Reconciling CA-Oblivious Encryption, Hidden Credentials, OSBE, and Secret Handshakes *

Internet Security Research Lab Technical Report 2006-5

Jason E. Holt and Kent E. Seamons

Internet Security Research Lab
Brigham Young University
seamons@cs.byu.edu

June 2006

Abstract

We compare the abstract specifications of four similar systems with similar capabilities, and give transformations which allow any implementation of a particular system to transform into certain other systems. This clarifies the relationships between the systems, which have often been casually grouped together in the literature.

1 Introduction

In 2003, three separate credential systems were introduced which have very similar capabilities. Most notably, they allow credential contents to be used directly in access control processes, leading to systems in which credentials can be used without ever being disclosed.

The first system proposed was called Secret Handshakes [2], and described a key agreement protocol useful for resolving policy cycles and maintaining privacy against anonymous peers on a network. Then came Oblivious Signature Based Envelopes (OSBE) [14], which allows messages to be encrypted against a certificate's signature. The signature itself serves as the credential, and needs never be disclosed to the message sender. Hidden Credentials [11] were introduced next, allowing messages to be encrypted against complex policies, protecting policies from leaking to unqualified recipients and allowing recipients to use combinations of credentials without even acknowledging their existance. A fourth scheme, CA-Oblivious Encryption, was proposed in 2004 [7], offering improved performance for Secret Handshakes and claiming in passing to be an alternative implementation of OSBE and Hidden Credentials as well. All four schemes give proofs of security in the random oracle model (ROM).

Traditional access control systems typically begin with a client who requests access to a resource. The client and server demonstrate ownership of credentials in order to satisfy implicit or explicit access control policies on each side, and if both are ultimately satisfied, the server sends the resource to the client. Each party may choose to withhold more sensitive credentials until an adequate level of trust has been established through the exchange of less sensitive credentials, but ultimately one party must be the first to reveal a credential to the other as-yet untrusted party.

In three of the related systems considered here, resources can be encrypted in such a way that the client gains access to the resource using her credentials, and does so without revealing to the server what her credentials are or whether she was successful in recovering the resource. The remaining system, Secret Handshakes, instead defines a key agreement protocol in which two parties mutually authenticate in such a

*This research was supported by funding from the National Science Foundation under grant no. CCR-0325951 and prime cooperative agreement no. IIS-0331707, and The Regents of the University of California.

way that a failed authentication reveals to neither party whether the other party had the required credential. In each of the systems, credentials need never be implicitly or explicitly revealed to unqualified parties.

All four systems can be implemented using pairing-based cryptography, a recent trend in cryptography which has facilitated construction of several interesting new constructs, most notably Identity-Based Encryption (IBE), first proposed by Shamir in 1984, but not successfully implemented until 2001.

A flurry of papers have been written in this new vein of research, most of which cite all three systems as related work. However, several have missed subtle but significant differences between them. For instance, there are a number of problems inherent in translating any of the other schemes into Hidden Credentials, as we examine in section 4.1.1.

In this paper, we examine each system individually (in alphabetical order), discuss its relation to each of the others, and in several cases detail previously unexplored ambiguities and incompatibilities. The concrete implementations of the systems offer a variety of interesting features which we attempt to briefly summarize. However, our focus is on building transformations between the systems based only on their abstract requirements.

The paper proceeds as follows:

- Section 2 discusses related work.
- Section 3 summarizes the abstract requirements of each system, discussing ambiguities in their specifications as well as possible ways of strengthening their security properties.
- Section 4 examines compatibility between the abstract requirements of each system, showing how implementations of one system can be used to implement other systems.

2 Related Work

Sending Bob a message which he can only read if he has a certain attribute is easy if all people who have that attribute share a common secret. For example, Bob could distribute an RSA private key to all the members of "Bob's Club." Alice could then send messages to club members without being a member herself. However, widely shared secrets tend to leak. The systems we consider here avoid this drawback by allowing Alice to use a public key which depends upon Bob's attribute *and his identity.* If Bob's secret is compromised, only messages to Bob become vulnerable.

Chaum was the first to propose privacy protecting digital credentials [8]. Brands [5] and others [6] later proposed systems with additional privacy protecting measures. These systems define a CA which blindly issues credentials to users, preventing the CA from colluding with the services that accept credentials issued by the CA to compile dossiers on system users. More advanced features include the ability of a user to prove satisfaction of a service's complex policy without revealing the particular satisfying set of credentials used, and the ability to prove attributes to multiple parties in an unlinkable way, preventing those parties from colluding among themselves to compile dossiers on specific users.

While the new systems we consider here offer fascinating new features not supplied by traditional private credential systems, it is interesting to note that none of the new systems was proposed with a blind issuing mechanism. Paradoxically, then, despite providing such interesting privacy features, most of the systems described here don't even allow users to generate their own private keys; credentials are issued and potentially logged by the Certifying Authorities (CAs), who have the ability to impersonate any user and eavesdrop on any transaction. Recent work by Xu and Yung [18] describes how to achieve unlinkable multi-show for the Secret Handshakes scheme, and Castelluccia et al. [7] briefly mentions a blind issuing scheme in Appendix C.2, but in general, the new schemes we describe lack several features taken for granted in traditional private credential schemes.

3 System Overviews

The most interesting common feature of the systems described here is their ability to integrate encryption with access control. Whereas traditional access control systems work by using cryptography to prove attribute values to other parties in order to enable release of a resource, such as opening a door or delivering a document, these systems work by making the attribute values themselves the keys to the service. This turns the tables in the honest users' favor, obviating conundrums about which party should have to be the first to disclose attributes, resolving policy deadlocks, and reducing both the cryptographic proofs and implicit

acknowledgements which must be entrusted to external, potentially untrustworthy parties with whom they nonetheless need to accomplish transactions.

In the following subsections, we describe the abstract requirements of each system, in several cases noting ambiguities and offering strengthened alternative properties. We also describe features offered by particular concrete implementations as space allows. Note that throughout this paper, we use variable naming conventions from the CA-Oblivious Encryption specification.

Table 1 gives condensed representations of each system's abstract definitions.

3.1 CA-Oblivious Encryption

Castellucia, Jarecki and Tsudik [7] define CA-Oblivious schemes in terms of what they call PKI-enabled cryptosystems, which are defined in terms of five functions. An *Initialize* routine sets up global parameters. *CAInit* establishes CA public and private values. *Certify* is used by CAs to issue a public certificate ω and secret trapdoor t corresponding to any string it wishes to certify. Message recipients provide ω along with an ID to message senders, who pass this value to *Recover*. *Recover* returns the public key PK required by encryption function *Enc*. The recipient then passes her secret value t and the ciphertext to *Dec* to recover the sender's message. PKI-enabled cryptosystems are required to have **One-Way Security**, in which attackers cannot decrypt encryptions of random plaintexts.

For a PKI-enabled cryptosystem to be CA-Oblivious, it must be both **Sender Oblivious** and **Receiver Oblivious**. Receiver obliviousness is intended to ensure that users can safely release their ω values without leaking information about which CAs issued their credentials. Sender obliviousness is intended to ensure that unqualified recipients cannot distinguish valid messages encrypted against a particular CA from messages encrypted against any other CA. Both properties are defined in terms of games in which an adversary tries to distinguish a message encrypted against a particular ω value or identity from a fake alternative.

Formally, [7] defines the receiver obliviousness game as follows. An adversary $\mathcal{A}$ tries to gain a significant advantage over random guessing in distinguishing between ω values issued by *Issue* and a simulated issuing process denoted $SIM_{(R)}$ for all possible identities ID_R. See table 1 for variable descriptions.

1. *Initialize* and *CAInit* are executed, and $\mathcal{A}$ is given the resulting *params* and G.
2. $\mathcal{A}$ receives as many (ID_i, ω_i) pairs as it wants for ID values of its choosing.
3. $\mathcal{A}$ receives one of the following values:
   (a) (ID_R, ω_R), where ω_R is the value returned by $Certify$ for some ID ID_R.
   (b) (ID_R, r), where r is the value returned by $SIM_{(R)}(params)$.
4. $\mathcal{A}$ receives any other (ID_i, ω_i) values it wants, such that $ID_i \neq ID_R$.
5. $\mathcal{A}$ wins if it correctly guesses which of the two values it received.

The game defined for sender obliviousness is similar to the previous game. In this game, the adversary is allowed to choose a challenge ID, ω pair and then must distinguish a random message encrypted against that pair from a bogus message generated by $SIM_{(S)}$.

