

Subset Membership Encryption and Its Applications to Oblivious Transfer

Fuchun Guo, Yi Mu, and Willy Susilo

Abstract—In this paper, we propose a novel cryptographic notion called subset membership encryption (SME), and provide a very efficient SME scheme. Given system parameter generated by an encryptor (Alice), a decryptor (Bob) generates a randomized privacy-preserved attribute token $P(G)$ from a set of attributes $G$. A message is encrypted using an attribute set $A$ chosen by Alice and $P(G)$ provided by Bob. It requires that the subset $A \subseteq G$ holds for Bob to decrypt the message. We propose a very efficient SME scheme, where both the size of $P(G)$ and ciphertext are short and independent of $G$ and $A$. In particular, it offers three useful and practical applications to oblivious transfer as follows.

1) $k$-out-of-$n$ oblivious transfer (OT). SME can be naturally applied to a two-round OT, which features a great communication efficiency especially for the receiver, where the receiver only sends two group elements to the message sender.

2) Priced oblivious transfer (POT). Our POT protocol allows a buyer to purchase any number of items in each transaction and hide selected items, price and balance from the vendor. In comparison with previous POT protocols, our protocol is more flexible and eliminates the restriction that a buyer can only purchase one item in a transaction. Our POT scheme is very efficient since it does not require any zero-knowledge proof or homomorphic encryption.

3) Restricted priced oblivious transfer (RPOT). We introduce a novel POT named RPOT where a vendor can set restrictions on items or prices in POT. For example, a seller could offer a discounted price to those buyers who have purchased some specific items previously from the same seller.

Index Terms—Encryption, Oblivious Transfer

I. INTRODUCTION

We introduce a novel concept of subset membership encryption (SME in short) as a useful cryptographic primitive. SME can be seen as a variant of public-key encryption. Given system parameter generated by an encryptor (Alice), a decryptor (Bob) generates a randomized privacy-preserved attribute token $P(G)$ from a set of attributes, denoted by $G$. Throughout this paper, $P(G)$ is referred to as an “attribute token”. When Alice encrypts a message with an attribute set $A$ chosen by herself and an attribute token $P(G)$ from Bob, Bob cannot decrypt the ciphertext unless the subset $A \subseteq G$ holds (See Figure 1 for description). This encryption notion allows Alice to define access policies of messages and Bob to hide other attributes in $G$.

SME is useful in the applications to oblivious transfer. In this paper, we show how to apply SME in the construction of two-round $k$-out-of-$n$ oblivious transfer, priced oblivious transfer and restricted priced oblivious transfer. With our proposed SME scheme, the SME-based oblivious transfer protocols exhibit more computational and communication efficiency compared to other oblivious transfer protocols in the literature.

A. Related Work

Subset membership encryption is related to the notion of membership encryption recently introduced by Guo, Mu, Susilo and Varadharajan in ACISP 2013 [1]. With the same $P(G)$ definition, in this notion, Alice (encryptor) encrypts a message with an attribute $A$ defined by herself and an attribute token $P(G)$ generated by Bob. Bob can only decrypt the ciphertext successfully if and only if the member relationship $A \in G$ holds. In comparison with a membership encryption, a subset membership encryption is a generalization of a membership encryption since a subset relationship implies a member relationship.

The authors in [1] gave a formal definition of membership encryption, and a construction modified from threshold attribute-based encryption [2]. They showed how to apply it to construct $k$-out-of-$n$ oblivious transfer. The proposed OT protocol achieves a constant-size communication cost (i.e. three group elements) from receiver to sender, which is independent of the number of $k$. However, the size of system parameter is $O(n^2)$.

Oblivious transfer is a two-part protocol where a message sender (Alice) holds $n$ messages and a receiver (Bob) retrieves $k$ of them, such that the sender does not know which messages the receiver obtains. Since Rabin first introduced this notion in [3], there have been many publications on oblivious transfer, such as adaptive OT (e.g. [4], [5], [6]), OT with fully simulatable security (e.g. [7]), OT with universally composable security (e.g. [8]), and OT with restrictions (e.g. [9], [10]).

Supposing the indices of messages and system parameter are known by the receiver. The ideal communication rounds can be two only, where the receiver sends privacy-preserving choices to the sender in the first round, and the sender responds on the receiver’s choice in the second round. Some proposed $k$-out-of-$n$ oblivious transfer protocols such as [11], [12], [5], [13], [1] have ideal communication rounds.

Fuchun Guo, Yi Mu and Willy Susilo are with the School of Computer Science and Software Engineering, University of Wollongong, NSW 2500, Australia.
Email: \{fuchun,ymu,wsusilo\}@uow.edu.au
Priced oblivious transfer (POT) was first introduced by Aiello, Ishai, and Reingold in [14] and later improved in [15], [16], [17], [18]. POT is a two-party protocol between a vendor (Alice) and a buyer (Bob) for privately buying digital goods. The vendor sells a set of items $M_1, M_2, \ldots, M_N$ with prices $p_1, p_2, \ldots, p_N$ respectively. A buyer initially deposits an amount of money into its account. In each transaction phase, the buyer selects one index $i \in \{1, 2, \ldots, N\}$ to buy item $M_i$ in such a way that (1) the buyer can get $M_i$ as long as the remaining balance is sufficient, (2) the remaining balance will be deducted with the amount of the item’s price, and (3) the vendor does not know the selected item, price, and remaining balance. The previous POT protocols are constructed from the combinations of many primitives such as homomorphic encryption and zero-knowledge proofs. Hence, they are rather impractical and inefficient in terms of their implementation.

### B. Our Contribution

Our contribution is twofold. First, we introduce the notion of subset membership encryption (SME). Subsequently, we present the efficient applications of SME in achieving oblivious transfer protocols.

