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

September 17, 2005

Abstract

We compare four recent systems which have often been cited together, yet which have significant, subtle differences. We argue that the systems are not as interchangeable as others have suggested, attempt to correct common misconceptions about the systems, and suggest several potentially rich avenues of future work.

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. All 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.

The first system proposed was called Secret Handshakes [1], 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) [9], 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. Finally, Hidden Credentials [7] were introduced, 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. All three schemes, as well as the CA-Oblivious Encryption scheme given in [4], give proofs of security in the random oracle model (ROM).

Since then, a flurry of papers have been written in this new vein of research, most of which cite all three systems as related work. However, many have missed subtle but significant differences between them. For instance, a paper titled "Secret Handshakes from CA-Oblivious Encryption" [4] gives a Computational Diffie-Hellman (CDH) implementation of Secret Handshakes based closely on the definition of OSBE, but requires a property unspecified in the OSBE definition, leaving it an open question whether OSBE's abstract requirements are sufficient to create Secret Handshakes. The paper also claims in passing to provide the needed ingredients for a Hidden Credentials implementation, a claim which we examine more closely 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 compatibilities and incompatibilities. Note that only Hidden Credentials and CA-Oblivious Encryption seem to fully provide the requirements of the other systems as specified. Also note that only OSBE and Hidden Credentials have been considered in the context of complex access control policies, and that while CA-Oblivious, OSBE and Hidden Credentials systems are all fundamentally based on preserving secrecy of plaintexts against unqualified recipients, Secret Handshakes are unique in being fundamentally a key agreement protocol.

This paper is not merely a survey of existing work, condensing and summarizing information already available in the literature. Rather, it summarizes existing work as a basis for demonstrating that several generalizations made in the literature are not in fact true, then proposing ways to increase compatibility between the systems.

In particular, we propose a complete implementation of Hidden Credentials from CA-Oblivous Encryption in section 4.1.1, which presents several intrinsic difficulties hitherto ignored in the literature.

2 Common Characteristics

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 we nonetheless need to accomplish transactions.

Paradoxically, 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. It has yet to be seen whether the privacy features taken for granted in traditional systems can be applied to these new systems as well.

3 System Overviews

In this section, we briefly describe the abstract requirements of each system, giving implementation details when they are necessary for the discussion in the next section, which discusses how the systems can be related to each other.

3.1 CA-Oblivious Encryption

CA-Oblivious schemes [4] are built on 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 token ω and secret trapdoor t corresponding to any attribute string it wishes to certify. Message recipients provide ω along with a *nym* 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. Because senders must obtain a public value from recipients before encrypting, table 1 lists this system as a public key cryptosystem.

For such a PKI-enabled cryptosystem to be CA-Oblivious, it must be both **Sender Oblivious** and **Receiver Oblivious**. Sender obliviousness ensures that users can safely release their ω values without leaking information about which CAs issued their credentials. Receiver obliviousness ensures that unqualified recipients cannot distinguish valid messages encrypted against a particular CA from random data.

The authors define indistinguishability games for these properties for a one way encryption system, then mention that such a system can then be extended to provide CPA and CCA security using standard transformations. Their implementation 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

	CA-Oblivious	Hidden Credentials	OSBE	Secret Handshakes
Encryption	Public key	Identity-based	Interactive	Key Agreement
Assumption	BDH,CDH	BDH,CDH(note 1)	BDH,CDH,QR,RSA	BDH,CDH,RSA
Roles/Attributes	√	√	√	√
Complex policy support		√	(note 2)	
Hidden Policy Support		√		
Non-omniscient CA	√			
Multi-show				√
Use with existing certs			√	
Traitor tracing				√
Can implement:				
Secret Handshakes	√	√		√
OSBE	√	√	√	
Hidden Credentials	(note 1)	√		
CA-Oblivious	√	√		

Table 1: Approximate feature comparison; see text for specifics. **Note 1:** See section 4.1.1 for details on implementing Hidden Credentials with CA-Oblivious Encryption. **Note 2:** Later systems GOSBE [10] and OACerts [8] added complex policy support and selective disclosure.

we examine here, which offer extremely good privacy protection for parties yet leave CAs almost entirely omnipotent.

