Secret Handshakes from CA-Oblivious Encryption

Claude Castelluccia[2,1], Stanisław Jarecki[1], and Gene Tsudik[1]

[1] Information and Computer Science, University of California, Irvine, CA 92697, USA
{ccastell,stasio,gts}@ics.uci.edu
[2] INRIA Rhône-Alpes, 655 Avenue de l'Europe, 38334 Saint Ismier CEDEX, France
claude.castelluccia@inrialpes.fr

Abstract. Secret handshakes were recently introduced [BDS+03] to allow members of the same group to authenticate each other *secretly*, in the sense that someone who is *not* a group member cannot tell, by engaging some party in the handshake protocol, whether that party is a member of this group. On the other hand, any two parties who *are* members of the same group will recognize each other as members. Thus, a secret handshake protocol can be used in any scenario where group members need to identify each other without revealing their group affiliations to outsiders.
The work of [BDS+03] constructed secret handshakes secure under the *Bilinear Diffie-Hellman* (BDH) assumption in the Random Oracle Model (ROM). We show how to build secret handshake protocols secure under a more standard cryptographic assumption of *Computational Diffie Hellman* (CDH), using a novel tool of *CA-oblivious* public key encryption, which is an encryption scheme s.t. neither the public key nor the ciphertext reveal any information about the Certification Authority (CA) which certified the public key. We construct such CA-oblivious encryption, and hence a handshake scheme, based on CDH (in ROM). The new scheme takes 3 communication rounds like the [BDS+03] scheme, but it is about twice cheaper computationally.

keywords: authentication, privacy, anonymity, encryption

1 Introduction

A secret handshake scheme, introduced by Balfanz et al. [BDS+03], 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 also a group member. For example, a CIA agent Alice might want to authenticate herself to Bob, but only if Bob is also a CIA agent. Moreover, if Bob is *not* a CIA agent, the protocol should not help Bob in determining whether Alice is a CIA agent or not. This secrecy property can be extended to ensure that group members' affiliations are revealed only to members who hold specific *roles* in the group. For example, Alice might want to authenticate herself as a CIA agent with security level one if *and only if* Bob is a CIA agent with security clearance two, and vice versa.

In other words, if A is a member of group G_a with role r_a and B is a member of G_b with role r_b, a secret handshake scheme guarantees the following [BDS+03]:

- A and B authenticate each other if and only if $G_a = G_b$.[3]
- If $G_a \neq G_b$ then both parties learn only *the sole fact* that $G_a \neq G_b$.
- A can choose not to reveal anything about herself unless B is a member with particular role r_b (and vice versa).[4]
- An eavesdropper or a man in the middle learn nothing from the protocol.

As observed in [BDS+03], secret handshakes seem to require new cryptographic protocols since they can not be easily obtained from existing tools in the "cryptographic toolbox". For example, group signatures [CVH91,ACJT00] might appear to be an attractive building block for secret handshakes. However, they offer anonymity and unlinkability of group members' signatures, not secrecy of membership itself. In the interactive variant of group signatures, called *identity escrow* [KP98], one party can prove to another its membership in a group in an anonymous fashion. However, what turns out to be quite difficult is the seemingly simple issue of two parties proving group membership to each other simultaneously, in such a way that one party never reveals its group membership to another unless the former is also a member of the same group.

Secret Handshake Scheme as a "CA-oblivious PKI". To be usable in practice, a secret handshake scheme must provide efficient revocation of any group member by the Group Authority (GA) which administers the group. To support this functionality we will consider secret handshake schemes which, like the scheme of [BDS+03], are similar to PKI's (Public Key Infrastructures), where the role of a group authority corresponds to that of a Certification Authority (CA) in a PKI. Namely, to become a member of a group a party needs the GA to issue a certificate on an ID bitstring which the CA agrees to assign to this party. The certificate must include a CA-specific *trapdoor* which corresponds to this ID.[5] To revoke some party, the CA puts that party's ID on a revocation list. To perform a handshake, two parties first exchange their ID's, and then proceed only if the ID of the other party is not on the revocation list of their CA. Since the secret handshake protocol must hide one's group affiliation from outsiders, the ID's will be random strings picked from the same domain by all the CA's.[6]

[3] However, as noted by [BDS+03], a handshake protocol cannot be *fair* in the sense that if $G_a = G_b$ then one party is going to learn about it first and could abort the protocol and thus withhold their group affiliation from the counterparty.

[4] To simplify the presentation, we will ignore roles for most of the paper. However, as we show in appendix A.1, they can be added easily.

[5] For example, in an identity based encryption scheme, the trapdoor is a secret key corresponding to the public key which can be recovered from ID and the public parameters associated with the CA. In a standard PKI system, this correspondence has an added level of indirection: The trapdoor t is a secret key corresponding to the public key PK which is in turn bound to the ID string by a signature of CA on the $(ID|PK)$ pair.

[6] To make protocol runs executed by the same party *unlinkable*, [BDS+03] propose that a single user gets multiple (ID,certificate) pairs, each to be used only once.

In this setting, constructing a secret handshake scheme amounts to solving the following protocol problem: For a given CA, Alice wants to prove to Bob that she possesses a trapdoor t_A issued by this CA on her ID_A, but only if Bob possesses a trapdoor t_B issued by *the same* CA on his ID_B (and vice versa). Moreover, the protocol must be "CA-oblivious" in the sense that if a cheating Bob is not in the group administered by a given CA, and hence does not hold a CA-specific trapdoor t_B associated with ID_B, then his interaction with Alice must not help him in guessing if Alice belongs to this group or not. (And vice versa for an honest Bob and a cheating Alice.) While this protocol problem can be solved in principle with general 2-party secure computation techniques, the issue remains whether it can be solved with a *practical* protocol, at a cost comparable to standard authentication protocols.