Roughly speaking, an SME involves two entities: an encryptor and a decryptor. Given system parameter generated by the encryptor, the decryptor selects a secret value $sk$ and then computes $P(G)$ from $sk$ and an attribute set $G$. The attributes in $G$ for the encryptor are anonymous from the view of $P(G)$. The encryption takes as input an attribute set $\mathcal{A}$ chosen by the encryptor and the attribute token $P(G)$. Decryption takes as input the ciphertext and $(sk, \mathcal{A}, G)$, and successful decryption requires that the subset $\mathcal{A} \subseteq G$ must hold. We also define two features of SME called accountability and provability of hidden subset relationship. The accountability feature refers to the fact that the upper bound attribute number of $P(G)$ is provable. That is, we can prove the number of attributes in $P(G)$ is not more than $k$ without leaking $G$, if $|G| \leq k$ is true. The provability of hidden subset relationship ensures that the subset relationship of two attribute sets $G, G'$ from their tokens $P(G), P(G')$ is provable. This means that if $G$ is a subset of $G'$, we can prove the attributes in $P(G)$ is a subset of $P(G')$ without leaking $G$ and $G'$.

We propose a very efficient SME scheme. In particular, $P(G)$ is represented by one group element only, which is independent of its attribute number. The ciphertext is very short composed of only two group elements. The proofs for accountability and provability of hidden subset relationship are also short with merely one group element only. The size of system parameter is linear in the upper bound of attribute number of $G$. In comparison with the membership encryption definition and construction [1], SME is a generalization notion of a membership encryption, and our proposed scheme has a shorter size in terms of system parameter, attribute token $P(G)$ and ciphertext.

Then, we also show how to apply SME to efficient applications, namely $k$-out-of-$n$ oblivious transfer, priced oblivious transfer and restricted priced oblivious transfer. The features of the above three applications are listed as follows.

- The $k$-out-of-$n$ oblivious transfer protocol is secure against malicious receiver and semi-honest sender defined in [5]. Using the proposed SME scheme, our protocol only requires the receiver to forward two group elements to the sender independent of the number $k$. The communication cost from the sender to the receiver are $n$ ciphertexts, where each ciphertext merely consists of two group elements.
- The priced oblivious transfer protocol not only captures privacy and security requirements of POT, but also allows a buyer to select any number of items in each transaction phase. By adopting our proposed SME scheme, the buyer only generates three group elements for the vendor in each transaction phase. In comparison with previous POT constructions, our POT protocol is quite compact and efficient without the need of zero-knowledge proof or homomorphic encryption.
- The restricted priced oblivious transfer (RPOT) is a particular POT protocol that allows the vendor to set items or prices with necessary restrictions. The restriction in our definition is based on buyers’ purchase
state, which items have been purchased previously. One of potential scenarios for RPOT is the situation where the seller wishes to sell goods with discount prices for bulk buyers. For instance, an author is selling his book entitled “Crypto Heaven” in an online bookstore for the price of $19.95. However, he would like to sell this book with the reduced price of $9.95, if the same buyer is also purchasing the two companion books: “Crypto Cloud” (for $7.95) and “Crypto in Summer” (for $4.95), which are also written by the same author.

II. Subset Membership Encryption

A. Notations

We provide the main notations used in the definitions and construction of SME for references.

- $A_i$: An attribute.
- $A$: An attribute set chosen by an encryptor.
- $G$: An attribute set selected by a decryptor.
- $|G| \leq k$: The number of attributes in $G$ is not more than $k$.
- $A \subseteq G$: $A$ is a subset of $G$.
- $P(G)$: A randomized privacy-preserved attribute token on $G$ without leaking $G$.
- $\Sigma_A$: A proof for accountability.
- $\Sigma_R$: A proof for hidden subset relationship.
- $M$: A message.
- $C$: A ciphertext.

B. Definitions

A subset membership encryption denoted by $A \subseteq P(G)$ is composed of the four algorithms:

- **Setup:** Taking as input a security parameter $1^\lambda$ and an integer $n$, the setup algorithm generates the system parameter $SP$. Here, $n$ denotes the upper bound attribute number in all attribute tokens. This algorithm is run by the encryptor.
- **TokenGen:** Taking as input the system parameter $SP$ and an attribute set $G = \{A_1, \ldots, A_k\}$ $(1 \leq k \leq n)$, the token generation algorithm returns an attribute token $P(G)$ and a secret key $sk$. Here, $P(G)$ is computed from $sk$ and $G$ by the decryptor.
- **Encrypt:** Taking as input the system parameter $SP$, an attribute set $A$, an attribute token $P(G)$ and a message $M$, the encryption algorithm returns a ciphertext $C$ of $M$. We define the ciphertext as $C \leftarrow \text{SME}[A, P(G), M]$.
- **Decrypt:** Taking as input $A$, $G$, the secret key $sk$ and the ciphertext $C$, the decryption algorithm returns the message $M$ or $\perp$. We define the decryption as $\{ M, \perp \} \leftarrow \text{SMD}[C, A, G, sk]$.

**Correctness:** The correctness of subset membership encryption must satisfy that for any system parameter $SP$, $(P(G), G, sk)$ and ciphertext $\text{SME}[A, P(G), M]$, if $A \subseteq G$, we have

$$\text{SMD}[\text{SME}[A, P(G), M], A, G, sk] = M.$$  

We also define two features of SME as follows.

**Definition 1:** We say that an SME captures accountability if the following two algorithms exist.

- **AProof:** Taking as input $(P(G), G, sk, k)$ where the number of attributes in $G$ is not more than $k$, the algorithm returns a proof $\Sigma_A$ to prove $|G| \leq k$.
- **AVerify:** Taking as input $(P(G), k, \Sigma_A)$, the verification algorithm accepts $|P(G)| \leq k$ if the attribute number in $G$ satisfies $|G| \leq k$.

**Definition 2:** We say that an SME captures provability of hidden subset relationship if the following two algorithms exist.

- **RProof:** Taking as input $(P(G), G, sk)$ and $(P(G'), G', sk')$ satisfying $G \subseteq G'$, the algorithm returns a proof $\Sigma_R$ for $P(G)$ and $P(G')$ to prove that $G$ is a subset of $G'$.
- **RVerify:** Taking as input $(P(G), P(G'), \Sigma_R)$, the verification algorithm accepts $P(G) \subseteq P(G')$ if the subset $G \subseteq G'$ holds.

C. Comparison

Our subset membership encryption is a generalization of membership encryption defined in [1]. In Table I, we list three differences in terms of definitions.