3.2 Hidden Credentials

Hidden Credentials schemes have four functions: $CA\_Create()$, $CA\_Issue(nym, attribute)$, $HC_E(M, nym, P)$, and $HC_D(C, Creds)$, which create a CA, issue users a secret corresponding to the certified *attribute* about *nym*, encrypt M based on a policy P of attributes which *nym* must possess as certified by specified CAs, and decrypt a ciphertext C using the credentials in *Creds*. To implement their system, they defined functions $C = HC_{simpleE}(R, nym, p)$ and $R = HC_{simpleD}(C, s)$, which encrypt and decrypt a resource R contingent upon a single term policy p that requires the recipient's knowledge of secret s from a particular CA corresponding to *nym* and a specified *attribute*. They then constructed a simple secret splitting scheme which securely implements the multi-term policy accepting HC_E given a secure single-term function $HC_{simpleE}$. Because message senders require only an identity string (*nym*) to encrypt, we classify this system as identity-based in table 1.

The unique security requirement of a Hidden Credentials system [7] is called **Credential Indistinguishability**, meaning that ciphertexts encrypted against different single-element policies using $HC_{simpleE}$ must be indistinguishable to an attacker not possessing any of the corresponding credentials. A later paper [3] formalized the notions of **Policy Indistinguishability**, in which ciphertexts encrypted against multiple-element policies are secure against unqualified attackers. Further work [6] makes even 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 are given a concrete implementation using the Boneh-Franklin IBE, which was then optimized in the later paper. That IBE is based on the Bilinear Diffie-Hellman (BDH) assumption, which is described along with the IBE in [2].

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 [9] 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**. It must be **oblivious**, meaning that the sender cannot distinguish between qualified and unqualified recipients (equivalent to the "sender oblivious" property defined for CA-Oblivious systems).

Later work specified Generalized OSBE (GOSBE) [10], which allows messages to be encrypted against a boolean policy, much like the original Hidden Credentials system. Even more recently, OACerts were introduced [8], which add more sophisticated policy semantics, selective disclosure and zero-knowledge proofs. See below for comparison with the policy support in Hidden Credentials.

OSBE has the most different implementations among the systems discussed here, including an RSA implementation as well as implementations under both the Boneh-Franklin and Cocks IBE systems, which operate under the BDH and Quadratic Residue (QR) assumptions, respectively.

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.

3.4 Secret Handshakes

The abstract definition for a secret handshake scheme as given in [1] comprises five functions: $SH.CreateGroup(G)$ creates a group of users G, returning the group secret $GroupSecret_G$. $SH.AddUser(U, G, GroupSecret_G)$ returns the secret $UserSecret_{U,G}$ corresponding to user U's membership in G. U may be a simple nym, or a concatenation of a nym and role. $SH.Handshake(A, B)$ ensures that B learns whether $A \in G$ only if $B \in G$, and that A learns whether $B \in G$ only if $A \in G$. $SH.TraceUser(T)$ given a transcript T, returns which users participated in the transaction. $SH.RemoveUser(RevokedUserList, U)$ adds U 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 [4]. Vergnaud also gave several variants of an RSA-based implementation of Secret Handshakes [11].

3.4.1 Secret Handshake Security

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.

Later, the authors also described 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. Unlinkability is trivially achieved by using one-time pseudonyms, and has also been achieved cryptographically [12].

4 Cross-system Implementations

Here we consider claims about which systems can be used to implement the others. Since Secret Handshakes alone require that both parties have a credential from the same issuer, they show no immediate promise in being used to implement the other systems.

4.1 CA-Oblivious Encryption

Secret Handshakes from CA-Oblivious Encryption is the title of the paper which 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 also point out that their specification of sender obliviousness corresponds directly with OSBE's obliviousness requirement, whereas OSBE has no corresponding receiver obliviousness property. Consequently, they claim their system (or, presumably, a transformed CPA-secure version thereof) is always a correct OSBE implementation.

