2010

Efficient Secret Handshaking Protocol

Maged Ibrahim, *Helwan University*

ISSN No. 0976-5697

Volume 2, No. 6, Nov-Dec 2011

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

Efficient Secret Handshaking Protocol

Maged Hamada Ibrahim
Department of Electronics, Communications and Computers Engineering
Faculty of Engineering, Helwan University
1, Sherif st., Helwan, Cairo; Egypt
mhii72@hotmail.com

Abstract: Secret handshaking protocols allow two members of the same group to identify each other secretly, i.e., any two parties who are members of the same group will recognize each other as members, yet, a party which is not a member of this group cannot tell, by engaging some party in the handshaking protocol, whether that party is a member of this group. Unlinkability is one of the main merits of secret handshaking protocols, that is, a party engaged in at least two handshakes must not be able to link any two different handshakes to a particular party. To achieve unlinkability, almost all protocols proposed so far rely on the one-time credentials technique, where each party can use her credential only once. Hence, each party must hold enough credentials allowing her to engage in the handshakes for enough period of time (e.g. a month) without referring to the group authority for renewal. There is a severe security problem when one-time credentials are employed, that is, an active adversary may initialize with an honest party as many handshaking sessions as she can and hence, depletes all the credentials held by this party, once a party runs out of credentials she will not be able to engage in handshaking no more (Denial of Service attack, DoS). At the same time, the group authority must be able to manage enormous number of issued credentials in data structures and certificate revocation lists (CRL). Thus, on the large scale implementation (large group population), one-time credentials become impractical. In this paper, we propose a provably secure two-party secret handshaking protocol which realizes the unlinkability property using only one permanent credential for each member and avoiding the inefficient one-time credentials. At the same time, our protocol provides immediate revocation of members by the group authority without relying heavily on CRL structures.

Keywords: Secret handshakes, authentication, one-time credentials, unlinkability, revocation, denial of service, anonymous RSA, mediated PKI.

I. INTRODUCTION

A secret handshaking scheme is a cryptographic primitive which allows two members of the same group to identify each other secretly, in the sense that each party reveals his/her affiliation to the other only if the other party is a member of the same group. If (*A*)lice is a member of group G_a and (*B*)ob is a member of group G_b, a secret handshaking scheme is supposed to guarantee the following [1]:

a. *A* and *B* authenticate each other if and only if $G_a = G_b$.
b. If $G_a \neq G_b$ then the only thing that either party learns is what could be implied from knowing that $G_a \neq G_b$.
c. *A* can choose not to reveal anything about herself unless *B* is a member of *A*'s group (and vice versa).
d. Given three parties *A*, *B* and *C* all in the same group, a handshaking protocol between *A* and *B*, and a handshaking protocol between *A* and *C*, then *A* cannot determine whether or not *B* = *C*.
e. An eavesdropper or a man in the middle learns nothing from the protocol.

The original secret handshaking protocol of [1] is based on bilinear maps, which can be constructed using Weil pairings on elliptic curves. The protocol of [1] builds on the non-interactive key-agreement scheme of [2].

A. The Problem:

We first describe a real life example for a secret handshaking protocol, this example was also introduced in [1] to clarify the idea of secret handshakes, and then we escalate the problems we deal with in this paper.

Consider a user (*A*)lice who lives in a country with a questionable human-rights record. The ministry of transportation (viewed as a group authority in possession of a secret master key) in that country issues driving licenses for citizens who have passed the driving test. For Alice, this license (credential) comes in the way of a one-time identity (pseudonym) and a private parameter (point on an elliptic curve generated by the master key) corresponding to this identity $(ID_{A,i}, priv_{A,i})$, $i = 1,\cdots,t$ where t is the number of one-time credentials given to Alice at a time. Alice can show her identity to anyone, but keeps her private point secret. The ministry of transportation also issues one-time credentials for traffic cops. (*B*)ob is such a traffic cop, and his one-time credential comes on the form $(ID_{B,i}, priv_{B,i})$, $i = 1,\cdots,t$. Bob demands to see Alice's driving license. Alice wants to make sure that Bob is a real cop, and not an impostor. To perform a secret handshake, Alice and Bob exchange their one-time identities and then, using their private parameters corresponding to these identities, both of them are able to compute a session key *K*. If both Alice and Bob belong to the same group (controlled by the ministry of transportation) then both of them compute the same session key *K*. Once a party has burned all her credentials, she contacts the group authority for a new set of credentials. Such contact to the group

authority must be performed over a reasonably long period of time (e.g. a month).