- The relation of our encryption notion is a subset relationship over an attribute set $A$ and another set of attributes $G$, denoted by $A \subseteq G$. It is more generic compared to the member relationship over an attribute set $A$ and an attribute set $G$, denoted by $A \in G$. We have the subset membership encryption implies the “member relationship” encryption when $A = \{A\}$.
- Our definition is proposed for unbounded attributes. The setup algorithm of our definition does not take as input attributes. The system parameter generated by the setup algorithm is available for all attributes. Notice that the definition in [1] must pre-define all attributes or have to expand the system parameter for new attributes.
- We define a new property called Provability of Hidden Subset Relationship for subset membership encryption. This property is used to prove the subset relationship of two attribute tokens $P(G), P(G')$. We adopt this property in the construction of priced oblivious transfer and restricted priced oblivious transfer.

D. Security Models

An SME with accountability and provability of hidden subset relationship must satisfy the following security requirements.

- (Privacy) Given $(P(G), k, \Sigma_A)$ and two attribute sets $G_0, G_1$ satisfying $|G_0|, |G_1| \leq k$, it is hard to distinguish whether $G = G_0$ or $G = G_1$.
- (Indistinguishability) Given $A, G, P(G)$, a ciphertext $\text{SME}[A, P(G), M]$ for any $A$ and $G$ and two messages
have with Indistinguishability. unconditionally preserves the privacy of attribute token if

\[ \text{Adv} = \Pr[c' = c] - 1/2. \]

**Definition 3:** An SME generated with a security parameter \(1^\lambda\) preserves the privacy of attribute token \(P(G)\) with \((t, \epsilon)\) if for all \(t\)-polynomial time adversaries, we have \(\epsilon = \text{Adv}\) is a negligible function of \(\lambda\). We say it unconditionally preserves the privacy of attribute token if \(\epsilon = 0\) for any all unlimited-computing-power adversaries.

**Indistinguishability.**

- **Setup:** The challenger runs the Setup algorithm to generate the system parameter \(SP\) and sends it to the adversary.
- **Query 1:** The adversary makes a query on any attribute set \(G\). The challenger responds by computing \((P(G), G, sk)\) and sending \(P(G)\) to the adversary.
- **Challenge:** The adversary gives the challenger an attribute set \(A^*\) and two messages \(M_0, M_1\) for challenge, where \(A^*\) is a subset of \(G\). The challenger responds by randomly choosing a coin \(c \in \{0, 1\}\), and generating the challenge ciphertext \(C^* = SME[A^*, P(G), M_c]\) which is forwarded to the adversary.
- **Query 2:** The adversary can make decryption queries on any \((A_i, C_i)\) except \((A^*, C^*)\). The challenger responds by computing \(SMD[C_i, A_i, G, sk]\) and sending the result to the adversary.
- **Win:** The adversary outputs a guess \(c'\) of \(c\) and wins the game if \(c' = c\).

We define the advantage of adversary as \(\text{Adv} = |\Pr[c' = c] - 1/2|\).

**Definition 4:** An SME generated with a security parameter \(1^\lambda\) captures indistinguishability and it is \((t, q_d, \epsilon)\)-secure against chosen-ciphertext attack, if for all \(t\)-polynomial time adversaries who make \(q_d\) decryption queries, we have \(\epsilon = \text{Adv}\) is a negligible function of \(\lambda\). We call it semantic security when \(q_d = 0\).

**Subset Relationship.**

- **Setup:** The challenger runs the Setup algorithm to generate the system parameter \(SP\) and sends it to the adversary.
- **Challenge:** The adversary gives the challenger two attribute sets \(G_0 = \{A_1, A_2, \ldots, A_{k_0}\}\), \(G_1 = \{A'_1, A'_2, \ldots, A'_{k_1}\}\) and an integer \(k\) satisfying \(k_0, k_1 \leq k\). The challenger responds by randomly choosing a coin \(c \in \{0, 1\}\), setting \(G = G_c\) and generating \((P(G), G, sk)\). The challenger computes and sends \((P(G), k, \Sigma_0)\) to the adversary.
- **Win:** The adversary outputs a guess \(c'\) of \(c\) and wins the game if \(c' = c\).

We define the advantage of adversary as \(\text{Adv} = |\Pr[c' = c] - 1/2|\).

**Definition 5:** An SME generated with a security parameter \(1^\lambda\) is \((t, \epsilon)\)-secure against subset relationship if for all \(t\)-polynomial time adversaries, we have \(\epsilon = \text{Adv}\) is a negligible function of \(\lambda\).

**Accountability.** The models of Accountability and Provability of Hidden Subset Relationship are defined to capture attacks, where an adversary can forge proofs for valid attribute tokens \(P(G)\). Here, the validness means the adversary knows \(G\) and \(sk\) of computing \(P(G)\).

- **Setup:** The challenger runs the Setup algorithm to generate the system parameter \(SP\), and sends it to the adversary.
- **Challenge:** The adversary outputs \((P(G), G, sk)\) and \(k\) for challenge, where \(P(G)\) was generated from \((G, sk)\) and \(|G| > k\). I.E., the number of attributes in \(G\) is larger than \(k\).
\begin{itemize}
\item \textbf{Win}: The adversary outputs \((P(G), k, \Sigma_A)\) and wins the game if \((P(G), k, \Sigma_A)\) passes the verification of AVerify.

We define the advantage of adversary as Adv in computing \((P(G), k, \Sigma_A)\).

\textbf{Definition 6}: An SME generated with a security parameter \(1^\lambda\) captures the accountability and it is \((t, \epsilon)\)-secure if for all \(t\)-polynomial time adversaries, we have \(\epsilon = \text{Adv}\) is a negligible function of \(\lambda\).

\textbf{Provability of Hidden Subset Relationship.}
\begin{itemize}
\item \textbf{Setup}: The challenger runs the Setup algorithm to generate the system parameter \(SP\), and sends it to the adversary.
\item \textbf{Challenge}: The adversary outputs \((P(G), G, sk)\) and \((P(G'), G', sk')\) for challenge, where \(P(G)\) was generated from \((G, sk)\), and \(P(G')\) was generated from \((G', sk')\). It requires that \(G \not\subseteq G'\).
\item \textbf{Win}: The adversary outputs \((P(G), P(G'), \Sigma_R)\) and wins the game if \((P(G), P(G'), \Sigma_R)\) passes the verification of RVerify.

