacy-preserving authentication called *secret handshake*, whereby two users (e.g., CIA agents) authenticate each other in a way that no one reveals its own membership (or credential) unless the handshake peer's (i.e., the handshake initiator or responder's) legitimacy was already ensured of, remains to be elusive because simultaneity of authentication must be guaranteed in the presence of an active adversary that may act as a handshake initiator or responder. The state-of-the-art secret handshake scheme is due to [3], which is very efficient, but imposes on the users the following restriction: either they have to use *one-time credentials*, or they have to suffer from the privacy degradation that all the sessions involving a same user (or credential) are trivially *linkable*. On one hand, adopting one-time credentials could severely limit their usefulness because an attacker can easily deplete an honest user's credentials (which is certainly true for the scheme of [3] and of a more recent work [9]), and therefore the honest users are forced to always have on-line access to the credential issuers. On the other hand, the necessity of unlinkability in anonymous transaction systems has been well-known (cf. [10, 11] and their follow-ons) mainly because anonymity could be completely compromised by correlating with information obtained through other sources.

1.1 Our Contributions

We present the *first* secret handshake schemes that achieve *unlinkability* with *reusable* credentials (i.e., unlinkability is not achieved by means of one-time credentials). Specifically, we introduce and formally define the notion of k-anonymous secret handshakes, where k is an adjustable parameter indicating the desired anonymity assurance. Intuitively, k-

anonymity means that, in the worst case, an adversary can only infer that a participant is one out of certain k users.

We present a detailed secret handshake scheme based on public key cryptosystems. The scheme is presented via two steps (for clarification): the first step is a basic scheme that incurs $O(1)$ computation (in terms of modular exponentiations) and $O(w)$ communication (in terms of ciphertexts), where w is a parameter that will be specified later; the second step is an improvement whereby we achieve a scheme that only incurs $O(1)$ computation and $O(1)$ communication. As an specific instantiation, this scheme can be based on a standard public key infrastructure (PKI), which means that this scheme may be seen as a natural *value-added* application of a standard PKI to provide *anonymity* without incurring any other cost in initializing the systems. We also sketch a secret handshake scheme based on symmetric key cryptosystems, which naturally incurs $O(1)$ communication without requiring any time-consuming computation of modular exponentiations. Security analysis of all the schemes does not resort to any random oracle, which means that if the underlying building blocks are secure in the standard model, so is the resulting secret handshake scheme.

On integrating secret handshakes into key exchange protocols. In the past years, the issue of protecting participants' anonymity from any unauthorized party has been raised in the IETF Internet Key Exchange scheme (IKE) [19, 22, 8] or its alternative [2]. For example, the schemes proposed in [22, 2] target at protecting the identities of the communication peers from an eavesdropper, and can be adapted to protect the identity of an initiator *or* a responder from an active attacker. It seems debatable whether it is possible for a protocol to protect the identities of both the initiator *and* the responder from an active attacker. The argument for the impossibility is that one of the participants must always "go first". However, we believe that secret handshakes can be integrated into key exchange protocols to protect both the initiator *and* the responder against an active attacker, at least in system settings where it is practical to tolerate the following *unfairness* between a pair of *legitimate* users: Alice learns that Bob is legitimate without guaranteeing that Bob also learns Alice's legitimacy.

1.2 Related Work

We only focus on some closely related works, and refer the reader to [3] for discussions on some loosely related ones (e.g., private authentication [1]).

Secret Handshake and Authenticated Key Exchange. The pioneering secret handshake scheme [3] is based on the protocol of [28], which indeed targets at the key exchange problem. As a matter of fact, a secret handshake can be appropriately turned into an authenticated key exchange, but an authenticated key exchange does not necessarily imply a secret handshake (e.g., the classic two-party Diffie-Hellman key agreement scheme [14] does not necessarily solve the problem of secret handshake; cf. [3] for attacks).

Very recently, [9] presented some interesting new secret handshake schemes which, however, inherited the limitations of [3].

Anonymous Credentials, Ring Signatures and Ring Authentication. There have been various anonymous credential schemes that aimed at authenticating an anonymous user. For example, group signatures [11] allow a verifier to determine whether a signature was generated by someone belonging to a certain group of users. Ring signatures [27], or their interactive variants [25, 24], achieve the same functionality without involving any complex initialization procedure. However, all these schemes only protect one participant's (i.e., the authenticator's) anonymity. This is true even if one combines them with the so-called designated verifier signatures [21], because the latter necessarily expose that the signer and the intended verifier are communicating with each other.

Automated Trust Negotiation. The concept of oblivious signature-based envelopes (OSBEs) [23] was introduced to solve a problem encountered in the context of automated trust negotiation. Informally, an OSBE scheme enables a sender to send an envelope (encrypted message) to a receiver, and has the following properties: the receiver can open the envelope if and only if it has a third party's (e.g., certification authority) signature on an agreed-upon message. An OSBE scheme is oblivious if at the end of the protocol the sender cannot tell whether the receiver has the intended signature or not. The notion of Hidden Credentials [20] was introduced to fulfill a similar but somewhat different problem also in the context of automated trust negotiation.

Both notions are related to secret handshakes. However, secret handshakes differ from oblivious signature-based envelopes in that both Alice and Bob need to know whether they belong to the same group. Hidden credentials differ from secret handshakes in that the latter requires Alice and Bob to mutually authenticate each other using credentials from the same issuer, whereas the former allows Alice to send Bob a message dependent only on Bob's credentials – Alice need not even have any credentials of her own.

k-anonymity. This notion was introduced to protect privacy in the context of database systems such that released information limits what can be revealed about properties of the entities that are to be protected [29]. Recently, it was adopted to help construct efficient anonymous messages transmission schemes [30]. In parallel to [30], we adopt the concept of k-anonymity to investigate secret handshake schemes.

This paper is organized as follows. In Section 2, we briefly review the utilized cryptographic tools. In Section 3 we present a system model and definition of k-anonymous secret handshake schemes. In Section 4 we present a basic k-anonymous secret handshake scheme based on public key cryptosystems, which is improved in Section 5 with a general method that can significantly reduce the communication complexity. In Section 6 we sketch an k-anonymous secret scheme based on symmetric key cryptosystems. We conclude the paper in Section 7. Due to space limitation, many details are deferred to the full version of this paper [31].

2. CRYPTOGRAPHIC PRELIMINARIES

A function $\epsilon : \mathbb{N} \rightarrow \mathbb{R}^+$ is negligible if for any c there exists κ_c such that $\forall \kappa > \kappa_c$ we have $\epsilon(\kappa) < 1/\kappa^c$. Below we briefly review the cryptographic primitives that will be utilized in our schemes.

Public Key Cryptosystems. A public key cryptosystem consists of three polynomial-time algorithms: GEN, ENC, and DEC. The probabilistic key generation algorithm GEN takes as input 1^κ and outputs a key pair (pk, sk). The prob-

abilistic encryption algorithm ENC takes as input a public key pk and a message m, and outputs a ciphertext c. The decryption algorithm DEC takes as input a ciphertext c and a private key sk, and returns a message m or $\perp$ (meaning that c is not valid). We will use public key cryptosystems that are secure against adaptive chosen ciphertext attack (i.e., IND-CCA2) [26, 15]. Basically, this means that an attacker $\mathcal{A}$, who is given pk and allowed to query the decryption oracle, generates two messages, X_0 and X_1, of equal length, and sends them to a challenge oracle. The oracle chooses $b \in_R \{0,1\}$ and returns $Y = ENC(pk, X_b)$. The task of the adversary $\mathcal{A}$, which is still allowed to query the decryption oracle by feeding any ciphertexts other than Y, is to output b'. We say $\mathcal{A}$ succeeds if $b = b'$, for which the probability should be negligibly (in κ) more than a random guess. Practical schemes are available in [5, 16, 13].