Existing Solutions Based on Bilinear Maps. The secret handshake protocol of [BDS$^+$03] is based on bilinear maps, which can be constructed using Weil pairings on elliptic curves [Jou02,Gag02]. The protocol of [BDS$^+$03] builds on the non-interactive key-agreement scheme of [SOK00], and works as follows. As in the identity based encryption scheme of [BF01], 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 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.

Thus, the "CA-obliviousness" property of the protocol of [BDS$^+$03] follows from two properties of cryptosystems built on bilinear maps: (1) that the receiver's public key can be recovered by the sender from the receiver's ID, and thus the receiver does not need to send any information revealing his CA affiliation to the sender, and (2) knowing their public keys, the two parties can establish a session key non-interactively, and thus they again do not reveal any CA-specific information. Given that the first property relies on identity based encryption, and that the only practical IBE known so far is based on bilinear maps [BF01], it seems that BDH is indeed needed for secret handshakes.

Our Contributions. In this paper we show that efficient secret handshake (SH) schemes can be built using weaker and more standard assumption than the BDH, namely the *Computational Diffie Hellman* (CDH) assumptions. However, our security arguments, just like those for the BDH-based scheme of [BDS$^+$03] remain in the so-called Random Oracle Model (ROM). Moreover, the proposed scheme is computationally at least twice cheaper than the scheme of [BDS$^+$03].

We show this in several steps: First, we generalize the IBE-based secret handshake solution sketched above by showing that an efficient *four-rounds* secret handshake protocol can be built using any *PKI-enabled* encryption with the additional property of *CA-obliviousness*. We define the notion of (chosen-*plaintext* secure) PKI-enabled encryption, which generalizes both the Identity Based Encryption schemes, and the standard encryption schemes used in the context of a PKI system like X.509. We define the CA-obliviousness property for this notion of PKI-enabled encryption, which requires that both the public-key-related information which the receiver provides to the sender, and the ciphertext sent from the sender to the receiver, do not reveal which CA issued the receiver's certificate. We then show that every CA-oblivious PKI-enabled encryption leads to a four-round secret handshake protocol whose cost is one decryption and one encryption for each party. We also show an alternative construction, which creates a three-round secret handshake protocol using any CA-oblivious PKI-enabled encryption equipped with the so-called zero-knowledge "signature of knowledge" [CS97] of the private decryption key.

Next, we combine ElGamal encryption and Schnorr signatures to construct a practical CA-oblivious PKI-enabled encryption secure under the CDH assumption (in ROM), which thus leads to a four-round secret handshake protocol secure under CDH. However, since this encryption admits a very practical (in ROM) ZK signature of knowledge of the private key, which is simply the Schnorr signature scheme itself, this results in a secret handshake scheme which takes three rounds, like the scheme of [BDS+03], and which involves one multiexponentiation and one or two exponentiations per player. Compared to the cost of the scheme of [BDS+03], where each player computes a pairing of two elements one of which is known in advance, this is about twice less expensive, according to the results of Barreto et al. [BKLS02].

We also improve the functionality of a secret handshake system by showing that our CDH-based SH schemes support "blinded" issuance of the member certificates in the sense that the CA does not learn the trapdoors included in the certificate, and thus, in contrast to the BDH-based SH scheme of [BDS+03], the CA cannot impersonate that member.

Finally, we note that the CA-oblivious encryption we devise can be also applied to provide a CDH-based solution to the *Hidden Credentials* problem [HBSO03], which generalizes the notion of secret handshakes to general PKI trust evaluations where two communicating partners are not necessarily certified by the same group/certification authority. This problem was also given only a BDH-based solution so far, in [HBSO03].

Related Work. As described in [BDS+03], existing anonymity tools such as anonymous credentials, group signatures, matchmaking protocols, or accumulators, have different goals than secret handshakes, and it is indeed unclear how to achieve a secret handshake scheme from any of them. Thus we will briefly discuss here only the new work of [LDB03], which proposes a new notion "oblivious signature-based envelopes", which is closely related to the secret handshake problem. The oblivious envelope notion they define is very similar to our notion of

PKI-enabled encryption, but with a weaker obliviousness property. Namely, they only require that the encrypting party does not know if the receiver possesses a CA-certified public/private key or not, but the protocol does not hide the identity of the CA itself from the receiver. In contrast, our CA-oblivious encryption notion requires the protocol to hide this identify. Thus, while our CA-oblivious encryption gives an oblivious signature-based envelope for Schnorr signatures, the other direction is not clear. In particular, it remains an open problem if CA-oblivious encryption and/or secret handshakes can be constructed based on the RSA assumption.[7]

Organization. In section 2 we revise the definitions of an SH scheme [BDS$^+$03], restricting them to "PKI-like" SH schemes we consider here. In section 3 we define the notion of a *PKI-enabled encryption*, and the *CA-obliviousness* property for such encryption. In section 4 we construct a CA-oblivious encryption secure under CDH in ROM. In section 5 we give two general constructions of SH schemes from any CA-oblivious encryption. In appendix A we show how to support roles and blinded issuing of CA certificates.

2 Definition of Secret Handshakes

We adapt the definition of a secure Secret Handshake [SH] scheme from [BDS$^+$03] to what we call "PKI-like" SH schemes. Our definitions might potentially restrict the notion of a secret handshake scheme, but both the SH scheme of [BDS$^+$03] and our SH schemes fall into this category. We define an SH scheme as a tuple of probabilistic algorithms Setup, $\mathsf{CreateGroup}$, $\mathsf{AddMember}$, and $\mathsf{Handshake}$ s.t.