We define the advantage of adversary as Adv in computing \((P(G), P(G'), \Sigma_R)\).

\textbf{Definition 7}: An SME generated with a security parameter \(1^\lambda\) captures the provability of hidden subset relationship and it is \((t, \epsilon)\)-secure if for all \(t\)-polynomial time adversaries, we have \(\epsilon = \text{Adv}\) is a negligible function of \(\lambda\).
\end{itemize}

\textbf{E. Proposed Scheme}

In this section, we propose an efficient SME with accountability and provability of hidden subset relationship. The proposed scheme here is semantically secure under the Indistinguishability model. We can naturally extend it to chosen-ciphertext attack using the technique due to Fujisaki-Okamoto [19], [20] in the random oracle model.

Our scheme is built from a pairing group \(PG = (G, G_T, e, p, g)\), where \(e: G \times G \rightarrow G_T\) is the bilinear map, \(g\) is a generator of \(G\) and \(p\) is the group order of both \(G\) and \(G_T\). The scheme is described as follows.

\textbf{Setup}: Taking as input a security parameter \(1^\lambda\), let \(n\) be the upper bound attribute number of groups, the setup algorithm chooses \(\alpha \in \mathbb{Z}_p\) at random and computes \(g^{\alpha}, g^{\alpha^2}, \cdots, g^{\alpha^n}\) in \(G\). The system parameter \(SP\) is

\[ SP = \left( P_G, g^{\alpha}, g^{\alpha^2}, \cdots, g^{\alpha^n} \right). \]

\textbf{TokenGen}: Taking as input an attribute set \(G = \{A_1, A_2, \cdots, A_k\} \in \mathbb{Z}_p\) for any \(1 \leq k \leq n\), the token generation algorithm randomly chooses \(s \in \mathbb{Z}_p\), and computes \((P(G), sk)\) as

\[(P(G), sk) = (g^{s \prod_{i=1}^{k} (\alpha + A_i)}, s) \in G \times \mathbb{Z}_p.\]

\textbf{Encrypt}: Taking as input an attribute set \(A\), an attribute token \(P(G) \neq 1_G\), a message \(M \in G_T\) and the system parameter, the encryption algorithm randomly chooses \(r\) from \(\mathbb{Z}_p\) and computes the ciphertext on the message \(M\) as

\[ C = (c_1, c_2) = \left( g^{\prod_{i=1}^{k} (\alpha + A_i)}, e(P(G), g)^{-r} \cdot M \right). \]

\textbf{Decrypt}: Taking as input the ciphertext \(C\), the secret key \(sk\), the attribute set \(A\), the attribute set \(G\) and the system parameter, the decryption algorithm computes \(M\) by

\[ c_2 \cdot e\left( c_1^k, g^{\prod_{i=1}^{k} (\alpha + A_i)} \right) = e\left( g^s \prod_{i=1}^{k} (\alpha + A_i), g^{\prod_{i=1}^{k} (\alpha + A_i)} \right)^{-r} \cdot M \]

\[ = e\left( g, g^{rs} \prod_{i=1}^{k} (\alpha + A_i) \right) \cdot e\left( g, g^{-rs} \prod_{i=1}^{k} (\alpha + A_i) \right)^{\alpha} \cdot M = M. \]

\(G/\hat{A}\) denotes the relative complement set of \(A\) in \(G\).

The algorithms for accountability are defined as follows.

\textbf{AProof}: Taking as input \((P(G), G, sk, k)\), if \(|G| \leq k\), the proof algorithm computes \(\Sigma_A = P(G)^{\alpha - k} = g^{\alpha^k} \cdot g^{\prod_{i=1}^{k} P(A_i)} \in G\), which is computable from system parameter.

\textbf{AVerify}: Taking as input \((P(G), k, \Sigma_A)\), the verification algorithm accepts \(|P(G)| \leq k\) if

\[ e\left( P(G), g^{\alpha^k} \right) = e\left( \Sigma_A, g^{\alpha^k} \right). \]

The algorithms for provability of hidden subset relationship are defined as follows.

\textbf{RProof}: Taking as input \((P(G), G, sk)\) and \((P(G'), G', sk')\), if \(G \subseteq G'\), the proof algorithm computes \(\Sigma_R = g^{\prod_{i=1}^{k} P(A_i)} \cdot g^{\prod_{i=1}^{k} P(A_i)} \in G\).

\(G'/\hat{G}\) denotes the relative complement set of \(G\) in \(G'\).

\textbf{RVerify}: Taking as input \((P(G), P(G'), \Sigma_R)\), the verification algorithm accepts \(P(G) \subseteq P(G')\) if

\[ e\left( P(G), \Sigma_R \right) = e\left( P(G'), g \right). \]

\textbf{III. Security Proof}

In this section, we prove the security of our SME scheme under the five security models defined in Section II. We prove that the privacy of our scheme is unconditionally preserved. The indistinguishability and subset relationship are based on the \(n\)-BDHE assumption [21]. We prove the two additional properties of accountability under the \((f, n)\)-DHE assumption [1], and the provability of hidden subset relationship under the \(n\)-BSDH assumption [22]. Since these assumptions have been proposed and analysed, we omit the intractability analysis and refer readers to the corresponding references.
**n-BDHE Problem [21]:**

Instance: \( g, g^a, g^{a^2}, \ldots, g^{a^n}, T \) such that \( g^{2a^n + 2}, g^{2a^{n+3}}, \ldots, g^{2a^{3n+1}} \).

Output: Return 1 if \( T = e(g, g)g^{a^{n+1}} \); otherwise, 0.

**Proof of Indistinguishability**

Then, \( n \) sends \( \lambda \), an \( \epsilon \)-negligible function of \( n \), to the adversary. \( n \) completes the proof and we obtain the Theorem 1.

**Output:**

Return \( (b, c, e(g, g)^{-1}) \) for any \( b \in \mathbb{Z}_p \).

