A Construction for General and Efficient Oblivious Commitment Based Envelope Protocols

Jiangtao Li[1]* and Ninghui Li[2]

[1] Intel Corporation, 2111 NE 25th Avenue, Hillsboro, OR 97124
jiangtao.li@intel.com
[2] Department of Computer Science, Purdue University, West Lafayette, IN 47907
ninghui@cs.purdue.edu

Abstract. The notion of Oblivious Commitment Based Envelope (OCBE) was recently proposed; it enables attribute-based access control without revealing any information about the attributes. Previous OCBE protocols are designed by taking zero-knowledge proof protocols that prove a committed value satisfies some property and changing the protocols so that instead of one party proving to the other party, the two parties compute two keys that agree if and only if the committed value indeed satisfy the property. In this paper, we introduce a more general approach for designing OCBE protocols that uses zero-knowledge proof protocols in a black-box fashion. We present a construction such that given a zero-knowledge proof protocol that proves a committed value satisfies a predicate, we have an OCBE protocol for that predicate with constant additional cost. Compared with previous OCBE protocols, our construction is more general, more efficient, and has wide applicability.

1 Introduction

In attribute-based access control systems, access decisions are based on attributes of the requester, which are established by digitally signed certificates through which certificate issuers assert their judgements about the attributes of entities. Each certificate associates a public key with the key holder's identity and/or attributes such as employer, group membership, credit card information, date of birth, citizenship, and so on. Because these certificates are digitally signed, they can serve to introduce strangers to one another without online contact with the attribute authorities. In many scenarios, the attribute information in a certificate is sensitive and needs to be protected. The requester may want to disclose only the information that is absolutely necessary to obtain the resource from the server.

Recently, Li and Li [20] proposed a cryptographic primitive called Oblivious Commitment Based Envelope (OCBE) that enables oblivious access control; that is, it enables attribute-based access control without revealing any information about the attributes. Informally, in an OCBE scheme, the receiver has a private attribute value a which has been committed to the sender; the sender has a public predicate b (i.e., her access control policy) and a private message M. The sender and the receiver engage in

* Most of this work was done while the author was at Purdue University.

an interactive protocol such that in the end, the receiver gets the message M if and only if her attribute value satisfies the predicate, i.e., $b(a) = \mathsf{true}$. Furthermore, the sender learns nothing about the receiver's attribute value. Formal definition of OCBE will be reviewed in Section 2.

Li and Li [20] developed OCBE protocols for the Pedersen commitment scheme [25] and predicates such as $=, \neq, <, >, \leq, \geq$ as well as logical combinations of them. Their approach for designing the OCBE protocols is to take zero-knowledge proof protocols that prove a committed value satisfies some property and change the protocols so that instead of one party proving to the other party, the two parties compute two keys that agree if and only if the committed value indeed satisfy the property.

In this paper, we introduce a more general approach for designing OCBE protocols. Rather than taking zero-knowledge proof protocols and finding ways to change them, we use these protocols in a black box fashion. The basic idea is as follows. The receiver first sends a commitment of another attribute value to the sender, and then uses zero-knowledge proof protocols to prove that the committed value in the new commitment satisfies the policy. Finally, the receiver and the sender run a protocol such that the receiver can retrieve the message if and only if the values in the two commitments are the same, and the sender does not learn whether the two committed values are the same.

We apply this approach to the commitment scheme introduced by Fujisaki and Okamoto [17] and later extended by Damgård and Fujisaki [14]. We prove that our protocol is secure under the Strong RSA assumption and the Computational Diffie-Hellman assumption modulo an RSA modulus in the Random Oracle Model.

Compared with previous work [20], our approach has the following advantages:

- Our OCBE construction achieves a general result; i.e., if there exists a zero-knowledge proof of knowledge protocol that can prove a committed value satisfies a predicate, then we can build an OCBE protocol for that predicate with constant additional cost. As a result, one does not need develop new OCBE protocols for each new families of predicates from the scratch and prove their security. Instead, one can directly take advantage of the existence of efficient zero-knowledge proof protocols for properties of committed values.
- Our OCBE protocol is more efficient than the previous OCBE protocols [20]. The OCBE protocols in [20] for the comparison predicates other than the equality predicates have linear computation and communication costs; i.e., both the sender and the receiver need to perform $O(\ell)$ modular exponentiations where ℓ is the maximum length of the receiver's attribute. In comparison, our OCBE protocol has constant computation cost for comparison predicates.
- Unlike the OCBE protocols in [20] where the input range for the receiver is limited to $\mathbb{Z}_q$, where q is a prime; the set of the receiver's committed input in our OCBE protocol can be $\mathbb{Z}$, the set of all integers. This feature is particularly important for linear relation predicates, i.e., to test whether a committed value satisfies a linear relation over $\mathbb{Z}$.
- Our OCBE protocols are compatible with the anonymous credentials [24, 8, 11]. The OCBE scheme in [20] is designed primarily for Oblivious Attribute Certificates (OACerts) [20] and is not compatible with the anonymous credential systems. The reason is that, in [20], the commitment is computed by the trusted third party rather

than the receiver. We modify the definition of OCBE to let the receiver commit, so that the new definition is compatible with the anonymous credentials.

The rest of this paper is organized as follows. We first give our notations and review the definition of OCBE in Section 2. We next review in Section 3 cryptographic assumptions and tools that we use. In Section 4, we present our OCBE protocol and prove that it is secure. In Section 5, we describe how the OCBE protocol can be applied in attribute-based access control systems and automated trust negotiation systems, in particular, we show how our OCBE protocol can be used together with the anonymous credentials. We discuss the related work in Section 6 and conclude our paper in Section 7.

2 Review the Definition of OCBE

2.1 Notation

In the rest of this paper, we use the following notations. We say that $\mu(k)$ is a negligible function, if for every polynomial $p(k)$ and for all sufficiently large k, $\mu(k) < 1/p(k)$. We say $\nu(k)$ is overwhelming if $1 - \nu(k)$ is negligible.