- Setup is an algorithm executed publicly on the high-enough security parameter k, to generate the public parameters params common to all subsequently generated groups.
- $\mathsf{CreateGroup}$ is a key generation algorithm executed by a GA, which, on input of params, outputs the group public key G, and the GA's private key t_G.
- $\mathsf{AddMember}$ is a protocol executed between a group member and the GA on GA's input t_G and shared inputs: params, G, and the bitstring ID (called a *pseudonym* in [BDS$^+$03]) of size regulated by params. The group member's private output is the trapdoor t produced by GA for the above ID.
- $\mathsf{Handshake}$ is the authentication protocol, i.e. the SH protocol itself, executed between players A, B on public input ID_A, ID_B, and params. The private input of A is (t_A, G_A) and the private input of B is (t_B, G_B). The output of the protocol for either party is either a *reject* or *accept*.

We note that $\mathsf{AddMember}$ can be executed multiple times for the same group member, resulting in multiple (ID, t) authentication tokens for that member. We

[7] In the poster advertising the preliminary version of these results in PODC'04, we erroneously claimed that we know how to get RSA-based CA-oblivious encryption scheme, but this claim was incorrect, and this issue is still an open problem.

also note that in all the SH schemes discussed here the output of the $\mathsf{Handshake}$ protocol can be extended to include an authenticated session key along with the "accept" decision.

2.1 Basic Security Properties

An SH scheme must be complete, impersonator resistant, and detector resistant:[8]

Completeness. If honest members A, B of the same group run $\mathsf{Handshake}$ with valid trapdoors t_A, t_B generated for their ID strings ID_A, ID_B and for the same group $G_A = G_B$, then both parties output "accept".

Impersonator Resistance. Intuitively, the impersonator resistance property is violated if an honest party V who is a member of group G authenticates an adversary $\mathcal{A}$ as a group member, even though $\mathcal{A}$ **is not** a member of G. Formally, we say that an SH scheme is *impersonator resistant* if every polynomially bounded adversary $\mathcal{A}$ has negligible probability of winning in the following game, for *any* string ID_V which models the ID string of the victim in the impersonation attack:

1. We execute $\mathsf{params} \leftarrow \mathsf{Setup}(1^k)$, and $(G, t_G) \leftarrow \mathsf{CreateGroup}(\mathsf{params})$.
2. $\mathcal{A}$, on input (G, ID_V), invokes the $\mathsf{AddMember}$ algorithm on any number of group members ID_i of his choice. (The GA's inputs are ID_i's, G, and t_G.)
3. $\mathcal{A}$ announces a new $ID_{\mathcal{A}}$ string, different from all the ID_i's above. (This models a situation where the ID_i's belong to group members who are malicious but who might be revoked.)
4. $\mathcal{A}$ interacts with the honest player V in the $\mathsf{Handshake}$ protocol, on common inputs $(ID_{\mathcal{A}}, ID_V)$, and on V's private inputs G and t_V, where $t_V \leftarrow \mathsf{AddMember}((G, ID_V), t_G)$.

We say that $\mathcal{A}$ *wins* if V outputs "accept" in the above $\mathsf{Handshake}$ instance.

We note that the above impersonator resistance property is rather weak, and that stronger versions of this property are possible, and indeed advisable. Namely, the attacker *should* be allowed to run the protocol several times against V, and be able to ask for additional trapdoors after each attempt, before he announces that he is ready for the true challenge. Also, the attacker *could* be allowed to ask for trapdoors on additional $ID_i \neq ID_A$ strings during the challenge protocol with V. We adopt the simplest and weakest definition here to reduce the level of formalism in the paper. Nevertheless, we believe that our schemes remain secure under these stronger notions as well.

Remark: We note that even such strengthened notion of impostor resistance is not strong enough to be used in practice. For example, the resulting notion

[8] Once we restrict the notion of SH schemes to the PKI-like SH schemes, the security properties defined originally in [BDS+03] can be stated in a simpler way. Specifically, their properties of *impersonator resistance* and *impersonator tracing* are subsumed by our *impersonator resistance*, and their *detector resistance and tracing* is subsumed by what we call *detector resistance.*

makes no claims of security against the man in the middle attacks, and no claims if the adversary triggers a handshake protocol with an honest owner of the $ID_{\mathcal{A}}$ identity at any time *before* the adversary tries to authenticate himself to V under this identity. Therefore we do not claim that the above impostor resistance property is sufficient in practice. Instead, the above *authentication*-like notion of impostor resistance has to be first extended to *Authenticated Key Agreement* [AKE]. We discuss this further in the Section 2.2 below.

Detector Resistance. Intuitively, an adversary $\mathcal{A}$ violates the detector resistance property if it can decide whether some honest party V is a member of some group G, even though $\mathcal{A}$ **is not** a member of G. Formally, we say that an SH scheme is *detector resistant* if there exists a probabilistic polynomial-time algorithm SIM, s.t. any polynomially bounded adversary $\mathcal{A}$ cannot distinguish between the following two games with the probability which is non-negligibly higher than 1/2, for *any* target ID string ID_V:

Steps 1-3 proceed as in the definition of *Impersonator Resistance*, i.e. on input ID_V and a randomly generated G, $\mathcal{A}$ queries GA on adaptively chosen ID_i's and announces some challenge string $ID_{\mathcal{A}}$, $ID_{\mathcal{A}} \neq ID_i$ for all i.