Key-Private Public Key Cryptosystems. While standard public key cryptosystems (as reviewed above) emphasize on the *data-privacy* properties such as "indistinguishability of the encrypted-content under adaptive chosen-ciphertext attack", the notion of *key-privacy* [4] ensures that a ciphertext does not expose the corresponding public key – a property not captured by the standard definition of public key cryptosystems. More specifically, let us briefly review the notion of key-privacy under adaptive chosen-ciphertext attack. Consider an adversary $\mathcal{A}$ that takes as input two public keys pk_0 and pk_1 (corresponding to the private keys sk_0 and sk_1, respectively) and outputs a message X together with some state information s. Then, $\mathcal{A}$ is given a challenge ciphertext $Y = ENC(pk_b, X)$, where $b \in_R \{0,1\}$. The task of the adversary $\mathcal{A}$, which is still allowed to query the two decryption oracles by feeding any ciphertexts other than Y, is to output b'. We say $\mathcal{A}$ breaks the *key-privacy* property if $b = b'$, for which the probability should be negligibly (in κ, namely the security parameter of the public key cryptosystems) more than a random guess. In [4], it is shown that the Cramer-Shoup cryptosystem [13] implements key-privacy under an adaptive chosen-ciphertext attack in the standard model, and RSA-OAEP [5] implements key-privacy under an adaptive chosen-ciphertext attack in the random oracle model (with a slightly different operation from the standard RSA-OAEP, namely that one needs to repeat the standard encryption procedure until the ciphertext falls into a certain "safe" range). Note that, of course, the key-private public key cryptosystems [13, 5] naturally preserve their data-privacy assurance.

Pseudorandom Functions. A pseudorandom function (PRF) family $\{f_k\}$ parameterized by a secret value k has the following property [17]: An adversary $\mathcal{A}$ cannot distinguish f_k, where $k \in_R \{0,1\}^\kappa$, from a perfect random function (with the same domain and range) with a non-negligible (in κ) probability.

Digital Signature Schemes. A digital signature scheme consists of three polynomial-time algorithms: GEN, $SIGN$, and VER. The probabilistic key generation algorithm GEN takes as input 1^κ and outputs a key pair (pk, sk). The signing algorithm $SIGN$ takes as input a message m and a private key sk, and outputs a signature σ. The verification algorithm VER takes as input a message m, a public key pk, and a candidate signature σ, returns "accept" if σ is a valid signature and "reject" otherwise. A signature scheme should be existentially unforgeable under an adaptive chosen-message attack [18]. Basically, this means that an adversary $\mathcal{A}$ cannot output a valid signature on a message that was not signed by the signing oracle with a non-negligible probability in κ.

3. k-Anonymous Secret Handshakes: Model and Definition

In this section, we specify the system model in which participants conduct secret handshakes. We then elaborate on the security of k-anonymous secret handshakes.

3.1 System Model and Definition

Let U be the set of all possible pseudonyms. Suppose there are l groups $\mathbf{G} = \{G_1, \cdots, G_l\}$ where each group[1] $G \in \mathbf{G}$ is a set of users (i.e., $G \subset \mathsf{U}$) and managed by an authority CA. In other words, a CA is responsible for admitting users into a group and revoking their memberships when the need arises – just like a certificate authority in a standard public key infrastructure (PKI). All the participants are modeled as probabilistic polynomial-time Turing machines. To simplify the presentation, we assume that each user is a member of exactly one group, while all the results can be naturally generalized to the case that users are allowed to join multiple groups.

A secret handshake scheme SHS consists of the following algorithms:

$\mathsf{SHS.CreateGroup}$ This algorithm is executed by an authority, CA, to establish a group G. It takes as input appropriate security parameters, and outputs a cryptographic context specific to this group. In particular, the context may include a data structure called certificate/membership revocation list, CRL, which is originally empty. The cryptographic context is made public.

$\mathsf{SHS.AdmitMember}$ This algorithm is run by a CA to admit a user to become a member of the group that is under the CA's jurisdiction. The CA admits members according to a certain policy, which is orthogonal to the focus of this paper. For example, the CA may interact with the user to verify its real identity and its ownership of the private key corresponding to a claimed public key. After executing this algorithm, the group state information (e.g., the list of the members' certificates) is appropriately updated, and the member holds some secret(s) as well as a membership certificate. We will identify an anonymous user through its pseudonym $U \in \mathsf{U}$, which can be included in its certificate.

$\mathsf{SHS.Handshake}(U, V)$ This protocol is executed by a pair of anonymous users, where $U, V \in \mathsf{U}$ are just placeholders, U plays the role of an initiator, and V plays the role of a responder. (It is even true that U does not know V's pseudonym before a successful handshake, and vice versa.) The input to this protocol includes the anonymous users' secrets, and possibly some public information corresponding to the current state of the system. The output of this protocol, upon completion, ensures that U discovers $V \in G$ if and only if

[1]The term "group" in this paper always refers to a set of users unless explicitly stated otherwise.

V discovers $U \in G$. In other words, it returns "1" if the handshake succeeds (i.e., both U and V belong to G), and "0" otherwise.

SHS.RemoveUser This algorithm is executed by an authority CA. It takes as input its current CRL and U's certificate/pseudonym. The output includes an updated CRL which includes the newly revoked certificate U, and perhaps the updated list of the members' certificates/pseudonyms.

3.2 Security

The security properties of k-anonymous secret handshake schemes include the following: `correctness`, `resistance to impersonation attacks`, and k-`anonymity` which is specified through three properties: k-`resistance to detection attacks`, k-`unlinkability`, and k-`indistinguishability to eavesdroppers`. Consider a probabilistic polynomial-time adversary $\mathcal{A}$ that may have access to the following oracles:

- $\mathcal{O}_{CG}(\cdot)$: This activates a new CA to create a new group via algorithm SHS.CreateGroup. The identity, CA, may be given by $\mathcal{A}$ as the input. We assume that a CA is not under $\mathcal{A}$'s control before the new group is established. However, the CA may be corrupt immediately after its establishment (i.e., before admitting any user into the group).

- $\mathcal{O}_{AM}(\cdot,\cdot)$: The input includes the identity of a CA and, optionally, the identity U of a user that is under $\mathcal{A}$'s control. In the case of $\mathcal{O}_{AM}(CA, U)$, the CA may admit the corrupt user U by executing algorithm SHS.AdmitMember; in the case of $\mathcal{O}_{AM}(CA)$, the CA executes algorithm SHS.AdmitMember to admit an honest user and assigning it with a unique pseudonym U.

- $\mathcal{O}_{HS}(\cdot,\cdot)$: The oracle will activate SHS.Handshake between U and V, where none, one, or both, of them may have been corrupt. A corrupt user will execute what the adversary is pleased of.

- $\mathcal{O}_{RU}(\cdot,\cdot)$: The input includes the identity of a CA and a pseudonym U. The oracle activates algorithm SHS.RemoveUser to insert U into the corresponding CRL, and the system state information is appropriately updated.

- $\mathcal{O}_{Corrupt}(\cdot,\cdot)$: The input includes the identity of a CA, and possibly a pseudonym U issued by the CA. In the case of $\mathcal{O}_{Corrupt}(CA, U)$, the oracle returns U's current internal state information (including all secrets) to $\mathcal{A}$; in the case of $\mathcal{O}_{Corrupt}(CA)$, the oracle returns CA's current internal state information (including all secrets) to $\mathcal{A}$. Once the CA or U is corrupt, it will execute what $\mathcal{A}$ is pleased of, until such a corruption is detected by some outside mechanism (e.g., intrusion detection systems). When the corruption of a user U is detected, it is excluded from the group via algorithm SHS.RemoveUser; when the corruption of an authority CA is detected, the corresponding group is simply excluded from the system.

