Oblivious Signature-Based Envelope

Ninghui Li*
Department of Computer Science
Stanford University
Gates 4B
Stanford, CA 94305-9045
ninghui.li@cs.stanford.edu

Wenliang Du†
Department of Electrical Engineering and Computer Science
Syracuse University
121 Link Hall
Syracuse, NY 13244
wedu@ecs.syr.edu

Dan Boneh‡
Department of Computer Science
Stanford University
Gates 4B
Stanford, CA 94305-9045
dabo@cs.stanford.edu

ABSTRACT

Exchange of digitally signed certificates is often used to establish mutual trust between strangers that wish to share resources or to conduct business transactions. Automated Trust Negotiation (ATN) is an approach to regulate the flow of sensitive information during such an exchange. Previous work on ATN are based on access control techniques, and cannot handle cyclic policy interdependency satisfactorily. We show that the problem can be modelled as a 2-party secure function evaluation (SFE) problem, and propose a scheme called oblivious signature-based envelope (OSBE) for efficiently solving the SFE problem. We develop a provably secure and efficient OSBE protocol for certificates signed using RSA signatures. We also build provably secure and efficient one-round OSBE for Rabin and BLS signatures from recent constructions for identity-based encryption. We also discuss other applications of OSBE.

1. INTRODUCTION

Consider the following scenario: user Alice has a certificate showing that she has top-secret clearance. To protect herself, Alice will only present the certificate to other parties who also have a top-secret clearance certificate. Similarly, user Bob has a top-secret certificate and he will only reveal his certificate to others who have top-secret clearance. Now imagine what happens when Alice and Bob wish to establish a secure session using automated trust negotiation techniques [17, 20, 21, 22, 24, 25]. Neither one is willing to present their certificate first. Consequently, they are stuck and cannot establish the session. We describe efficient cryptographic solutions to this problem. Our solutions work with standard certificate formats.

Exchanging digitally signed certificates is an increasingly popular approach for authentication and authorization in distributed systems. These certificates associate public keys with key holders' identity and/or attributes such as employer, membership of associations, credit card information, security clearance, and so on. Often, the attribute information contained in certificates is sensitive. The goal of a growing body of work on *automated trust negotiation (ATN)* [17, 20, 21, 22, 24, 25] is to protect this information. In ATN, each party establishes access control (AC) policies to regulate not only the granting of resources, but also the disclosure of certificates to opponents. (Engaging in a discussion about secret information can be viewed as an abstract resource protected by the AC policy that requires secret clearance certificates.) A negotiation begins when a requester requests to access a resource protected by an AC policy. The negotiation process consists of a sequence of exchanges of certificates and possibly AC policies. In the beginning, certificates that are not sensitive are disclosed. As certificates flow, higher levels of mutual trust are established, and AC policies for more sensitive certificates are satisfied, enabling these certificates also to flow. In successful negotiations, certificates eventually flow to satisfy the AC policy of the desired resource. A security requirement on ATN is that no certificate should flow to a party who does not satisfy the AC policy established for the certificate.

In the scenario we described in the beginning of this paper, current ATN protocols would conclude negotiation failure, because there is cyclic interdependency between two negotiators' AC policies. Existing ATN protocols require one negotiator to reveal its certificate first; however, if the receiver does not have top-secret clearance, the AC policy is violated. Reporting negotiation failure in this scenario is not very satisfactory, since both parties have top-secret clearance and it would be more productive for them to proceed. How to break this policy cycle? Observe that, in many cases, the secret information in a certificate is the signature created by the certificate authority. For example, Alice's certificate may contain her public key and some string representing "top-secret clearance"; these are often public information, but the fact that a trusted authority signed the certificate is sensitive. Using this observation, the cycle can be broken as follows: First, Bob sends the content, including the Certificate Authority's (CA) public key but not the signature, of his certificate to Alice.[1] Alice verifies that the content satisfies her re-

*Supported by DARPA through SPAWAR contract N66001-00-C-8015, by DOD MURI program under ONR Grant N00014-97-1-0505, and by DOD URI program under ONR Grant N00014-01-1-0795.
†Supported in part by Grant ISS-0219560 from the National Science Foundation.
‡Supported in part by NSF ITR and the Packard foundation.

PODC'03, July 13–16, 2003, Boston, Massachusetts, USA.

[1]To prevent Alice from guessing whether Bob has top-secret clear-

quirement, then conducts a joint computation with Bob such that in the end Bob sees Alice's certificate if and only if Bob has the CA's signature on the content he sent earlier. Bob concludes negotiation success and proceeds with Alice if he has the signature and successfully verifies that Alice has the right certificate. Bob aborts the negotiation process when he does not have the signature or when Alice does not have the right certificate. Bob learns whether Alice has the certificate only when he has the required certificate, and vice versa. This approach for breaking policy cycles requires solving the following 2-party Secure Function Evaluation (SFE) problem.

Problem 1. Let PK be a public key (the CA's public key). Let M and P be two messages. (M is the content of Bob's certificate without the CA's signature; P is Alice's complete certificate.) Let Verify be the verification algorithm of a signature scheme such that $\mathsf{Verify}_{PK}(M, \sigma) = \mathsf{true}$ when σ is PK's signature on M. Alice and Bob want to compute a family F of functions, parameterized by Verify, M and PK. Both parties have M and PK. Alice has private input P (Alice's certificate). Bob has private input σ (the CA's signature on M). The function F is defined as follows.

$$F[\mathsf{Verify}, M, PK]_{Alice}(P, \sigma) = \perp$$
$$F[\mathsf{Verify}, M, PK]_{Bob}(P, \sigma) = \begin{cases} P & \text{if } \mathsf{Verify}_{PK}(M, \sigma) = \mathsf{true}; \\ \perp & \text{otherwise.} \end{cases}$$

where $F[\mathsf{Verify}, M, PK]_{Alice}$ represents Alice's output, $F[\mathsf{Verify}, M, PK]_{Bob}$ represents Bob's output, and $\perp$ is a special symbol. In other words, our goal is that Alice learns nothing and Bob learns $F[\mathsf{Verify}, M, PK]_{Bob}(P, \sigma)$ without learning anything else.