**Definition 8:** We say that the \( n\)-BDHE/\( f, n\)-DHE/\( n\)-BSDH problems generated from a security parameter \( \lambda \) are \((t, \epsilon)\)-hard if given as input a corresponding instance, an adversary can solve it with \( \epsilon \) advantage. Here, \( \epsilon \) is a negligible function of \( \lambda \).

**A. Proof of Privacy**

**Theorem 1 (Privacy):** Our SME unconditionally preserves the privacy of attribute token.

**Proof:** Let \( \mathcal{P}(G, k, \Sigma) \) be generated from \( G = G_0 = \{ A_1, A_2, \ldots, A_k \} \) (k0 \( \leq k \)). We have

\[
\mathbb{P}(G) = g^{sk}\Pi_{i=1}^{k_0}(a + A_i), \quad \Sigma_A = g^{\alpha - k \cdot sk}\Pi_{i=1}^{k_0}(a + A_i).
\]

For any distinct set \( G_1 = \{ A'_1, A'_2, \ldots, A'_{k_1} \} \) (k1 \( \leq k \)), let \( sk' \in \mathbb{Z}_p \) be defined as

\[
sk' = sk \cdot \Pi_{i=1}^{k_0}(a + A_i) / \Pi_{i=1}^{k_1}(a + A'_i).
\]

We have

\[
\mathbb{P}(G) = g^{sk}\Pi_{i=1}^{k_0}(a + A_i) = g^{sk'}\Pi_{i=1}^{k_1}(a + A'_i) = \mathbb{P}(G_1).
\]

\[
\Sigma_A = g^{\alpha - k \cdot sk}\Pi_{i=1}^{k_0}(a + A_i) = g^{\alpha - k \cdot sk'}\Pi_{i=1}^{k_1}(a + A'_i) = \Sigma_A.
\]

Such that \( \mathbb{P}(G_0, k, \Sigma_A) = \mathbb{P}(G_1, k, \Sigma_A) \). Since \( sk \) is randomly chosen from \( \mathbb{Z}_p \), we have \( sk' \) is also universally random in \( \mathbb{Z}_p \). The distributions of \( \mathbb{P}(G_i, k, \Sigma_A) \) for both \( G_0 \) and \( G_1 \) are identical, and therefore the adversary has no advantage in guessing the attribute set in \( \mathbb{P}(G) \). This completes the proof and we obtain the Theorem 1.

**B. Proof of Indistinguishability**

**Theorem 2 (Indistinguishability):** Our SME is \((t', \epsilon')\)-semantically secure if the \( n\)-BDHE problem is \((t, \epsilon)\)-hard, where \( t' = t - O(n t_c) \), \( \epsilon' \leq n \epsilon \) and \( t_c \) is the average time of a point multiplication in \( \mathbb{G} \).

**Proof:** Suppose there exists an adversary who can break the indistinguishability. We construct an algorithm \( B \) that solves the \( n\)-BDHE problem. Given as input a challenge instance of \( n\)-BDHE problem, \( B \) interacts with the adversary as the follows.

**Setup:** \( B \) sets \( \alpha = a \) in the simulation of system parameter. Then, \( SP = \{ \mathcal{P}(G), g^{\alpha}, g^{\alpha^2}, \ldots, g^{\alpha^n} \} \) is computable. \( B \) sends \( SP \) to the adversary.

**Query 1:** For the query on \( G = \{ A_1, A_2, \ldots, A_k \} \) \( (k \leq n) \), the challenger randomly chooses \( j^* \in \{ 1, 2, \ldots, k \} \), \( s \in \mathbb{Z}_p \) and computes

\[
g^{s \cdot \Pi_{i=1}^{k}(a + A_i)} = g^{s \cdot \Pi_{i=1}^{k}(a + A_{j^*})}.
\]

Let the secret key of \( \mathcal{P}(G) \) be \( sk = \frac{s}{a + A_{j^*}} \). We have

\[
\mathbb{P}(G) = g^{sk}\Pi_{i=1}^{k}(a + A_i) = g^{s \cdot \Pi_{i=1}^{k}(a + A_{j^*})}.
\]

The algorithm \( B \) sends \( \mathbb{P}(G) \) to the adversary.

**Challenge:** The adversary gives an attribute set \( \mathbb{A}^* \) and two messages \( M_0, M_1 \) for challenge, where \( \mathbb{A}^* \subseteq \mathbb{G} \). If \( A_{j^*} \notin \mathbb{A}^* \), abort. Otherwise, \( B \) randomly chooses \( c \in \{ 0, 1 \} \) and \( r \in \mathbb{Z}_p \). Let \( f_i \) be the coefficient of \( x^i \) in the following polynomial

\[
f(x) = \prod_{i=1, i \neq j^*}^{k} (x + A_i) \cdot \frac{x^{2n+2} - A_{j^*} \cdot 2n+2}{x + A_{j^*}}
\]

whose degree is \( 2n + k \). It is not hard to verify that the coefficient of \( f_{2n+1} \) is \( \prod_{i=1, i \neq j^*}^{k} (-A_{j^*} + A_i) \) that is non-zero. \( B \) computes the challenge ciphertext \( C^* = (c_1, c_2) \) as

\[
c_1 = g^{r \cdot \Pi_{A_i \notin \mathbb{A}^*}(a + A_i)(a^{2n+2} - A_{j^*} \cdot 2n+2)}
\]

\[
c_2 = e \left( \prod_{i=0}^{n} g_{f_i(a)}^{a^i}, g \right) \cdot e \left( \prod_{i=n+1}^{2n} g_{f_i(a^{i-n})}^{a^{i-n}}, g \right) \cdot e \left( \prod_{i=2n+2}^{2n+k} g_{f_i(a)}^{a^i}, g \right) \cdot M_c.
\]

Let \( r' = r \cdot a^{2n+2} - A_{j^*} \cdot 2n+2 \). If \( T = e(g, g)^{a^{2n+1}} \), we have

\[
c_1 = g^{r \cdot \Pi_{A_i \notin \mathbb{A}^*}(a + A_i)(a^{2n+2} - A_{j^*} \cdot 2n+2)}
\]

\[
c_2 = e \left( \prod_{i=0}^{n} g_{f_i(a)}^{a^i}, g \right) \cdot e \left( \prod_{i=n+1}^{2n} g_{f_i(a^{i-n})}^{a^{i-n}}, g \right) \cdot e \left( \prod_{i=2n+2}^{2n+k} g_{f_i(a)}^{a^i}, g \right) \cdot M_c.
\]

Therefore, \( C^* \) is a valid challenge ciphertext.