Now we ask two questions in the context of the above example. First, "In a country, how many cops are employed by the ministry of transportation and how many citizens in possession of a driving license?" Second, "How many handshakes a cop is supposed to perform with the drivers per month?" The answer to the first question is predicted to be "millions", while the answer to the second question is supposed to be in the order of "hundreds". Hence, the group authority is supposed to manage trillions of one-time credentials in data structures and revocation lists.

One-time credentials also suffer from possible DoS attacks in the sense that, an adversary may initialize as many handshaking sessions as she can (although unsuccessful) with some honest member and is able to deplete all the credentials held by an honest member. Once the member runs out of credentials he will not be able to engage in future handshakes for the rest of the period before renewal. Therefore, the idea of using one-time credentials is almost impractical. Yet, in all previous contributions, one-time credentials were the only way to achieve unlinkability in a handshaking protocol. That is, given three parties *A*, *B* and *C*, a handshaking protocol between *A* and *B*, and a handshaking protocol between *A* and *C*, then *A* cannot determine whether or not *B* = *C*.

B. Our solution: A bird's eye view:

Since the idea of one-time credentials becomes impractical on the large scale. Our solution aims to allow secret handshakes using only one permanent credential for each member to allow easy management of credentials in data structures and at the same time preserve the unlinkability property which is one of the main merits of secret handshakes. Our solution is based on the idea of mediated public-key infrastructure (mPKI) introduced in [9]. Our construction has much similarity to the efficient mediated-RSA (mRSA) [9].

The main difference is that, the role of the CA in an mPKI will be played by the group authority (GA) and we do not consider the users' personal public/private keys generated by this CA, only group keys are considered.

mRSA was invented as a method to achieve fast revocation in RSA PKI. mRSA involves a special entity, called the SEM (SEcurity Mediator), an on-line partially trusted server, to help signing or decrypting messages. The CA generates the private key d corresponding to Bob's (the receiver's) public key e and splits this private key into two pieces. One piece (d_{SEM}) is delivered to the SEM and the other piece (d_{Bob}) is delivered to Bob. The pair (e, N) is the usual RSA public key, where N is the RSA public modulus. To decrypt a received ciphertext *C*, each party (Bob and SEM) performs his/her partial decryption on *C*; finally the partial decryptions are combined to recover the plaintext message *M*. To revoke Bob's ability to sign or decrypt messages, the CA instructs the SEM to stop issuing partial decryptions or signatures (spoken of as tokens) for Bobs public key. At this instant, Bob's signature and/or decryption capabilities are revoked. The functionality is equivalent to (and indistinguishable from) standard RSA due to the fact that the splitting of the private key is transparent to the outside, i.e., to those who use the corresponding public key. Also, knowledge of a half-key cannot be used to derive the entire private key. Therefore, neither Bob nor the SEM can decrypt or sign a message without mutual consent.

Our solution assumes that the CA of the mPKI is the group authority (GA), this is efficient since we are dealing with large scale implementation (e.g. country wide system). The GA generates a public/private key pair for the group, publishes the public key and keeps the group private key secret. For each group member, the GA splits the private key into two large pieces (roughly speaking) and delivers one piece to the SEM and the other piece to the member as his secret parameter. This secret parameter together with the member's ID forms the member's credential for this group. It is important to notice that, even if all secret parameters held by the group members are compromised by a certain adversary, as long as the SEM is not compromised, this adversary still has no information (in the information theoretic sense) about the group private key held by the GA since, each pair of pieces (member, SEM) are picked independently.

Our solution also assumes that the parties involved in the handshaking protocol never exchange their ID's. An ID is used only to authenticate a party to her own SEM. Consequently, there is no reason for a party to renew her ID since this ID is not shown in the handshaking protocol to anyone but the SEM.