The SFE problem can be solved using general solutions to 2-party SFE [23]; however, the general solutions are not efficient, since signature verification is done within the SFE. We propose the Oblivious Signature-Based Envelope (OSBE) scheme that solves the above 2-party SFE problem efficiently. Formal definition of OSBE will be given in Section 3. Informally, an OSBE scheme enables a sender to send an envelope (encrypted message) to a receiver, and has the following properties: the receiver can open the envelope if and only if it has a third party's (e.g. certification authority) signature on an agreed-upon message M. An OSBE scheme is *secure against the receiver* if a receiver who does not have the third-party's signature on M cannot open the envelope. An OSBE scheme is *oblivious* if at the end of the protocol the sender cannot tell whether the receiver has the signature on M or not.

In this paper, our focus is to find efficient OSBE constructions for existing signature schemes, rather than to develop new signature schemes that make OSBE easy. In addition, we look for protocols that do not involve any interaction with (trusted or semi-trusted) third parties, except for the generation of signatures on certificates. We present OSBE protocols for three existing signature schemes: RSA [16], Rabin [15], and BLS [6]. The RSA-OSBE protocol is two-round: one message from the receiver followed by one message from the sender. The receiver and the sender each computes two exponentiations. We prove in the Random Oracle Model [3] that our RSA-OSBE protocol is as secure as RSA signatures. We also show that any Identity Based public key Encryption (IBE) [18, 5, 11] scheme gives rise to an OSBE scheme for the signature scheme corresponding to the IBE scheme. We use IBE to build one-round OSBE protocols for Rabin and BLS. These two protocols involve only one message from the sender to the receiver.

The rest of this paper is organized as follows. We discuss other applications of OSBE and related work in Section 2, and give formal definition of OSBE and its security requirements in Section 3. In Section 4, we describe an OSBE protocol for RSA signatures and prove its security. In Section 5 we build a one round OSBE for Rabin and BLS signatures. We conclude in Section 6.

2. OTHER APPLICATIONS AND RELATED CONCEPTS OF OSBE

Our original motivation for OSBE comes from automated trust negotiation; however, OSBE can be used for other purposes. An OSBE scheme enables the sender to send a message with the assurance that it can be seen only by the receiver if it has appropriate certificates while at the same time protecting the receiver's privacy such that the sender does not know whether the receiver has the required certificates or not. In other words, OSBE performs access control on a message in an oblivious (or privacy preserving) fashion. We envision that OSBE could be used in other contexts (possibly in conjunction with other protocols) to provide such oblivious access control.

One application of OSBE is Oblivious Subscription. Consider an online publishing service that gives access of various documents to members of several organizations. Users need membership certificates to gain access to specific documents. OSBE enables users to gain access without disclosing which organizations they are members of. To do so, the publishing service encrypts all documents with distinct keys. When a user requests to access a document, it sends contents of some membership certificates it may or may not possess, and runs multiple rounds of OSBE protocol with the publishing service. The publishing service delivers decryption keys of the documents in corresponding envelopes. Only a user that has the required certificate can open the envelope and obtain keys to decrypt documents. The publishing service does not know what memberships the user has.

OSBE might also be used in the context of Private Information Retrieval (PIR) [8, 9, 10, 12, 14] to provide access control on the information being retrieved.

A problem related to OSBE that has been studied in the literature is Fair Exchange of Signatures (FES) [1, 2], which enables two parties to exchange signatures such that either both parties obtain the other parties' signature or no party obtains the other party's signature. FES protocols are useful in contract signing and other e-commerce transactions. A common approach to FES is verifiable encryption of signatures, i.e., a signature encrypted in a way such that one can verify that the right signature is being encrypted, one can also go to a trusted third party (TTP) to obtain the signature when necessary, but one cannot retrieve the signature without the TTP. The TTP is involved only if one party tries to cheat. There are several differences between OSBE and FES. First, the signatures involved in OSBE are not generated by the two parties involved in the protocols, but rather generated by certification authorities before the OSBE protocol is used. Second, in FES protocols, at some stage, one party learns that the other party has a signature without obtaining that signature. This does not satisfy the security require-

ance or not, Bob should follow the protocol to send the same thing even if he does not have the top-secret clearance certificate. This is possible because the content of certificate is not secret.

ments of OSBE. Because of the above two reasons, FES protocols cannot be used directly to achieve OSBE. Third, OSBE does not require a fair exchange of signatures. It is allowed that the receiver gets the sender's signature without sending its own signature, as long as the receiver has the required signature. In this sense, OSBE is weaker than fair exchange of signatures. This weaker requirement enables efficient OSBE protocols that do not involve third parties.

Another piece of related work is Brands' private certificates [7]. There, the main goal is that certificates can be used anonymously. Our goal is different; we want a simultaneous exchange of attribute information that works with current standards, e.g., X.509 attribute certificates with RSA signatures.

3. OBLIVIOUS SIGNATURE-BASED ENVELOPE (OSBE): DEFINITION

In this section, we give formal definition of OSBE. We will use the following terminology. A function is *negligible* in the security parameter t if, for every polynomial p, $f(t)$ is smaller than $1/|p(k)|$ for k large enough; otherwise, it is *nonnegligible*. An *adversary* is a probabilistic interactive Turing Machine.