If S is a probability space, then the probability assignment $x \leftarrow S$ means that an element x is chosen at random according to S. If S is a finite set, then $x \leftarrow S$ denotes that x is chosen uniformly from S. Let A be an algorithm, we use $y \leftarrow A(x)$ to denote that y is obtained by running A on input x. In case A is deterministic, then y is unique; if A is probabilistic, then y is a random variable. Let A and B be interactive Turing machines, we use $(a \leftarrow A(\cdot) \leftrightarrow B(\cdot) \rightarrow b)$ to denote that a and b are two random variables corresponding to the outputs of A and B as a result of their joint computation.

Let p be a predicate and $A_1, A_2, \ldots, A_n$ be n algorithms, We use $\Pr[\{x_i \leftarrow A_i(y_i)\}_{1 \leq i \leq n} : p(x_1, \cdots, x_n)]$ to denote the probability that $p(x_1, \cdots, x_n)$ will be true after running sequentially algorithms $A_1, \ldots, A_n$ on inputs $y_1, \ldots, y_n$.

2.2 Definition of OCBE

We now briefly review the definition of OCBE [20]. We slightly modify the definition to make the OCBE scheme compatible with the anonymous credential schemes. The difference between our new definition and the OCBE definition in [20] is that, in our new definition, we let the sender to commit rather than the trusted third party. Please refer to Section 5 for the detailed explanation. Note that, our OCBE protocol presented in Section 4 works both this new definition and the original definition in [20].

Definition 1 (OCBE). An OCBE scheme is parameterized by a commitment scheme commit. An OCBE scheme involves a sender, a receiver, and a trusted third party, and has the following four phases:

Setup The trusted third party takes a security parameter k and outputs public parameters P for commit, a set S of possible values, and a set B of predicates (i.e., boolean functions). Each predicate $b \in B$ maps each element in S to either true or false. The domain of commit contains S as a subset. The trusted third party sends $\langle P, S, B \rangle$ to the sender and the receiver.

Initialization The receiver chooses a value $a \in S$, computes the commitment $c = \mathsf{commit}(a, r)$ where r is a random number, and sends c to the sender. The sender chooses a message $M \in \{0,1\}^*$ and a predicate $b \in B$ and then reveals b to the receiver.

The sender has b, c, and M. The receiver has a, b, c, and r.

Interaction The sender and the receiver run an interactive protocol, during which an envelope containing an encryption of M is delivered to the receiver.

Open After the interaction phase, if $b(a)$ is true, the receiver outputs the message M. Otherwise, the receiver does nothing.

Note that, in the initialization phase, it is crucial for the sender to reveal the predicate b after the receiver has committed a. Otherwise, the receiver could choose a value a' that satisfies the predicate and commit a' rather than a, so that she can always open the envelope and obtain M in the end.

An OCBE scheme must satisfy the following three properties [20]. It must be sound, oblivious, and semantically secure against the receiver.

Sound. An OCBE scheme is *sound* if in the case that $b(a)$ is true and both the sender and the receiver are honest, the receiver can output the message M with overwhelming probability.

Oblivious. An OCBE scheme is *oblivious* if the sender learns nothing about a. More precisely, no adversary sender $\mathcal{A}$ has a non-negligible advantage against the challenger $\mathcal{C}$ in the following game: $\mathcal{C}$ first runs the setup program and sends $\langle P, S, B\rangle$ to $\mathcal{A}$. Then $\mathcal{A}$ chooses a random message $M \in \{0,1\}^*$, two values $a_0, a_1 \in S$, and a predicate $b \in B$, and sends a_0, a_1, b to $\mathcal{C}$. Next $\mathcal{C}$ chooses randomly $a \in \{a_0, a_1\}$, computes the commitment c for a, and interacts with $\mathcal{A}$ by emulating the receiver. In the end, $\mathcal{A}$ outputs $a' \in \{a_0, a_1\}$. The adversary wins the game if $a = a'$. An OCBE scheme is oblivious if

$$| \Pr\left[a' \leftarrow \mathcal{A}(1^k) \leftrightarrow \mathcal{C}(1^k) \rightarrow a : a' = a\right] - 1/2 | = \mu(k)$$

where the symbols $\leftarrow, \leftrightarrow, \rightarrow$ are defined in Section 2.1, and $\mu(k)$ is negligible function in k. In other words, the adversary cannot do substantially better than random guessing whether the receiver's committed value is a_0 or a_1.

Secure against the receiver. An OCBE scheme is *secure against the receiver* if the receiver learns nothing about M when $b(a)$ is false. More precisely, no adversary sender $\mathcal{A}$ has a non-negligible advantage against the challenger $\mathcal{C}$ in the following game: $\mathcal{C}$ first runs the setup program and sends $\langle P, S, B\rangle$ to $\mathcal{A}$. Then $\mathcal{A}$ chooses a value $a \in S$ and a value r[3], computes the commitment $c = \mathsf{commit}(a, r)$, and sends a, r, c to $\mathcal{C}$. $\mathcal{A}$ also chooses two equal length messages $M_0, M_1 \in \{0,1\}^*$ and sends M_0, M_1 to $\mathcal{C}$. Next, $\mathcal{C}$ chooses a predicate $b \in B$ such that $b(a)$ equals to false, chooses randomly a message $M \in \{M_0, M_1\}$, and interacts with $\mathcal{A}$ by emulating the sender. In the end, $\mathcal{A}$ outputs

[3] In [20], r is chosen by the challenger rather than the adversary. This is because in the original definition of OCBE [20], the trusted third party chooses r; whereas in our new definition, the receiver chooses r and computes the commitment.

$M' \in \{M_0, M_1\}$. The adversary wins the game if $M = M'$. An OCBE scheme is secure against the receiver if

$$| \Pr \left[M' \leftarrow \mathcal{A}(1^k) \leftrightarrow \mathcal{C}(1^k) \rightarrow M : M' = M \right] - 1/2 | = \mu(k)$$

where $\mu(k)$ is negligible function in k. In other words, even if we give the adversary the power to pick two equal-length messages M_0 and M_1 of her choice, she still cannot distinguish an envelope containing M_0 from one containing M_1.

3 Cryptographic Assumptions and Tools

In this section, we first review some standard assumptions in cryptography that we use, then review two cryptographic tools that shall be used in our OCBE protocol, one is the integer commitment scheme, the other is zero-knowledge proofs of knowledge.