**Win:** The adversary outputs a guess \( c' \) of \( c \) and \( B \) outputs \( c' \) as the solution to the \( n\)-BDHE problem.
This completes the description of our simulation. The
time cost is mainly dominated by $O(n)$ point multipli-
cations in $G$ for the challenge ciphertext simulation. Since
$j^*$ is randomly chosen from $\{1, 2, \ldots, k\}$ for any $k \leq n$
and $k^* \neq 0$, we have probability $\frac{1}{n}$ at least for no abortion
during the simulation. If $T = e(g, g)^{a^{2n+1}}$, the adversary
guesses $c$ correctly with probability $\frac{1}{2} + c$; otherwise, we have $c_2$ is uniformly random and independent of $c_1$, such that the adversary can only guess $c$ correctly with probability $\frac{1}{2}$. We therefore obtain the theorem 2. ■

C. Proof of Subset Relationship

**Theorem 3 (Subset Relationship):** Our SME is $(t', e')$-
secure against subset relationship if the $n$-BDHE problem is
$(t, e)$-hard, where $t' = t - O(n_{H_2})$, $e' = e$ and $t_e$ is the
average time of a point multiplication in $G$.

**Proof:** Suppose there exists an adversary who can
break the membership. We construct an algorithm $\mathcal{B}$
that solves the $n$-BDHE problem. Given as input a challenge in-
stance of $n$-BDHE problem, $\mathcal{B}$ interacts with the adversary
as the follows.

**Setup:** $\mathcal{B}$ sets $\alpha = a$ in the simulation of system parameter.
We have $SP = (\mathbb{G}, g^a, g^{a^2}, \ldots, g^{a^n})$ that is computable.
$\mathcal{B}$ sends $SP$ to the adversary.

**Challenge:** The adversary outputs $(h^*, \mathcal{P}(G^*), G^*,
\sigma^*, M_0, M_1)$ for challenge where $h^* \notin G^*$. Let $|G^*| = k,
\sigma^* = s$, $A \in h^*$, $A \notin G^*$ and $f_i$ be the coefficient of $x^i$
in the following polynomial

$$f(x) = \prod_{A_i \in G^*} \left( x + A_i \right) \cdot \frac{x^{2n+2} - A^{2n+2}}{x + A}$$

whose degree is $2n+1+k$. It it not hard to verify $f_{2n+1} = \prod_{i=1}^{k} (-A + A_i) \neq 0$. $\mathcal{B}$ randomly chooses $c \in \{0, 1\}$, $r \in 
\mathbb{Z}_p$ and computes the challenge ciphertext $C^* = (c_1, c_2)$ as

$$c_1 = g^{r \prod_{A_i \in h^*} f_i A_i} \cdot (a^{2n+2} - A^{2n+2})$$

$$c_2 = e\left( \prod_{i=1}^{n} g^{f_i A_i}, g \right) \cdot e\left( \prod_{i=n+1}^{2n} g^{f_i A_i}, g^{a^n} \right) \cdot T^{f_{2n+1}} \cdot e\left( \prod_{i=2n+2}^{2n+1+k} g^{f_i A_i}, g \right)^{rs} \cdot M_c.$$

Let $r' = r \cdot \frac{a^{2n+2} - A^{2n+2}}{a + A}$. If $T = e(g, g)^{a^{2n+1}}$, we have

$$c_1 = g^{r' \prod_{A_i \in h^*} f_i A_i} \cdot (a^{2n+2} - A^{2n+2})$$

$$c_2 = e\left( \prod_{i=1}^{n} g^{f_i A_i}, g \right) \cdot e\left( \prod_{i=n+1}^{2n} g^{f_i A_i}, g^{a^n} \right) \cdot T^{f_{2n+1}} \cdot e\left( \prod_{i=2n+2}^{2n+1+k} g^{f_i A_i}, g \right)^{rs} \cdot M_c.$$

Therefore, $C^*$ is a valid challenge ciphertext.

**Win:** The adversary outputs a guess $c'$ of $c$ and $\mathcal{B}$ outputs
$c'$ as the solution to the $n$-BDHE problem.

This completes the description of our simulation. The
time cost is mainly dominated by $O(n)$ point multipli-
cations in $G$ for the challenge ciphertext simulation. Since
$j^*$ is randomly chosen from $\{1, 2, \ldots, k\}$ for any $k \leq n$
and $k^* \neq 0$, we have probability $\frac{1}{n}$ at least for no abortion
during the simulation. If $T = e(g, g)^{a^{2n+1}}$, the adversary
guesses $c$ correctly with probability $\frac{1}{2} + c$; otherwise, we have $c_2$ is uniformly random and independent of $c_1$, such that the adversary can only guess $c$ correctly with probability $\frac{1}{2}$. We therefore obtain the theorem 3. ■

D. Proof of Accountability

We note that the security of accountability is based on the fact that if $|G| > k$, for example $|G| = k + 1$, the corresponding proof is $g^{\sigma^* - k \sum_{A_i \in G^*} f_i A_i}$, which contains the group element $g^{a^n}$ that is not provided in
$SP$. The proof is completed in the following theorem.

**Theorem 4 (Accountability):** Our SME captures the ac-
countability under the hard assumption of $(f, n)$-DHE
problem.

**Proof:** Suppose there exists an adversary who can
break the security of accountability. We construct an al-
gorithm $\mathcal{B}$ that solves the $(f, n)$-DHE problem. Given an instance of $(f, n)$-DHE problem, $\mathcal{B}$ interacts with the adversary
as the follows.

**Setup:** $\mathcal{B}$ sets $\alpha = a$ in the simulation of system parameter.
Then, we have $SP = (\mathbb{G}, g^a, g^{a^2}, \ldots, g^{a^n})$ that is computable. $\mathcal{B}$ sends $SP$ to the adversary.