In the following definition of OSBE, we use one sender S and two receivers R_1 and R_2. Receiver R_1 has a third party's signature on some message M. Receiver R_2 does not have the signature. In each protocol run, the sender S interacts with either R_1 or R_2.

Definition 1. [Oblivious Signature-Based Envelope (OSBE)]

An Oblivious Signature-Based Envelope (OSBE) scheme is parameterized by a signature scheme Sig. It involves a sender S and two receivers R_1 and R_2. An OSBE scheme has the following three phases:

Setup The Setup algorithm takes a security parameter t and creates system parameters, which include a signing key whose public key is denoted by PK. Two messages M and P are chosen. PK and M are given to all three parties, namely, S, R_1, and R_2. In addition, the sender S is given P and the receiver R_1 is given the signature $\sigma = \mathsf{Sig}_{PK}(M)$.

Interaction One of R_1 and R_2 is chosen as R, without S knowing which one. S and R run an interactive protocol.

Open After the interaction phase, if $R = R_1$, i.e., R_1 was chosen in the interaction phase, R outputs the message P. (R can do that because it knows $\mathsf{Sig}_{PK}(M)$.) Otherwise, when $R = R_2$, R does nothing.

An OSBE must satisfy three properties defined below. It must be sound, oblivious, and semantically secure against the receiver.

Sound. An OSBE scheme is *sound* if in the open phase, R_1 can output the message P with overwhelming probability, that is, the probability that R_1 cannot output P is negligible.

Oblivious. An OSBE scheme is *oblivious* if the sender S does not learn whether it is communicating with R_1 or R_2. More precisely, no adversarial sender $\mathcal{A}$ has a nonnegligible advantage against the Challenger in the following game: The Challenger finishes the setup phase, and sends PK, M, P to the adversary. The Challenger picks random $b \in \{0,1\}$, then interacts with the adversary by emulating R_b. Finally, the adversary outputs $b' \in \{0,1\}$. The adversary wins the game if $b = b'$. In other words, an OSBE scheme is *oblivious* if for every probabilistic interactive Turing Machine $\mathcal{A}$, $|\mathsf{Pr}[\mathcal{A}$ wins the above game$] - \frac{1}{2}| \leq f(t)$, where f is a negligible function in t. (The adversary cannot do substantially better than random guessing.)

Semantically secure against the receiver. An OSBE scheme is *semantically secure against the receiver* if R_2 learns nothing about P. More precisely, no polynomially bounded adversary $\mathcal{A}$ has a nonnegligible advantage against the Challenger in the following game: The Challenger finishes the setup phase, and sends PK and M to the adversary. The adversary responds with two messages P_0 and P_1. The Challenger picks a random $b \in \{0,1\}$, then interacts with the adversary by emulating the sender S using message $P = P_b$. Finally, the adversary outputs $b' \in \{0,1\}$. The adversary wins the game if $b = b'$. In other words, even if we give the adversary the power to pick two messages P_0 and P_1 of its choice, it still cannot distinguish an envelope containing P_0 from one containing P_1. This formalizes the intuitive notion that the envelope leaks no information about its content.

We now argue that OSBE is an adequate solution to the 2-party SFE problem in Problem 1, by showing intuitively that the above security properties defined for OSBE suffice to prove that the scheme protects the privacy of the participants in the malicious model. Observe that our definitions allow arbitrary adversaries, rather than just those following the protocol. The oblivious property guarantees that the sender's view of any protocol run can be simulated using just the sender's input, because one can simulate a protocol run between S and R_2, who has no private input. Soundness and semantic security against the receiver guarantee that the receiver's view can be simulated using just the receiver's input and output. If the receiver has the signature, then the message P is in the output, one can therefore simulates the sender S. If the receiver does not have the signature, one can simulate the sender S with a arbitrary message P' and no polynomially bounded receiver can tell the difference.

We assume that OSBE is executed on top of a secure communication channel that the sender and the receiver has already established. This assumption is common in secure multiparty computation literature. In the context of automated trust negotiation, this assumption is also valid, since secure communication is already required to protect against eavesdroppers. Technically, an SSL connection can be established between the sender Alice and receiver Bob using self-signed certificates. When Alice and Bob wants to use OSBE to break a policy cycle, Bob first sends M (the content of Bob's certificate) to Alice. At this time, Alice verifies that the public key in M is the same as the one Bob used to establish the communication channel and then runs the OSBE protocol to send P (Alice's certificate) to Bob. At the end of the OSBE, Bob verifies that the public key in P is the same as the one Alice used to establish the communication channel. A man-in-the-middle attack during the OSBE will not be a problem.

In our proofs, we often use the random oracle model, which is an idealized security model introduced by Bellare and Rogaway [3] to analyze the security of certain natural cryptographic constructions. Roughly speaking, a random oracle is a function $H : X \rightarrow Y$ chosen uniformly at random from the set of all functions $\{h : X \rightarrow Y\}$ (we assume Y is a finite set). An algorithm can query the ran-

dom oracle at any point $x \in X$ and receive the value $H(x)$ in response. Random oracles are used to model cryptographic hash functions such as SHA-1. Note that security in the random oracle model does not imply security in the real world. Nevertheless, the random oracle model is a useful tool for validating natural cryptographic constructions. Security proofs in this model prove security against attackers that are confined to the random oracle world.