Now we are ready to present the definitions of the security properties.

`Correctness`. If two users U and V belong to the same group, then SHS.Handshake(U, V) always returns "1"; otherwise, it returns "0".

`Resistance to impersonation attacks`. Intuitively, this property captures that $\mathcal{A}$, who does not belong to, or does not corrupt a legitimate member of, a group G managed by an incorrupt CA, has only a negligible probability in convincing an honest user $U \in G$ that $\mathcal{A}$ is also a member of G. Formally, consider the experiment specified in Fig. 1. Let $\mathsf{AdvRIA}_{\mathsf{SHS}}(\mathcal{A}) = \Pr[\mathsf{RIA}_{\mathsf{SHS}}(\mathcal{A})$ returns "1"$]$, which is the probability that the experiment $\mathsf{RIA}_{\mathsf{SHS}}(\mathcal{A})$ returns "1", where probability is taken over all the tossed coins (including those utilized in generating the cryptosystem instances, those utilized by the honest users, and those utilized by the adversary). A secret handshake scheme SHS is "`resistant to impersonation attacks`" if for $\forall\mathcal{A}$, $\mathsf{AdvRIA}_{\mathsf{SHS}}(\mathcal{A})$ is negligible in the security parameter of SHS.

k-`resistance to detection attacks`. Suppose there are β groups such that none of their members or CAs is corrupt. Ideally, "`resistance to detection attacks`" means that no adversary $\mathcal{A}$, who does not belong to any of the β groups, can successfully guess the membership of an anonymous (and honest) handshaking peer U with a non-negligible advantage over $1/\beta$. In this paper, we pursue a weaker, but practical and useful (see Section 4.4 for discussions), notion we call "k-`resistance to detection attacks`", where $1 \leq k \leq \beta$ is a parameter indicating the desired anonymity assurance. Intuitively, it means that no adversary $\mathcal{A}$, who does not belong to any of the k groups, can successfully guess the membership of an anonymous (and honest) handshake peer U with a non-negligible advantage over $1/k$. Formally, consider the experiment specified in Fig. 2. Let $\mathsf{AdvRDA}_{\mathsf{SHS}}(\mathcal{A}) = |\Pr[\mathsf{RDA}_{\mathsf{SHS}}(\mathcal{A})$ returns "1"$] - 1/k|$, which is the advantage that $\mathcal{A}$ successfully guesses the membership of the anonymous handshake peer X. The probability is taken over all the tossed coins. A scheme SHS is "k-`resistant to detection attacks`" if it holds that for $\forall\mathcal{A}$, $\mathsf{AdvRDA}_{\mathsf{SHS}}(\mathcal{A})$ is negligible in the security parameter of SHS.

k-`unlinkability`. Suppose there are β groups such that none of their members or CAs is corrupt. Ideally, the property of "`unlinkability`" means that no adversary $\mathcal{A}$, who does not belong to any of the β groups, can successfully associate two sessions involving a same honest user with a non-negligible advantage over $1/\beta$. In this paper, we pursue a weaker, but practical and useful notion we call "k-`unlinkability`", where $1 \leq k \leq \beta$ is a parameter indicating the desired anonymity assurance. Intuitively, it means that no adversary $\mathcal{A}$, who does not belong to any of the k groups, can successfully associate two sessions, τ_1 and τ_2, involving a same honest user with a non-negligible advantage over $1/k$. Formally, consider the experiment specified in Fig. 3. Let $\mathsf{AdvUnlink}_{\mathsf{SHS}}(\mathcal{A}) = |\Pr[\mathsf{Unlink}_{\mathsf{SHS}}(\mathcal{A})$ returns "1"$] - 1/k|$, which is the advantage that $\mathcal{A}$ successfully associates two handshake sessions to a same honest user. The probability is taken over all the tossed coins. A scheme SHS is "k-`unlinkable`" if it holds that for $\forall\mathcal{A}$, $\mathsf{AdvUnlink}_{\mathsf{SHS}}(\mathcal{A})$ is negligible in the security parameter of SHS.

k-`indistinguishability to eavesdroppers`. Consider an adversary $\mathcal{A}$ who corrupts some users, interacts with some others, and observes a session of SHS.Handshake for a pair

Experiment $\mathsf{RIA}_{\mathsf{SHS}}(\mathcal{A})$:
$(CA, U, b) \leftarrow \mathcal{A}^{\mathcal{O}_{CG}(\cdot),\mathcal{O}_{AM}(\cdot,\cdot),\mathcal{O}_{HS}(\cdot,\cdot),\mathcal{O}_{RU}(\cdot,\cdot),\mathcal{O}_{Corrupt}(\cdot,\cdot)}(\cdot)$
Return "1" if the following hold and "0" otherwise:
(1) U belongs to the group managed by CA
(2) There is no $\mathcal{O}_{Corrupt}(CA)$ query
(3) There is no $\mathcal{O}_{RU}(CA, U)$ query
(4) If there is an $\mathcal{O}_{AM}(CA, X \in \mathsf{U})$ query, then there is also an $\mathcal{O}_{RU}(CA, X)$ query
(5) If there is an $\mathcal{O}_{Corrupt}(CA, X \in \mathsf{U})$ query, then there is also an $\mathcal{O}_{RU}(CA, X)$ query
(6) $\mathsf{SHS.Handshake}(\mathcal{A}, U)$ returns "1" if $b = 0$, or $\mathsf{SHS.Handshake}(U, \mathcal{A})$ returns "1" if $b = 1$

Figure 1: An experiment specifying "`resistance to impersonation attacks`"

Experiment $\mathsf{RDA}_{\mathsf{SHS}}(\mathcal{A})$:
$(\mathsf{stateInfo}, b) \leftarrow \mathcal{A}^{\mathcal{O}_{CG}(\cdot),\mathcal{O}_{AM}(\cdot,\cdot),\mathcal{O}_{HS}(\cdot,\cdot),\mathcal{O}_{RU}(\cdot,\cdot),\mathcal{O}_{Corrupt}(\cdot,\cdot)}(\cdot)$
If $b = 0$ execute $\mathsf{SHS.Handshake}(\mathcal{A}, X)$ else execute $\mathsf{SHS.Handshake}(X, \mathcal{A})$
where X is a placeholder for an anonymous (and honest) user
Let CA_i be the authority of group G_i, to which X belongs
Return "1" if the following hold and "0" otherwise:
(1) There is no $\mathcal{O}_{RU}(CA_i, X)$ query
(2) There are at least k groups (including G_i) such that for each group G managed by CA:
(2.1) There is no $\mathcal{O}_{Corrupt}(CA)$ query
(2.2) If there is an $\mathcal{O}_{AM}(CA, Y)$ query, then there is also an $\mathcal{O}_{RU}(CA, Y)$ query
(2.3) If there is an $\mathcal{O}_{Corrupt}(CA, Y)$ query, then there is also an $\mathcal{O}_{RU}(CA, Y)$ query
(3) $\mathcal{A}$ outputs i

Figure 2: An experiment specifying "k-`resistance to detection attacks`"

of incorrupt users, U and V. Suppose there are at least k groups such that none of their members or CAs is corrupt. Intuitively, this property means that $\mathcal{A}$ should not be able to learn from this handshake session anything that it did not already know (including whether U and V belong to the same group). In order to capture this property, consider a simulated transcript of a handshake session, which is obtained by substituting all the strings derived from cryptographic secrets with random strings of appropriate lengths (i.e., no cryptographic secrets or memberships are involved). Yet, such a substitution cannot be detected by $\mathcal{A}$. Formally, consider the experiment specified in Fig. 4. Let $\mathsf{AdvINDeav}_{\mathsf{SHS}}(\mathcal{A}) = |\Pr[\mathsf{INDeav}_{\mathsf{SHS}}(\mathcal{A})$ returns "1"$|b = 0] - \Pr[\mathsf{INDeav}_{\mathsf{SHS}}(\mathcal{A})$ returns "1"$|b = 1]|$, which is the advantage that the adversary $\mathcal{A}$ successfully distinguishes a real transcript from a simulated one. The probability is taken over all the tossed coins. A scheme SHS is "k-`indistinguishable to eavesdroppers`" if it holds that for $\forall\mathcal{A}$, $\mathsf{AdvINDeav}_{\mathsf{SHS}}(\mathcal{A})$ is negligible in the security parameter of SHS.