4.1.1 Hidden Credentials from CA-Oblivious Encryption

The authors of the CA-Oblivious scheme claim in passing that their scheme can be used to implement Hidden Credentials. While Receiver Obliviousness is virtually identical to the Hidden Credentials definition of Credential Indistinguishability, the ω values used by CA-Oblivious encryption present a problem.

In the Hidden Credentials protocol given in section 6 of [7], Alice and Bob first exchange nyms. Then Alice encrypts her resource request using HC_E against Bob's nym and a policy specifying what credentials Bob must possess if he is to understand her potentially very sensitive request. Bob responds with the resource Alice requested, encrypted against Alice's nym, the policy protecting the resource, and any policies protecting Bob's credentials which he has implicitly revealed by demonstrating that he understood Alice's request. Throughout the protocol, it is assumed that each participant's credentials were all issued using the same nym.

Implementing the protocol using CA-Oblivious Hidden Credentials, Alice and Bob can still 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 sender obliviousness of the CA-Oblivious scheme guarantees that these values do not leak information about the issuing CAs. However, in doing so they disclose the number of credentials they possess. This type of leak is not formally defined in the Hidden Credentials system, but does present an uncomfortable disclosure in a system designed for extremely sensitive credentials and access control policies. It may be possible for Alice and Bob to add additional, bogus values of ω to their message, converting the disclosure from a quantifier to an upper bound in exchange for additional network and computational overhead. If we accept this disclosure, then CA-Oblivious encryption's defined functions can be used to implement the required Hidden Credentials functions, listed in bold:

- **CA_Create**: 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.
- $\mathbf{CA\_Issue(nym, attribute)}$: Return $\langle t, \omega\rangle = Certify(join(nym, attribute))$.
- $\mathbf{HC_{simpleE}(R, nym, p, \Omega)}$: Let $\langle attribute, CA\_pub\rangle = p$.
  Return $\mathcal{C} = \langle c_1 ... c_n\rangle$
  $|c_i = Enc_{PK_i}(R)$
  $|PK_i = Recover(CA\_pub, join(nym, attribute), \omega_i)\forall\omega_i \in \Omega$
- $\mathbf{HC_{simpleD}}(\mathcal{C}, \mathcal{T})$: Return $\bigcup Dec(c_i, t_i)\forall\langle t_i, \omega_i\rangle$.
- $\mathbf{HC_E(R, nym, P, \Omega)}$: Call $\mathbf{HC_{simpleE}}$ for each $p \in P$ as required by the secret splitting scheme to produce ciphertext C.

- $\mathbf{HC_D}(\mathbf{C}, \mathcal{T})$: Also unchanged. Returns R iff $\mathcal{T}$ contains a satisfying set for P.

Note the addition of Ω to $HC_{simpleE}$ and HC_E. The requirement of Ω exchange prevents the implementation's use in applications described in [7] where the pseudonym exchange step can be omitted due to conventions such as setting *nym* to each user's IP address or domain name, and reflects our classification of CA-Oblivious encryption as a public key cryptosystem rather than an identity-based system.

After receiving the Ω values, the sender creates a ciphertext for each $\omega \in \Omega$ each time $HC_{simpleE}$ is called. For the improved secret splitting scheme given in [3], this produces the expected $O(n)$ increase in space. But the original scheme in [7] uses nested calls to $HC_{simpleE}$ to implement AND operations in policy expressions, progressively encrypting the resource against each of the required attributes. This causes an exponential blowup in ciphertext size for this implementation which can be avoided by modifying $HC_{simpleE}$ to return a single encryption of R under a random key along with a vector of encryptions of the random key, instead of a vector of encryptions which must each be at least as long as the input plaintext.

Since *Enc* has CCA2 security and Receiver Obliviousness, $HC_{simpleE}$ has the requisite secrecy and Credential Indistinguishability, and can safely be used with either the original or improved secret splitting schemes to construct HC_E.