4. AN OSBE SCHEME FOR RSA SIGNATURES

In this section, we present an OSBE scheme for RSA signatures (i.e. when user certificates are signed using RSA). The RSA signature scheme [16] is as follows. The key space $\mathcal{K}$ is defined to be the following set:

$$\{(n,e,d) \mid n = pq,\ p,q \text{ equal size primes}, ed \equiv 1 \pmod{\phi(n)}\}$$

The values n and e are public, and the value d is secret.
For $K = (n,e,d)$, message M, and a message digest function $H : \{0,1\}^* \to \mathbb{Z}_n$, define

$$\mathsf{Sig}_K(M) = H(M)^d \bmod n$$
and $$\mathsf{Verify}_K(M,\sigma) = \mathsf{true} \iff H(M) \equiv \sigma^e \pmod{n}$$

Our RSA-OSBE scheme runs a Diffie-Hellman style key agreement protocol. If it is run between S and R_1, then R_1 can derive the shared secret. If it is run between S and R_2, then R_2 cannot derive the shared secret. Let $h = H(M)$, then the signature on the message M is $\sigma = (h^d \bmod n)$. R_1 sends to S a blinded version of the signature $\eta = (\sigma h^x \bmod n)$ for some random x. S then computes $\eta^e h^{-1} \bmod n$, which should be $h^{ex} \bmod n$. S now holds $(h^e)^x$ such that only R_1 knows the value x. This achieves half of the Diffie-Hellman key agreement protocol, with h^e as the base. S then does the other half and creates the envelope using a symmetric key derived from the shared secret.

Definition 2. [RSA-OSBE] Let H be the message digest function used in the signature. Let $\mathcal{E}$ be a semantically secure symmetric encryption scheme. Let H' be a function (e.g., a cryptographic hash function) that extracts a key for the symmetric encryption scheme from a shared secret.

Setup The setup algorithm takes a security parameter t and runs the RSA key generation algorithm to create an RSA key (n,e,d); in addition, it generates two security parameters t_1 and t_2, which are linear in t. In practice, $t_1 = t_2 = 128$ suffices. Two messages M and P are chosen. Party S is given (n,e), M, and P. Party R_1 is given (n,e), M, and $\sigma = (H(M)^d \bmod n)$. Party R_2 is given (n,e) and M.

Interaction We use $x \leftarrow [1..2^{t_1}n]$ to denote that x is randomly chosen from $[1..2^{t_1}n]$. In the following protocol, we describe actions for S, R_1, and R_2. However, during each protocol run, only one of R_1 and R_2 is involved as the receiver R.

- R_1 sends to S: $\eta = (\sigma h^x \bmod n)$, in which $x \leftarrow [1..2^{t_1}n]$.
  R_2 sends to S: $\eta = (h^{x'} \bmod n)$, in which $x' \leftarrow [1..2^{t_1}n]$.
- S receives η, checks that $\eta \notin \{0,1,n-1\}$, picks $y \leftarrow [1..2^{t_2}n]$, computes $r = (\eta^{ey} h^{-y} \bmod n)$ and then sends to R the pair: $\langle \zeta = (h^{ye} \bmod n), C = \mathcal{E}_{H'(r)}[P] \rangle$.

Open R_1 receives $\langle \zeta, C \rangle$ from the interaction phase; it computes $r' = (\zeta^x \bmod n)$, and decrypts C using $H'(r')$.

To see that this scheme is sound, observe that $\zeta = (h^{ye} \bmod n)$ and when R is R_1, $\eta = (h^{d+x} \bmod n)$; therefore:

$$r = \eta^{ye} h^{-y} = h^{(d+x)ey}\, h^{-y} = h^{dey}\, h^{xey}\, h^{-y}$$
$$h^{xye} = \zeta^x = r' \pmod{n}$$

Thus S and R_1 share the same symmetric key.

The key idea of the RSA-OSBE scheme is that it converts R_1's knowledge of the e'th root of h to the knowledge of a discrete log with base h^e. The sender S then uses this fact to do a Diffie-Hellman style key agreement with R_1.

Before proving the oblivious property of RSA-OSBE, we introduce the following terminology. Two distribution families $\delta^0(t)$ and $\delta^1(t)$ are *statistically indistinguishable* if

$$\textstyle\sum_y |\mathsf{Pr}_{x \in \delta^0(t)}[x = y] - \mathsf{Pr}_{x \in \delta^1(t)}[x = y]| \text{ is negligible in } t.$$

If two distribution families are statistically indistinguishable, then there exists no algorithm that can distinguish the two distribution families with nonnegligible advantage by sampling from them.

THEOREM 1. *RSA-OSBE is oblivious.*

PROOF. It suffices to show that what R_1 and R_2 send in the first step are drawn from two distribution families that are statistically indistinguishable, i.e., for all h, n, and d, the two distribution families $\delta^0(t_1) = \{h^{d+x} \bmod n \mid x \leftarrow [1..2^{t_1}n]\}$ and $\delta^1(t_1) = \{h^{x'} \bmod n \mid x' \leftarrow [1..2^{t_1}n]\}$ are statistically indistinguishable.

Let o be the order of h, i.e., the smallest number j such that $h^j \equiv 1 (\bmod n)$. For any fixed t_1, both distributions have o points. The probability difference on any point is at most $1/(2^{t_1}n)$; the total difference is thus at most $o/(2^{t_1}n)$. Since $o \leq \phi(n) < n$, the statistical difference between the two distributions is less than $1/2^{t_1}$, which is negligible in t_1. Since t_1 is linear in t, the statistical difference is also negligible in t. □

THEOREM 2. *Assuming that there exists no polynomial algorithm that can forge an RSA signature on a message M with nonnegligible probability, and H' is modelled as a random oracle, RSA-OSBE is secure against the receiver.*

PROOF. RSA-OSBE uses a semantically secure symmetric encryption algorithm. When H' is modelled as a random oracle, RSA-OSBE is secure against the receiver when no receiver who does not have the signature can compute with nonnegligible probability the secret that the sender uses to derive the encryption key. More precisely, RSA-OSBE is secure against the receiver if no polynomially bounded adversary wins the following game against the Challenger with nonnegligible probability: The Challenger randomly picks a public key (n,e) and a message M, and gives them to the adversary. The adversary responds with a η such that $\eta \notin$

$\{0, 1, n-1\}$. The Challenger then pick a random y from $[1..2^{t_2}n]$ and sends the adversary $H(M)^{y^e} \bmod n$. The adversary then outputs r, and the adversary wins the game if $r = \eta^{ey} h^{-y} \bmod n$.