II. RELATED WORK

A secret handshaking scheme is a cryptographic primitive originally introduced in [1] and then studied in several publications [3, 4, 5, 6, 7]. Among these contributions, the work in [5, 6, 7] focused on the extension of the secret handshakes schemes to the multiparty case (more than two parties are involved in the handshaking) and hence, they are beyond the scope of this paper. It is worth noting that, the work in [5] is an inefficient attempt to allow reuse of credentials for several times, their scheme offer somewhat weak anonymity to the members who furthermore must be aware of the information of other groups and hence still such attempt quite a burden for a member.

Considering the two-party case, the protocol proposed in [1] (under the Bilinear Diffie-Helman assumption) is a simple adaptation of the non-interactive key-agreement scheme of [2] and works as follows: As in the identity based encryption scheme of [8], *A* and *B* can compute each other's public keys from each other's ID's and from the public parameters associated with the CA. If Alice is a group member, she can use her trapdoor t_A which is a secret point on the elliptic curve corresponding to pk_A to non-interactively compute a session key from (t_A, pk_B). Similarly, if Bob is a group member he can compute the same session key from (t_B, pk_A). The two parties can then verify if they computed the same key via a standard MAC-based challenge-response protocol. Under the Bilinear Diffie-Hellman (BDH) assumption, it is easy to show (in the Random Oracle Model)

that an attacker who does not hold the correct trapdoor cannot compute the session key. Moreover, the MAC-based challenge response confirmation protocol has the needed property that without the knowledge of the key, one learns nothing from the counterparty's responses.

The work in [3] showed how to build secret handshake protocols using a tool spoken of as "CA-oblivious public key encryption" which is an encryption scheme such that neither the public key nor the ciphertext reveal any information about the Certification Authority. Their schemes are secure under a standard cryptographic assumption: the hardness of the classical computational Diffie-Hellman problem. They showed that identity based encryption [8, 11] and hence the protocols of [1] are special case of their CA-oblivious encryption technique. The work in [4] proposed three RSA-based constructions of secret handshake protocol and the security treatment of them. The schemes assume the hardness of the RSA problem. The work in [4] is a response to an open problem raised in [3]. Almost all the two-party secret handshakes protocols proposed so far rely on one-time credentials to insure that instances of the handshake protocol performed by the same party cannot be linked.

III. MOTIVATIONS AND CONTRIBUTIONS

A. Motivations:

The work in this paper is motivated by the observation that the two-party secret handshakes schemes proposed so far suffer from the one-time credentials as a common weakness, that is, for each handshaking session, the member must show a new identity and hence, the member must hold and securely store a large amount of one-time identities enough for all the handshaking sessions he is going to perform without referring to the group authority side by side with all the secret parameters associated with each one-time credential, which is quite a burden for the member. One-time credentials technique is vulnerable to DoS attacks in the sense that, an adversary may initialize as many handshaking sessions as she can (although unsuccessful) with some honest member and is able to deplete all the credentials held by an honest member. Once the member runs out of credentials he will not be able to engage in future handshakes for the rest of the period before renewal. Also, these protocols rely heavily on the CRL's to revoke members.

Hence, each member must frequently download CRL updates or update his information using Online Certificate Status Protocol (OCSP), which is again quite a burden for a member, specially in case of high population.

B. Contributions:

The contributions of this paper is to propose a two-party secret handshaking protocol allowing the group member to hold and securely store only a single permanent credential, this credential can be efficiently used to perform as many handshaking sessions as he wants without referring to the group authority unless there are regular periodic key updates and at the same time preserves the unlinkability property. Our protocol avoids the need to download and access CRL's by the members and provide immediate revocation of members by the group authority. The protocol avoids the DoS attacks. Our protocol is a three-round protocol and hence the computations and communications complexity of our protocol are almost the same as the original protocol and is one round less than those proposed in [3, 4].