3.1 Security Assumptions

In the rest of this paper, we shall use the following cryptographic assumptions and models, namely, the computational Diffie-Hellman assumption, the strong RSA assumption, and the random oracle model. The strong RSA assumption was introduced by Barić and Pfitzmann [3] and has been used in proving the security of many cryptographic schemes (e.g., [17, 10], to list a few).

Computational Diffie-Hellman (CDH) Assumption Given a finite cyclic group G, a group generator g, and group elements g^a, g^b; there exists no polynomial-time algorithm that can compute g^{ab} with non-negligible probability.

Strong RSA Assumption Given an RSA modulus n and a random value x in $\mathbb{Z}_n^*$, there exists no polynomial-time algorithm that can compute $e > 1$ and $y \in \mathbb{Z}_n^*$ with non-negligible probability such that $y^e = x \bmod n$.

Random Oracle Model The random oracle model is an idealized security model introduced by Bellare and Rogaway [5]. Roughly speaking, a random oracle is a function $H : X \rightarrow Y$ chosen uniformly at random from the set of all functions $\{h : X \rightarrow Y\}$. Random oracles are used to model cryptographic hash functions such as SHA-1.

3.2 Integer Commitment Scheme

Our OCBE protocol use the following integer commitment scheme introduced by Fujisaki and Okamoto [17] and later extended by Damgård and Fujisaki [14]. The reasons we choose this integer commitment scheme instead of the Pedersen commitment scheme [25] used in [20] are that: first, the input domain of this commitment scheme is the set of all integers rather than a set of values in a group; and second, this commitment scheme supports efficient proof that a committed value lies in a given interval [6]. This second feature shall be used to construct efficient OCBE protocol for greater-than-or-equal-to predicates.

Definition 2 (Integer Commitment Scheme).

Setup This step takes a security parameter k and outputs a special RSA modulus $n = pq$, such that $p = 2p'+1$ and $q = 2q'+1$ where p, q, p', q' are primes. It also outputs $h \leftarrow QR_n$ and $g \leftarrow \langle h \rangle$, where QR_n is the set of quadratic residues modulo n and $\langle h \rangle$ is the group generated by h. The public parameters of this commitment scheme are (n, g, h).

Commit The domain of the committed values is $\mathbb{Z}$. To commit an integer $a \in \mathbb{Z}$, the prover chooses $r \leftarrow \mathbb{Z}$ and computes the commitment $c = \mathsf{commit}(a, r) = g^a h^r \bmod n$.

Open To open a commitment c, the prover reveals a and r; then the verifier verifies whether $c = g^a h^r \bmod n$.

The integer commitment scheme is *statistically hiding*: under the factoring assumption, $\mathsf{commit}(a, r)$ statistically reveals no information to the verifier. More formally, there exists a simulator which outputs simulated commitments to a which are statistically indistinguishable from true commitments to a. This commitment scheme is *computationally binding*: the prover cannot commit herself two distinct values a_0 and a_1 by the same commitment unless she can factorize n. In other words, under the factoring assumption, it is computationally infeasible for the prover to compute a_0, a_1, r_0, r_1 where $a_0 \neq a_1$ such that $\mathsf{commit}(a_0, r_0) = \mathsf{commit}(a_1, r_1)$.

3.3 Zero-Knowledge Proofs of Knowledge

We now list a few known proof of knowledge protocols based on the Fujisaki and Okamoto commitment scheme. In the rest of this paper, all the computations are modulo n unless explicitly specified.

– Proof of knowledge on how to open a commitment [14]. That is, given the parameters (n, g, h) of the integer commitment scheme and a commitment c, the prover proves the knowledge of a and r such that $c = g^a h^r$. We denote the protocol as

$$\mathrm{PK}\{(a, r) : c = g^a h^r\}$$

– Proof that a committed value is equal to a given integer [14]. That is, given the parameters (n, g, h) of the integer commitment scheme, a commitment c, and an integer a_0; the prover proves the knowledge of a and r such that $c = g^a h^r$ and $a = a_0$. We denote the protocol as

$$\mathrm{PK}\{(a, r) : c = g^a h^r \ \wedge \ a = a_0\}$$

– Proof that a committed value lies in a given integer interval [6]. That is, given the parameters (n, g, h) of the integer commitment scheme, a commitment c, and integers a_0 and a_1; the prover proves the knowledge of a and r such that $c = g^a h^r$ and $a_0 \leq a \leq a_1$. We denote the protocol as

$$\mathrm{PK}\{(a, r) : c = g^a h^r \ \wedge \ a_0 \leq a \leq a_1\}$$

– Proof that a committed value is the product of two other committed values [14]. That is, given the parameters (n, g, h) of the integer commitment scheme and three commitments c, c_0, c_1, the prover proves the knowledge of a_0, a_1, r, r_0, r_1 such that $c = g^{a_0 a_1} h^r$, $c_0 = g^{a_0} h^{r_0}$, and $c_1 = g^{a_1} h^{r_1}$. We denote the protocol as

$$\mathrm{PK}\{(a_0, a_1, r, r_0, r_1) : c = g^{a_0 a_1} h^r \ \wedge \ c_0 = g^{a_0} h^{r_0} \ \wedge \ c_1 = g^{a_1} h^{r_1}\}$$

– Proof that a committed value has a linear relation with n committed values [14]. That is, given the parameters (n, g, h) of the integer commitment scheme, $n + 1$ commitments $c, c_1, \ldots, c_n$, and integers $z_0, z_1, \ldots, z_n$; the prover proves the committed values $a, a_1, \ldots, a_n$ satisfy the equation $a = z_0 + a_1 z_1 + \cdots + a_n z_n$. We denote the protocol as

$$\mathrm{PK}\{ \begin{matrix} (a, r, a_1, r_1, \ldots, a_n, r_n) : \{c_i = g^{a_i} h^{r_i}\}_{1 \le i \le n} \ \wedge \ c = g^a h^r \ \wedge \\ a = z_0 + a_1 z_1 + \cdots + a_n z_n \end{matrix} \}$$

All the above described protocols have constant computational and communication costs. These protocols are secure under the strong RSA assumption.

4 Our OCBE Protocol

In this section, we first present a construction that turns any zero-knowledge proof protocol that proves a committed value satisfies some predicate into an OCBE protocol for that predicate. Then we prove that our protocol is secure and compare our protocol with the several OCBE protocols described in [20].