Given an attacker $\mathcal{A}$ that wins the above game with probability ϵ. We construct another attacker $\mathcal{B}$ that can successfully forge the RSA signature $H(M)^d \bmod n$ with probability ϵ', where $|\epsilon - \epsilon'|$ is negligible. $\mathcal{B}$ does the following (all arithmetic is mod n):

1. $\mathcal{B}$, when given (n, e) and M, gives (n, e) and M to $\mathcal{A}$ and gets η back.
2. $\mathcal{B}$ then computes $h = H(M)$, picks a random z from $[1..2^{t_2}n]$ and sends h^{1+ez} to $\mathcal{A}$. Note that $h^{1+ez} = h^{ed+ez} = h^{e(d+z)}$. Then $\mathcal{B}$ can get $r = \eta^{e(d+z)} h^{-(d+z)}$ from $\mathcal{A}$.
3. Note that $r = \eta^{1+ez} h^{-d} h^{-z}$. Since $\mathcal{B}$ knows η, h, e, and z, then $\mathcal{B}$ can compute h^d.

$\mathcal{B}$ succeeds in forging an RSA signature if and only if $\mathcal{A}$ wins the above game, i.e., successfully compute $(\eta^{ey} h^{-y} \bmod n)$. What $\mathcal{A}$ receives from the Challenger in the game is drawn from the distribution family $\{h^{e(d+z)} \mid z \leftarrow [1..n2^{t_2}]\}$. What $\mathcal{A}$ receives from $\mathcal{B}$ are drawn from $\{h^{ey} \mid y \leftarrow [1..n2^{t_2}]\}$. Using an argument similar to that in the proof of Theorem 1, it is easy to show that these two distribution families are statistically indistinguishable. Therefore, the difference between $\mathcal{A}$'s success probabilities in the two cases is negligible. □

RSA-OSBE does a Diffie-Hellman style key agreement that has the added twist that one party can recover the shared key only when knowing the signature. This construction may be useful for other purposes, in which case the following property of the RSA-OSBE scheme could be useful: no eavesdropping attacker against RSA-OSBE can recover the shared secret with nonnegligible probability, even if the eavesdropper knows the signature h^d. (This property is not required for OSBE because we assume secure communication channels.) We base the security on the CDH (Computational Diffie-Hellman) problem in $\mathbb{Z}_n^*$. The CDH problem is the following: given a finite cyclic group G, a generator $g \in G$, and group elements g^a, g^b, find g^{ab}. The difficulty of this problem is the security foundation of Diffie-Hellman key agreement protocol and many other protocols. The *CDH assumption* is that there exists no polynomial probabilistic algorithm that can solve the CDH problem. It is known that if the CDH problem in $\mathbb{Z}_n^*$ can be solved in polynomial time for a nonnegligible portion of all bases $g \in \mathbb{Z}_n^*$, then n can be factored in expected polynomial time [4].

THEOREM 3. *Under the CDH assumption on $\mathbb{Z}_n^*$, no eavesdropping attacker against RSA-OSBE can recover the shared secret with nonnegligible probability.*

PROOF. We prove that there exists no polynomial bounded algorithm that can solve the following problem with non-negligible probability (all arithmetic is mod n): given an RSA public key (n, e), which has corresponding private key d, and the following tuple $\langle h,\ h^d,\ h^{d+x},\ h^{ey} \rangle$, compute h^{exy}.

Given an algorithm $\mathcal{A}$ that solves the above problem, we construct another algorithm $\mathcal{B}$ that can solve the CDH problem in $\mathbb{Z}_n^*$. $\mathcal{B}$, when given $(g,\ g^a,\ g^b)$, picks a small prime e and outputs $\mathcal{A}((n, e), \langle h = g^e,\ g,\ h_2 = gg^a,\ h_3 = (g^b)^e \rangle)$. Let x denote $(ad \bmod \phi(n))$ and y denote $(bd \bmod \phi(n))$. Observe that $h_2 = (h)^{d+x}$, $h_3 = h^{ey}$; therefore, $h^{exy} = g^{e^2 d^2 ab} = g^{ab}$.

□

5. ONE-ROUND OSBE USING IDENTITY BASED ENCRYPTION

Next, we show how to implement a one-round OSBE using any Identity Based public key Encryption scheme (IBE). The one-round refers to the fact that during the interaction phase there is only one message — the sender sends a ciphertext to the recipient. As usual, the recipient is only able to decrypt if she has a third party's signature on some predefined message M. Using IBE we build a one-round OSBE where user certificates are signed using a Rabin [15] or BLS [6] signature.

Before we describe the one-round OSBE we briefly review the concept of Identity Based Encryption. IBE was first proposed by Shamir [18], but the first usable IBE systems were found only very recently [5, 11]. An IBE public key encryption scheme is a standard public key system with the added twist that any string can function as a public key. In such a system there is a third party that has a secret master-key that enables it to generate the private key corresponding to any public key string. This third party plays the role of a Certificate Authority (CA) in a standard PKI. There are also global IBE system parameters given to all users, as is the CA's root certificate in a standard PKI. Shamir's idea was that user Alice uses her name (or email address) as a public key, thus avoiding the need for a public key certificate. Alice obtains her private key from the third party. More details on using IBE can be found in [5].