IV. OUR DEFINITIONS

In a secret handshake scheme, there are group authorities managing groups of members, each group has a public key and a matching master private key. They can provide any member with his secret parameter (partial key) and his unique identity. The members can then identify themselves in a protocol in which the parties involved begin by knowing only the groups public keys and their own secret parameters and identities provided by their authorities. We emphasize that the unique ID's given by a GA to her group members are used to identify the members to their security mediator (SEM) associated with this GA.

A secret handshake protocol **SH** consists of the following protocols/algorithms: (**Setup**, **CreateGroup**, **AddMember**, **RemoveMember**, **Handshake**) such that:

a. **SH.Setup** is a PPTM which takes a security parameter k as input. The outputs are the public parameters.

b. **SH.CreateGroup** is a PPTM which takes the public parameters as input and outputs a pair of group keys (pk_G, sk_G). It may also output a data structure CRL called a certificate revocation list (held by the SEM) which is originally empty.

c. **SH.AddMember** is a polynomial time two-party protocol (**Member, Group**) where

i. **SH.AddMember.Member** takes the public parameters, a bit string ID and a group public key pk_G as inputs.

ii. **SH.AddMember.Group** takes the public parameters, ID and the matching group secret key sk_G as inputs.

SH.AddMember.Member outputs the secret parameters associated with this unique ID.

a. **SH.RemoveMember** is a PPTM which takes the public parameters, a bit string ID, a group pair of keys (pk_G, sk_G) and the corresponding current CRL (of the SEM) as inputs. It outputs an updated CRL which includes the newly revoked ID.

b. **SH.Handshake** is a polynomial time two-party protocol ((*A*)lice, (*B*)ob) where:

i. **SH.Handshake.A** takes the public parameters, Alice's group public key pk_G, Alice's secret parameter d_A and Alice's unique identity ID_A as inputs.

ii. **SH.Handshake.B** takes the public parameters, Bob's group public key pk_G, Bob's secret parameter d_B and Bob's unique identity ID_B as inputs.

The algorithms jointly output **Accept** if $(G_a = G_b) \wedge \{ID_A, ID_B\} \cap CRL = \phi$; and output **Fail** otherwise.

V. SECURITY REQUIREMENTS

a. *Completeness:* if two members engage in the protocol **SH.Handshake** with valid pair of keys associated with the same group public key, then both parties output **Accept** at the end of the protocol.
b. *Impersonator resistance:* Given a group public key, it is computationally infeasible without the knowledge of some part of the master private key associated with it to successfully execute the protocol **SH.Handshake** with a member of this group.
c. *Detector resistance:* Given a group public key pk_G it is computationally infeasible to determine whether a member is associated with pk_G without the knowledge of a partial private key associated with pk_G.
d. *Unlinkability:* Given three members *A*, *B* and *B'* which are in the same group, assume a handshaking achieved between two members *A* and *B* and a handshaking achieved between two members *A* and *B*', then *A* must not be able to determine whether *B*=*B*'.
e. *Indistinguishability to an eavesdropper:* Given any two members, it is computationally infeasible to distinguish a successful handshake between those members from an unsuccessful one.

Remark. We distinguish between unlinkability and full-unlinkability in the following manner: In the former, no adversary is able to associate two handshakes involving the same honest member even if it is in the group of this member and participated in both executions. This remains to be true even if the adversary plays the roles of multiple participants. In the later, no adversary is able to associate two handshakes involving the same honest member even if it is in the group of this member and participated in both executions, and the member has been corrupt later on. This remains to be true even if the adversary plays the roles of multiple participants. We emphasize that our protocol satisfies the requirements of full-unlinkability.

VI. THE UNDERLYING PRIMITIVES

In this section we describe in some details the mRSA PKI and show how to achieve anonymity in RSA encryption.

A. Mediated RSA:

Mediated RSA was invented as a simple method to achieve fast revocation in public-key cryptosystem. As usual, a trusted certificate authority (CA) sets up the RSA modulus *N*, the public exponent e and the private exponent d for the user. Next, instead of delivering d to the user, the CA splits *d* into two pieces d_{SEM} and d_{user} such that $d = d_{SEM} + d_{user}$ modulo φ where φ is the RSA Euler totient. Finally, the CA secretly delivers d_{user} to the user and d_{SEM} to the SEM.