4.1 Construction of OCBE Protocol

In [20], Li and Li developed several OCBE protocols for comparison predicates, i.e., one OCBE protocol for each type of predicates. In this paper, we present a more general and more efficient result:

Our result Let S be a set of integers and B be a set of predicates, such that each predicate $b \in B : S \to \{\mathsf{true}, \mathsf{false}\}$. Let commit be an integer commitment scheme described in Section 3.2. Suppose for every predicate $b \in B$, there exists an efficient zero-knowledge proof of knowledge protocol that can prove a committed value $a \in S$ satisfies the predicate b, i.e., $\mathrm{PK}\{(a, r) : c = \mathsf{commit}(a, r) \ \wedge \ b(a) = \mathsf{true}\}$. Then with constant additional cost, we can develop an efficient OCBE protocol for predicates set B.

Before we present the OCBE protocol, we briefly describe the intuition how the protocol works. Let (n, g, h) be the public parameters of the integer commitment scheme, a be the receiver's private input, c be the corresponding commitment. Consider the following scheme: If $b(a)$ is true, The receiver first proves to the sender that the value committed in c satisfies the predicate b; then the sender sends the message M to the receiver. Such scheme is secure against the receiver, but not oblivious; i.e., the sender learns some information about a.

In our proposed scheme, if $b(a)$ is true, then the receiver chooses $a' = a$; otherwise, she chooses a' such that $b(a') = \mathsf{true}$. Then the receiver commits a' to the sender; i.e., she chooses r, computes $c' = g^{a'} h^{r'}$, and gives c' to the sender. The receiver also proves that the value committed in c' satisfies the predicate b. Observe that the sender learns nothing about a — all she learns so far is that the value committed under c' satisfies b. Also note that, if $b(a)$ is true, then $a = a'$ and $c/c' = g^a h^r / g^{a'} h^{r'} = h^{r-r'}$; otherwise, $c/c' = g^{a-a'} h^{r-r'}$. In other words, if $b(a)$ is true, the receiver has the knowledge of $\log_h(c/c')$. The sender can use this fact to build a Diffie-Hellman style key-agreement, so that only $b(a)$ is true can the receiver obtain the encryption key.

Protocol 1 (OCBE) Let E be a semantically secure symmetric encryption scheme with keyspace $\{0,1\}^s$. Let $H : \mathbb{Z}_n \to \{0,1\}^s$ be a cryptographic hash function that extracts a key for E from an element in the group $\mathbb{Z}_n$. The OCBE protocol involves a sender, a receiver, and a trust third party.

Setup The trusted third party takes a security parameter k and runs the setup algorithm of the integer commitment scheme in Section 3.2 to create $P = (n, g, h)$. The third party also outputs an integer set $S \subseteq \mathbb{Z}$, and a set B of predicates. For each $b \in B$, $b : S \to \{\mathsf{true}, \mathsf{false}\}$. The trusted third party sends $\langle P, S, B \rangle$ to the sender and the receiver.

Suppose for each $a \in S$ and $b \in B$, if $b(a) = \mathsf{true}$, there is an efficient zero-knowledge proof of knowledge protocol $\mathrm{PK}\{(a, r) : c = g^a h^r \ \wedge \ b(a) = \mathsf{true}\}$.

Initialization The receiver chooses a value $a \in S$ and a value $r \in \mathbb{Z}$, computes the commitment $c = g^a h^r$, and sends c to the sender. The sender chooses a message $M \in \{0,1\}^*$ and a predicate $b \in B$ and then reveals b to the receiver.

The sender has b, c, and M. The receiver has a, b, c, and r.

Interaction The sender and the receiver run the following steps:

1. If $\forall a \in S, b(a) = \mathsf{false}$; then the sender and the receiver terminate the protocol immediately.
2. If $b(a) = \mathsf{true}$, the receiver set $a' = a$, otherwise the receiver randomly chooses a' such that $b(a') = \mathsf{true}$. The receiver chooses $r' \leftarrow \mathbb{Z}$ and computes the commitments $c' = g^{a'} h^{r'}$. The receiver sends c' to the sender.
3. The sender and the receiver runs the zero-knowledge proof of knowledge protocol
$$\mathrm{PK}\{(a', r') : c' = g^{a'} h^{r'} \ \wedge \ b(a') = \mathsf{true}\}$$
to prove that the value committed in c' satisfies the predicate b.
4. The sender picks $y \leftarrow \mathbb{Z}_n$, computes $\sigma = (c/c')^y$, and then sends to the receiver the pair $\langle \eta = h^y, C = E_{H(\sigma)}[M] \rangle$.

Open The receiver receives $\langle \eta, C \rangle$ from the interaction phase. If $b(a) = \mathsf{true}$, in other words $a = a'$, the receiver computes $\sigma' = \eta^{r-r'}$, decrypts C using $H(\sigma')$, and obtains M.

Observe that the computational cost of the above OCBE protocol is close to the cost of the zero-knowledge proof sub-protocol. More specifically, suppose in the zero-knowledge proof sub-protocol, the sender and the receiver need to perform ℓ_s and ℓ_v modular exponentiations, respectively; then in the OCBE protocol, the sender and the receiver need to perform at most $\ell_s + 2$ and $\ell_v + 1$ modular exponentiations, respectively.

4.2 Security Proofs

Our OCBE protocol invokes the zero-knowledge proof protocol $\mathrm{PK}\{(a, r) : c = g^a h^r \ \wedge \ b(a) = \mathsf{true}\}$ as a sub-protocol. We here need to examine the security properties of this zero-knowledge proof protocol, before we prove the security of the OCBE protocol. The security definition of a zero-knowledge proof of knowledge protocol derives from Bellare and Goldreich [4].