Any secure IBE system gives rise to a signature scheme [5]: to sign a message M we view M as an IBE public key; the signature on M is the private key corresponding to the public key M. Here the signer has the IBE master-key that enables it to generate the signature on any message M. The main point is that this signature on M can also function as an IBE decryption key. For the two recently proposed IBE systems the associated signature schemes are Rabin signatures and BLS signatures.

We show how to build an OSBE from any IBE system. As usual, both the sender S and the receiver R have a certain message M. The sender wants to send an encrypted message P to the receiver R so that R is able to recover P only if R has the third party's signature on M. The OSBE based on a generic IBE system works as follows:

Setup. Run the setup algorithm of the IBE system to generate the third party's master-key and the global IBE system parameters, which are viewed as PK. Let M and P be two messages and let $\mathrm{Sig}_{PK}(M)$ be the IBE private key corresponding to M when M is viewed as a public key. The sender is given M and P. The receiver is given $\mathrm{Sig}_{PK}(M)$.

Interaction. The sender wants to send P to the receiver so that the receiver can only obtain P if she has the signature $\mathrm{Sig}_{PK}(M)$ on M. The sender encrypts P using M as an IBE public key and sends the resulting ciphertext C to the receiver.

Open. The receiver, using the private key $\mathrm{Sig}_{PK}(M)$ can decrypt C to obtain P.

The OSBE described above is clearly oblivious since S receives no

information from R. The semantic security of this OSBE follows from the security of the IBE system. We summarize this in the following theorem. The theorem refers to the standard notion of security for IBE systems (IND-ID-CCA) defined in [5].

THEOREM 4. *Let $\mathcal{E}_{IBE}$ be an IBE system that is semantically secure under a chosen ciphertext attack (IND-ID-CCA). Then the resulting OSBE is sound, oblivious, and secure against the receiver.*

PROOF. The oblivious property is trivial, as the sender receives no information at all during the interaction phase, and thus cannot tell whether the receiver has the signature or not.

Since $\mathrm{Sig}_{PK}(M)$ is the private key corresponding to M. The soundness property of the resulting OSBE scheme is immediate from the soundness property of the IBE scheme (given a private key and a message encrypted under the corresponding public key, one can decrypt the message).

In addition, if the resulting OSBE is not semantically secure against the receiver, then there exists an adversary $\mathcal{A}$ that wins the following game against the Challenger with nonnegligible probability: The Challenger gives PK and M to the adversary. The adversary responds with two messages P_0 and P_1. The Challenger picks a random $b \in \{0,1\}$ and gives the adversary C, which is the IBE encryption of P_b with M as the public key. The adversary outputs $b' \in \{0,1\}$ and wins if $b' = b$. $\mathcal{A}$ is a direct attacker against the semantic security of the IBE scheme. Therefore, the OSBE is semantically secure when the IBE system is semantically secure. □

In Appendix A, we describe an OSBE for Rabin signatures, using Cocks' IBE system [11]. In this OSBE, communication during the interaction phase is quite large. This is because encryption in Cocks' IBE is done bit by bit, and the ciphertext for each bit is a number in $\mathbb{Z}_n$ (about 1024 bits in a typical setting). In the rest of this section, we describe an OSBE for BLS signatures [6], using an IBE system due to Boneh and Franklin [5]. With this OSBE, the amount of communication during the interaction phase is small.

The BLS short signature scheme [6] is based on bilinear maps. A number of recent cryptographic constructions make use of such maps [13, 5, 19]. Let G_1, G_2 be two groups of prime order q. A bilinear map $e : G_1 \times G_1 \to G_2$ satisfies $e(g^x, g^y) = e(g,g)^{xy}$ for any $g \in G_1$ and $x, y \in \mathbb{Z}_q$. Using elliptic curves one can give examples of bilinear maps $e : G_1 \times G_1 \to G_2$ where the Computational Diffie-Hellman problem (CDH) in G_1 is believed to be hard. Throughout this section we let g be a generator of G_1.

The BLS signature scheme works as follows: the public key is $h = g^x \in G_1$ and the private key is $x \in \mathbb{Z}_q^*$. Let H be a hash function from $\{0,1\}^*$ to G_1. To sign a message M the signer computes $\sigma = H(M)^x \in G_1$. To verify a signature on M test that $e(g,\sigma) = e(h, H(M))$. When H is modelled as a random oracle the system is existentially unforgeable under a chosen message attack assuming CDH in G_1 is hard [6]. Note that a BLS signature is a single element of G_1. Using certain elliptic curves, elements in G_1 are represented as short strings, resulting in very short signatures.

To build an OSBE using BLS signatures we use the Boneh-Franklin IBE system [5]. We do not describe the system here, but note that in this IBE system, the private key corresponding to a public key $M \in \{0,1\}^*$ is exactly a BLS signature on M. Thus we can build a one-round OSBE out of this system as we did in the previous section. The advantage of this IBE system is that the encryption of a 128-bit message key results in a short ciphertext (two elements in a finite field). Encryption and decryption are also more efficient than in Cocks' system.

Given a bilinear map $e : G_1 \times G_1 \to G_2$, the OSBE works as follows:

Setup. Pick a random $x \in \mathbb{Z}_q^*$ and compute $h = g^x \in G_1$. The third party is given x. Let M and P be two messages. Let $\mathrm{Sig}(M)$ be the BLS signature on M, i.e. $\mathrm{Sig}(M) = H(M)^x \in G_1$. The sender is given h, M and P. The receiver is given h and $\mathrm{Sig}(M)$.