1. *Initialize* and *CAInit* are executed, and $\mathcal{A}$ is given the resulting *params* and G.
2. $\mathcal{A}$ receives as many $\langle t_i, \omega_i \rangle$ pairs as it wants corresponding to ID values of its choosing.
3. $\mathcal{A}$ decides on a challenge ID $\langle ID_R, \omega_R \rangle$.
4. $\mathcal{A}$ receives one of the following values:
   (a) $C = Enc(M, PK)$, for a random message M, and $PK = Recover(G, ID_R, \omega_R)$.
   (b) C', a bogus ciphertext returned by $SIM_{(S)}(params)$.
5. $\mathcal{A}$ receives any other $\langle t_i, \omega_i \rangle$ values it wants, such that $ID_i \neq ID_R$.
6. $\mathcal{A}$ wins if it correctly guesses which value it received.

CA-Oblivious	
Setup:	$params = Initialize(k)$; $\langle G, t_G \rangle = CAInit(params)$
Issuing:	$\langle t, \omega \rangle = Certify(G, t_G, ID)$
Encryption:	$PK = Recover(G, ID, \omega)$; $C = Enc(M, PK)$
Decryption:	$M = Dec(C, t)$
Properties:	Sender & Receiver Obliviousness, One-Way Security

Hidden Credentials	
Setup:	$\langle G, t_G \rangle = CreateCA(params)$
Issuing:	$t = Issue(ID, attr)$
Encryption:	$C = Encrypt(M, ID, \mathcal{P})$
Decryption:	$M = Decrypt(C, \langle t_1, t_2, ... \rangle)$
Properties:	Credential & Policy Indistinguishability, CCA for *Encrypt*

OSBE	
Setup:	$\langle params, G, t_G, t \rangle = Setup(k, ID, M)$ Party $R1$ gets t, party S gets M, all get ID
Interaction:	S interacts with $R1$ or $R2$
Open:	$R1$ outputs M iff S interacted with $R1$ $R2$ outputs nothing
Properties:	Sound, Oblivious, Semantically Secure Against the Receiver

Secret Handshakes	
Setup:	$\langle G, t_G \rangle = SH.CreateGroup(Group)$
Issuing:	$t = SH.AddUser(ID, G, t_G)$
Handshake:	$SHS.Handshake(A, B)$ informs A, B of group membership only if both are members
Traitor Trace:	$U = SHS.TraceUser(T)$ reveals user involved in transaction T
Revoke:	$SHS.RemoveUser(CRL, U)$ adds user U to revocation list CRL
Properties:	Impersonation & Detection Resistance Imposter & Detector Tracing

	$params$:	Various system parameters
	G, t_G:	CA public and private keys
	t, ID:	Secret t issued by CA for identity ID
	ω:	Public value corresponding to t
Variables:	$ID, attr$:	Hidden Credentials breaks ID into a separate ID and attribute
	$\mathcal{P}$:	Policy expression determines which t values must be used for decryption
	M, C:	Plaintext and ciphertext

Table 1: Summary of abstract specifications, using variable naming conventions from CA-Oblivious Encryption. [7] gives protocols for converting any CA-Oblivious Encryption scheme into a valid OSBE or Secret Handshake scheme. We describe how any Hidden Credentials scheme can easily be used as CA-Oblivious scheme, but using CA-Oblivious schemes to produce Hidden Credentials is more difficult.

The only difference in the One-Way Security game is that $\mathcal{A}$ is always given C, and must return M to win, rather than distinguishing C from C'. This similarity highlights the deficiency we consider in section 3.1.1, since there is little about $SIM_{(S)}$ to suggest that it presents a challenge related to the game's intuitive definition of forcing $\mathcal{A}$ to distinguish between ciphertexts encrypted using different CA public keys.

The other difficulty we had with [7] is that the authors define indistinguishability games for these properties only for a one way encryption system. They mention that such a system can then be transformed to provide semantic (or chosen plaintext, CPA) and chosen ciphertext (CCA) security using standard transformations, but although these general-purpose transformations elegantly achieve their goal of protecting plaintexts, they are not required to guarantee secrecy of values such as the ID and CA public key used during encryption. While it seems unlikely that a One-Way Secure scheme with the requisite obliviousness properties would lose them through application of such a transform, we recommend a more formal exploration for future work.

Finally, the implementation in [7] is unique in relying on the long-standing Computational Diffie Hellman (CDH) assumption, as well as being trivially implemented under the Bilinear Diffie Hellman (BDH) assumption used by identity-based cryptosystems. In passing, the authors also suggest a construction which allows CAs to certify a credential without learning the trapdoor secret. This feature is an important consideration among the systems we examine here, which offer extremely good privacy protection for parties yet leave CAs almost entirely omnipotent.

3.1.1 Replacing Sender and Receiver Obliviousness

The *SIM* functions used in the obliviousness games above cause some difficulties when trying to create transformations from CA-Oblivious schemes to the other schemes. Formally, the properties are said to exist for an implementation if there exist *SIM* functions for which no polynomially-bounded adversary can win the games with nonnegligible probability.

Consider the following alternative to sender obliviousness, the "Modified Sender Oblivious" property. Its only change is to replace $SIM_{(S)}$ with a concrete function, namely the *Enc* function itself running with a different CA public key. This directly embodies the intuitive explanation given for the game: message recipients should not be able to discern what CA public key was used by *Enc* when creating the message, unless they have the necessary decryption key.

A CA-Oblivous Encryption scheme is said to be Modified Sender Oblivious if no polynomially-bounded adversary $\mathcal{A}$ has a non-negligible advantage in winning the following game:

1. *Initialize* and *CAInit* are executed, and $\mathcal{A}$ is given the resulting *params* and G.
2. $\mathcal{A}$ receives as many $\langle t_i, \omega_i \rangle$ pairs as it wants corresponding to ID values of its choosing.
3. $\mathcal{A}$ decides on a challenge ID $\langle ID_R, \omega_R \rangle$.
4. $\mathcal{A}$ receives one of the following values:
   (a) $C = Enc(M, PK)$, for a random message M, and $PK = Recover(G, ID_R, \omega_R)$.
   (b) $C' = SIM_{(S)}(params) = Enc(M', PK')$ for a random message M', and $PK' = Recover(G', ID', \omega_R)$ where ID' is chosen at random and G' results from another call to $CAInit(params)$.
5. $\mathcal{A}$ receives any other $\langle t_i, \omega_i \rangle$ values it wants, such that $ID_i \neq ID_R$.
6. $\mathcal{A}$ wins if it correctly guesses which value it received.

If a system has this property, then adversaries $\mathcal{A}$ have no non-negligible advantage in winning the game for our definition of $SIM_{(S)}$. Thus, all systems which are Modified Sender Oblivious are also Sender Oblivious, since sender obliviousness requires only that $SIM_{(S)}$ exists, while the modified game lists one explicitly.

However, we can offer no proof that all systems which are Sender Oblivious are also Modified Sender Oblivious, in which case the two properties would be exactly equivalent. If they are not equivalent, then there must be some implementation in which a polynomial-time algorithm ${SIM_{(S)}}'$ defeats all polynomially-bounded adversaries, but in which there does exist an adversary which can win against our specific $SIM_{(S)}$ in the modified game. Thus, without a proof that sender obliviousness implies modified sender obliviousness, there remains the possibility that an implementation could exist which has the Sender Oblivious property, but in which attackers could distinguish between the $C = Enc(M, PK)$ and $C' = Enc(M', PK')$ values in our modified game.

While we cannot prove equivalence, nor can we offer an example of a system which is Sender Oblivious but not Modified Sender Oblivious. However, it is interesting to note that if we allow $SIM_{(S)}$ to also see the values G, ID_R, ω_R, it is easy to show how even broken implementations could be considered Sender Oblivious. If we set C' to have the same definition as C:

$$C' = SIM_{(S)}(params, G, ID_R, \omega_R) = Enc(M, PK),$$
$$PK = Recover(G, ID_R, \omega_R)$$

then the adversary's choice becomes meaningless, since no adversary can distinguish between two copies of the same algorithm. We could even create a system in which *Enc* returns a ciphertext of the form $\langle C, ID, G\rangle$, explicitly revealing the ID and CA public key used during encryption. Since both our modified $SIM_{(S)}$ and its counterpart would both return $\langle C, ID_R, G_R\rangle$, the attacker would have nothing meaningful to distinguish, and our system would appear to be Sender Oblivious.

Although it might seem paradoxical, the better *SIM* is at imitating its counterpart, the weaker the guarantees are about the system, since better imitation makes it harder for an adversary to win the games. This is why the open-ended definition in the Sender Oblivious game potentially admits *more* system implementation than its modified counterpart. Fortunately, hiding the challenge ID and CA public key by providing only *params* to *SIM* implicitly sets an upper limit on its ability to imitate the valid alternative, and indeed could effectively make the properties equivalent.