4-1. In game 1, $\mathcal{A}$ interacts with an algorithm for the honest player V in the $\mathsf{Handshake}$ protocol, on common inputs $(ID_{\mathcal{A}}, ID_V)$, and on V's private inputs G and $t_V = \mathsf{AddMember}((G, ID_V), t_G)$.
4-2. In game 2, $\mathcal{A}$ interacts with SIM on common inputs $(ID_{\mathcal{A}}, ID_V)$.
5. $\mathcal{A}$ can query GA on additional strings $ID_i \neq ID_{\mathcal{A}}$.
6. $\mathcal{A}$ outputs "1" or "2", making a judgment about which game he saw.

Similarly to impersonator resistance, stronger notions of detector resistance are possible and indeed advisable. In particular, the adversary should be able to trigger several executions of the handshake protocol with player V, and he should be able to interleave these instances with instances executed with the rightful owner of the $ID_{\mathcal{A}}$ identity. We adopt the above weak notion for simplicity, but our schemes satisfy these stronger notion as well.

2.2 Extensions and Other Security Properties

Authenticated Key Exchange. As mentioned in the previous section, the impostor resistance property defined above is only a weak authentication-like property which does not give sufficient guarantees in practice. Moreover, in practice one would like to extend the notion of a secret handshake from one where participants' outputs are binary decisions "accept" / "reject", to authenticated key exchange, where parties output instead either "reject" or a secure session *key*. We believe that the SH schemes we propose, just like the original SH protocol of [BDS+03], can be easily extended to AKE protocols using the standard AKE protocol techniques. However, the formal security analysis of the resulting protocols requires adoption of AKE formalism [BR93,CK02,Sho99], which is beyond the scope of this paper.

Group-Affiliation Secrecy against Eavesdroppers. Our schemes also protect secrecy of participants' group affiliations against eavesdroppers, even if the eavesdropper is a malicious member of the same group. An observer of our SH protocols does not even learn if the participants belong to the same group or not. We do not formally define security against eavesdroppers, because it is very similar to the security against active attackers which we do define, the impersonator and detector resistance. Moreover, if the protocol participants first establish a secure anonymous session, e.g. using SSL or IKE, and then run the SH protocol over it, the resulting protocol is trivially secure against eavesdroppers.

Unlinkability. A potentially desirable property identified in [BDS$^+$03], is *unlinkability*, which extends privacy protection for group members by requiring that instances of the handshake protocol performed by the same party cannot be efficiently linked. This can be achieved trivially (but inefficiently) by issuing to each group member a list of one-time certificates, each issued on a randomly chosen ID, to be discarded after a single use. Unfortunately, an honest member's supply of one-time certificates can be depleted by an active attacker who initiates the handshake protocol enough times. Indeed, while one can run our SH schemes using multiple certificates to offer some heuristic protections against linking, constructing an efficient and perfectly unlinkable SH scheme remains an open problem.

3 Definition of PKI-enabled CA-oblivious Encryption

We define the notion of *PKI-enabled* encryption, which models the use of standard encryption in the context of a PKI system, and also generalizes Identity Based Encryption. We define *one-way security* for PKI-enabled encryption, adapting a standard (although weak) notion of one-way security of encryption to our context, and we define a novel *CA-obliviousness* property for such schemes.

A PKI-enabled encryption is defined by the following algorithms:

- $\mathsf{Initialize}$ is run on a high-enough security parameter, k, to generate the public parameters params common to all subsequently generated Certification Authorities (CAs).
- CAInit is a key generation algorithm executed by a CA. It takes as inputs the system parameters params and returns the public key G and the private key t_G of the CA.
- $\mathsf{Certify}$ is a protocol executed between a CA and a user who needs to be certified by this CA. It takes CA's private input t_G, and public inputs G (assume that G encodes params) and string ID which identifies the user, and returns *trapdoor* t and *certificate* ω as the user's outputs.
- $\mathsf{Recover}$ is an algorithm used by a *sender*, a party who wants to send an encrypted message to a user identified by some string ID, to recover that user's public key. It takes inputs (G, ID, ω) and outputs a public key PK.
- Enc is the actual encryption algorithm which takes inputs message m and the public key PK (assume that PK encodes params and G), and outputs a ciphertext c.

– Dec is the decryption algorithm which takes as inputs the ciphertext c and the trapdoor t (as well as possibly params, G, ID, and ω, all of which can be encoded in t), and returns m.

The above algorithms must satisfy the obvious *correctness* property that the decryption procedure always inverts encryption correctly.

It is easy to see (see footnote 5) that this notion of encryption indeed models both regular encryption schemes in the PKI context as well as the Identity Based encryption schemes.

One-Way Security. We define the security of PKI-enabled encryption only in the relatively weak sense of so-called *one-way* security, namely that the attacker who does not own a trapdoor for some public key cannot decrypt an encryption of a random message. This is a weaker notion than the standard *semantic* security for an encryption, but we adopt it here because (1) it simplifies the definition of security, (2) one-way security is all we need in our construction of a secure SH scheme, and (3) in the Random Oracle Model, it is always possible to convert a one-way secure encryption into a semantically secure encryption, or even a CCA-secure encryption using the method of Fujisaki and Okamoto [FO99].