Interaction. The sender encrypts P using M as the public key and sends the resulting ciphertext CT to the receiver. The public key M is only used to encrypt a message key k which is then used to encrypt P.

Open. The receiver, using the private key $\mathrm{Sig}(M)$, decrypts the ciphertext CT to obtain P.

The security of this OSBE follows from the security of BLS signatures [3] and the security of the Boneh-Franklin IBE [11]. We summarize this in the following corollary of Theorem 4.

COROLLARY 5. *The OSBE above is sound, oblivious, and secure against the receiver, assuming that the bilinear Diffie-Hellman problem is hard for $e : G_1 \times G_1 \to G_2$.*

6. CONCLUSION

Automated Trust Negotiation (ATN) is an approach to regulate the flow of sensitive information. Previous work on ATN, which only uses access control techniques, cannot deal with cyclic policy interdependency satisfactorily. We showed that cyclic policy interdependency in ATN can be handled by solving a particular 2-party Secure Function Evaluation (SFE) problem. We introduced oblivious signature-based envelope (OSBE) as a solution to the SFE problem and mentioned that OSBE can be used in other privacy sensitive applications as well. We developed an OSBE protocol for RSA signatures. The protocol does not involve a third party, is provably secure and quite efficient. We also showed that identity-based encryption can be used to build efficient one-round OSBE for Rabin and BLS signatures.

An open problem is to find an efficient and provably secure OSBE scheme for DSA signatures. We are also investigating other applications of the OSBE concept.

Acknowledgement

We would like to thank Will Winsborough for helpful discussions and the anonymous reviewers for their helpful comments.

7. REFERENCES

[1] N. Asokan, Victor Shoup, and Michael Waidner. Optimistic fair exchange of digital signatures. *IEEE Journal on Selected Areas in Communications*, 18(4):591–610, April 2000.

[2] Feng Bao, Robert H. Deng, and Wenbo Mao. Efficient and practical fair exchange protocols with off-line TTP. In *Proceedings of the 1998 IEEE Symposium on Security and Privacy*, pages 77–89. IEEE Computer Society Press, May 1998.

[3] Mihir Bellare and Phillip Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In *Proceedings of the 1st ACM Conference on Computer and Communications Security*, pages 62–73. ACM Press, 1993.

[4] Eli Biham, Dan Boneh, and Omer Reingold. Breaking generalized Diffie-Hellman modulo a composite is no easier than factoring. *Information Processing Letters*, 70(2):83–87, 1999.

[5] Dan Boneh and Matt Franklin. Identity-Based Encryption from the Weil Pairing. In *Proceedings of Crypto 2001*, volume 2139 of *LNCS*, pages 213–229. Springer, 2001.

[6] Dan Boneh, Ben Lynn, and Hovav Shacham. Short Signatures from the Weil Pairing. In *Proceedings of Asiacrypt 2001*, volume 2248 of *LNCS*, pages 514–32. Springer, 2001.

[7] Stefan A. Brands. *Rethinking Public Key Infrastructures and Digital Certificates: Building in Privacy*. MIT Press, August 2000.

[8] Christian Cachin, Silvio Micali, and Markus Stadler. Computationally private information retrieval with polylogarithmic communication. In *Advances in Cryptology: EUROCRYPT '99*, volume 1592 of *LNCS*, pages 402–414. Springer, 1999.

[9] Benny Chor and Niv Gilboa. Computationally private information retrieval (extended abstract). In *Proceedings of the 29th Annual ACM Symposium on Theory of Computing*, pages 304–313. ACM Press, May 1997.

[10] Benny Chor, Oded Goldreich, Eyal Kushilevitz, and Madhu Sudan. Private information retrieval. In *Proceedings of the 36th IEEE Symposium on Foundations of Computer Science*, pages 41–50. IEEE Computer Society Press, October 1995.

[11] Clifford Cocks. An identity based encryption scheme based on quadratic residues. In *Eighth IMA International Conference on Cryptography and Coding*, number 2260 in LNCS, pages 360–363. Springer, December 2001.

[12] Yael Gertner, Yuval Ishai, Eyal Kushilevitz, and Tal Malkin. Protecting data privacy in private information retrieval schemes. In *Proceedings of the 30th ACM Symposium on Theory of Computing*, pages 151–160. ACM Press, May 1998.

[13] Antoine Joux. A one round protocol for tripartite Diffie-Hellman. In *Proceddings of the 4th Algorithmic Number Theory Symposium*, volume 1838 of *LNCS*, pages 385–394. Springer, 2000.

[14] Eyal Kushilevitz and Rafail Ostrovsky. Replication is not needed: Single database, computationally-private information retrieval. In *Proceedings of the 38th IEEE Symposium on Foundation of Computer Science*, pages 364–373. IEEE Computer Society Press, October 1997.

[15] Michael O. Rabin. Digitalized signatures as intractable as factorization. Technical Report MIT/LCS/TR-212, MIT Laboratory for Computer Science, January 1979.

[16] Ronald L. Rivest, Adi Shamir, and Leonard M. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, 21:120–126, 1978.

[17] Kent E. Seamons, Marianne Winslett, and Ting Yu. Limiting the disclosure of access control policies during automated trust negotiation. In *Proceedings of the Symposium on Network and Distributed System Security (NDSS'01)*, February 2001.

[18] Adi Shamir. Identity-based cryptosystems and signature schemes. In *Advances in Cryptology – Crypto'84*, volume 196 of *LNCS*, pages 47–53. Springer, 1984.

[19] Eric R. Verheul. Self-blindable credential certificates from the weil pairing. In *Advances in Cryptology: AsiaCrypt 2001*, number 2248 in LNCS, pages 533–551. Springer, 2001.