Finally, we note one other consequence of providing only *params* to $SIM_{(S)}$. Since $SIM_{(S)}$ knows neither the CA public key nor the challenge ID_R chosen by the attacker, we must assume that the simulator chooses both a different ID and CA public key when constructing its challenge ciphertext. Thus, the game cannot consider cases in which an attacker might benefit from comparing ciphertexts encrypted against the same ID but differing CAs, or ciphertexts encrypted against different IDs using the same CA public key.

In the case of the Receiver Oblivious game, $SIM_{(R)}$ knows everything its counterpart does except for the private key for the challenge identity ID_R. $SIM_{(R)}$'s definition as a polynomial-time algorithm precludes it from being specified as an alogrihtm which first breaks the private key of the CA by brute force, then recovers its own private key for ID_R (in which case it can likewise exactly imitate its counterpart, causing all systems to be Receiver Oblivious), but otherwise we have only the intuitive definition of the property to indicate that $SIM_{(R)}$ is intended to simulate a receiver with a private key issued by a different CA. We found that having an explicit definition based on the intuitive definition of the property shows us exactly what we can assume about the adversary's abilities, and gives us something more concrete to work with when translating CA-Oblivious schemes into other systems.

With that aim, then, and to further strengthen the obliviousness properties, we offer two new properties to replace sender and receiver obliviousness, called **Certificate Privacy** and **Key Privacy**, respectively. These properties ensure that information about the CA is protected, as required by the intuitive definitions of the obliviousness properties. They also protect the ID string used, as is required in the Credential Indistinguishability property in Hidden Credentials. Note that neither these new properties nor our modified game above are guaranteed to exist in implementations based on the original specification. Examination of existing implementations is an avenue for future work.

It is simple to show that possession of these properties is sufficient to show possession of the corresponding obliviousness properties, just as we showed that our Modified Sender Oblivious property implies satisfaction of sender obliviousness, since they provide specific instances of the *SIM* algorithms and offer adversaries all the facilities required by the obliviousness games.

A CA-Oblivous Encryption scheme is said to have **Certificate Privacy** if no polynomially-bounded adversary $\mathcal{A}$ has a non-negligible advantage in winning the following game:

1. A value $params = Initialize(k)$ is published to all parties.
2. $\mathcal{A}$ makes any number of requests for:
    - A randomly generated CA public key G. The challenger keeps the corresponding private key t_G to itself. Or:
    - A keypair $\langle t, \omega\rangle = Certify'(G, t_G, ID)$ for an ID of his choosing and G from among the randomly generated CA keys. Or:
    - A public value ω returned by $Certify'$ for an ID of his choosing and G from among the randomly generated CA keys.

3. $\mathcal{A}$ then chooses two challenge identities $\langle ID_0, \omega_0, G_0\rangle, \langle ID_1, \omega_1, G_1\rangle$ for which he has not received the corresponding private keys. He sends these identities to the challenger.

4. The challenger chooses a random bit $b \in \{0, 1\}$ and sends ω_b to $\mathcal{A}$, where $\langle t_b, \omega_b\rangle = \mathit{Certify}(G_b, t_{Gb}, ID_b)$.

5. $\mathcal{A}$ makes any number of additional requests, as in step 2, except that he may not request a keypair or ω value for $\langle ID_0, G_0\rangle$ or $\langle ID_1, G_1\rangle$.

6. Eventually, $\mathcal{A}$ outputs a guess $b' \in \{0, 1\}$ for b. $\mathcal{A}$ wins if $b' = b$.

A CA-Oblivous Encryption scheme is said to have **Key Privacy under Chosen Plaintext Attack** if no polynomially-bounded adversary $\mathcal{A}$ has a non-negligible advantage in winning the following game:

1. A value $\mathit{params} = \mathit{Initialize}(k)$ is published to all parties.

2. $\mathcal{A}$ makes any number of requests for:

   - A randomly generated CA public key G. The challenger keeps the corresponding private key t_G to itself. Or:
   - A keypair $\langle t, \omega\rangle = \mathit{Certify}'(G, t_G, ID)$ for an ID of his choosing and G from among the randomly generated CA keys, or:
   - A public value ω returned by $\mathit{Certify}'$ for an ID of his choosing and G from among the randomly generated CA keys.

3. $\mathcal{A}$ then chooses two challenge identities $\langle ID_0, \omega_0, G_0\rangle, \langle ID_1, \omega_1, G_1\rangle$ for which he has not received the corresponding t values. He sends these identities and a message M of his choosing to the challenger.

4. The challenger chooses a random bit $b \in \{0, 1\}$ and sends $C = \mathit{Enc}(M, PK)$ to $\mathcal{A}$, where $PK = \mathit{Recover}(G_b, ID_b, \omega_b)$.

5. $\mathcal{A}$ makes any number of additional requests, as in step 2, except that he may not request a keypair for $\langle ID_0, G_0\rangle$ or $\langle ID_1, G_1\rangle$.

6. Eventually, $\mathcal{A}$ outputs a guess $b' \in \{0, 1\}$ for b. $\mathcal{A}$ wins if $b' = b$.

Note the addition of the ability for the attacker to request only the resulting ω value when *Certify* is run on an ID. The sender obliviousness game given in [7] required $\mathcal{A}$ to provide an $\langle ID, \omega\rangle$ challenge pair, but provided no explicit way for $\mathcal{A}$ to legitimately obtain an ω value for use in the challenge.

Also note the attacker's ability to explicitly choose the CA public keys and challenge identities he must distinguish. In the Receiver Oblivious game, the attacker is not given an explicit choice of challenge identity. If we assume that the challenge identity is chosen at random, then this game would not be able to detect weaknesses in systems having less than $log(k)$ weak identities, even if the attacker knew how to derive and exploit them, since the chances of a weak identity showing up would be lower than a polynomial bound would allow. Since the game specifies that the adversary should fail for "*any* target ID string ID_R", we assume in section 4.1.2 that this means the attacker does in fact get to choose. Our Certificate Privacy property makes this choice explicit.

In the Sender Oblivious game, the attacker chooses the challenge ID for use by *Enc*, but has no control over $SIM_{(S)}$. Thus, in this game the attacker is unable to take advantage of weaknesses appearing only when pairs of identities are chosen from very small sets.

Finally, note that our game's facility allowing requests for randomly generated CA public keys can be shown to be equivalent to a security game in which only two CAs are generated. Note that the attacker can generate CAs on its own, a polynomially-bounded adversary can only request a polynomially-bounded number of keys, and the number of games expected to occur before the first two CA keys happen to be the keys the attacker finds suitable can be expressed as a polynomial in the number of CA public keys he would request in our games. Thus, an attacker's advantage in our game is polynomially related to his advantage in the more restricted game.

3.2 Hidden Credentials

Hidden Credentials schemes are defined in terms of four functions: *CreateCA*, *Issue*, *Encrypt*, and *Decrypt*, which create a CA, issue users a secret corresponding to the certified attribute, encrypt a message based on a policy of attributes which the recipient must possess as certified by specified CAs, and decrypt a ciphertext using their credentials.

The unique security requirement of a Hidden Credentials system is **Credential Indistinguishability**, meaning that ciphertexts encrypted against different simple (single-element) policies must be indistinguishable to an attacker not possessing any of the corresponding credentials. **Policy Indistinguishability** was given later in [4], and specifies how ciphertexts encrypted against multiple-element policies protect privacy against unqualified attackers. Hidden Credentials are not formally required to possess any level of Policy Indistinguishability, but the property exemplifies the tendency in Hidden Credentials work to protect policy contents in addition to credential contents.

Formally, [4] specifies Credential Indistinguishability as follows. No polynomial-time bounded adversary $\mathcal{A}$ has a non-negligible advantage in winning the following game against a challenger. Note that a simple policy is just a pair of values specifying an attribute and a CA public key. The recipient of a message encrypted against a simple policy must know the corresponding private value issued by the CA for that attribute in order to decrypt the message.

1. $\mathcal{A}$ makes any number of requests for a randomly generated CA public key or private key corresponding to an identity of his choosing.
2. $\mathcal{A}$ then chooses a challenge identity ID and two simple policies p_0 and p_1 for which he has not received the corresponding credentials. He sends these two policies, along with ID and a message M of his choosing, to the challenger.
3. The challenger chooses a random bit $b \in \{0, 1\}$ and encrypts M against p_b and ID. He returns the resulting ciphertext to $\mathcal{A}$.
4. $\mathcal{A}$ makes any number of requests for additional CA keys or private keys for identities other than the challenge values, then eventually outputs a guess $b' \in \{0, 1\}$ for b. $\mathcal{A}$ wins if $b' = b$.