Encryption. For Alice to encrypt a message $M \in Z_N$ to Bob, she uses Bob's public pair (N, e) to compute the usual RSA ciphertext $C = M^e \bmod N$ and sends *C* to Bob.

Decryption. On the reception of *C* by Bob, the decryption process is as follows:

a. Bob delivers *C* to the SEM.
b. If Bob's key is revoked, the SEM returns Error and aborts, else,
c. The SEM computes her partial decryption $PD_{SEM} = C^{d_{SEM}} \bmod N$ to Bob.
d. Bob computes his partial decryption $PD_{Bob} = C^{d_{Bob}} \bmod N$ and extracts $M = PD_{SEM} PD_{Bob} \bmod N$.

Remarks:

a. It is important to notice that the SEM gains no information about the decrypted message *M* [9].
b. Although in mRSA PKI the CA distributes personal public/private keys for each user, in our construction, the GA only generates one public/private key pair for the whole group.

B. Anonymous RSA:

A simple observation to the RSA encryption above is that standard RSA does not provide anonymity, even if all moduli in the system have the same length. One approach to anonymize RSA, suggested by Desmedt [10], is to add random multiples of the modulus N to the cipher- text. This padding removes any information about the size of N and does not interfere with the reduction of the value modulo N. In our protocol, we assume that such a technique is adopted and that the adversary gains no information on the RSA modulus involved in some protocol (in a statistical sense) from the encoding used in the transcript [4].

C. Assumptions and Model:

Our model has much similarity to the mediated PKI (mPKI). Our model follows the mRSA PKI introduced in [9]. The CA of the mPKI plays the role of a group authority. We have a set $GA = \{GA_1, \cdots, GA_g\}$ of g group authorities, each authority is assumed to be fully trusted by all members in this group. Using the system-wide public parameters and policy which are common to all authorities, each authority GA_i generates her own public/private key pair (pk_{G_i}, sk_{G_i}), publishes pk_{G_i} as the public-key of group G_i and keeps sk_{G_i} as the corresponding master private key. One may think of any group G_i as a domain (or system).

For each group G_i there is a SEcurity Mediator server (SEM_{G_i}) associated with group authority GA_i. SEM_{G_i} is assumed semi-trusted (honest-but-curious) in the sense that, it follows the execution steps of the protocol word for word but it is willing to learn any information leaked during execution.

SEM_{G_i} interacts with each member in the group G_i via a private and authenticated one-to-one channel.

A member $M_j \in G_i$ holds a unique identity ID_{Mj} as in a mPKI and is able to prove his identity to SEM_{G_i}. Although, in a mPKI, a user has his own personal public/private key pair, these keys are not needed in our handshaking protocol.

VII. OUR IDEA AND PROTOCOL OUTLINES

To add a member M_j to group G_i, the group authority GA_i splits the group secret key sk_{G_i} into two random large pieces $m_i^{(j)}$ and $s_i^{(j)}$ GA_i secretly delivers $m_i^{(j)}$ to M_j as his secret parameter and $s_i^{(j)}$ to the SEM_{G_i} as the M_j 's partial private key. Revoking a member $M_j \in G_i$ is very simple; GA_i instructs SEM_{G_i} not to help M_j in performing any decryptions using his corresponding partial key $s_i^{(j)}$ and hence, since M_j will not be able to perform any decryptions without the piece $s_i^{(j)}$ held by SEM_{G_i}, M_j is immediately revoked.

Now consider the two parties (*A*)lice and (*B*)ob where $A \in G_a$ and $B \in G_b$ for arbitrary groups G_a and G_b. Performing a secret handshaking between *A* and *B* is briefly outlined as follows: *A* (using her own group public key pk_a encrypts a random nonce r_a to *B* , while *B* (using his own group public key pk_b encrypts a random nonce r_b to *A*. Next, each party contacts his/her own SEM for decryption: *A* communicates with SEM_{G_i} to decrypt what she received from *B*, at the end, *A* gets r_b', on the other side, *B* communicates with SEM_{G_b} to decrypt what she received from *A*, at the end, *B* gets r_a' . Notice that:

a. Neither of the two parties will get the partial decryption from their SEM unless they successfully prove themselves to their SEM. That is *A* will not be able to decrypt unless $A \in G_a$ and *B* will not be able to decrypt unless $B \in G_b$.

b. $r_a = r_a'$ if and only if the encryption was performed using pk_b and $B \in G_b$. $r_b = r_b'$ if and only if the encryption was performed using pk_a and $A \in G_a$.