3.3 Discussion

Our definition does not specify a $\mathsf{SHS.TraceUser}$ algorithm [3]; we believe that such an escrow capability is orthogonal to secret handshakes. We remark that our schemes actually achieve a stronger version of the above definitions of k-`resistance to detection attack`, k-`unlinkability`, and k-`indistinguishability to eavesdroppers`, because they only require at least k (pairs of) drafted users, rather than *all* the users in the k drafted groups, to be incorrupt. Nevertheless, there could be a scheme that achieves the weaker, but not the stronger, version of these properties.

In the definition of k-`unlinkability`, we actually capture the worst case scenario that even if an adversary has somehow compromised the anonymity in one session (e.g., through some occasional attack against the underlying system components such as anonymous communication), it is still unable to associate another session conducted by the same honest user (of course, under the premise that there is no attack against the underlying system components with respect to this session).

4. k-Anonymous Secret Handshake based on Public Key Cryptosystems

This section is organized as follows. In Section 4.1 we start with the intuition underlying our basic scheme and then present the scheme itself. In Section 4.2 and 4.3 we analyze its efficiency and security. In Section 4.4 we discuss some extensions and practical issues.

Notations. Denote by $\mathbf{G} = \{G_1, \cdots, G_l\}$ a set of groups, where G_z $(1 \leq z \leq l)$ is a set of users whose public keys are certified by an authority CA_z (via certificates) using a secure signature scheme. Without loss of generality, we assume that both the groups $G_1, \cdots, G_l$ and the users in a group $G \in \mathbf{G}$ are in an appropriate order (e.g., alphabetic), based on which a partition can be naturally defined. If X is a user, let $Group(X)$ denote the group to which X belongs, and $Cert_X$ denote X's public key certificate. If $Cert$ is a certificate, let $CA(Cert)$ denote the authority which issues it. For example, a user $X \in G_i$ owning a public key pk_X will be issued a certificate $Cert_X$ by authority CA_i. Moreover, $Group(X)$ returns G_i and $CA(Cert_X)$ returns CA_i.

Let $\kappa_0, \kappa_1, \kappa_2$ be additional security parameters, and q be a prime of length κ_0. Suppose $f_0 : \{0,1\}^* \rightarrow \{0,1\}^{\kappa_0}$ and $f_1 : \{0,1\}^* \rightarrow \mathbb{Z}_q^*$ are secure pseudorandom functions.

Experiment $\mathsf{Unlink}_{\mathsf{SHS}}(\mathcal{A})$:
$(\tau_1, \tau_2) \leftarrow \mathcal{A}^{\mathcal{O}_{CG}(\cdot),\mathcal{O}_{AM}(\cdot,\cdot),\mathcal{O}_{HS}(\cdot,\cdot),\mathcal{O}_{RU}(\cdot,\cdot),\mathcal{O}_{Corrupt}(\cdot,\cdot)}(\cdot)$
Return "1" if the following hold and "0" otherwise:
(1) τ_1 and τ_2 involve a same user $U \in G_i$, where G_i is managed by CA_i
(2) There is no $\mathcal{O}_{RU}(CA_i, U)$ query
(3) There are at least k groups (including G_i) such that for each group G managed by CA:
(3.1) There is no $\mathcal{O}_{Corrupt}(CA)$ query
(3.2) If there is an $\mathcal{O}_{AM}(CA, Y)$ query, then there is also an $\mathcal{O}_{RU}(CA, Y)$ query
(3.3) If there is an $\mathcal{O}_{Corrupt}(CA, Y)$ query, then there is also an $\mathcal{O}_{RU}(CA, Y)$ query

Figure 3: An experiment specifying "k-unlinkability"

Experiment $\mathsf{INDeav}_{\mathsf{SHS}}(\mathcal{A})$:
Let $R \notin \mathsf{U}$ be an algorithm (i.e., simulator) having no access to any cryptographic secrets
$\mathsf{stateInfo} \leftarrow \mathcal{A}^{\mathcal{O}_{CG}(\cdot),\mathcal{O}_{AM}(\cdot,\cdot),\mathcal{O}_{HS}(\cdot,\cdot),\mathcal{O}_{RU}(\cdot,\cdot),\mathcal{O}_{Corrupt}(\cdot,\cdot)}(\cdot)$
Flip a random coin b
If $b = 0$ then give $\mathcal{A}$ a $\mathsf{transcript}$ of $\mathsf{SHS.Handshake}(U, V)$
else give $\mathcal{A}$ a simulated $\mathsf{transcript}$ of $\mathsf{SHS.Handshake}(R, R)$
$b' \leftarrow \mathcal{A}^{\mathcal{O}_{CG}(\cdot),\mathcal{O}_{AM}(\cdot,\cdot),\mathcal{O}_{HS}(\cdot,\cdot),\mathcal{O}_{RU}(\cdot,\cdot),\mathcal{O}_{Corrupt}(\cdot,\cdot)}(\mathsf{stateinfo}, \mathsf{transcript})$
Suppose $U \in G_i$, where G_i is managed by CA_i
Suppose $V \in G_i$, where G_j is managed by CA_j
Return "1" if the following hold and "0" otherwise:
(1) $b' = b$
(2) There is no $\mathcal{O}_{RU}(CA_i, U)$ or $\mathcal{O}_{RU}(CA_j, V)$ query
(3) There are at least k groups (including G_i and G_j) such that for each group G managed by CA:
(3.1) There is no $\mathcal{O}_{Corrupt}(CA)$ query
(3.2) If there is an $\mathcal{O}_{AM}(CA, Y)$ query, then there is also an $\mathcal{O}_{RU}(CA, Y)$ query
(3.3) If there is an $\mathcal{O}_{Corrupt}(CA, Y)$ query, then there is also an $\mathcal{O}_{RU}(CA, Y)$ query

Figure 4: An experiment specifying "k-indistinguishability to eavesdroppers"

4.1 The Scheme

The basic idea is to let each handshake peer (i.e., handshake initiator or responder) "draft" users on-the-fly from a certain number of groups to form a "crowd". Then, each handshake peer encrypts some random strings using the public keys of the drafted users, respectively. In order to prevent certain attacks and make the scheme efficient, we utilize two twists:

1. The drafting algorithms must not leak information regarding the drafting user's membership. This is trivial if *all* the users are honest, which is however unrealistic. The twist deals with this situation is the following: Let the drafting user correspond to a point, which then specify a line such that all the drafted users correspond to some points on the line.

2. A simple-minded scheme would let a user decrypt all the ciphertexts presented by a handshake peer. The twist avoiding this search is to let the ciphertexts correspond to different plaintexts, so that each user only needs to decrypt the ciphertext generated using its own public key.

Since all the algorithms other than $\mathsf{SHS.Handshake}(U, V)$ are just like in a standard public key infrastructure (e.g., SHS.CreateGroup corresponding to the establishment of a CA, SHS.AdmitMember corresponding to the issuing of a new public key certificate, SHS.RemoveUser corresponding to the revocation of a public key certificate), we only specify $\mathsf{SHS.Handshake}(U, V)$ in Fig. 5, where the parameter w will be determined in Section 4.3. Without loss of generality, we assume $w|l$. The protocol $\mathsf{SHS.Handshake}(U, V)$ calls four subroutines specified in Fig. 6: rSelect for selecting w groups from $\mathbf{G} = \{G_1, \cdots, G_l\}$, mSelect for selecting w members from the w groups, rSelectVer and mSelectVer for verifying that the selections are appropriately done.