An independent paper on Hidden Credentials [10] makes more extreme privacy guarantees, using oblivious transfer and secure function evaluation to constrain the information even qualified recipients can infer from a transaction. Hidden Credentials systems are unique among the systems presented here in taking pains in the abstract definition to hide complex access policies from unqualified recipients.

3.2.1 Formal CCA Security for Hidden Credentials

Although the implmentations in [11] and [4] are both shown to have chosen ciphertext security, and [4] specifies that "Encrypt should be secure against chosen ciphertext attacks", neither paper gives a formal, abstract security game. Consequently, we propose the following HC-IND-CCA-SIMPLE game for simple policies to make this property explicit. HC-IND-CCA-SIMPLE is analogous to the IND-ID-CCA game given in [3], which establishes CCA security for IBE systems.

- HC-IND-CCA-SIMPLE: A Hidden Credentials scheme is said to have HC-IND-CCA-SIMPLE security if no polynomially-bounded adversary $\mathcal{A}$ has a non-negligible advantage in the following game:
  1. The challenger publishes a set of system parameters *params*, and a CA public key G, keeping the CA private key t_G to itself.
  2. $\mathcal{A}$ makes any number of requests for values $t = Issue(ID, attr)$ corresponding to ID and *attr* values of its choosing.
  3. Additionally, $\mathcal{A}$ may make any number of requests for the result of $Decrypt(C, t)$ by specifying the ciphertext C and value ID from which to derive t.
  4. $\mathcal{A}$ then chooses two equal length messages M, M', a challenge identity ID and attribute *attr* for which he has not received the corresponding t value. He sends $\langle M, M', ID, attr \rangle$ to the challenger.
  5. The challenger chooses a random bit $b \in \{0, 1\}$ and returns the resulting ciphertext $C = Encrypt(M_b, ID, p)$ to $\mathcal{A}$, where $p = \langle attr, G \rangle$.

6. $\mathcal{A}$ makes any number of additional requests for private keys as in step 2, with the exclusion that he may not request the private key for the challenge $\langle ID, attr \rangle$ pair.
7. $\mathcal{A}$ may also request any number of additional decryptions as in step 3, as long as the request $\langle C', ID' \rangle$ is not equal to the challenge ciphertext and ID $\langle C, ID \rangle$.
8. Eventually, $\mathcal{A}$ outputs a guess $b' \in \{0, 1\}$ for b. $\mathcal{A}$ wins if $b' = b$.

HC-IND-CCA-SIMPLE is sufficient for our needs in this paper, but we leave a formal definition of a CCA game for complex policies for future work. We suggest that for such a game, the decryption oracle be augmented to accept a ciphertext and multiple ID, G pairs specifying the t values to be used in $Decrypt(C, \langle t_1, t_2, ... \rangle)$. It should then accept decryption queries after the challenge as long as the set of t values resulting from the specified ID, G pairs does not constitute a set of values satisfying the challenge policy.

Although neither [11] or [4] gives proof of chosen ciphertext security directly in terms of our HC-IND-CCA-SIMPLE game, we consider their arguments sufficient to assume that their implementations do indeed possess this attribute, particularly since HC-IND-CCA-SIMPLE is modeled on the IND-ID-CCA game for the system in [3] on which both Hidden Credentials systems are based. Future work may consider it worthwhile to formally evaluate the systems with respect to HC-IND-CCA-SIMPLE before proposing a corresponding game for complex policies.

3.3 Oblivious Signature-Based Envelopes

Whereas Secret Handshakes are defined as a key agreement protocol and Hidden Credentials are defined as an encryption function, OSBE is defined as an interactive protocol. The original paper [14] defines four parties, a CA, a message sender S, a qualified recipient $R1$ and an unqualified recipient $R2$.

A message M is sent in a three phase process. In the **Setup** phase, the CA distributes system parameters and a secret to $R1$. In the **Interaction** phase, S attempts to send M to either $R1$ or $R2$. In the **Open** phase, the recipient attempts to decrypt M.

An OSBE scheme must satisfy three properties. It must be **Sound**, meaning that qualified recipients can successfully recover messages they are qualified to receive. It must be **Semantically Secure Against the Receiver**, just as ordinary encryption functions are required to be semantically secure. It must also be **Oblivious**, meaning that the sender cannot distinguish between interactions with qualified and unqualified recipients.

Formally, an OSBE scheme [15] is Oblivious if no polynomially-bounded adversary $\mathcal{A}$ has a non-negligible advantage in the following game:

1. $\mathcal{A}$ receives the public system parameters $params, G$ and the recipient's ID string ID (called M in [15]).
2. $\mathcal{A}$ chooses a challenge message M, and interacts with the challenger, who chooses to emulate either $R1$ or $R2$ at random.
3. $\mathcal{A}$ wins if it correctly guesses which party was emulated by the challenger.

Later work specified Generalized OSBE (GOSBE) [15], which allows messages to be encrypted against a boolean policy, much like the original Hidden Credentials system. Even more recently, OACerts were introduced [12], which add more sophisticated policy semantics, selective disclosure and zero-knowledge proofs. See section 4.3.2 for a comparison with the policy support in Hidden Credentials.

OSBE has the most different implementations among the systems discussed here, including an RSA implementation and implementations under both the Boneh-Franklin and Cocks IBE systems, which operate under the BDH and Quadratic Residue (QR) assumptions, respectively. Nasserian and Tsudik [16] propose even more implementations based on other signature algorithms, while Bagga and Molva [1] describe a generalized implementation of OSBE that also supports complex policies.

OSBE's RSA-based implementation means it can be used with existing, traditional RSA-signed certificates and trust negotiation protocols to resolve policy cycles and obtain some of the privacy advantages offered by these new systems.

The significance of these various OSBE implementations to this paper is a demonstration that even though a system may have a "weaker" definition, it may be implemented in situations where other systems with more rigorous properties cannot. This is an important consideration when comparing the systems described in this paper.

System	Transforms to	Using	Caveat	Section
Hidden Credentials	IBE	HC-TO-IBE	-	4.2.1
Hidden Credentials	CA-Oblivious	HC-TO-CAOE	-	4.2.2
Hidden Credentials	OSBE	HC-TO-OSBE	-	4.2.3
CA-Oblivious	OSBE	CAOE-TO-OSBE	-	4.1.2
CA-Oblivious	Secret Handshakes	Given in [7]	CA-like	4.1.3
Hidden Credentials	Secret Handshakes	via HC-TO-CAOE	CA-like	4.2
CA-Oblivious	Hidden Credentials	CAOE-TO-PKHC	PKHC	4.1.1
OSBE	Secret Handshakes	See note 1	-	-
OSBE	Hidden Credentials	-	-	4.3
OSBE	CA-Oblivious	-	-	4.3
Secret Handshakes	*	See note 2	-	4

Table 2: Transformations between systems, in approximate order of difficulty. Note the HC-TO-IBE transform, which helps explain the difficulty of transforming the other systems to Hidden Credentials. The HC-TO-IBE, CAOE-TO-OSBE, HC-TO-IBE and HC-TO-CAOE transforms involve only minor ambiguities in the specifications of the original systems. CA-Oblivious Encryption and Hidden Credentials transform to a slightly constrained definition of Secret Handshakes called CA-like Secret Handshakes. Transforming CA-Oblivious systems to Hidden Credentials requires a significant Public Key-like restriction which we call Public Key Hidden Credentials (PKHC), and the transformation may require the CA-Oblivious scheme to have additional properties. Note 1: OSBE does not require certain properties needed by CA-Oblivious Encryption and Hidden Credentials, although [7] suggests that OSBE might be suitable for implementing Secret Handshakes. Note 2: Secret Handshakes are not transformable in the general case since they assume senders and recipients will have credentials from the same CA.

3.4 Secret Handshakes

The abstract definition for a secret handshake scheme as given in [2] comprises five functions: $SH.CreateGroup$ ($Group$) creates a group of users $Group$, returning the group secret t_G. $SH.AddUser(ID, Group, t_G)$ returns the secret t corresponding to ID's membership in G. ID may be a simple nym, or a concatenation of a nym and role. $SH.Handshake(A, B)$ ensures that B learns whether $A \in Group$ only if $B \in Group$, and that A learns whether $B \in Group$ only if $A \in Group$. $SH.TraceUser(T)$ given a transcript T, returns which users participated in the transaction. $SH.RemoveUser(RevokedUserList, ID)$ adds ID to the list of revoked users.