**Challenge:** The adversary outputs $(\mathcal{P}(G^*), G^*, \sigma^*, k)$ for challenge, where $|G^*| > k$ and

$$\mathcal{P}(G^*) = g^{\sigma^*} \cdot \prod_{A_i \in G^*} (a + A_i).$$

**Win:** The adversary outputs $(\mathcal{P}(G^*), k, \Sigma_A)$, and wins the game if the tuple $(\mathcal{P}(G^*), k, \Sigma_A)$ passes the verification.
In this case, we have

$$\Sigma_A = g^{\sigma^* - k \cdot \sigma^*} \cdot \prod_{A_i \in G^*} (x + A_i).$$

Let $f(x) = \sigma^* x^{n-k} \prod_{A_i \in G^*} (x + A_i)$. We have $f(x)$ is polynomial function in $\mathbb{Z}_p$ whose degree is larger than $n$. $\mathcal{B}$ outputs $(f(x), \Sigma_A)$ as the solution to the $(f, n)$-DHE problem. This completes the description of our simulation and we obtain the Theorem 4. ■
E. Proof of Provability of Hidden Subset Relationship

Theorem 5 (Provability of Hidden Subset Relationship): Our SME captures the provability of hidden subset relationship under the hard assumption of \( n \)-BSDH problem.

Proof: Suppose there exists an adversary who can break the security of provability of hidden subset relationship. We construct an algorithm \( B \) that solves the \( n \)-BSDH problem. Given an instance of \( n \)-BSDH problem, \( B \) interacts with the adversary as the follows.

Setup. \( B \) sets \( \alpha = a \) in the simulation of system parameter. Then, we have \( SP = (P^G, g^a, g^{2a}, \cdots, g^{na}) \) that is computable. \( B \) sends \( SP \) to the adversary.

Challenge. The adversary outputs \((P(G), G, sk)\), and \((P(G'), G', sk')\) for challenge, where \( P(G) \) is generated from \((G, sk)\), \( P(G') \) is generated from \((G', sk')\) and \( G \) is not a subset of \( G' \).

Win: The adversary outputs \((P(G), P(G'), \Sigma_R)\) and wins the game if the tuple \((P(G), P(G'), \Sigma_R)\) passes the verification.

Let the two sets be \( G = S_0 \cup S_1, G' = S_1 \cup S_2 \), where \( S_0, S_1, S_2 \) are three attribute sets satisfying \( S_0 \cap S_1 = S_0 \cap S_2 = 0 \). If \( G \nsubseteq G' \), we have \( S_0 \neq \emptyset \). If \((P(G), P(G'), \Sigma_R)\) passes the verification, we have
\[
\Sigma_R = g^{\sum_{\alpha \in G } \prod_{A \in G} (\alpha + A)} = g^{\sum_{\alpha \in G' } \prod_{A \in G'} (\alpha + A) + \sum_{\alpha \in S_0 \cap S_1} \prod_{A \in G} (\alpha + A)}.
\]

Without loss of generality, let \( S_0 = \{A_1, A_2\}, B \) computes
\[
eq e(g^{\alpha + A_2}, \Sigma_R) = e(g, g) \frac{\prod\prod_{A \in G} (\alpha + A)}{\alpha + A_1},
\]
which can be re-written into
\[
eq e(g, g) \frac{\prod_{A \in S_2} (\alpha + A)}{\alpha + A_1}
\]
where \( f_n, f_{n-1}, \cdots, f_1, f_0, d \) are integers from \( \mathbb{Z}_p \) and \( d \) is non-zero. \( B \) extracts \((g, g)\) from the above using \( g, g^a, \cdots, g^{na} \) and \( f_n, \cdots, f_1, f_0, d \), and outputs \((A_1, e(g, g)^{\frac{\sum_{A \in S_0 \cap S_1} \prod_{A \in G} (\alpha + A)}{\alpha + A_1}})\) as the solution to the \( n \)-BSDH problem. This completes the description of our simulation and we obtain the Theorem 5.

IV. APPLICATIONS OF SME

A. Notations

We provide the following notations used in the applications of SME for references.

\[ M_i \]: The message that the receiver (buyer) wants to get, where the index is \( i \).
\[ p_i \]: The price of item \( M_i \) is \( p_i \).
\[ l_B \]: The initial amount of deposited coins.
\[ CB_i \]: The current balance after the \( i \)-th transaction.
\[ G_i \]: The purchasing state after the \( i \)-th transaction.
\[ sk_i \]: The secret key of SME chosen in the \( i \)-th transaction for \( G_i \).

B. Oblivious Transfer

The authors in [1] showed how to construct a (non-adaptive) \( k \)-out-of-\( n \) oblivious transfer protocol from membership encryption with accountability. Since the SME is a generalization of membership encryption, we can definitely also apply our encryption in OT protocol.

Suppose a sender has \( n \) messages \( M_1, M_2, \cdots, M_n \) and a receiver wants to get messages \( M_{i_1}, M_{i_2}, \cdots, M_{i_k} \), where \( \{i_1, i_2, \cdots, i_k\} \) is the set of indices of messages. Briefly, the OT protocol from SME with accountability works as follows.

- The receiver sets \( G = \{l_1, l_2, \cdots, l_k\} \) and generates \((P(G), G, sk)\) and \((P(G), k, \Sigma_A)\). The receiver forwards \((P(G), \Sigma_A)\) to the sender.
- The sender firstly verifies \((P(G), k, \Sigma_A)\). If it is correct, computes
\[
C_i = SME[A_i, P(G), M_i], \text{ for all } i = 1, 2, \cdots, n,
\]
where \( A_i = \{i\} \) and here the index “\( i \)” is an attribute. All ciphertexts are forwarded to the receiver.
- The receiver decrypts
\[
M_i = SME[C_i, A_i, G, sk], \text{ for all } i \in \{l_1, l_2, \cdots, l_k\}.
\]

It is not hard to check that the proposed OT protocol is secure against malicious receiver and semi-honest sender under the model in [5], when the SME is secure under the defined security models. The OT protocol preserves the receiver’s privacy because \( P(G) \) hides the message choice in \( G \). The OT protocol also protects sender’s security since the indistinguishability and the accountability make sure the receiver generated \( P(G) \) correctly with \( k \) choices at most in \( G \), and the security of subset relationship only allows the receiver to decrypt those chosen messages. Therefore, the receiver can only decrypt \( k \) number of chosen messages.