4.2 Efficiency Analysis

The handshake protocol itself has 2 round-trips excluding the exchange of the plaintext nonces n_U and n_V, which could have been done before the handshake protocol is executed (e.g., when they exchange their cipher suits). Each user needs to execute w encryptions but only *a single* decryption. If the public key cryptosystems are based on (say) low public-exponent RSA, then the scheme is computationally efficient. The communication complexity is $O(w)$ ciphertexts. We remark that the complexity inccured in running the subroutines can be minimized by letting the users cache the valid public key certificates.

4.3 Security Analysis

We show that our scheme achieves the security goals specified in Section 3. To simplify the discussion, suppose each user has the same probability p of being corrupt. A pair of users (U, V) are incorrupt, if none of them is corrupt.

1. Let U select and send $n_U \in_R \{0,1\}^{\kappa_2}$ to V, and V select and send $n_V \in_R \{0,1\}^{\kappa_2}$ to U. This is for them to exchange their nonces, and typically could have been done before the handshake protocol is activated.

2. U executes as follows:

   (a) Run $\mathsf{rSelect}(\mathbf{G}, U, w, n_U, n_V)$, which returns $(\mathbf{G}^* = \{G_{z_{s_z}}\}_{z=0}^{w-1}, \theta_1)$, where $Group(U) = G_{i_{s_i}}$.

   (b) Run $\mathsf{mSelect}(\mathbf{G}^*, U, w, n_U, n_V)$, which returns $(\mathbf{X}^*, \theta_2)$. In order to clarify presentation, parse $\mathbf{X}^*$ also as $\{U_{z_{s_z,\lambda_z}}\}_{z=0}^{w-1}$, where $U_{z_{s_z,\lambda_z}} \in G_{z_{s_z}}$ and $0 \le \lambda_z \le |G_{z_{s_z}}| - 1$.

   (c) Send $(\mathbf{G}^*, \mathbf{X}^*, \theta_1, \theta_2, \{Cert_{U_{z_{s_z,\lambda_z}}}\}_{z=0}^{w-1})$ to V.

3. Upon receiving $(\mathbf{G}^*, \mathbf{X}^*, \theta_1, \theta_2, \Delta = \{Cert_{U_{z_{s_z,\lambda_z}}}\}_{z=0}^{w-1})$, V checks if the selected groups (consistently indicated by $\mathbf{G}^*$, $\mathbf{X}^*$, and the certificates) are distinct, if $CA(Cert_V) = CA(Cert_{U_{j_{s_j,\lambda_j}}})$ for some $0 \le j \le w-1$, if the groups are appropriately selected by running $\mathsf{rSelectVer}(\mathbf{G}, \mathbf{G}^*, w, n_U, n_V, \theta_1)$, if $Cert_V \notin \Delta$, and if the members are appropriately selected by running $\mathsf{mSelectVer}(\mathbf{G}^*, \mathbf{X}^*, w, n_U, n_V, \theta_2)$. If any condition is not satisfied, it simply quits; otherwise, it executes as follows:

   (a) Set $\mathbf{G}' = \{G'_{z_{s_z}}\}_{z=0}^{w-1}$, where $G'_{z_{s_z}} = G_{z_{s_z}} - \{U_{z_{s_z,\lambda_z}}\}$.

   (b) Run $\mathsf{mSelect}(\mathbf{G}', V, w, n_U, n_V)$, which returns $(\mathbf{X}', \theta'_2)$. In order to clarify presentation, parse $\mathbf{X}'$ also as $\{V_{z_{s'_z,\lambda'_z}}\}_{z=0}^{w-1}$, where $V_{z_{s'_z,\lambda'_z}} \in G'_{z_{s_z}}$ and $0 \le \lambda'_z \le |G'_{z_{s_z}}| - 1$.

   (c) For $z = 0$ to $w-1$, choose $\delta_z \in_R \{0,1\}^{\kappa_1}$ and encrypt it using $pk_{U_{z_{s_z,\lambda_z}}}$ (certified via $Cert_{U_{z_{s_z,\lambda_z}}}$) to obtain $\alpha_{U_{z_{s_z,\lambda_z}}} = ENC(pk_{U_{z_{s_z,\lambda_z}}}, \delta_z)$.

   (d) Send $(\mathbf{X}', \theta'_2, \{Cert_{V_{z_{s'_z,\lambda'_z}}}\}_{z=0}^{w-1}, \{\alpha_{U_{z_{s_z,\lambda_z}}}\}_{z=0}^{w-1})$ to U.

4. Upon receiving $(\mathbf{X}', \theta'_2, \{Cert_{V_{z_{s'_z,\lambda'_z}}}\}_{z=0}^{w-1}, \{\alpha_{U_{z_{s_z,\lambda_z}}}\}_{z=0}^{w-1})$, U checks if $CA(Cert_{V_{z_{s'_z,\lambda'_z}}}) = CA(Cert_{U_{z_{s_z,\lambda_z}}})$ for $0 \le z \le w-1$, and if the members are appropriately selected by running $\mathsf{mSelectVer}(\mathbf{G}', \mathbf{X}', w, n_U, n_V, \theta'_2)$, where $\mathbf{G}' = \{G'_{z_{s_z}}\}_{z=0}^{w-1}$ and $G'_{z_{s_z}} = G_{z_{s_z}} - \{U_{z_{s_z,\lambda_z}}\}$. If not, it quits; otherwise, it executes as follows:

   (a) Decrypt $\alpha_{U_{i_{s_i,\lambda_i}}}$ to get δ_i because $Group(U) = G_{i_{s_i}}$.

   (b) For $z = 0$ to $w-1$, choose $\gamma_z \in_R \{0,1\}^{\kappa_1}$ and encrypt it using $pk_{V_{z_{s'_z,\lambda'_z}}}$ (certified via $Cert_{V_{z_{s'_z,\lambda'_z}}}$) to obtain $\alpha_{V_{z_{s'_z,\lambda'_z}}} = ENC(pk_{V_{z_{s'_z,\lambda'_z}}}, \gamma_z)$.

   (c) Set $\sigma_0 = f_0(\gamma_i, \delta_i, 0)$ because $Group(U) = G_{i_{s_i}}$, and send $(\{\alpha_{V_{z_{s'_z,\lambda'_z}}}\}_{z=0}^{w-1}, \sigma_0)$ to V.

5. Upon receiving $(\{\alpha_{V_{z_{s'_z,\lambda'_z}}}\}_{z=0}^{w-1}, \sigma_0)$, V decrypts $\alpha_{V_{j_{s'_j,\lambda'_j}}}$ to get γ_j because $Group(V) = G_{j_{s_j}}$, and checks if $\sigma_0 = f_0(\gamma_j, \delta_j, 0)$. If it holds, V returns $\sigma_1 = f_0(\gamma_j, \delta_j, 1)$; otherwise, V chooses $r \in_R \{0,1\}^{2\kappa_1}$ and returns $\sigma_1 = f_0(r, 1)$.

6. Upon receiving σ_1, U checks if $\sigma_1 = f_0(\gamma_i, \delta_i, 1)$. If so, $Group(V) = G(U)$; otherwise, $Group(V) \neq G(U)$.