From the above discussion, given at least one of the two parties is honest (encrypts using his own group public key), then: $(r_a = r_a') \wedge (r_b = r_b')$ if and only if $G_a = G_b$.

VIII.CONCRETE DESCRIPTION OF OUR PROTOCOL

In **SH.Setup** and **SH.Creatgroup** each Group authority G_i takes the security parameter *k* as an input and outputs the public parameters, the group G_i's public key pk_{G_i} and the group G_i's private key sk_{G_i}. In case of RSA, $pk_{G_i} = (e_{G_i}, N_{G_i})$ where e_{G_i} is G_i's RSA public exponent and N_{G_i} is G_i's RSA public modulus, and $sk_{G_i} = (d_{G_i}, N_{G_i})$ where d_{G_i} is G_i's RSA private exponent. To add a member M_j to group G_i (**SH.AddMember**), the member approaches GA_i which creates an identity ID_{M_j} for him. Now for the new member M_j , GA_i splits d_{G_i} into two pieces, $(d_{G_i}^{(M_j)} \in_r Z_{\varphi_{Gi}})$ and $(d_{G_i}^{(SEM,M_j)} \in_r Z_{\varphi_{Gi}})$ such that $d_{G_i}^{(M_j)} + d_{G_i}^{(SEM,M_j)} = d_{G_i} \bmod \varphi_{G_i}$ where φ_{G_i} is the RSA Euler totient. GA_i secretly delivers $d_{G_i}^{(M_j)}$ to M_j and secretly delivers $d_{G_i}^{(SEM,M_j)}$ side by side with ID_{M_j} to SEM_{G_i}.

We assume the existence of a strong hash function $H : \{0,1\}^* \rightarrow \{0,1\}^k$ (e.g. SHA-256), modeling a random oracle (RO).

Now we are ready to describe the handshaking protocol **SH.Handshake**. Assume a member *A* and a member *B* where $A \in G_a$ and $B \in G_b$. By the notation $X \rightarrow Y$ we mean, "*X* computes and sends to *Y* ". The protocol **SH.Handshake** between *A* and *B* is as follows:

a. $A \rightarrow B$:
   i. Picks a large $r_a \in_r Z_{N_{Ga}}$.
   ii. Computes the ciphertext $C_a = E_{pk_{Ga}}(r_a) = r_a^{e_{Ga}} \bmod N_{G_a}$.
   iii. Picks a challenge $ch_a \in_r \{0,1\}^k$.
   iv. Sends C_a and ch_a.

b. $B \rightarrow A$:
   i. Picks a large $r_b \in_r Z_{N_{Gb}}$
   ii. Computes the ciphertext $C_b = E_{pk_{Gb}}(r_b) = r_b^{e_{Gb}} \bmod N_{G_b}$.
   iii. With the help of SEM_{G_b}, obtains r_a as a decryption of C_a.
   iv. Computes the challenge response $h_b = H(r_a, r_b, ch_a)$.

v. Picks a challenge $ch_b \in_r \{0,1\}^k$.

vi. Sends C_b, ch_b and h_b.

c. $A \rightarrow B$:

i. Contacts SEM_{G_a} to obtain r_b as a decryption of C_b.

ii. Computes $h_a = H(r_a, r_b, ch_b)$.

iii. Sends h_a.

iv. Checks whether h_b equals $H(r_a, r_b, ch_a)$, if equality holds then output **Accept**, otherwise output **Fail**.

d. B:

i. Checks whether h_a equals $H(r_a, r_b, ch_b)$, if equality holds then output **Accept**, otherwise output **Fail**.

Remark. Notice that our protocol can be easily extended to an authenticated key agreement protocol. Simply, each party computes the session key as $K = H(r_a, r_b)$. Obviously, if both parties output **Accept** in the secret hand-shaking protocol, then both of them compute the same session key K.