The definition of security for PKI-enabled encryption is very similar to the definition of security of an IBE scheme: We say that a PKI-enabled encryption scheme is *One-Way* (OW) secure on message space $\mathcal{M}$ under *Chosen-Plaintext Attack* (CPA), if every polynomially-bounded adversary $\mathcal{A}$ has only negligible probability of winning the following game:

1. The $\mathsf{Initialize}$ and CAInit algorithms are run, and the resulting public key G is given to $\mathcal{A}$.
2. $\mathcal{A}$ repeatedly triggers the $\mathsf{Certify}$ protocol under the public key G, on ID strings ID_i of $\mathcal{A}$'s choice. In each instance $\mathcal{A}$ receives (t_i, ω_i) from the CA.
3. $\mathcal{A}$ announces a pair $(ID_{\mathcal{A}}, \omega)$, where $ID_{\mathcal{A}} \neq ID_i$ for all ID_i's queried above.
4. $\mathcal{A}$ receives $c = \mathsf{Enc}_{PK}(m)$ for a random message $m \in \mathcal{M}$ and $PK = \mathsf{Recover}(G, ID_{\mathcal{A}}, \omega)$.
5. $\mathcal{A}$ is allowed to trigger the $\mathsf{Certify}$ algorithm on new $ID_i \neq ID_{\mathcal{A}}$ strings of his choice, getting additional (t_i, ω_i) pairs from the CA.
6. $\mathcal{A}$ outputs a message m'. If $m' = m$ then we say that $\mathcal{A}$ *wins*.

CA-Obliviousness. Informally, PKI-enabled encryption is CA-oblivious if (1) the receiver's message to the sender, i.e., the pair (ID, ω), hides the identity of the CA which certified this ID; and (2) the sender's messages to the receiver, i.e., ciphertexts, do not leak any information about the CA which the *sender* assumed in computing the receiver's public key. Consequently, in a standard exchange of messages between the receiver and the sender, neither party can guess which CA is assumed by the other one. Formally, we call a PKI-enabled encryption scheme *CA-oblivious* under two conditions:

(I) It is *"Receiver CA-oblivious"*, i.e., if there exists a probabilistic polynomial-time algorithm $SIM_{(R)}$, s.t. no polynomially-bounded adversary $\mathcal{A}$ can distinguish between the following two games with probability non-negligibly higher than 1/2, for *any* target ID string ID_R:

1. The $\mathsf{Initialize}$ and CAInit algorithms are executed, and the resulting parameters params and the public key G is given to $\mathcal{A}$.
2. $\mathcal{A}$ can trigger the $\mathsf{Certify}$ protocol on any number of ID_i's.

3-1. In game 1, $\mathcal{A}$ gets (ID_R, ω_R), where ω_R is output by the $\mathsf{Certify}$ protocol on G and ID_R.

3-2. In game 2, $\mathcal{A}$ gets (ID_R, r) where $r = SIM_{(R)}(\mathsf{params})$.

4. $\mathcal{A}$ can trigger the $\mathsf{Certify}$ protocol some more on any $ID_i \neq ID_R$.
5. $\mathcal{A}$ outputs "1" or "2", making a judgment about which game he saw.

(II) It is *"Sender CA-oblivious"*, i.e., if there exists a probabilistic polynomial-time algorithm $SIM_{(S)}$ s.t. no polynomially-bounded adversary $\mathcal{A}$ can distinguish between the following two games, with probability non-negligibly higher than 1/2:

1. The $\mathsf{Initialize}$ and CAInit algorithms are executed, and the resulting parameters params and the public key G is given to $\mathcal{A}$.
2. $\mathcal{A}$ can trigger the $\mathsf{Certify}$ protocol any number of times, for public key G and group members ID_i's of $\mathcal{A}$'s choice.
3. $\mathcal{A}$ announces pair (ID_R, ω_R) on which he wants to be tested, where $ID_R \neq ID_i$ for all i.

4-1. In game 1, $\mathcal{A}$ gets $c = \mathsf{Enc}_{PK_R}(m)$ for random $m \in \mathcal{M}$ and $PK_R = \mathsf{Recover}(G, ID_R, \omega_R)$.

4-2. In game 2, $\mathcal{A}$ gets $c = SIM_{(S)}(\mathsf{params})$.

5. $\mathcal{A}$ can query GA on some more ID_i's s.t. $\forall_i, ID_i \neq ID_R$.
6. $\mathcal{A}$ outputs "1" or "2", making a judgment about which game he saw.

4 Construction of CA-Oblivious Encryption

We construct a CA-oblivious PKI-enabled encryption scheme secure based on the CDH assumption in the Random Oracle Model.[9]

- $\mathsf{Initialize}$ picks the standard discrete logarithm parameters (p, q, g) of security k, i.e., primes p, q of size polynomial in k, s.t. g is a generator of a subgroup in $\mathbb{Z}_p^*$ of order q. $\mathsf{Initialize}$ also defines hash functions $H : \{0,1\}^* \rightarrow \mathbb{Z}_q$ and $H' : \{0,1\}^* \rightarrow \{0,1\}^k$. (Both hash functions are modeled as random oracles, but we note that H' is not essential in this construction and can be easily removed.)
- CAInit picks random private key $x \in \mathbb{Z}_q$ and public key $y = g^x \mathrm{mod} p$.
- In $\mathsf{Certify}$ on public inputs (y, ID), the CA computes the Schnorr signature on string ID under the key y [Sch89], i.e., a pair $(\omega, t) \in (\mathbb{Z}_p^*, \mathbb{Z}_q)$ s.t. $g^t = \omega y^{H(\omega, ID)} \bmod p$. The user's outputs are the trapdoor t and the certificate ω. The signature is computed as $\omega = g^r \bmod p$, and $t = r + xH(\omega, ID) \bmod q$, for random $r \leftarrow \mathbb{Z}_q$.

[9] We remark that since the Identity Based Encryption scheme of [BF01] is also a CA-oblivious PKI-based encryption scheme, the SH construction of Section 5 applied to that encryption scheme implies efficient BDH-based SH schemes.

– $\mathsf{Recover}(y, ID, \omega)$ outputs $PK = \omega y^{H(\omega,ID)} \bmod p$.
– $\mathsf{Enc}_{PK}(m)$ is an ElGamal encryption of message $m \in \{0,1\}^k$ under the public key PK: It outputs a ciphertext $[c_1, c_2] = [g^r \bmod p, m \oplus H'(PK^r \bmod p)]$, for random $r \in \mathbb{Z}_q$.
– Dec is an ElGamal decryption, outputing $m = c_2 \oplus H'(c_1^t \bmod p)$.