$SH.Handshake$ is given a concrete implementation for pairing-based key agreements, $PBH.Handshake$, which is based on the BDH assumption and involves a very simple protocol that outputs a shared secret upon successful completion. The CA-Oblivious scheme already discussed was designed to implement Secret Handshakes [7]. Vergnaud also gave several variants of an RSA-based implementation of Secret Handshakes [17].

The security properties defined in [2] are **Impersonation Resistance**, **Imposter Tracing**, **Detector Resistance** and **Detector Tracing**.

Impersonation Resistance implies that any polynomial time bounded adversary that has corrupted no users from the group has a negligible advantage in convincing a valid user that it is a member of the group.

A Secret Handshake scheme with Imposter Tracing is one in which, given the transcript of a session between an adversary and a valid user, group administrators have approximately the same probability of detecting what user secrets have been compromised as the adversary has in impersonating a valid user.

A scheme has Detection Resistance if adversaries have negligible chances of distinguishing group members from nonmembers. Detector Tracing is then defined analogously to Imposter Tracing.

[2] goes on to list some additional, but not mandatory, security properties. The authors define Forward Repudiability, Indistinguishability to Eavesdroppers, Collusion Resistance and Unlinkability. Forward Repudiability means that users are not left with cryptographic proof of a partner's group membership after a transaction. Indistinguishability to Eavesdroppers and Collusion Resistance follow from the earlier properties.

Multi-show Unlinkability is described in terms of the trivial approach of using one-time pseudonyms, but was later also achieved cryptographically [18].

4 Transformations Between Systems

While the systems we consider tend to be cited as a group, and indeed share many similarities in their specifications, the systems are not fundamentally interchangeable based on their abstract definitions. Here we consider the difficulties inherent in transforming systems of one type into systems of another based only on their abstract requirements. Table 2 summarizes these results.

Since Secret Handshakes alone require that both parties have a credential from the same issuer, there is no guarantee that any implementation can be transformed into one of the other systems described here.

Of course, a particular implementation of any system may embody the requirements of another system, but here we only consider whether all compliant implementations of a given system can be expected to satisfy the requirements of another.

4.1 Transforming CA-Oblivious Encryption

[7] introduces CA-Oblivious Encryption. The authors give a generalized four-round protocol for implementing Secret Handshakes, then offer a three-round protocol which works using a zero-knowledge signature of knowledge of t. The authors claim in passing that their system also meets the needs of Hidden Credentials and OSBE. We consider those claims in the following subsections.

4.1.1 CA-Oblivious Encryption to Hidden Credentials

The authors of [7] claim that CA-Oblivious Encryption schemes can be used to implement Hidden Credentials. Here we describe the difficulties inherent in this claim, and sketch the process by which CA-Oblivious schemes with additional security properties could be transformed into a restricted form of Hidden Credentials we call Public Key Hidden Credentials (PKHC). The applications available to Hidden Credentials systems restricted to allow transformations from CA-Oblivious and OSBE schemes are considered briefly in section 4.3.2.

The first challenge in transforming CA-Oblivious systems to Hidden Credentials is the CA-Oblivious use of ω values. As specified, Hidden Credentials are built to model identity-based cryptosystems, in which message recipients need not publish public values (or even obtain private keys) before senders can encrypt messages for them. Consequently, adapting CA-Oblivious schemes requires defining a limited form of Hidden Credentials in which recipients receive private keys and send the corresponding ω values to senders before encryption can take place. Accepting that limitation, and the need for Hidden Credentials systems to protect ID strings used during encryption (something not specified in the CA-Oblivious specification), a transform from CA-Oblivious schemes with the requisite additional properties to PKHC could proceed as follows:

1. Define a constrained variation of Hidden Credentials which operates like a public-key cryptosystem, called Public Key Hidden Credentials (PKHC). This system could operate by specifying that credential holders publish the public keys in a directory, but this makes it hard to implement one-time credentials, in which credential holders use a different copy of their credentials for each transaction to avoid linkability. Rather, we suggest the addition of a $TransmitKeys$ protocol which allows credential holders to deliver their public keys to message senders in an appropriate way. This protocol would be required to happen after *Issue* certifies an attribute, but before *Encrypt* is used. Since *Encrypt* may involve encryption against multiple credentials, *TransmitKeys* must allow users to send multiple public keys to a message sender, and to be true to the aims of Hidden Credentials, should also allow recipients to send additional bogus values to prevent senders from learning how many credentials they possess. Since Hidden Credentials also aims to protect sender policies, care must be taken to ensure that senders aren't required to specify what public values the recipient needs to provide. If at all possible, ensure that PKHC implementations form a proper subset of Hidden Credentials implementations. In other words, it should be relatively easy to construct a HC-TO-PKHC transformation which operates on all compliant Hidden Credentials implementations.

2. Update the Credential Indistinguishability game to reflect possible privacy attacks opened by *TransmitKeys*. An attacker must not be able to expose CA public keys, IDs, or policies and attributes used by *Encrypt* by his choice of values sent during *TransmitKeys*.

3. Establish a PKHC property similar to the Certificate Privacy we proposed in section 3.1.1 to ensure that recipients don't reveal details about what credentials they hold. Even if *TransmitKeys* is defined as we recommend above, users may have to reveal an upper bound on the number of credentials they are willing to use in a transaction.

4. Since malicious recipients may be able to force senders to encrypt messages against potentially bogus public keys, ciphertext indistinguishability properties (such as the HC-IND-CCA-SIMPLE property defined in section 3.2.1) must also be carefully constructed to ensure that abuses of *TransmitKeys* cannot result in message leaks in *Encrypt*.

5. Once Public Key Hidden Credentials (PKHC) have been properly specified, construct a transform CAOE-TO-PKHC which allows CA-Oblivious systems to implement the new PKHC scheme. Appendix A gives a sketch of this transform. Note that the transform needs to provide for the complex policy support required by *Encrypt*.

6. Next, proofs must be constructed like those in section 4.2.2 to show that violating the security properties of PKHC requires violating required properties of the underlying CA-Oblivious system. This may require defining constraining properties for CA-Oblivious encryption which an implementation must possess in order to be safely used in the CAOE-TO-PKHC transform. The properties given in section 3.1.1 were designed with this in mind.

4.1.2 CA-Oblivious Encryption to OSBE

[7] also claims in passing that CA-Oblivious Encryption is sufficient to construct OSBE, but here we consider this proposition formally and offer proof.

- CAOE-TO-OSBE transform: Given a CA-Oblivious Encryption scheme with functions *Initialize*, *CAInit*, *Certify*, *Enc*, *Dec*, an OSBE scheme can be constructed as follows.
  - Let parties $CA, S, R1, R2$ be defined as required by the OSBE specification.
  - *SetupPhase*: $\langle params, G, t_G, t\rangle = Setup(k, ID, M)$ is implemented as follows: $params = Initialize(k)$; $\langle G, t_G\rangle = CAInit(params)$; $\langle t, \omega\rangle = Certify(ID)$. $R1$ gets $\langle t, \omega\rangle$, S gets M and all parties get ID as required.
  - *InteractionPhase*: If S interacts with $R1$, $R1$ sends ω, and S responds with $C = Enc(M, PK)$, where $PK = Recover(G, ID, \omega)$. If S interacts with $R2$, $R2$ sends ω_R as output by $SIM_{(R)}(params)$, defined in the Receiver Oblivious game for the CA-Oblivious scheme.
  - *OpenPhase*: If S and $R1$ interacted, $R1$ outputs $M = Dec(C, t)$. $R2$ always outputs nothing.

An OSBE scheme must be Sound, Oblivious, and Semantically Secure Against the Receiver. Soundness is obviously supplied by a sound CA-Oblivious scheme. Proving semantic security against the receiver requires transforming the CA-Oblivious scheme into a semantically secure scheme (from its weaker default of One-Way Security) first. While we see no reason to believe that such a transformed scheme would not provide the requisite semantic security for OSBE, as we described in section 3.1, constructing a proof for CAOE-TO-OSBE based on the transformed CA-Oblivious scheme is left as an avenue for future work.

The CA-Oblivious scheme's Receiver Oblivious property (given in section 3.1) and OSBE's Oblivious property (section 3.3) are quite similar, leading to a proof in which we almost trivially transform the Receiver Oblivious game into an OSBE Oblivious game. However, we note that the OSBE game allows the adversary to choose a challenge ID, while the Receiver Oblivious game simply gives the adversary a value ID_R. The Receiver Oblivious game specifies that it must work for all values of ID_R, so if we take this to mean that the adversary can in fact choose ID_R (as would be necessary for an adversary to win if it found a very small class of weak ID values), the following proof works. The Certificate Privacy property defined in section 3.1.1 implies receiver obliviousness, and also works with this proof, but without the need to make assumptions about the ability to choose ID values.