Figure 5: A secret handshake protocol based on public key cryptosystems

LEMMA 4.1. *Suppose each user has the same probability p of being corrupt. If $w = 1 + \frac{2(k-1)}{(1-p)^2}$, then $\mathsf{SHS.Handshake}(U, V)$ involves k pairs of incorrupt users (including the handshaking peers) with a high probability.*

PROOF. For any pair of $(U \in G_i, V \in G_i)$, denote by X the random variable indicating the number of incorrupt pairs of members in the drafted $w-1$ pairs, then we have $E[X] = (w-1)(1-p)^2$. If we set $w = 1 + \frac{2(k-1)}{(1-p)^2}$, then $E[X] = 2(k-1)$. This means that the expected number of incorrupt pairs of members (out of the $w-1$ pairs) is $2(k-1)$. A multiplicative Chernoff bound shows that for independently selected $w-1$ pairs of members, $\Pr[X < (k-1)] \le e^{-(k-1)/4}$. That is, the probability that the crowds have less than k incorrupt pairs of users (including the handshaking peers) is no greater than $e^{-(k-1)/4}$. □

LEMMA 4.2. *If $w = 1 + \frac{2(k-1)}{(1-p)^2}$, then with a high probability there are at least k incorrupt users among the w users output by* $\mathsf{mSelect}$.

PROOF. Let X be the random variable indicating the number of incorrupt users in the $w-1$ drafted users. Then we have $E[X] = (w-1)(1-p) \ge (w-1)(1-p)^2$. Lemma 4.1 immediately implies the conclusion. □

THEOREM 1. *Assume that the public key cryptosystems, the signature schemes, and the pseudorandom functions are secure (as specified in Section 2). Then the above scheme is a secure k-anonymous secret handshake scheme.*

PROOF. (sketch) It is easy to check `correctness`: if the handshake peers belong to the same group, the handshake always succeeds; otherwise, the handshake fails except for a negligible probability.

The algorithm $\mathsf{rSelect}(\mathbf{G}, U, w, n_1, n_2)$ has the following steps:

1. Partition $\mathbf{G}$ into $\mathbf{G}_0, \cdots, \mathbf{G}_{w-1}$ where $\mathbf{G}_z = \{G_{z_0}, \cdots, G_{z_{l/w-1}}\}$ for $0 \le z \le w-1$. The handshake initiator U belongs to G_{i_u} for some $0 \le i \le w-1$ and $0 \le u \le l/w-1$.
2. Set $\eta = f_1(n_1, n_2, 0)$ and $x = f_1(n_1, n_2, 1, i)$. Choose $r \in_R \{0, 1, \cdots, \lfloor (q+1)w/l \rfloor\}$ and set $y = u + r \cdot l/w$ (in $\mathbb{Z}$). Solve $y = \eta \cdot x + \theta_1 \mod q$ to get θ_1.
3. For $z = 0$ to $w-1$ (except $z = i$), set $y = \eta \cdot f_1(n_1, n_2, 1, z) + \theta_1 \mod q$, and $s_z = y \mod l/w$.
4. Output $(\mathbf{G}^* = \{G_{z_{s_z}}\}_{z=0}^{w-1}, \theta_1)$, where $G_{z_{s_z}}$ is the group selected from $\mathbf{G}_z$ (in particular, $s_i = u$).

The algorithm $\mathsf{rSelectVer}(\mathbf{G}, \mathbf{G}^*, w, n_1, n_2, \theta_1)$ has the following steps:

1. Partition $\mathbf{G}$ into $\mathbf{G}_0, \cdots, \mathbf{G}_{w-1}$ where $\mathbf{G}_z = \{G_{z_0}, \cdots, G_{z_{l/w-1}}\}$ for $0 \le z \le w-1$.
2. Parse $\mathbf{G}^*$ as $\{G_{z_{s_z}}\}_{z=0}^{w-1}$, and set $\eta = f_1(n_1, n_2, 0)$.
3. Accept if for $z = 0$ to $w-1$, it holds that $s_z = y \mod l/w$ where $y = \eta \cdot f_1(n_1, n_2, 1, z) + \theta_1 \mod q$; reject otherwise.

The algorithm $\mathsf{mSelect}(\bar{\mathbf{G}}, X, w, n_1, n_2)$ has the following steps:

1. Parse $\bar{\mathbf{G}}$ as $\{\bar{G}_{z_{s_z}}\}_{z=0}^{w-1}$. Suppose X is the λ-th member of group $\bar{G}_{a_{s_a}}$ for some $0 \le a \le w-1$ and $0 \le \lambda \le |\bar{G}_{a_{s_a}}| - 1$.
2. Set $\eta = f_1(n_1, n_2, 2)$ and $x = f_1(n_1, n_2, 3, a, s_a)$. Choose $r \in_R \{0, 1, \cdots, \lfloor (q+1)/|\bar{G}_{a_{s_a}}| \rfloor\}$, and set $y = \lambda + r \cdot |\bar{G}_{a_{s_a}}|$ (in $\mathbb{Z}$). Solve $y = \eta \cdot x + \theta_2 \mod q$ to get θ_2.
3. For $z = 0$ to $w-1$ (except $z = a$), set $y = \eta \cdot f_1(n_1, n_2, 3, z, s_z) + \theta_2 \mod q$, and $\lambda_z = y \mod |\bar{G}_{z_{s_z}}|$.
4. Output $(\mathbf{X} = \{X_{z_{s_z}, \lambda_z}\}_{z=0}^{w-1}, \theta_2)$, where $\lambda_a = \lambda$ and $X_{z_{s_z}, \lambda_z}$ is the λ_z-th member of group $\bar{G}_{z_{s_z}}$.

The algorithm $\mathsf{mSelectVer}(\bar{\mathbf{G}}, \mathbf{X}, w, n_1, n_2, \theta_2)$ has the following steps:

1. Parse $\bar{\mathbf{G}}$ as $\{\bar{G}_{z_{s_z}}\}_{z=0}^{w-1}$, and parse $\mathbf{X}$ as $\{X_{z_{s_z}, \lambda_z}\}_{z=0}^{w-1}$.
2. Accept if for $z = 0$ to $w-1$, it holds that $\lambda_z = y \mod |\bar{G}_{z_{s_z}}|$ where $y = \eta \cdot f_1(n_1, n_2, 3, z, s_z) + \theta_2 \mod q$; reject otherwise.

Figure 6: Subroutines rSelect, rSelectVer, mSelect, and mSelectVer

`Resistance to impersonation attacks`. We assumed that the signature schemes are existentially unforgeable under an adaptive chosen-message attack. We claim that no adversary can fake a certificate (i.e., an honest user will only deal with those certificates issued by the CAs). Suppose there is an adversary $\mathcal{A}$ that is able to impersonate an honest user $X \in G$ with a non-negligible probability, which means that $\mathcal{A}$ can successfully provide $f_0(\gamma_i, \delta_i, 0)$ or $f_0(\gamma_j, \delta_j, 1)$ without knowing X's private key. We claim that either the encryption scheme or the pseudorandom function f_0 is broken. To see this, consider an experiment $\mathsf{RIA}^*_{\mathsf{SHS}}(\mathcal{A})$, which is exactly the same as the experiment $\mathsf{RIA}_{\mathsf{SHS}}(\mathcal{A})$, except that the encryption under X's public key is substituted with the encryption of an all-zero-bit string of the same length, and that f_0 is replaced by a random function. Clearly, $\mathcal{A}$ has only negligible probability in impersonating X. Therefore, a standard hybrid argument shows that either the encryption scheme or the pseudorandom function is broken.