Given a commitment c and a predicate b as input. Let R be a polynomially computable relation defined as, given a and r, $R(a, r, c, b) = 1$ if and only if $c = g^a h^r$ and $b(a) = \mathsf{true}$. A zero-knowledge proof of knowledge of witness (a, r) such that $R(a, r, c, b) = 1$ is a probabilistic polynomial-time protocol between a prover P and a verifier V with the following properties:

1. Completeness: For all possible a and r such that $R(a, r, c, b) = 1$

$$\Pr\left[P(a, r, c, b) \leftrightarrow V(c, b) \rightarrow x : x = \mathsf{accept}\right] = 1 - \mu(k),$$

   where $\mu(k)$ is a negligible function.
2. Zero-knowledge: Intuitively, the verifier should not learn any information about a and r other than the fact that $R(a, r, c, b) = 1$. Formally speaking, there exists a probabilistic, expected polynomial-time simulator S such that, for every probabilistic polynomial-time verifier V_k, for all $(a, r, c, b) \in R$, the distribution $P(a, r, c, b) \leftrightarrow V_k(c, b)$ and $S(c, b) \rightarrowtail V_k(c, b)$ are computationally indistinguishable, where $S \rightarrowtail V$ means S has black-box access to algorithm V.
3. Soundness: Intuitively, if the prover does not know (a, r) such that $R(a, r, c, b) = 1$, the probability that the prover can convince the honest verifier is negligible. More formally, for all (c, b), for all probabilistic polynomial-time prover P_k, if there is a function $\epsilon(k)$ such that

$$\Pr\left[P_k(c, b) \leftrightarrow V(c, b) \rightarrow x : x = \mathsf{accept}\right] = \epsilon(k);$$

   then there exists a polynomial-time extractor E and a negligible function $\mu(k)$ such that

$$\Pr\left[P_k(c, b) \leftarrowtail E(c, b) \rightarrow (a, r) : R(a, r, c, b) = 1\right] = \epsilon(k) - \mu(k).$$

   In other words, if the prover can convince the verifier with probability $\epsilon(k)$, then the extractor can compute (a, r) with probability close to $\epsilon(k)$.

Theorem 1. *The OCBE protocol in Section 4.1 is sound.*

Proof. To show the OCBE protocol is sound, we consider two cases. In the first case, suppose for any $a \in S \Rightarrow b(a) = \mathsf{false}$, then both the sender and the receiver know that the receiver should not receive M, as the receiver's input will never satisfy the predicate b. Therefore, the sender and the receiver simply quit the protocol on step 1 of the interaction phase. In the second case, there exists some $a \in S$ such that $b(a) = \mathsf{true}$. In this case, steps 2-4 of the interaction phase will be executed. By the completeness property of the zero-knowledge proof protocol, the zero-knowledge proof protocol almost

always succeeds. If the value a committed by the receiver satisfies the predicate b, then $a' = a$. It follows that

$$\sigma = (c/c')^y = (g^{a-a'}h^{r-r'})^y = (h^{r-r'})^y = (h^y)^{r-r'} = \eta^{r-r'} = \sigma'.$$

The sender and the receiver share the same symmetric key that is used to encrypt the message M. Therefore, if $b(a) = \mathsf{true}$, the receiver can obtain the message M.

Theorem 2. *The OCBE protocol in Section 4.1 is oblivious.*

Proof. An OCBE scheme is oblivious if the sender learns nothing about the value the receiver committed. Let us first see which messages the sender receives during the initialization and interaction phases. In the initialization phase, the sender receives c, the commitment of a, from the receiver. In the interaction phase, the sender receives from the receiver c', the commitment of a'. The sender also involves in the zero-knowledge proof protocol $\mathrm{PK}\{(a', r') : c' = g^{a'}h^{r'} \ \wedge \ b(a') = \mathsf{true}\}$ as the verifier.

More formally, an OCBE scheme is oblivious if we can show that the view of the sender can be simulated. We build a simulator $\mathcal{S}$ as follows: Given a set S of integers and a predicate b, $\mathcal{S}$ chooses a random number $a_0 \leftarrow S$ and computes the commitment $c_0 = \mathsf{commit}(a_0)$. Then $\mathcal{S}$ chooses another value $a'_0 \leftarrow S$ such that $b(a'_0) = \mathsf{true}$ and computes the commitment $c'_0 = \mathsf{commit}(a'_0)$. Finally, $\mathcal{S}$ calls the simulator for the zero-knowledge proof protocol. As the commitment scheme is statistically hiding, the joint distribution of (c, c') is statistically indistinguishable from the joint distribution of (c_0, c'_0). By the zero-knowledge property of the zero-knowledge proof protocol, it is easy that the transcripts generated by the sender when interacting with the receiver is computationally indistinguishable with the transcripts generated by the simulator $\mathcal{S}$. Therefore, the OCBE protocol is oblivious.

Theorem 3. *The OCBE protocol in Section 4.1 is secure against the receiver.*

Proof. The preceding OCBE protocol uses a semantically secure symmetric encryption algorithm. Suppose H is modeled as a random oracle, the OCBE protocol is secure against the receiver when no receiver, whose committed value does not satisfy the predicate b, can compute with non-negligible probability the secret that the sender uses to derive the encryption key.

More precisely, the OCBE protocol is secure against the receiver if no polynomial-time adversary wins the following game against the challenger with non-negligible probability: The challenger runs the setup phase and sends $\langle P, S, B\rangle$ to the adversary, where $P = (n, g, h)$ is the public parameter of the integer commitment scheme. The adversary picks integers $a \in S$ and $r \in \mathbb{Z}$, computes the commitment $c = g^a h^r$, and sends a, r, and c to the challenger. The challenger responds by picking a predicate $b \in B$ such that $b(a) = \mathsf{false}$. The adversary chooses a value a' such that $b(a') = \mathsf{true}$, chooses $r \in \mathbb{Z}$, and computes $c' = g^{a'}h^{r'}$. The adversary sends c' to the challenger and also runs $\mathrm{PK}\{(a', r') : c' = g^{a'}h^{r'} \ \wedge \ b(a') = \mathsf{true}\}$ to prove that the value committed in c' satisfies the predicate b. The challenger then picks $y \leftarrow \mathbb{Z}_n$ and sends to the adversary h^y. The adversary then outputs σ, and the adversary wins the game if $\sigma = (c/c')^y$.

By the soundness property of the zero-knowledge proof protocol, the challenger can extract a' and r' from the zero-knowledge proof using the standard rewinding technique. That is, we can replace the zero-knowledge proof protocol in the above game with that the adversary sends a', r', and c' to the challenger. Note that, as $b(a) = \mathsf{false}$ and $b(a') = \mathsf{true}$, it follows that $a \neq a'$. Also note that, since the challenger runs the setup program of the integer commitment scheme, she knows how to factorize n.