First we define a game transformation **RO-TO-O** which transforms a Receiver Oblivious game into the Oblivious game from the OSBE definition: An intermediary $\mathcal{I}$ acts as adversary in the Receiver Oblivious game, and also implements the challenger in the OSBE Oblivious game against an adversary $\mathcal{A}$. (For convenience, note that $\mathcal{I}$ will essentially be emulating $R1$ or $R2$ in the CAOE-TO-OSBE specification above, while $\mathcal{A}$ plays the role of S.) $\mathcal{I}$ passes along the system parameters *params* to $\mathcal{A}$, then selects $\mathcal{A}$'s choice of ID as its own challenge identity. $\mathcal{I}$ passes along the ω or r value it receives, then accepts and discards whatever C value is returned by $\mathcal{A}$. $\mathcal{I}$ makes the same guess $\mathcal{A}$ makes, and wins its game if $\mathcal{A}$ wins.

Theorem 4.1. *Any OSBE scheme produced using CAOE-TO-OSBE is Oblivious.*

Proof. Every CA-Oblivious Encryption scheme must be Receiver Oblivious. SO-TO-O allows $\mathcal{I}$ to implement a proper OSBE Oblivious game for attacker $\mathcal{A}$ using the OSBE scheme resulting from CAOE-TO-OSBE, and allows $\mathcal{I}$ to play the attacker in a proper Sender Oblivious game for the underlying CA-Oblivious system. $\mathcal{I}$ wins its game iff $\mathcal{A}$ wins its game. Thus, any attacker $\mathcal{A}$ with a nonnegligible advantage in winning its game implies the existance of an attacker $\mathcal{I}$ with nonnegligible advantage in winning the Sender Oblivious game. Since $\mathcal{I}$ has no abilities unavailable to any other attacker in its game, and no such attacker exists for a valid CA-Oblivious system given the assumptions of the implementation's underlying hard problem, there must not exist an attacker $\mathcal{A}$ for the OSBE system produced by CAOE-TO-OSBE. □

4.1.3 CA-Oblivious Encryption to Secret Handshakes

Secret Handshakes from CA-Oblivious Encryption is the title of the paper which introduces CA-Oblivious Encryption [7]. The authors give a generalized four-round protocol for implementing a slightly constrained yet reasonable definition Secret Handshakes which they denote "CA-Like Secret Handshakes". The only differences in this system are that the CA publishes a public key, and the issuing process is defined as a protocol between the user and CA, rather than a single function which outputs the user secret. The authors also offer a three-round protocol for implementing CA-like Secret Handshakes which works using a zero-knowledge signature of knowledge of t.

Since we show in section 4.2.2 how any Hidden Credentials system can be used to derive a CA-Oblivious scheme, the transformation given in [7] is sufficient to show, by transitivity, that Hidden Credentials can also be transformed into a Secret Handshake scheme.

4.2 Transforming Hidden Credentials

Here we give explicit transformations allowing any Hidden Credentials implementation to implement CA-Oblivious Encryption and OSBE. The ability to use CA-Oblivious Encryption to implement CA-like Secret Handshakes means that, by transitivity, Hidden Credentials implementations are guaranteed to lead to CA-like Secret Handshakes. Thus, we give no explicit transform from Hidden Credentials to CA-like Secret Handshakes.

To point out one of the major differences between Hidden Credentials and the other systems, we also give a simple transformation from any Hidden Credentials scheme to Identity-Based Encryption. The other systems generally allow their implementations to include cryptographic values which message recipients must provide to senders before messages can be encrypted. By omitting such values, Hidden Credentials can be used in IBE-like applications where senders can create messages even before the recipient has received any private keys. But conversely, our transformation shows that Hidden Credentials are at least as hard to implement as IBE. Thus, a Hidden Credentials implementation based on a traditional assumption such as CDH or RSA would be much more remarkable than the implementations of the other systems based on such assumptions, since it would imply an IBE implementation based on that assumption (whereas the only known secure IBE implementations today are based on the less established Bilinear Diffie-Hellman (BDH) and Quadratic Residue (QR) problems).

4.2.1 Hidden Credentials to Identity-Based Encryption

- HC-TO-IBE transform: Given a Hidden Credentials scheme with functions *CreateCA*, *Issue*, *Encrypt*, *Decrypt*, the functions making up an Identity-Based Encryption scheme (as defined in [3]) are constructed as follows.
  - *Setup*(k): Returns $\langle params, t_G \rangle$, where *params* includes both the system values implicitly chosen in the Hidden Credentials scheme and the CA public key G returned by $\langle G, t_G \rangle = CAInit(params)$. Only the CA gets t_G.
  - *Extract*($params, t_G, ID$): $t = Issue(\emptyset, ID)$.
  - *Encrypt*($params, ID, M$): $C = Encrypt(M, \emptyset, p)$ where $p = \langle ID, G \rangle$.
  - *Decrypt*($params, C, t$): $M = Decrypt(C, \langle t \rangle)$

Hidden Credentials are required to be CCA-secure. Since the HC-IND-CCA-SIMPLE game for chosen ciphertext security is modeled directly on the IND-ID-CCA game for Identity-Based Encryption, it is easy to show that HC-IND-CCA-SIMPLE security in the Hidden Credentials scheme implies IND-ID-CCA security

in the IBE scheme. Both games specify equivalent private key extraction and decryption queries, require the attacker to choose two equal-length challenge messages, allow the attacker more queries, and then require the attacker to decide which message was encrypted. A proof that success in the IBE game implies success in the Hidden Credentials game then proceeds just as in our proofs for the other transforms.

4.2.2 Hidden Credentials to CA-Oblivious Encryption

Here we give a transform which shows how to construct a CA-Oblivious Encryption system from any Hidden Credentials system. Note that the policy p passed to *Encrypt* is just a simple policy, consisting of a single (attribute, CA public key) tuple.

- HC-TO-CAOE transform: Given a Hidden Credentials scheme with functions *CreateCA*, *Issue*, *Encrypt*, *Decrypt*, the functions making up a CA-Oblivious scheme are constructed as follows.
  - *Initialize*: A set of parameters *params* is chosen (This is done implicitly in the Hidden Credentials scheme)
  - $CAInit(params)$: $\langle G, t_G \rangle = CreateCA(params)$
  - $Certify(G, t_G, ID)$: $\langle t, \omega \rangle = \langle Issue(\emptyset, ID), \emptyset \rangle$
  - $Recover(G, ID, \omega)$: $PK = \langle G, ID \rangle$
  - $Enc(M, PK)$: $C = Encrypt(M, \emptyset, p)$ where $p = \langle ID, G \rangle$
  - $Dec(C, t)$: $M = Decrypt(C, \langle t \rangle)$

Every CA-Oblivious cryptosystem must be **Sender Oblivious**, **Receiver Oblivious**, and **One-Way Secure**.

Since receiver obliviousness deals with an attacker's inability to learn about the receiver's credentials based on their ω values, and the HC-TO-CAOE transform sets these values to $\emptyset$ for all transactions, the resulting CA-Oblivious scheme is trivially Receiver Oblivious.

To prove that HC-TO-CAOE produces a scheme with sender obliviousness, we introduce an attacker $\mathcal{I}$ in the Credential Indistinguishability game defined in section 3.2 which also acts as the challenger in a sender obliviousness game with $\mathcal{A}$. Here is the specification for the game $\mathcal{A}$ plays with $\mathcal{I}$, which we call CI-TO-SO, since it transforms a Credential Indistinguishability game into a Sender Oblivious game:

CI-TO-SO:

1. Let $\mathcal{A}$ be the attacker in a Sender Oblivious game for a CA-Oblivous Encryption scheme constructed using HC-TO-CAOE. Let $\mathcal{I}$ be the attacker in the Credential Indistinguishability game for the Hidden Credentials scheme used to produce the CA-Oblivious scheme.
2. $\mathcal{I}$ requests two CA public keys G, G' and provides G to $\mathcal{A}$.
3. Each time $\mathcal{A}$ requests $\langle t_i, \omega_i \rangle$ corresponding to an identity ID_i, $\mathcal{I}$ likewise requests a private key for the ID $\emptyset$ and attribute ID_i, and returns $\langle t_i, \emptyset \rangle$ to $\mathcal{A}$.
4. When $\mathcal{A}$ decides on a challenge ID (ID_R, ω_R), $\mathcal{I}$ sets its challenge identity $ID = \emptyset$, $p_0 = \langle ID_R, G \rangle$, $p_1 = \langle ID'_R, G' \rangle$ for some randomly chosen value ID'_R, and chooses a random challenge message M.
5. $\mathcal{I}$ receives its challenge ciphertext, which is either $C = Encrypt(M, ID, p_0)$ or $C = Encrypt(M, ID, p_1)$. It passes C along to $\mathcal{A}$.
6. $\mathcal{I}$ passes along additional requests for ID certification as before.
7. When $\mathcal{A}$ makes its guess, $\mathcal{I}$ passes the same value along as its own guess, and wins its game iff $\mathcal{A}$ wins.