4.2 Hidden Credentials

While Hidden Credentials are most difficult to implement, they provide the simplest implementations of the other three systems. Since Hidden Credentials implement CA-Oblivious encryption, and CA-Oblivious encryption implements OSBE and Secret Handshakes, Hidden Credentials can obviously also implement these systems. OSBE's fundamental soundness and semantic security against the receiver are trivially provided by Hidden Credentials. OSBE's **obliviousness** property is virtually identical to the Sender obliviousness required by CA-Oblivious systems, and is thus also trivially achieved by Hidden Credentials systems.

4.3 CA-Oblivious Encryption from Hidden Credentials

The security properties required to implement Hidden Credentials are almost exactly the same as those required for CA-Oblivious encryption. Every CA-Oblivious cryptosystem must be both **Sender Oblivious** and **Receiver Oblivious**.

Sender obliviousness means that message senders cannot learn what CAs have issued the credentials held by message recipients. Sender obliviousness is necessary in the implementation given in [4] because recipients must provide a value ω to message senders allowing them to construct the recipient's public key, and this value is mathematically related to the recipient's credential. Since the Hidden Credentials encryption function requires no such value, and in fact involves no interaction with message recipients, sender obliviousness is trivially achieved by defining *Recover* and ω to be null.

Receiver obliviousness, conveniently, is a direct analog to the Credential Indistinguishability required by Hidden Credentials. Thus, any Hidden Credentials system trivially implements CA-Oblivious encryption.

4.4 CA-Oblivious Encryption from OSBE

Since OSBE defines no notion comparable with the "receiver oblivious" property in [4], implementing CA-Oblivious and Hidden Credentials encryption is immediately problematic. 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'.

However, techniques proposed my Desmedt [5] might prove useful in patching this leak.

4.5 Hidden Credentials from OSBE

Like the CA-Oblivious scheme, some OSBE implementations assume that users provide tokens which correspond to their credentials, causing further problems for Hidden Credentials implementations as described in section 4.1.1.

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 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.

4.6 Secret Handshakes from OSBE

Vergnaud gives an RSA-based implementation [11] 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.

5 Conclusion

Our results suggest that Hidden Credentials are most versatile in implementing other systems, but correspondingly have the most demanding specifications to meet. Hidden Credentials also most aggressively protect elements of a transaction such as the size of the sender's policy and the receiver's number of credentials. CA-Oblivious encryption provides the most reliable underlying assumption and has the potential to implement each of the other systems, while OSBE offers the largest range of underlying assumptions as well as the most richly varied set of policy operations. Secret Handshakes show promise in having unlinkable multi-show credentials.

In each case, the systems have significant differences from each other, and while they can sometimes be used to implement each other, no one system is a direct drop-in replacement for another. Authors should take care when choosing systems and characterizing them in related work summaries to avoid misappraising their feature sets.

6 Future Work

Hidden Credentials would greatly benefit from CA-Oblivious Encryption's underlying CDH assumption and the potential for issuing without omniscient CAs, although the transformation may come at a significant computational and communications cost, providing another avenue for future work. With strengthened requirements, OSBE's policy expressiveness could be used to strengthen any of the other systems. k-Anonymity features from Secret Handshakes would also be a great boon to each of the other systems. Hidden Credentials' attention to privacy suggests that the other systems might benefit from additional scrutiny as to details implicitly leaked by a transaction, and the techniques in [6] might be combined with the features suggested in [8] to create even richer policy semantics than are currently available.

References

[1] D. Balfanz, G. Durfee, N. Shankar, D.K. Smetters, J. Staddon, and H.C. Wong. Secret handshakes from pairing-based key agreements. In *Proceedings of the 2003 IEEE Symposium on*

Security and Privacy, pages 180–196, Oakland, CA, May 2003.

[2] 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.

[3] 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.

[4] 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.

[5] 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.

[6] 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.

[7] 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.

[8] J. Li and N. Li. OACerts: Oblivious Attribute Certificates. CERIAS TR 2005-26.

[9] 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.

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

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

[12] 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.