Given an attacker $\mathcal{A}$ that wins the above game with probability ϵ, we construct another attacker $\mathcal{B}$ that can solve the computational Diffie-Hellman problem in $\langle h \rangle$, the group generated by h, with the same probability. $\mathcal{B}$ does the following:

1. $\mathcal{B}$, when given p, q, h, h^x, h^y where $n = pq$ is a special RSA modulus used in the commitment scheme and h is an element in QR_n, gives $P = (n, h^x, h)$, $S = \mathbb{Z}$, and B to $\mathcal{A}$. We use g to denote h^x.
2. $\mathcal{B}$ receives a, r, and c from $\mathcal{A}$, and verifies that $c = g^a h^r$. $\mathcal{B}$ chooses a predicate $b \in B$ such that $b(a) = \mathsf{false}$ and sends b to $\mathcal{A}$.
3. $\mathcal{B}$ receives a', r', and c' from $\mathcal{A}$. $\mathcal{B}$ verifies that $c' = g^{a'} h^{r'}$ and $b(a') = \mathsf{true}$. Then $\mathcal{B}$ sends h^y to $\mathcal{A}$.
4. $\mathcal{B}$ receives σ from $\mathcal{A}$, computes $\tau = \sigma h^{(r'-r)y}$ and outputs $\tau^{(a-a')^{-1} \bmod \phi(n)}$.
   If $\mathcal{A}$ wins the game, we get

$$\sigma = (c/c')^y = (g^{a-a'} h^{r-r'})^y = (g^y)^{a-a'} h^{(r-r')y}$$

$$\tau = \sigma h^{(r'-r)y} = (g^y)^{a-a'} = (h^{xy})^{a-a'}$$

   $\mathcal{B}$ outputs $\tau^{(a-a')^{-1} \bmod \phi(n)} = h^{xy}$.

$\mathcal{B}$ succeeds in solving the computational Diffie-Hellman problem if and only if $\mathcal{A}$ wins the above game, i.e., successfully compute $(c/c')^y$. Therefore, under the CDH assumption, the OCBE protocol is secure against the receiver.

4.3 Comparison with Previous OCBE Protocols

To compare the OCBE protocol with the OCBE protocols proposed by Li and Li [20], we first list all families of predicates that our OCBE protocol supports. All the zero-knowledge proof protocols listed below are summarized in Section 3.3; and they have constant computation and communication costs.

– *Equality Predicates*: Let B be the family of equality predicates. Each predicate b in B takes a_0 as a parameter, and $b(a)$ is equal to true if and only if $a = a_0$. As there exists an efficient zero-knowledge proof protocol

$$\mathrm{PK}\{(a, r) : c = g^a h^r \ \wedge \ a = a_0\},$$

  we can build an efficient OCBE protocol for this family of predicates. In this OCBE protocol, the receiver can open the sender's message if and only if her committed number a is equal to a_0.

– *Greater-Than-Or-Equal-To Predicates*: Let B be the family of Greater-Than-Or-Equal-To predicates. Each predicate b in B takes a_0 as a parameter, and $b(a)$ is equal to true if and only if $a \geq a_0$. As there exists an efficient zero-knowledge proof protocol

$$\mathrm{PK}\{(a, r) : c = g^a h^r \ \wedge \ a \geq a_0\},$$

we can build an efficient OCBE protocol for this family of predicates. In this OCBE protocol, the receiver can open the sender's message if and only if her committed number a is greater than or equal to a_0.

– *Other Comparison Predicates*: Besides $=$ and $\geq$, the other comparison operations include $<, >, \leq, \neq$. Since there exists efficient zero-knowledge protocols for these comparison operations, the corresponding OCBE protocols can be built.

– *Range Predicates*: Let B be the family of range predicates. Each predicate b in B takes a_0 and a_1 as parameters, and $b(a)$ is equal to true if and only if $a_0 \leq a \leq a_1$. As there exists an efficient zero-knowledge proof protocol

$$\mathrm{PK}\{(a, r) : c = g^a h^r \ \wedge \ a_0 \leq a \leq a_1\},$$

we can build an efficient OCBE protocol for this family of predicates. In this OCBE protocol, the receiver can open the sender's message if and only if her committed number a is in the range of (a_0, a_1).

– *Square Predicate*: Given an input a, this predicate outputs true if and only if a is a square, i.e., $a = x^2$ for some integer x. Note that

$$\mathrm{PK}\{(a, r, x) : c = g^a h^r \ \wedge \ a = x^2\}$$

can be constructed using the zero-knowledge proof that a committed value is the product of two other committed values as

$$\mathrm{PK}\{(a, r, x, r_x) : c = g^a h^r \ \wedge \ c_x = g^x h^{r_x} \ \wedge \ a = x \cdot x\}$$

Therefore, we can build an efficient OCBE protocol for the square predicate. In this OCBE protocol, the receiver can open the sender's message if and only if her committed number a is a square number.

– *Modular Equality Predicates*: Let B be the family of range predicates. Each predicate b in B takes z_0 and z_1 as parameters, and $b(a)$ is equal to true if and only if $a \equiv z_0 \pmod{z_1}$. For example, if $z_0 = 0$ and $z_1 = 2$, this predicate tests whether a is an even number. Note that

$$\mathrm{PK}\{(a, r) : c = g^a h^r \ \wedge \ a \equiv z_0 \pmod{z_1}\}$$

can be constructed using the zero-knowledge proof that a committed value has a linear relation with another committed value as

$$\mathrm{PK}\{(a, r, x, r_x) : c = g^a h^r \ \wedge \ c_x = g^x h^{r_x} \ \wedge \ a = z_0 + x z_1\}$$

Therefore, we can build an efficient OCBE protocol for this family of predicates. In this OCBE protocol, the receiver can open the sender's message if and only if her committed number a is equal to z_0 modulo z_1.

In Table 1, we give a detailed comparison between our OCBE protocols and the OCBE protocols developed by Li and Li [20]. For equality predicates, the OCBE protocol in [20] is slightly better than ours, although they have the same complexity. For other comparison predicates and range predicates, the OCBE protocols in [20] are more expensive than ours, i.e., their protocols have linear cost whereas ours require only constant computation. Also note that the OCBE protocols for the square predicate and the modular equality predicates are first developed in this paper.