Theorem 1. *The above encryption scheme is CA-oblivious and One-Way secure under the CDH assumption in the Random Oracle Model.*

Proof (of One-Way Security). Assume that an adversary $\mathcal{A}$ breaks one-wayness of this encryption scheme. This means that after receiving n Schnorr signatures (t_i, ω_i) on ID_i's of his choice, $\mathcal{A}$ sends a tuple (ID, ω) s.t. $ID \neq ID_i$ for all the above ID_i's, and (in ROM), to break one-wayness $\mathcal{A}$ must query the H' oracle on $c_1^t \bmod p$ where $g^t = \omega y^{H(\omega,ID)} \bmod p$. Therefore, $\mathcal{A}$ must exponentiate a random element c_1 it received to the exponent t. Hence, what we need to argue that, even though $\mathcal{A}$ receives n signatures (t_i, ω_i) on her ID_i's, she cannot produce a new pair (ID, ω) s.t. she can exponentiate a random elements c_1 to exponent t where $g^t = \omega * y^{h(\omega,ID)}$. Now, this is very similar to proving the chosen message attack security of the underlying Schnorr signature scheme, where one argues that, after receiving n signatures, $\mathcal{A}$ cannot produce a new triple (ID, ω, t) s.t. $g^t = \omega * y^{h(\omega,ID)}$. Hence, our proof is very similar to the forking-lemma proof for Schnorr signature security in [PS96]. However, here we reduce the successful attack not to computing discrete logarithm, but to breaking the CDH assumption by computing m^x on input $y = g^x$ and a random value m.

To reduce $\mathcal{A}$'s ability to succeed in this protocol to computing m^x on the Diffie-Hellman challenge (g, g^x, m), we first simulate, as in the proof of Schnorr signature security, the signatures (t_i, ω_i) that $\mathcal{A}$ gets on her ID_i's, by taking random t_i, c_i, computing $\omega_i = g^{t_i} * y^{-c_i} \bmod p$, and assigning $H(\omega_i, ID_i)$ to c_i. Since the verification equation is satisfied and t_i, c_i are picked at random, this is indistinguishable from receiving real signatures. Then, as in the forking lemma argument of [PS96], we can argue that if $\mathcal{A}$'s probability of success is ϵ, the probability that $\mathcal{A}$ executed twice in a row succeeds in *both* executions *and* sends the *same* (ID, ω) challenge in both of them, is at least ϵ^2/q_h where q_h is the number of queries $\mathcal{A}$ makes to the hash function H (see [PS96]). The forking lemma used in the security proof of the Schnorr signature scheme shows that if two conversations with an adversary produce triples (t, ω, ID) and (t', ω, ID), where in first conversation $H(\omega, ID) = c$ and in the second $H(\omega, ID) = c'$ for some random c, c', then $x = DL_g(y)$ can be computed as $x = (s-s')/(c-c') \bmod q$, because $g^t = \omega * y^c$ and $g^{t'} = \omega * y^{c'}$. By applying the same forking lemma to our case, adversary $\mathcal{A}$ produces two *exponentiations* m^t and $m^{t'}$, instead of forgeries t, t', but still we have that $x = DL_g(y) = (t - t')/(c - c')$. Therefore, with probability ϵ^2/q_h we can break the CDH challenge and compute $m^x = m^{(t-t')/(c-c')} = (m^t/m^{t'})^{1/(c-c')} \bmod p$.

Note that if the success probability ϵ is higher than negligible, and if $\mathcal{A}^*$ is an efficient algorithm and hence the number of queries q_h is polynomial, then the probability of CDH break ϵ^2/q_h is non-negligible as well.

Proof (of CA-Obliviousness). It is easy to see that neither ω nor the ciphertext $C = [c_1, c_2]$ reveal any information about the CA: Since $\omega = g^r$ for random r, ω is independent from CA's public key y, and hence the scheme is receiver CA-oblivious. Ciphertext $C = [c_1, c_2]$ on a random message m is also independent from the group key y, because $c_1 = g^r$ for random r and c_2 is computed by xoring $H'(PK^r)$ with the random m.

5 Secret Handshakes from CA-Oblivious Encryption

We first show how to built a secure four-rounds SH scheme using CA-oblivious PKI-enabled encryption. Given a CA-oblivious one-way secure PKI-enabled encryption scheme (Initialize, CAInit, Certify, Recover, Enc, Dec), and a hash function $H : \{0,1\}^* \to \{0,1\}^k$ modeled as a random oracle, we specify a secret handshake scheme as follows: Algorithms Setup, CreateGroup, and AddMember, are simply set to Initialize, CAInit, and Certify, respectively, while algorithm Handshake proceeds as follows. A's inputs are (ID_a, ω_a, t_a) and B's inputs are (ID_b, ω_b, t_b).[10]