k-`resistance to detection attacks`. Lemma 4.2 shows that there are always k incorrupt users in the crowd drafted by an incorrupt handshake peer X. If $\mathcal{A}$ breaks this property, we claim that either the encryption scheme or the pseudorandom function f_0 is broken. To see this, consider an experiment $\mathsf{RDA}^*_{\mathsf{SHS}}(\mathcal{A})$ that is the same as $\mathsf{RDA}_{\mathsf{SHS}}(\mathcal{A})$, except that all the encryptions under the public keys of the k incorrupt users are substituted with the encryptions of all-zero-bit string of the same length, and that f_0 is replaced by a random function. Clearly, $\mathcal{A}$ breaks the property in $\mathsf{RDA}^*_{\mathsf{SHS}}(\mathcal{A})$ with a negligible probability. A standard hybrid argument shows that either the encryption scheme or the pseudorandom function is broken.

k-`unlinkability`. Suppose $\mathcal{A}$ breaks this property. Lemma 4.1 shows that there are always k incorrupt pairs of users in the crowds drafted by the incorrupt handshake peers in sessions τ_1 and τ_2. Then consider an experiment $\mathsf{Unlink}^*_{\mathsf{SHS}}(\mathcal{A})$ that is the same as $\mathsf{Unlink}_{\mathsf{SHS}}(\mathcal{A})$ except for the following: without loss of generality, substitute τ_1 with a simulation in which σ_0 and σ_1 are substituted with $f_0(r_1, 0)$ and $f_0(r_2, 1)$, where $r_1 \in_R \{0,1\}^{2\kappa_1}$ and $r_2 \in_R \{0,1\}^{2\kappa_1}$. If the parameters are appropriately chosen, then $\mathcal{A}$ cannot distinguish this simulation from the real world transcript; otherwise, a standard hybrid argument shows that the encryption scheme(s) with respect to the drafted pairs of incorrupt users or the pseudorandom function is broken. However, this means that $\mathcal{A}$ made the wrong conclusion.

k-`indistinguishability to eavesdroppers`. Suppose $\mathcal{A}$ breaks this property. Lemma 4.1 shows that there are al-

ways k incorrupt pairs of users in the selected w pairs of users. Then, $\mathcal{A}$ can distinguish a simulation from a real world transcript. By a standard hybrid argument one shows that either the encryption schemes with respect to the incorrupt users or the pseudorandom function is broken. □

4.4 Discussion and Extension

Practical issues. There are several deployment issues that need to be taken care of. First, if there exists only one group that uses a secret handshake scheme, then an adversary can trivially figure out that the handshake peers belong to that group. In fact, if a secret handshake scheme is implemented as a TLS or IKE cipher suite, then the two parties will exchange a cipher suite designator that clearly shows that they wish to engage in a secret handshake. Second, in any secret handshake scheme, utilizing one-time or reusable credentials alike, it is assumed that there is no easy way to figure out the user who sent/received a certain message; otherwise it is easy for an adversary to figure out who is interacting with whom. This assumption is actually also true in previous privacy-preserving authentication mechanisms [22, 2, 8, 10, 11, 27, 24]. Third, if an adversary can observe that the handshake peers continue talking with each other after finishing the handshake protocol, then it can deduce that the users belong to the same group. Fourth, if an adversary can completely compromise a certain large number of groups (i.e., $p = 1$ for those groups) so that all the drafted users are corrupt, the adversary can easily figure out which pair of users are conducting the handshake protocol.

Those issues can be mitigated by various means. First, it is reasonable to assume that there are many groups, as long as it is not illegal to conduct secret handshakes. Second, there may be settings where the identity of a party is not directly derivable from the routing address that must appear in the clear in the protocol messages. A common example is the case of mobile devices wishing to prevent an attacker from correlating their (changing) locations with the logical identity of the devices (or users) [22]. Further, some form of anonymous communication could make it hard to find out exactly who is engaging in a secret handshake. Third, protection against traffic analysis (e.g., an adversary simply observing if there is a continued communication after a secret handshake session) could be achieved by utilizing mechanisms such as steganographic techniques, or some anonymous communication channels. Fourth, if some groups managed by some CAs are notoriously known as accommodating bad guys (or the CAs are even completely under the adversary's control), then the honest users could simply avoid involving any of such groups.

In summary, if the abovementioned assumptions are satisfied, then our secret handshake scheme (as well as [3, 22, 2, 8]) can provide provable privacy-preserving authentications whereby two participants authenticate each other's membership *simultaneously*; otherwise, all the schemes implement heuristic *best-effort* anonymity.

On the usefulness of k-anonymous secret handshakes. While k-anonymity is a weaker guarantee than full-anonymity (which is indeed the special case of k being the number of groups), it is still sufficient for a variety of applications. For example, in the US legal system, 2-anonymity would be enough to cast "reasonable doubt", thus invalidating a criminal charge, while 3-anonymity would be enough to invalidate a civil charge, in the absence of other evidence.

On the relationship with PKI and the extension to accommodate roles. Our secret handshake scheme can be naturally based on a public key infrastructure with many Certification Authorities (CAs), each of which manages a group of users and its own Certificate Revocation List (CRL). Note that the certificates can be naturally extended to specify roles so that a user can decide, according to a certain policy, whether to initiate, or respond to, a secret handshake request.

How should a session key be derived, if desired? In the basic scheme, we set γ and δ to be random strings. They can be substituted with a Diffie-Hellman instance so that forward-security of the session keys can be ensured. Nevertheless, our scheme achieves a guarantee that is stronger than the `forward-repudiability` which captures the following [3]: at time t_1 honest users U and V interacted; then at time $t_2 > t_1$, it should not be possible for V to prove to a third party that U indeed interacted with V at time t_1, even if V reveals its own secrets. This is so because any (even outside) user can perfectly fake a secret handshake transcript.

5. IMPROVED SCHEME WITH CONSTANT COMMUNICATION COMPLEXITY

In the last section, we presented our basic k-anonymous secret handshake scheme based on public key cryptosystems, which however incurs $O(w)$ communication, and $O(1)$ computation if the public key cryptosystems are based on low public-exponent RSA. In this section, we show how we achieve a scheme that incurs only $O(1)$ communication, and $O(1)$ computation *even if* the public key cryptosystems are *not* RSA-based. The ideas behind the improvement are:

- By utilizing *key-private* public key cryptosystems (reviewed in Section 2), a single ciphertext is sufficient in hiding the identity of the handshake peers. We stress that adopting key-private public key cryptosystems does not really restrict the application of our secret handshake scheme, because the most popular public key cryptosystems such as Cramer-Shoup and RSA-OAEP enjoy this property.

- Instead of letting the handshake peers send the selected w public keys (or certificates), we let them send the polynomial of degree 1, which is used to select the public keys.

Due to space limitation, the improved scheme is left to the full version of this paper [31].

6. k-Anonymous Secret Handshake based on Symmetric Key Cryptosystems

In this section we sketch a secret handshake scheme based on symmetric key cryptosystems, whose detailed description, performance as well as security analysis are left to the full version of this paper [31]. This scheme is based on the key distribution scheme of [7, 6] which we now briefly review: A trusted third party chooses a bivariate symmetric polynomial $f(x, y)$ of degree t over $\mathbb{F}_q$, where t is the tolerated number of corrupt users. A user with a unique identity i holds $f(i, y)$. Then two users i and j can derive a common secret $f(i, j) = f(j, i)$.

The k-anonymous secret handshake scheme based on symmetric key cryptosystems is essentially the same as the one based on public key cryptosystems (i.e., each group is managed by a trusted third party just as in the key distribution case, the users utilize the rSelect and mSelect algorithms to form crowds, no trusted third party is involved in a secret handshake, etc.), except that a trusted third party may also adopt a CRL to publish the identities that have been revoked. We remark that this secret handshake scheme actually has a nice property we may call it `hunter-resilience`, which cannot be achieved in a scheme based on public key cryptosystems. Intuitively, it means that even a successful handshake between an honest user $U \in G$ and an adversary $\mathcal{A}$ that has corrupted a user $V \in G$ does not necessarily convince $\mathcal{A}$ that the handshake peer is U, because V could have leaked its secret to someone other than $\mathcal{A}$.