Theorem 4.2. *Any CA-Oblivious Encryption scheme produced using HC-TO-CAOE is Sender Oblivious.*

Proof. This proof corresponds directly to the proof of theorem 4.1. Every Hidden Credentials system is required to have Credential Indistinguishability. CI-TO-SO allows $\mathcal{I}$ to implement a proper Sender Oblivious game for attacker $\mathcal{A}$ using the CA-Oblivious scheme produced by HC-TO-CAOE, and allows $\mathcal{I}$ to play the attacker in a proper Credential Indistinguishability game for the underlying Hidden Credentials system. $\mathcal{I}$ wins its game iff $\mathcal{A}$ wins its game. Thus, any attacker $\mathcal{A}$ with a nonnegligible advantage in winning its game implies the existance of an attacker $\mathcal{I}$ with nonnegligible advantage in winning the Credential

Indistinguishability game. Since $\mathcal{I}$ has no abilities unavailable to any other attacker in the Credential Indistinguishability game, and no such attacker exists for a valid Hidden Credentials system given the assumptions of the implementation's underlying hard problem, there must not exist an attacker $\mathcal{A}$ for the CA-Oblivious system produced by HC-TO-CAOE. □

Although we discussed ambiguities in the definition of the Sender Oblivious game in section 3.1, note that our construction of the game satisfies both its precise definition and its intuitive meaning, since we define $SIM_{(S)}$ as an algorithm which differs from its counterpart by virtue of encrypting using a different CA public key. Our original HC-TO-CAOE construction set ID in the CA-Oblivious scheme to ID in the Hidden Credentials scheme, but this prevented the CI-TO-SO game transform from working properly. As we noted in section 3.1, the definition of $SIM_{(S)}$ does not explicitly include passing in the challenge identity, and assuming that it can know values not explicitly passed in breaks it entirely. Thus, we must assume that $SIM_{(S)}$ uses both a different CA public key and a different challenge ID from its counterpart. Setting $ID = \emptyset$ in the Hidden Credentials scheme and instead using the CA-Oblivious ID as the policy attribute allows $\mathcal{I}$ to construct challenge policies p_0, p_1 which differ in both identity string and CA public key as required.

We defined the Key Privacy and Certificate Privacy properties for CA-Oblivious Encryption with our HC-TO-CAOE transform in mind. Certificate Privacy is trivially achieved in CA-Oblivious schemes produced by HC-TO-CAOE, since all the ω values are set to $\emptyset$. To prove Key Privacy, however, the CI-TO-SO transform must be modified slightly to give $\mathcal{A}$ the extra abilities required by the Key Privacy game. Most notably, $\mathcal{I}$ passes along both challenge values chosen by $\mathcal{A}$, rather than choosing the second at random. By design, these abilities line up with the abilities $\mathcal{I}$ has in its own game. The proof then works just as the one given above.

4.2.3 Hidden Credentials to OSBE

Although we originally intended to achieve Hidden Credentials to OSBE transformations transitively, simply invoking a transform from Hidden Credentials to CA-Oblivious Encryption, and then transforming that system using CAOE-TO-OSBE, demonstrating that CAOE-TO-OSBE satisfies the requisite OSBE properties involved some ambiguity, as we described in section 4.1.2. This ambiguity is relieved when the CA-Oblivious scheme has Certificate Privacy, and CA-Oblivious schemes produced using our HC-TO-CAOE transform are guaranteed to have Certificate Privacy. Nevertheless, we give a direct HC-TO-OSBE transform in this section.

- HC-TO-OSBE transform: Given a Hidden Credentials scheme with functions $Setup, Issue, Encrypt$, $Decrypt$, an OSBE scheme can be constructed as follows.
  - Let parties $S, R1, R2$ be defined as required by the OSBE specification.
  - *SetupPhase*: $\langle params, G, t_G, t\rangle = Setup(k, ID, M)$ is implemented as follows: *params* are chosen as usual for the Hidden Credentials scheme. $\langle G, t_G\rangle = CreateCA(params); t = Issue(\emptyset, ID)$. $R1$ gets t, S gets M and all parties get ID as required.
  - *InteractionPhase*: S sends the value $C = Encrypt(M, \emptyset, p)$, where $p = \langle ID, G\rangle$, to either $R1$ or $R2$.
  - *OpenPhase*: If S and $R1$ interacted, $R1$ outputs $M = Decrypt(C, \langle t\rangle)$. $R2$ always outputs nothing.

OSBE schemes must be Sound, Oblivious, and Semantically Secure Against the Receiver. Soundness is obviously supplied by a sound Hidden Credentials scheme. Since obliviousness is defined as a game in which S tries to distinguish between interactions with $R1$ and $R2$, and in our specification neither $R1$ nor $R2$ send any values to S, obliviousness is trivially achieved. Semantic security against the receiver is likewise trivially achieved by the CCA security required of *Encrypt*, particularly if we assume fulfillment of the HC-IND-CCA-SIMPLE property defined in section 3.2.1.

4.3 Transforming OSBE

4.3.1 OSBE to CA-Oblivious Encryption

Since OSBE does not require that messages hide the ID and CA public value used during encryption, there is no guarantee that all OSBE implementations will be able to satisfy the Sender Oblivious requirement of CA-Oblivious Encryption or the Credential Indistinguishability required by Hidden Credentials. In fact, while

the OSBE paper gives a straightforward implementation using IBE, and both the CA-Oblivious Encryption and Hidden Credentials papers discuss their relation to IBE at length, it is worth noting that the RSA-OSBE is trivially shown not to be receiver oblivious. Given two CAs with RSA moduli n, n', where $n > n'$, any passive observer has an advantage distinguishing between messages reduced by the different moduli (as required by the encryption process) since some ciphertexts reduced modulo n will be greater than n'. Techniques proposed by Desmedt [9] might prove useful in patching this leak for RSA-OSBE, but OSBE's abstract definition remains insufficient to guarantee that any OSBE implementation will implement Hidden Credentials or CA-Oblivious Encryption. We give further notes on compatibility in the following subsections.

4.3.2 OSBE to Hidden Credentials

Like the CA-Oblivious scheme, some OSBE implementations assume that users provide tokens which correspond to their credentials, causing problems for Hidden Credentials implementations as described in section 4.1.1. But note that while CA-Oblivious Encryption needs to add a *TransmitKeys* protocol to Hidden Credentials before ciphertexts can be created, OSBE allows the encryption step itself to be specified as an interactive protocol. Thus, while Hidden Credentials implementations can be used in applications where message senders deliver ciphertexts to users who may at some point in the future receive the private keys necessary for decryption, and Public Key Hidden Credentials schemes built from CA-Oblivious schemes would allow encrypted email to be sent to a recipient once *TransmitKeys* was executed, a Hidden Credentials-like scheme specified to accomodate transformations from any OSBE scheme would require that sender and recipient be available for an interactive protocol which might take any number of rounds.

The OSBE and GOSBE protocols also specify that message recipients provide the text of their certificates minus the CA signature, or fabricate a certificate if they don't have one, whenever a message sender wishes to deliver a message. This assumes that the recipient knows what credential the sender is looking for, implying that the sender is willing to disclose his policy before initiating the OSBE protocol. In contrast, Hidden Credentials systems go to great lengths to protect even implicit characteristics of policies from being disclosed to unqualified recipients, and assume that clients may have credentials they are unwilling to even acknowledge they possess.

OACerts [12] add unique policy operators and selective disclosure features not found in base Hidden Credentials systems, but still assume that policies and certificate contents (which may in this case contain only obscured commitments to actual values) are disclosed before the protocol commences, suggesting that although OSBE and Hidden Credentials are superficially similar, they ultimately serve different privacy needs. However, recent work using secure function evaluation [13] begins to bridge this gap, using a protocol in which a server inputs a policy, clients input certificates, and both ultimately learn only whether the client satisfied the server's policy.