1. $(B \longrightarrow A)$: $ID_b,\ \omega_b$
   - A obtains $PK_b = \mathsf{Recover}(G, ID_b, \omega_b)$
   - A picks $r_a \leftarrow \mathcal{M}$ and $ch_a \leftarrow \{0,1\}^k$
   - A computes $C_a = \mathsf{Enc}_{PK_b}(r_a)$
2. $(A \longrightarrow B)$: $ID_a,\ \omega_a,\ C_a,\ ch_a$
   - B obtains $PK_a = \mathsf{Recover}(G, ID_a, \omega_a)$
   - B obtains $r_a = \mathsf{Dec}_{t_b}(C_a)$
   - B picks $r_b \leftarrow \mathcal{M}$ and $ch_b \leftarrow \{0,1\}^k$
   - B computes $C_b = \mathsf{Enc}_{PK_a}(r_b)$
   - B computes $resp_b = H(r_a, r_b, ch_a)$
3. $(B \longrightarrow A)$: $C_b,\ resp_b,\ ch_b$
   - A obtains $r_b = \mathsf{Dec}_{t_a}(C_b)$
   - if $resp_b \neq H(r_a, r_b, ch_a)$, A outputs FAIL; otherwise A outputs ACCEPT.
   - A computes $resp_a = H(r_a, r_b, ch_b)$
4. $(A \longrightarrow B)$: $resp_a$
   - if $resp_a \neq H(r_a, r_b, ch_b)$, B outputs FAIL; otherwise B outputs ACCEPT.

We note that the above protocol can be easily turned into an Authenticated Key Exchange (AKE) protocol (secure in the ROM model) if the two parties compute their authenticated session key as $K = H(r_a, r_b)$.

Theorem 2. *If the PKI-enabled encryption is CA-oblivious and One-Way secure, the above construction yields a Secret Handshake scheme secure in the Random Oracle Model (ROM).*

[10] Group member's trapdoor on string ID in this SH scheme is a *pair* (ω, t) produced by the Certify protocol. We can also assume that (ID_a, ID_b) are public inputs.

Proof (of Impersonator Resistance). Assume that $\mathcal{A}$ violates with non-negligible probability ϵ the impersonator resistance property against some honest member V identified by ID_V. Assume that $\mathcal{A}$ plays the role of A and V plays the role of B (the other case is easier because B has to speak first). Therefore with prob. ϵ, $\mathcal{A}$ sends a valid $resp_a = H(r_a, r_b, ch_b)$ response to B. In the ROM model, that can happen with non-negligible probability only if $\mathcal{A}$ querries the oracle for $H(\cdot)$ on the input (r_a, r_b, ch_b) s.t., in particular, r_b was the value picked by V and sent to $\mathcal{A}$ in the form of a ciphertext $C_b = \mathsf{Enc}_{PK_a}(r_b)$ for $PK_a = \mathsf{Recover}(G, ID_a, \omega_a)$, where (ID_a, ω_a) are sent by $\mathcal{A}$ in its first message to V. Therefore, in ROM, we can use $\mathcal{A}$ to create a break $\mathcal{A}'$ against the one-way security of the encryption scheme:

On input G, $\mathcal{A}'$ passes the public key G to $\mathcal{A}$. When $\mathcal{A}$ can makes a querry ID_i, so does $\mathcal{A}'$, passing back (ω_i, t_i) to $\mathcal{A}$. When $\mathcal{A}$ announces that he is ready for the impersonation challenge against V, $\mathcal{A}'$ passes as *his* encryption challenge the pair (ID_a, ω_a) sent by $\mathcal{A}$ in his first message to V. On encryption challenge $c = \mathsf{Enc}_{PK_a}(m)$ where m is chosen at random in $\mathcal{M}$, $\mathcal{A}'$ passes the same challenge as its response $C_b = c$ to $\mathcal{A}$, together with a random challenge value ch_b and $resp_b$ picked at random. The only way $\mathcal{A}$ can tell between this communication and a conversation with an honest V is by querying H on (r_a, r_b, ch_a) for $r_b = \mathsf{Dec}_{t_a}(C_b) = m$. Otherwise, as we argued above, he queries H on (r_a, r_b, ch_b) with probability almost ϵ. In either case, since $\mathcal{A}$ can make only polynomially-many queries to H, $\mathcal{A}'$ can pick one such query at random, and $\mathcal{A}'$ will have a non-negligible chance of outputing $r_b = m$. Thus $\mathcal{A}'$ breaks the one-wayness of the encryption scheme.

Proof (of Detector Resistance). We will show a simulator SIM s.t. if $\mathcal{A}$ distinguishes between interactions with SIM and interactions with a group member, we can break the one-way security of the encryption scheme. Assume again that the adversary $\mathcal{A}$ plays the role of A and V plays the role of B. Assume that the underlying encryption scheme is CA-oblivious, and therefore there exist simulators $SIM_{(S)}$ and $SIM_{(R)}$ which satisfy the two CA-obliviousness criteria. We define a simulator SIM, running on input $(ID_A, ID_V, \mathsf{params})$, as follows: (1) To simulate V's first message SH-1, SIM sends $ID_b = ID_V$ together with $\omega_b = SIM_{(R)}(\mathsf{params})$, (2) To simulate B's second message SH-3, SIM sends $resp_b$ and ch_b picked at random, and $C_b = SIM_{(S)}(\mathsf{params})$.

If $\mathcal{A}$ can distinguish a conversation with such SIM from a conversation with a true group member V, then by a standard hybrid argument, since the $SIM_{(S)}$ and $SIM_{(R)}$ simulators produce messages which are indistinguishable from the messages of an honest B, it must be that $\mathcal{A}$ distinguishes random values $resp_b$ chosen by SIM from values $resp_b = H(r_a, r_b, ch_a)$ computed by a real player. But this can happen only if $\mathcal{A}$ makes an oracle query on the triple (r_a, r_b, ch_a), in which case we can use $\mathcal{A}$, exactly in the same manner as we did in the proof of impersonator resistance, to attack the one-way security of the underlying encryption scheme.