	OCBE Protocols in [20]	Our OCBE Protocols
Equality Predicates	$O(1)$	$O(1)$
Other Comparison Predicates	$O(\ell)$	$O(1)$
Range Predicates	$O(\ell)$	$O(1)$
Square Predicate	$*$	$O(1)$
Modular Equality Predicates	$*$	$O(1)$

Table 1. Comparison between the OCBE protocols in [20] with our OCBE protocol in terms of computation and communication costs. In the table, ℓ is the length of the committed value, $*$ denotes that such predicates are not supported in the OCBE protocols.

5 Applications of Our OCBE Protocol

In this section, we discuss the applications of OCBE to attribute-based access control, right after we present how the OCBE protocol is used in access control systems.

5.1 How to Use the OCBE Protocol

Recall that in the initialization phase of an OCBE scheme, the receiver sends the commitment of her attribute to the sender. To use OCBE in attribute-based access control, the sender needs to make sure that the commitment from the receiver is indeed the commitment of the receiver's attribute. In other words, the receiver needs to show that the attribute committed to the sender is certified by a certificate authority. This can be done using either OACerts [20] or anonymous credentials [24, 8, 11].

OACerts OACerts is a certificate scheme developed by Li and Li [20]. In the OACerts scheme, instead of storing attribute values directly in the certificate, a certificate authority stores the commitments of these values in the certificate. These commitments are computed by the certificate authority. When the receiver interacts with the sender, she first reveals her OACerts; then she runs the OCBE protocol with the sender based on the commitments in the certificates.

Anonymous Credentials An anonymous credential system [24, 8, 11] is a credential system in which the transactions carried out by the same user cannot be linked. In the anonymous credential system proposed by Camenisch and Lysyanskaya [8], the attributes of a user are signed by a certificate authority using a specially designed signature scheme. To show an attribute, the user commits the attribute and proves that the

attribute in the commitment is the same as the attribute in the anonymous credential. When the receiver interacts with the sender, she first commits her attribute, then proves that the committed attribute is the same as in the anonymous credentials. In the end, the receiver runs the OCBE protocol with the sender based on that commitment. Note that the attribute is committed by the receiver, instead of by the certificate authority. That is the reason that, in the new definition of OCBE, the receiver chooses r and computes the commitment.

5.2 Applications to Attribute-Based Access Control

We list two applications of OCBE to attribute-based access control. The ideas of these applications come from [20]; we only sketch here.

Oblivious Access Control Suppose Alice is a client and Bob is a server. Alice wants to access some resources from Bob whose policy is based on Alice's attribute. Alice can first show her OACerts or anonymous credentials, then based on commitment of her attribute, she runs an OCBE protocol with Bob. In the OCBE protocol, Bob sends Alice an encrypted envelope such that Alice could open the envelope only if her attribute satisfies Bob's policy. That is, Bob can perform access control based on Alice's attribute values while being oblivious about Alice's attribute information.

Breaking Policy Cycles OCBE can be used to break policy cycles (see [23] for definition) in automated trust negotiation [29, 28, 30]. Consider the following scenario: Alice's policy is based on Bob's attribute and Bob's policy is based on Alice's attribute. As a result, none of them wants to reveal their attributes. Alice and Bob can run an OCBE protocol to break such cycles. See [22] for detailed discussions on how OCBE can be integrated into automated trust negotiation.

6 Related Work

An OCBE scheme can be seen as a special oblivious transfer. The oblivious transfer protocol was first introduced by Rabin [26]. In an oblivious transfer between Alice and Bob, Alice wants to send a message to Bob in such a way that with half probability Bob will receive the message, and with half probability Bob will receive nothing. Furthermore, Alice does not know which of the two events really happened.

Crescenzo, Ostrovsky, and Rajagopalan [13] introduced a variant of oblivious transfer called conditional oblivious transfer; in which Alice has a private input x_a and Bob has a private input x_b, they and shares with each other a public predicate b that evaluates over x_a and x_b. In the conditional oblivious transfer of a message M from Alice to Bob, Bob receives M only when the predicate holds, i.e., $b(x_a, x_b) = \mathsf{true}$; furthermore, Alice learns nothing about x_b or $b(x_a, x_b)$. OCBE is different from conditional oblivious transfer in that, in OCBE, Bob's private input is committed to Alice. Besides, the conditional oblivious transfer protocol for great-than-or-equal-to predicates [13] has the computation cost linear to the size of x_a and x_b, whereas our OCBE protocol for great-than-or-equal-to predicates has constant time performance.

Crépeau [12] introduced the notion of committed oblivious transfer. In committed oblivious transfer, Alice commits two bits: a_0 and a_1, and Bob commits a bit b. All three

commitments are shared by Alice and Bob. In the end, Bob learns a_b without learning anything else, and Alice learns nothing. Garay, MacKenzie, and Yang [18] gave an efficient construction of committed oblivious transfer under the universal composability framework. OCBE differs from committed oblivious transfer in that Bob's input is an arbitrary integer rather than a single bit.

Our work is also somewhat related to several cryptographic schemes that have been recently proposed for attribute-based access control. For example, oblivious signature based envelopes [23], hidden credentials [7, 19], secret handshakes [2, 9], pairing-based cryptography [27], anonymous identification [15], certified input private policy evaluation [21], hidden policies with hidden credentials [16], and policy-based cryptography [1] were proposed to address the privacy issues in access control, in particular, these schemes can be used to protect the requester's identities or attributes.

7 Conclusion

The OCBE scheme has been proved to be a useful primitive for privacy protection in attribute-based access control. In this paper, we improved the OCBE protocols in [20] and gave an efficient and general construction of OCBE. Our construction replies on the existence of efficient zero-knowledge proof of knowledge protocols that prove a committed value satisfying certain predicate. Our construction is secure under the CDH assumption and the strong RSA assumption in the random oracle model.

Acknowledgement

This work is supported by NSF IIS-0430274, NSF CCR-0325951, and sponsors of CERIAS. We thank the anonymous reviewers for their helpful comments.