4.3.3 OSBE to Secret Handshakes

Vergnaud gives an RSA-based implementation [17] of Secret Handshakes, suggesting that perhaps RSA-OSBE could also lead to a Secret Handshake scheme with or without satisfying the receiver obliviousness requirement of CA-Oblivious Encryption. Nasserian and Tsudik [16] also claim without proof that the OSBE protocol may be exectued twice to achieve Secret Handshakes, and describe the development of the idea as a major avenue of future work.

5 Conclusion

The four systems described here form a class of systems which all allow digital credentials to be used directly in authorization processes.

While Hidden Credentials is the only system in which any implementation can be transformed to any of the other systems, our HC-TO-IBE transformation shows that Hidden Credentials themselves are at least as hard to implement as Identity-Based Encryption. CA-Oblivious Encryption fills a middle ground, with implementations based on the well-established Computational Diffie-Hellman assumption, implementing OSBE and Secret Handshakes directly, and having the potential to implement a restricted form of Hidden Credentials we call Public-Key Hidden Credentials. OSBE allows encryption to take the form of an interactive protocol, and its weaker privacy assumptions admit a wide variety of implementations, including implementations based on the RSA assumption. Secret Handshakes are more limited, assuming that both parties to a

transaction will have credentials from the same CA, however they are unique in having been shown to have efficient multi-show unlinkability [18].

In each case, the systems have significant differences from each other in abstract specifications as well as in implementation-specific features. Authors should take care when choosing systems and characterizing them in related work summaries to avoid misappraising their properties.

6 Future Work

Our work provides firm footing for future work on these systems. We identified Public-Key Hidden Credentials as a major avenue for future work, since they have the potential to provide many of the privacy features found in Hidden Credentials, but can likely be implemented using a much wider variety of assumptions than are currently available to Hidden Credentials. Likewise, transforming OSBE to Secret Handshakes will likely prove fruitful. And of course, new implementations of any of the systems are always useful, and in many cases offer unique privacy features. We recommend that implementors carefully specify the abstract requirements of such features, so that future work can consider whether they can be achieved via transformations from other systems. Future work may also consider compatibility between more complex features such as those offered by GOSBE [15], OACerts [12] and complex policy support in Hidden Credentials [4, 10].

References

[1] Walid Bagga and Refik Molva. Policy-based cryptography and applications. In *Financial Cryptography*, pages 72–87, 2005.

[2] D. Balfanz, G. Durfee, N. Shankar, D.K. Smetters, J. Staddon, and H.C. Wong. Secret handshakes from pairing-based key agreements. In *Proceedings of the 2003 IEEE Symposium on Security and Privacy*, pages 180–196, Oakland, CA, May 2003.

[3] D. Boneh and M. Franklin. Identity-based encryption from the Weil pairing. In *Proceedings of Crypto 2001*, volume 2139 of *Lecture Notes in Computer Science*, pages 213–229. Springer, 2001.

[4] R. Bradshaw, J. Holt, and K. E. Seamons. Concealing complex policies with hidden credentials. In *Eleventh ACM Conference on Computer and Communications Security*, Washington D.C., Oct 2004. ACM Press.

[5] S. Brands. *Rethinking Public Key Infrastructures and Digital Certificates: Building in Privacy.* The MIT Press, 2000.

[6] J. Camenisch and A. Lysyanskaya. Efficient Non-Transferable Anonymous Multi-Show Credential system with optional anonymity revocation. In *EUROCRYPT 2001*, volume 2045 of *Lecture Notes in Computer Science*. Springer, 2001.

[7] C. Castelluccia, S. Jarecki, and G. Tsudik. Secret handshakes from ca-oblivious encryption. In *Advances in Cryptology - ASIACRYPT 2004: 10th International Conference on the Theory and Application of Cryptology and Information Security*, volume 3329. Springer-Verlag GmbH, December 2004.

[8] D. Chaum. Security without Identification: Transactions Systems to Make Big Brother obsolete. *Communications of the ACM*, 24(2), 1985.

[9] Y. Desmedt. Securing Traceability of Ciphertexts - Towards a Secure Software Key Escrow System (Extended Abstract). In *Advances in Cryptology - Eurocrypt '95*, volume 921 of *Lecture Notes in Computer Science*. Springer, 1995.

[10] K. Frikken, M. Atallah, and J. Li. Hidden access control policies with hidden credentials. In *3rd Annual Workshop on Privacy in the Electronic Society (WPES)*, pages 27–28, 2004.

[11] J. Holt, R. Bradshaw, K. E. Seamons, and H. Orman. Hidden credentials. In *2nd ACM Workshop on Privacy in the Electronic Society*, pages 1–8, Washington, DC, October 2003. ACM Press.

[12] J. Li and N. Li. OACerts: Oblivious Attribute Certificates. In *Proceedings of 3rd Conference on Applied Cryptography and Network Security (ACNS)*, pages 301–317, June 2005.

[13] J. Li and N. Li. Policy-Hiding Access Control in Open Environment. In *24th ACM Symposium on Principles of Distributed Computing (PODC)*, Las Vegas, NV, 2005.

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

[15] N. Li, W. Du, and D. Boneh. Oblivious signature-based envelope. In *Distributed Computing*. Springer-Verlag GmbH, November 2004.

[16] S Nasserian and G Tsudik. Revisiting Oblivious Signature-Based Envelopes. In *Pre-print*, March 2006.

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

[18] Shouhuai Xu and Moti Yung. k-anonymous secret handshakes with reusable credentials. In *Eleventh ACM Conference on Computer and Communications Security*, Washington D.C., Oct 2004. ACM Press.

A Appendix: A Sketch of CAOE-TO-PKHC

When implementing Hidden Credentials using CA-Oblivious Encryption, Alice and Bob can have their credentials issued to a consistent nym, but each credential will have a different value ω. Alice and Bob can each send their n values of ω along with their nyms, incurring an $O(n)$ overhead, and the receiver obliviousness of the CA-Oblivious scheme guarantees that these values do not leak information about the issuing CAs. The CA-Oblivious scheme must also be constrained to have Certificate Privacy, so that the ω values do not reveal identity or attribute values. It should be possible for Alice and Bob to add additional, bogus values of ω to their message, converting the disclosure from a quantifier of the number of credentials they possess to an upper bound, in exchange for additional network and computational overhead.

Note the inclusion of the helper functions $HC_{simpleE}$ and $HC_{simpleD}$ used to handle encryption and decryption of a single secret share using a single credential, as specified in the original Hidden Credentials paper [11].

- OSBE-TO-PKHC transform.
  - **CreateCA**: Call *Initialize*, then *CAInit*. Define a one to one function $ID = join(nym, attribute)$ that maps the $\langle nym, attribute\rangle$ pairs used by Hidden Credentials to the single-string values ID used by CA-Oblivious encryption. (A similar function is defined in the appendix of [7], which deals with role support in CA-Oblivious Encryption.)
  - $Issue(nym, attribute)$: Return $\langle t, \omega\rangle = Certify(join(nym, attribute))$.
  - $HC_{simpleE}(\mathbf{R}, nym, \mathbf{p}, \mathbf{\Omega})$: Let $\langle attribute, CA\_pub\rangle = p$.
    Return $\mathcal{C} = \langle c_1 ... c_n\rangle$
    $|c_i = Enc_{PK_i}(R)$
    $|PK_i = Recover(CA\_pub, join(nym, attribute), \omega_i) \forall \omega_i \in \Omega$
  - $HC_{simpleD}(\mathcal{C}, \mathcal{T})$: Return $\bigcup Dec(c_i, t_i) \forall \langle t_i, \omega_i\rangle$.
  - $Encrypt(\mathbf{R}, \mathbf{nym}, \mathbf{P}, \mathbf{\Omega})$: Call $HC_{simpleE}$ for each $p \in P$ as required by the secret splitting scheme to produce ciphertext C.
  - $Decrypt(\mathbf{C}, \mathcal{T})$: Also unchanged. Returns R iff $\mathcal{T}$ contains a satisfying set for P.

Note the addition of Ω to $HC_{simpleE}$ and *Encrypt*. The requirement of Ω exchange prevents the implementation's use in applications described in [11] where message senders can send an encrypted message to recipients without any prior or subsequent interaction. The applications described assume that *nym* will be a value derivable by the sender, such as the recipient's IP address or domain name, whereas ω values are not guaranteed to be derivable using publicly available information (and indeed are not derivable in the implementation given in [7]).

After receiving the Ω values, the sender creates a ciphertext for each $\omega \in \Omega$ each time $HC_{simpleE}$ is called. This produces the expected $O(|\Omega|)$ increase in space.

If *Enc* has CCA2 security and Credential Indistinguishability, $HC_{simpleE}$ can safely be used with either the original [11] or improved [4] secret splitting schemes to construct *Encrypt*.