IX. SECURITY PROOF

In this section we prove the security of our handshaking protocol. We prove the impersonation-resistance security, detection-resistance security, unlinkability and indistinguishability to eavesdroppers of our protocol. We have to emphasize that: First, the ciphertexts C_a and C_b sent by *A* and (respectively) *B* reveal no information about which public keys are chosen for encrypting r_a and (respectively) r_b. Second, if an adversary is able to compromise all the secret parameters held by all members in a certain group, as long as the SEM is not compromised, she is unable (in the information theoretic sense) to gain any information about the master private key held by the group authority.

Lemma 1. Under the assumption that the underlying encryption scheme of the mPKI is one-way (OW) secure, then our SH protocol is impersonator-resistance secure in the random oracle model (ROM).

Proof. Assume that an adversary **A** violates (with non-negligible probability ε) the impersonation resistance property against some honest member *V* identified by ID_V. Assume that **A** plays the role of *A* while *V* plays the role of *B*. Consider the worst case situation that (among the published public keys of the groups) **A** has picked the *V* 's group public key pk_{G_v} and sent $C_a = E_{pk_v}(r_a)$ side by side with ch_a where r_a and ch_a are picked by **A**. On the reception of C_a and ch_a by *V*, *V* decrypts for r_a correctly (since the encryption is performed using *V* 's group public key) and responds by sending $C_v = E_{pk_v}(r_v)$ side by side with ch_v and $h_v = H(r_a, r_v, ch_a)$ where r_v, ch_v are picked by *V*. For **A** to obtain a decryption of C_v, she must prove her identity ID_A to SEM_{G_v}. Since **A** is not a member in G_v, she fails to prove her identity to the SEM_{G_v} (this follows from the security of the mPKI). Notice that **A** may use *A*'s identity but, in this case she does not hold the secret partial private key $d_{G_v}^{(V)}$ corresponding to $d_{G_v}^{(SEM_{G_v}, V)}$ held by SEM_{G_v} for *A*, consequently even if **A** succeeded in faking the identity of *A* he will fail to perform the partial decryption to obtain r_v. In this case, in the ROM, **A** can send a valid response h_a to *V* (with non-negligible probability) only if **A** queries the oracle $H(.)$ on the input (r_a, r_v, ch_v) s.t. in particular, r_v was the value picked by *V* and sent to **A** in the form of C_v. Thus, if so, we can use **A** to create another adversary **B** that (with non-negligible probability) breaks the OW property of the encryption scheme: On encryption challenge $C_x = E_{pk_v}(x)$ where x is chosen at random from the message space **M**, **B** passes the same challenge as its response $C_v = C_x$ to **A**. **B** also passes ch_v and h_v picked at random. The only way **A** can tell between this communication and a conversation with an honest *V* is by querying *H* on (r_a, r_v, ch_a) for $r_v = x$ is exactly the decryption of C_v. Since **A** can make only polynomial queries to *H*, **B** can pick one such query at random, and **B** will have a non-negligible chance to output $r_v = x$. Thus **B** breaks the (OW)ness of the encryption scheme, which contradicts our assumption.

Lemma 2. Under the assumption that the underlying encryption scheme of the mPKI is one-way (OW) secure, then our SH protocol is detector-resistance secure in the random oracle model.

Proof. This lemma could be proved via simulation, by showing that if an adversary **A** distinguishes between an interaction with a simulator (SIM) and an interaction with a group member *V*, the OW security of the underlying encryption scheme is broken. To make such distinguishability, it must be that **A** distinguishes random values C_v, ch_v and h_v chosen by the SIM from values $C_v = E_{pk_v}(r_v)$, ch_v and $h_v = H(r_a, r_v, ch_a)$ honestly computed by V. But this can happen only if **A** makes an oracle query on the triple r_a, r_v, ch_a itself. Again, if **A** is to know r_v, in this case, **A** cloud be used to break (with non-negligible probability) the OW security of the underlying encryption scheme thus contradicting the assumption stated in the lemma.

Lemma 3. Our SH protocol is fully unlinkable.