[20] William H. Winsborough and Ninghui Li. Towards practical automated trust negotiation. In *Proceedings of the Third International Workshop on Policies for Distributed Systems and Networks (Policy 2002)*, pages 92–103. IEEE Computer Society Press, June 2002.

[21] William H. Winsborough, Kent E. Seamons, and Vicki E. Jones. Automated trust negotiation. In *DARPA Information Survivability Conference and Exposition*, volume I, pages 88–102. IEEE Press, January 2000.

[22] Marianne Winslett, Ting Yu, Kent E. Seamons, Adam Hess, Jared Jacobson, Ryan Jarvis, Bryan Smith, and Lina Yu. Negotiating trust on the web. *IEEE Internet Computing*, 6(6):30–37, November/December 2002.

[23] Andrew C. Yao. How to generate and exchange secrets. In *Proceedings of the 27th IEEE Symposium on Foundations of Computer Science*, pages 162–167. IEEE Computer Society Press, 1986.

[24] Ting Yu, Xiaosong Ma, and Marianne Winslett. Prunes: An efficient and complete strategy for trust negotiation over the internet. In *Proceedings of the 7th ACM Conference on Computer and Communications Security (CCS-7)*, pages 210–219. ACM Press, November 2000.

[25] Ting Yu, Marianne Winslett, and Kent E. Seamons. Supporting structured credentials and sensitive policies through interoperable strategies for automated trust negotiation. *ACM Transactions on Information and System Security (TISSEC)*, 6(1), February 2003.

APPENDIX

A. ONE-ROUND OSBE WITH RABIN SIGNATURES

The Rabin signature scheme is similar to RSA, but one uses a public exponent $e = 2$, i.e. a signature on a message M is $H(M)^{1/2} \bmod N$. One just has to make sure that the square root exists.

To define Rabin signatures [15], let $n = pq$ be an RSA modulus with $p = q = 3 \bmod 4$. The public key is n and the signing key is p, q. Let $Q \subseteq \mathbb{Z}_n^*$ be the subset of $\mathbb{Z}_n^*$ containing all elements with Jacobi symbol 1. We know that the size of Q is approximately $n/2$. Let H be a hash function from $\{0,1\}^*$ to Q. Then for any $M \in \{0,1\}^*$ exactly one of $H(M)$ or $-H(M)$ are quadratic residues in $\mathbb{Z}_n^*$. To sign a message M the signer computes $\mathsf{Sig}(M) = (\pm H(M))^{1/2} \bmod n$ where the sign of $H(M)$ is chosen so that the square root exists. To verify the signature, test that $(\mathsf{Sig}(M))^2 = \pm H(M) \bmod n$. When H is modelled as a random oracle the system is existentially unforgeable under a chosen message attack assuming factoring RSA moduli is hard [3].

To build an OSBE using Rabin signatures we use Cocks' IBE system [11]. A private key in this system can be viewed as a Rabin signature of the public key. Cocks' IBE works as follows: the global parameters are simply "n" where $n = pq$ is an RSA modulus with $p = q = 3 \bmod 4$. The master-key is p, q. The private key corresponding to a public key $M \in \{0,1\}^*$ is $s = (\pm H(M))^{1/2} \bmod n$ (the sign of $H(M)$ is chosen so that the square root exists). To encrypt a plaintext bit $b \in \{0,1\}$ using the public key M one picks two random numbers $x_0, x_1 \in \mathbb{Z}_n^*$ such that the Jacobi symbols $\left(\frac{x_0}{n}\right) = \left(\frac{x_1}{n}\right) = (-1)^b$. The ciphertext is a pair (C_0, C_1) where $C_i = x_i + ((-1)^i H(M)/x_i) \bmod n$ for $i = 0, 1$. Suppose $H(M)$ is a quadratic residue in $\mathbb{Z}_n^*$. Then to decrypt a ciphertext (C_0, C_1), one computes the Jacobi symbol $\left(\frac{C_0 + 2s}{n}\right)$ which one can show is equal to $(-1)^b$ as required. If $-H(M)$ is a quadratic residue we use C_1 instead. The system can be shown to be semantically secure under a chosen ciphertext attack (IND-ID-CCA) in the random oracle model assuming that the problem of distinguishing quadratic residues from non-residues in Q is hard.

Note that in this system encryption of a plaintext P is done bit-by-bit. Thus, encrypting a 128-bit message key results in a long ciphertext – the ciphertext contains 256 elements in $\mathbb{Z}_n^*$. Nevertheless, this system gives a one-round OSBE using Rabin signatures.

The OSBE works as follows:

Setup. Generate an RSA modulus $n = pq$ where $p = q = 3 \bmod 4$. The third party is given the factorization of N. Let M and P be two messages. Let $\mathsf{Sig}(M)$ be the Rabin signature on M, i.e. $\mathsf{Sig}(M) = (\pm H(M)^{1/2}) \bmod n$. The sender is given n, M and P. The receiver is given n and $\mathsf{Sig}(M)$.

Interaction. The sender encrypts P bit-by-bit using M as the public key in Cocks' IBE and sends the resulting ciphertext CT to the receiver. For efficiency, one could pick a random block cipher message key k, encrypt P using k, and then encrypt k bit-by-bit using M as the public key.

Open. The receiver, using the private key $\mathsf{Sig}(M)$, decrypts the ciphertext CT to obtain P.

The security of this OSBE follows from the security of Rabin signatures [3] and the security of Cocks' IBE [11]. We summarize this in the following corollary of Theorem 4.

COROLLARY 6. *The OSBE above is sound, oblivious, and secure against the receiver, assuming that the problem of distinguishing quadratic residue from non-residues in Q is hard.*