It is natural that the strategy utilized to reduce the communication complexity in the handshake scheme based on public key cryptosystems also applies to the above handshake scheme based on symmetric key cryptosystems.

7. CONCLUSION

We presented the first unlinkable secret handshake schemes with reusable credentials. We gave a detailed construction based on public key cryptosystems, and sketched the other based on symmetric key cryptosystems. Both schemes are quite efficient, and their security analysis does not resort to any random oracle.

8. ACKNOWLEDGMENTS

We thank the anonymous reviewers for their useful comments.

9. REFERENCES

[1] M. Abadi. Private authentication. In *Proceedings of the 2002 Workshop on Privacy Enhancing Technologies*, pages 27–40, 2003.

[2] W. Aiello, S. Bellovin, M. Blaze, J. Ioannidis, O. Reingold, R. Canetti, and A. Keromytis. Efficient, dos-resistant, secure key exchange for internet protocols. In V. Atluri, editor, *Proc. of the 9th ACM Conference on Computer and Communications Security*, pages 48–58. ACM Press, 2002.

[3] D. Balfanz, G. Durfee, N. Shankar, D. Smetters, J. Staddon, and H. Wong. Secret handshakes from pairing-based key agreements. In *24th IEEE Symposium on Security and Privacy*, Oakland, CA, May 2003.

[4] M. Bellare, A. Boldyreva, A. Desai, and D. Pointcheval. Key-privacy in public-key encryption. In C. Boyd, editor, *Advances in Cryptology–ASIACRYPT '2001*, pages 566–582. Springer-Verlag, 2001. Lecture Notes in Computer Science No. 2248.

[5] M. Bellare and P. Rogaway. Optimal asymmetric encryption. In A. D. Santis, editor, *EUROCRYPT94*, pages 92–111. Springer, 1995. Lecture Notes in Computer Science No. 950.

[6] R. Blom. An optimal class of symmetric key generation systems. In T. Beth, N. Cot, and I. Ingemarsson, editors, *Proc. EUROCRYPT 84*, pages 335–338. Springer-Verlag, 1985. Lecture Notes in Computer Science No. 209.

[7] C. Blundo, A. DeSantis, A. Herzberg, S. Kutten, U. Vaccaro, and M. Yung. Perfectly-secure key distribution for dynamic conferences. In E. F. Brickell, editor, *Proc. CRYPTO 92*, pages 471–486. Springer-Verlag, 1992. Lecture Notes in Computer Science No. 740.

[8] C. Boyd, W. Mao, and K. Paterson. Deniable authenticated key establishment for internet protocols. In the Proceedings of Security Protocols, 2003.

[9] C. Castelluccia, S. Jarecki, and G. Tsudik. Secret handshakes from ca-oblivious encryption. In P. Lee, editor, *Advances in Cryptology - ASIACRYPT 2004*, volume ??? of *Lecture Notes in Computer Science*, pages ???–??? Springer, 2004.

[10] D. Chaum. Blind signatures for untraceable payments. In R. L. Rivest, A. Sherman, and D. Chaum, editors, *Proc. CRYPTO 82*, pages 199–203, New York, 1983. Plenum Press.

[11] D. Chaum and E. V. Heyst. Group signatures. In D. W. Davies, editor, *Advances in Cryptology — Eurocrypt '91*, pages 257–265, Berlin, 1991. Springer-Verlag. Lecture Notes in Computer Science No. 547.

[12] R. Cramer, I. Damgård, and B. Schoenmakers. Proofs of partial knowledge and simplified design of witness hiding protocols. In Y. G. Desmedt, editor, *Proc. CRYPTO 94*, pages 174–187. Springer, 1994. Lecture Notes in Computer Science No. 839.

[13] R. Cramer and V. Shoup. A practical public key cryptosystem provably secure against adaptive chosen ciphertext attack. In H. Krawczyk, editor, *Proc. CRYPTO 98*, pages 13–25. Springer-Verlag, 1998. Lecture Notes in Computer Science No. 1462.

[14] W. Diffie and M. E. Hellman. New directions in cryptography. *IEEE Trans. Inform. Theory*, IT-22:644–654, Nov. 1976.

[15] D. Dolev, C. Dwork, and M. Naor. Non-malleable cryptography. In *Proc. 23rd ACM Symp. on Theory of Computing*, pages 542–552. ACM, 1991.

[16] E. Fujisaki, T. Okamoto, D. Pointcheval, and J. Stern. Rsa-oaep is secure under the rsa assumption. In J. Kilian, editor, *Proc. CRYPTO 2001*, volume 2139 of *Lecture Notes in Computer Science*, pages 260–274. Springer-Verlag, 2001.

[17] O. Goldreich, S. Goldwasser, and S. Micali. How to construct random functions. *Journal of the ACM*, 33(4):792–807, Oct. 1986.

[18] S. Goldwasser, S. Micali, and R. Rivest. A digital signature scheme secure against adaptive chosen-message attacks. *SIAM J. Computing*, 17(2):281–308, Apr. 1988.

[19] D. Harkins and D. Carrel. *RFC 2409: The Internet Key Exchange (IKE)*. Internet Activities Board, 1998.

[20] J. Holt, R. Bradshaw, K. Seamons, and H. Orman. Hidden credentials. In *Proceedings of 2nd ACM Workshop on Privacy in the Electronic Society*, pages 147–173, 2003.

[21] M. Jakobsson, K. Sako, and R. Impagliazzo. Designated verifier proofs and their applications. In *Proc. EUROCRYPT 96*, pages 143–154.

[22] H. Krawczyk. Sigma: The 'sign-and-mac' approach to authenticated diffie-hellman and its use in the ike-protocols. In D. Boneh, editor, *Proc. CRYPTO 2003*, volume 2729 of *Lecture Notes in Computer Science*, pages 400–425. Springer-Verlag, 2002.

[23] N. Li, W. Du, and D. Boneh. Oblivious signature-based envelope. In *Proceedings of 22nd ACM Symposium on Principles of Distributed Computing (PODC)*, pages 182–189. ACM, 2003.

[24] M. Naor. Deniable ring authentication. In M. Yung, editor, *Proc. CRYPTO 2002*, volume 2442 of *Lecture Notes in Computer Science*, pages 481–498. Springer-Verlag, 2002.

[25] P. Persiano and I. Visconti. User privacy issues regarding certificates and the tls protocol: the design and implementation of the spsl protocol. In *ACM Conference on Computer and Communications Security*, pages 53–62. ACM, 2000.

[26] C. Rackoff and D. R. Simon. Non-interactive zero-knowledge proof of knowledge and chosen ciphertext attack. In J. Feigenbaum, editor, *Proc. CRYPTO 91*, pages 433–444. Springer, 1992. Lecture Notes in Computer Science No. 576.

[27] R. Rivest, A. Shamir, and Y. Tauman. How to leak a secret. In *Advances in Cryptology–ASIACRYPT '2001*, pages 552–565.

[28] R. Sakai, K. Ohgishi, and M. Kasahara. Cryptosystems based on pairing. In *Proceedings of the Symposium on Cryptography and Information Security (SCIS)*, 2002.

[29] L. Sweeney. k-anonymity: A model for protecting privacy. *International Journal on Uncertainty, Fuzziness and Knowledge-based Systems*, 10(5):557–570, 2002.

[30] L. von Ahn, A. Bortz, and N. Hopper. k-anonymous message transmission. In V. Atluri, editor, *Proc. of ACM-CCS 03*, pages 122–130, Washington D.C., USA, October 2003. ACM, ACM Press.

[31] S. Xu and M. Yung. k-anonymous secret handshakes with reusable credentials. Full version of the present paper.