Proof. Notice that during the handshaking protocol, *A* and *B* never exchange their identities. A member identity is used

only to authenticate a member to his SEM so that he is able to obtain the partial decryption of the ciphertext he receives from the other party. Thus, if A (for example) performed a handshake with B using random values r_b and ch_b and a handshake with B' using fresh independent random values r_b' and ch_b', assuming also that A, B and are in the same group, then **A** cannot distinguish whether $B = B'$. A cannot distinguish the execution even if he corrupted any of them at a later time. The last statement is true assuming that the parties securely erase their local randomness after execution, or else, we are facing an adaptive security problem. One may refer to [12] for more about this issue.

Lemma 4. Under the assumption that the underlying encryption scheme of the mPKI is one-way (OW) secure, Our SH protocol is indistinguishable to eavesdroppers in the ROM.

Lemma 4 is easily proved by noticing that, whether or not the handshaking protocol is successful (i.e. no, one or both parties output **Fail**), the number of messages and the distribution of the transferred messages are the same to an eavesdropper. Hence, a channel observer cannot distinguish whether or not the handshaking is successful unless he is able to break the OW security of the underlying encryption scheme.

X. CONCLUSIONS

In this paper we introduced an efficient secret handshaking scheme that overcomes the security breaches resulting from using one-time credentials as a way to achieve unlinkability and to overcome all the management difficulties associated with such techniques. Our protocol is the first to achieve full unlinkability without relying on one-time credentials.

XI. REFERENCES

[1] D. Balfanz, G. Durfee, N. Shankar, D.K. Smetters, J. Staddon, and H.C. Wong, Secret handshakes from pairing-based key agreements," in IEEE Symposium on Security and Privacy, 2003.

[2] R. Sakai, K. Ohgishi, and M. Kasahara, Cryptosystems Based on Pairings., Proceedings of the Symposium on Cryptography and Information Security (SCIS 2000), 2000.

[3] C. Castelluccia, S. Jarecki, and G. Tsudik, Secret Handshakes from CA-Oblivious Encryption., Advances in Cryptology - Asiacrypt 2004 (P. J. Lee, ed.), Lect. Notes Comput. Sci., vol. 3329, Springer, 2004, pp. 293-307.

[4] D. Vergnaud. RSA-based secret handshakes. In International Workshop on Coding and Cryptography, Bergen, Norway, March 2005.

[5] Xu, S., and Yung, M. *k*-anonymous secret handshakes with reusable credentials. In 11th CCS (Washington D.C., USA, 2004), ACM, pp. 158-167.

[6] G. Tsudik and S. Xu. A Flexible Framework for Secret Handshakes. In Privacy-Enhancing TechnologiesWork- shop (PET06), June 2006. Earlier version appeared as a Brief Announcement in ACM PODC05, August 2005.

[7] S. Jarecki, J. Kim, and G. Tsudik. Authentication for Paranoids: Multi-Party Secret Handshakes. In ACNS06, June 2006.

[8] D. Boneh and M. Franklin, Identity based encryption from weil pairing," in Advances in Cryptography, CRYPTO2001, Santa Barbara, CA, August 2001.

[9] D. Boneh, X. Ding, G. Tsudik, and M. Wong, A Method for Fast Revocation of Public Key Certi_cates and Security Capabilities, in proceedings of the 10th USENIX Security Symposium, pp. 297-308.

[10] Y. Desmedt, Securing Traceability of Ciphertexts - Towards a Secure Software Key Escrow System., Advances in Cryptology - Eurocrypt95 (L. C. Guillou and J.-J. Quisquater, eds.), Lect. Notes Comput. Sci., vol. 921, Springer, 1995, pp. 147-157.

[11] C. Cocks, An identity based encryption scheme based on quadratic residues, Cryptography and Coding, Lecture Notes in Computer Science vol. 2260, Springer-Verlag, Berlin (2001), pp. 360-364.

[12] Ran Canetti , Uri Feige , Oded Goldreich , Moni Naor, Adaptively secure multi-party computation, Proceedings of the twenty-eighth annual ACM symposium on Theory of computing, p.639-648, 1996.