References

1. Walid Bagga and Refik Molva. Policy-based cryptography and applications. In *Proceedings of the 9th International Conference on Financial Cryptography and Data Security*, February 2005.
2. Dirk Balfanz, Glenn Durfee, Narendar Shankar, Diana Smetters, Jessica Staddon, and Hao-Chi Wong. Secret handshakes from pairing-based key agreements. In *Proceedings of the IEEE Symposium and Security and Privacy*, pages 180–196, May 2003.
3. Niko Barić and Birgit Pfitzmann. Collision-free accumulators and fail-stop signature schemes without trees. In *Advances in Cryptology: EUROCRYPT '97*, volume 1233 of *LNCS*, pages 480–494. Springer, 1997.
4. Mihir Bellare and Oded Goldreich. On defining proofs of knowledge. In *Advances in Cryptology: CRYPTO '92*, volume 740 of *LNCS*, pages 390–420. Springer, 1992.
5. 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.
6. Fabrice Boudot. Efficient proofs that a committed number lies in an interval. In *Advances in Cryptology: EUROCRYPT '00*, volume 1807 of *LNCS*, pages 431–444, May 2000.

7. Robert Bradshaw, Jason Holt, and Kent Seamons. Concealing complex policies with hidden credentials. In *Proceedings of 11th ACM Conference on Computer and Communications Security*, pages 146–157, October 2004.
8. Jan Camenisch and Anna Lysyanskaya. An efficient system for non-transferable anonymous credentials with optional anonymity revocation. In *Advances in Cryptology: EUROCRYPT '01*, volume 2045 of *LNCS*, pages 93–118. Springer, 2001.
9. Claude Castelluccia, Stanislaw Jarecki, and Gene Tsudik. Secret handshakes from CA-oblivious encryption. In *Advances in Cryptology: ASIACRYPT '04*, pages 293–307, December 2004.
10. Ronald Cramer and Victor Shoup. Signature schemes based on the strong RSA assumption. In *Proceedings of the 6th ACM Conference on Computer and Communications Security*, pages 46–51. ACM Press, November 1999.
11. Jason Crampton and George Loizou. Administrative scope and role hierarchy operations. In *Proceedings of 7th ACM Symposium on Access Control Models and Technologies*, pages 145–154, June 2002.
12. Claude Crépeau. Verifiable disclosure of secrets and applications (abstract). In *Advances in Cryptology: EUROCRYPT '89*, volume 434 of *LNCS*, pages 150–154. Springer, 1990.
13. Giovanni Di Crescenzo, Rafail Ostrovsky, and S. Rajagopalan. Conditional oblivious transfer and timed-release encryption. In *Advances in Cryptology: EUROCRYPT '99*, volume 1592 of *LNCS*, pages 74–89, March 1999.
14. Ivan Damgård and Eiichiro Fujisaki. An integer commitment scheme based on groups with hidden order. In *Advances in Cryptology: ASIACRYPT '02*, volume 2501 of *LNCS*, pages 125–142. Springer, December 2002.
15. Yevgeniy Dodis, Aggelos Kiayias, Antonio Nicolosi, and Victor Shoup. Anonymous identification in ad hoc groups. In *Advances in Cryptology: EUROCRYPT 2004*, pages 609–626, May 2004.
16. Keith B. Frikken, Mikhail J. Atallah, and Jiangtao Li. Hidden access control policies with hidden credentials. In *Proceedings of the 3rd ACM Workshop on Privacy in the Electronic Society*, October 2004.
17. Eiichiro Fujisaki and Tatsuaki Okamoto. Statistical zero knowledge protocols to prove modular polynomial relations. In *Advances in Cryptology: CRYPTO '97*, volume 1294 of *LNCS*, pages 16–30. Springer, 1997.
18. Juan Garay, Philip MacKenzie, and Ke Yang. Efficient and universally composable committed oblivious transfer and applications. In *Theory of Cryptography, TCC 2004*, volume 2951 of *LNCS*, pages 297–316. Springer, 2004.
19. Jason E. Holt, Robert W. Bradshaw, Kent E. Seamons, and Hilarie Orman. Hidden credentials. In *Proceedings of the 2nd ACM Workshop on Privacy in the Electronic Society*, pages 1–8, October 2003.
20. Jiangtao Li and Ninghui Li. OACerts: Oblivious attribute certificates. In *Proceedings of the 3rd Conference on Applied Cryptography and Network Security*, volume 3531 of *LNCS*, pages 301–317. Springer, June 2005.
21. Jiangtao Li and Ninghui Li. Policy-hiding access control in open environment. In *Proceedings of the 24nd ACM Symposium on Principles of Distributed Computing*, pages 29–38. ACM Press, July 2005.
22. Jiangtao Li, Ninghui Li, and William H. Winsborough. Automated trust negotiation using cryptographic credentials. In *Proceedings of the 12th ACM Conference on Computer and Communications Security*, pages 46–57. ACM Press, November 2005.
23. Ninghui Li, Wenliang Du, and Dan Boneh. Oblivious signature-based envelope. In *Proceedings of the 22nd ACM Symposium on Principles of Distributed Computing*, pages 182–189. ACM Press, July 2003.

24. Anna Lysyanskaya, Ronald L. Rivest, Amit Sahai, and Stefan Wolf. Pseudonym systems. In *Proceedings of the 6th Workshop on Selected Areas in Cryptography*, volume 1758 of *LNCS*, pages 184–199. Springer, 1999.
25. Torben P. Pedersen. Non-interactive and information-theoretic secure verifiable secret sharing. In *Advances in Cryptology: CRYPTO '91*, volume 576 of *LNCS*, pages 129–140. Springer, 1991.
26. Michael O. Rabin. How to exchange secrets by oblivious transfer. Technical Report Memo TR-81, Aiken Computation Laboratory, Harvard University, 1981.
27. Nigel Smart. Access control using pairing based cryptography. In *Proceedings of the Cryptographers' Track at the RSA Conference 2003*, pages 111–121. Springer-Verlag LNCS 2612, April 2003.
28. William H. Winsborough and Ninghui Li. Safety in automated trust negotiation. In *Proceedings of the IEEE Symposium on Security and Privacy*, pages 147–160, May 2004.
29. 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.
30. 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*, 6(1):1–42, February 2003.