5.1 Three-Round Secret Handshake Scheme

We can eliminate one communication round in the above protocol using the zero-knowledge signature of knowledge [CS97] of the trapdoor t that corresponds to the public key $PK = \mathsf{Recover}(G, ID, \omega)$, which we will denote $\mathsf{sig}_t(m)$. One can easily construct such signatures in ROM if this relation admits a 3-round honest-verifier special-soundness proof system [CS97]. The protocol proceeds as follows, using the same notation as above:

1. $(B \longrightarrow A)$: (ID_b, ω_b, ch_b)
   – A computes $PK_b = \mathsf{Recover}(G, ID_b, \omega_b)$ and $c = \mathsf{Enc}_{PK_b}(r_a, \mathsf{sig}_{t_a}(ch_b))$
2. $(A \longrightarrow B)$: $(ID_a, \omega_a, cha_a, c)$
   – B accepts if c decrypts to (r_a, sig) where sig verifies as a signature on ch_b under the public key $PK_a = \mathsf{Recover}(G, ID_a, \omega_a)$
3. $(B \longrightarrow A)$: $resp_b = H(r_a, ch_a)$
   – A accepts if $resp_b = H(r_a, ch_a)$

In the case of the CDH-based encryption of Section 4, the above signature of knowledge is simply a Schnorr signature, and the resulting computational cost is one or two exponentiation and one multiexponentiation per player.

References

[ACJT00] G. Ateniese, J. Camenisch, M. Joye, and G. Tsudik. A practical and provably secure coalition-resistant group signature scheme. In *CRYPTO'2000*, 2000.

[BDS+03] 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.

[BF01] D. Boneh and M. Franklin. Identity based encryption from weil pairing. In *Advances in Cryptography - CRYPT0 2001*, Santa Barbara, CA, August 2001.

[BKLS02] P. Barreto, H. Kim, B. Lynn, and M. Scott. Efficient algorithms for pairing-based cryptosystems. In *Advances in Cryptography - CRYPT0 2002*, pages 354–368, Santa Barbara, CA, August 2002.

[BR93] M. Bellare and P. Rogaway. Entity authentication and key distribution. In *Advances in Cryptology - CRYPTO'93*, 1993.

[CK02] R. Canetti and H. Krawczyk. Universally composable notions of key exchange and secure channels. In *Advances in Cryptology - EUROCRYPT 2002*, 2002.

[CS97] Jan Camenisch and Markus Stadler. Proof systems for general statements about discrete logarithms. Technical Report TR 260, ETH Zurich, 1997.

[CVH91] D. Chaum and E. Van Heyst. Group signatures. In Springer-Verlag, editor, *Advances in Cryptology - EUROCRYPT'91*, volume 547, pages 257–265, 1991.

[FO99] E. Fujisaki and T. Okamoto. Secure integration of asymmetric and symmetric encryption schemes. In *Advances in Cryptology-CRYPTO'99*, pages 537–554, August 1999.

[Gag02] Martin Gagne. Applications of bilinear maps in cryptography. Master's thesis, University of Waterloo, 2002.

[HBSO03] J. Holt, R. Bradshaw, K. E. Seamons, and H. Orman. Hidden credentials. In *2nd ACM Workshop on Privacy in the Electronic Society*, October 2003.

[Jou02] A. Joux. The weil and tate pairings as building blocks for public key cryptosystems. In *Proceedings of the 5th International Symposium on Algorithmic Number Theory*, 2002.

[KP98] J. Kilian and E. Petrank. Identity escrow. In *Advances in Cryptography - CRYPT0 1998*, Santa Barbara, CA, August 1998.

[LDB03] N. Li, W. Du, and D. Boneh. Oblivious signature-based enevelope. In *Proceedings of 22nd ACM Symposium on Principles of Distributed Computing (PODC 2003)*, Boston, Massachusetts, July 13-16 2003.

[PS96] D. Pointcheval and J. Stern. Security proofs for signatures. In *Eurocrypt'96*, pages 387 – 398, 1996.

[Sch89] C. Schnorr. Efficient identification and signatures for smart cards. In *Advances in Cryptography - CRYPT0 1989*, Santa Barbara, CA, August 1989.

[Sho99] V. Shoup. On formal models for secure key exchange. Technical Report RZ3120, IBM, April 1999.

[SOK00] R. Sakai, K. Ohgishi, and M. Kasahara. Cryptosystems based on pairing. In *Symposium in Cryptography and Information Security*, Okinawa, Japan, January 2000.

A Achieving Additional Properties

A.1 Roles

Our schemes can easily be extended to handle group member roles (as in the SH scheme of [BDS+03]), in a way that a member can choose not to reveal anything about herself unless the other party is a member with a particular role r (and vice versa). This functionality can be provided by modifiying the AddMember and Recover procedures as follows:

- AddMember: takes as inputs $params$, G, t_G and an arbitrary string $ID \in \{0,1\}^*$ and returns (ω, t) where t is a trapdoor and ω is a public parameter. (ω, t) are constructed using the string $ID|r$ (instead of ID as in the original procedure), where r is the role that the CA is assigning to the user.
- Recover: takes as input $params$, G, ID and ω (provided by another user B). It outputs a public key PK using as input $ID|r$ (instead of ID as in the original Recover procedure), where r is the role that $\mathcal{A}$ chooses to have a secret hanshake with.

A.2 Trapdoor Secrecy

Since CA computes the user's trapdoor t, it can impersonate that user. Would that be problematic, AddMember can easily be modified to blind the trapdoor if in the AddMember protocol the user supplies the CA with $b = g^\delta \bmod p$, where δ is the user's temporary secret. The CA can then reply with $\omega = g^k * b \bmod p$, where k is a random value in $\mathbb{Z}_q$, and $t' = k + H(\omega, ID) * t_G \bmod q$, and the user computes his trapdoor as $t = t' + \delta \bmod q$.