We compare the efficiency of OT protocols from two different membership encryption. The comparison result is given in Table II. Both OT protocols are completed in two rounds, and the communication cost of the receiver is independent of \( k \). However, our OT protocol exhibits a much shorter system parameter and less communication cost for both receiver and sender.

C. Priced Oblivious Transfer

Priced oblivious transfer (POT) is a two-party protocol between buyer and vendor that provides privacy of buyers in purchasing digital goods. Suppose an initial amount of money is deposited into a buyer’s account. A POT protocol exhibits the following properties.

- (Price) It allows the vendor to sell digital goods with distinct price.
- (Purchasing) The buyer selects one item in each transaction phase.
- (Correctness) As long as the buyer has a sufficient fund, the selected item will be successfully retrieved and the balance will be deducted by the amount of item price.
• (Privacy and Security) It hides the selected item, the price of item and the remaining balance from the vendor. It also prevents the buyer from retrieving an item whose cost exceeds the remaining balance.

Existing POT protocols are constructed from the combinations of many primitives, such as homomorphic encryption and zero-knowledge proof. They are rather impractical or inefficient. In this section, we show how to construct a very simple POT protocol from SME with accountability and provability of hidden subset relationship.

Our POT construction does not need any other cryptographic primitives. Besides its compactness and efficiency, our SME-based POT protocol exhibits the nice feature that it allows a buyer to purchase any number of items in a transaction phase. Notice that existing POT protocols restrict buyers in selecting one item only in each transaction phase. If a buyer wants to buy more than one item at the same time, he/she has to buy items one by one by re-running the POT protocol. Furthermore, using previous POT protocols, the buyer has to purchase some “dummy” items [14] in order to hide the number of purchased items. This definitely increases the communication cost especially for the buyer. Different from previous POT protocols, ours allows buyers to purchase any number of items in a transaction as long as the remaining fund is sufficient.

To construct a POT protocol from an SME, we firstly set each attribute to denote a coin (e.g., 10 cents) such that the buyer can store any number of coins. The initial deposit is also an integer number. We define the algorithm as as follows.

To construct a POT protocol from SME with accountability and provability of hidden subset relationship.

Initialization Phase: Let $I_D$ be the initial amount of coins deposited into a buyer’s account. Let $CB_i$ be the remaining balance after the $i$-th transaction phase. The buyer initiates the secret state with $(I_D, CB_0, G_0, sk_0)$, where $CB_0 = I_D$, $G_0 = \emptyset$ and $sk_0 = 1$. The vendor initiates the state with $(I_D, P(G_0)) = (I_D, P(\emptyset))$.

$i$-th Transaction Phase: Let the buyer’s secret state be $(I_D, CB_{i-1}, G_{i-1}, sk_{i-1})$ and the vendor’s state be $(I_D, P(G_{i-1}))$. Suppose $(M_{i_1}, p_{i_1}), (M_{i_2}, p_{i_2}), \ldots, (M_{i_l}, p_{i_l})$ are the items to be purchased in the $i$-th phase. Let $A_i = \text{Att}[M_{i_l}, p_{i_l}]$. The buyer and the vendor interacts as follows.

Buyer: If $CB_{i-1} \geq p_{i_1} + p_{i_2} + \cdots + p_{i_l}$, compute as follows.

- Set $G_i = G_{i-1} \cup \text{Att}[M_{i_1}, p_{i_1}] \cup \cdots \cup \text{Att}[M_{i_l}, p_{i_l}]$.
- Run the TokenGen to compute $(P(G_i), G_i, sk_i)$.
- Run the AProof to generate $(P(G_i), I_D, \Sigma_{A_i})$.
- Run the RProof to generate $(P(G_{i-1}), P(G_i), \Sigma_{R_i})$.
- Compute $CB_i = CB_{i-1} - (p_{i_1} + p_{i_2} + \cdots + p_{i_l})$.
- Send $(P(G_i), \Sigma_{A_i}, \Sigma_{R_i})$ to the vendor.
- Update the state of buyer $(I_D, CB_i, G_i, sk_i)$.

Vendor: Upon receiving $(P(G_i), \Sigma_{A_i}, \Sigma_{R_i})$ from the buyer, verify and compute as follows.

- Run the AVVerify to verify $|P(G_i)| \leq I_D$.
- Run the RVVerify to verify $P(G_{i-1}) \subseteq P(G_i)$.
- If both are correct, compute ciphertexts

\[ \text{SME}[A_{j}, P(G_i), M_j] \text{ for all } j = 1, 2, \ldots, N. \]

- Send all ciphertexts to the buyer.
- Update the state of buyer $(I_D, P(G_i))$.

Buyer: Upon receiving all ciphertexts, the buyer can get the item $M_j$ by running the decryption $\text{SME}[C_j, A_{j}, G_i, sk_i]$. Since $\text{Att}[M_j, p_j] = A_j \subseteq G_i$ holds for all $j \in \{i_1, i_2, \ldots, i_l\}$.

Remark 1: In our POT protocol, $G_i$ is an attribute set computed from all purchased items and their prices after the $i$-th transaction. To reduce its size, if $\text{Att}[M_j, p_j] \subseteq G_i$, we can store $j$ and $p_j$ instead of all attributes in $\text{Att}[M_j, p_j]$.
Remark 2: In our POT protocol, once a buyer has bought an item, he/she can still decrypt the item in future transactions. This property allows the vendor to update purchased items for their buyers. For example, in the pay-TV system, suppose the programmes of the i-th channel are encrypted with the encryption key $M_i$. The POT protocol preserves the privacy of subscribers in purchasing their interest channels. When the vendor updates encryption keys for security purpose, subscribers can still get all purchased channels' new encryption keys.

Efficiency. There is no inefficient primitive such as zero-knowledge proof or homomorphic encryption in our POT protocol. The buyer computes and sends $(P(G_i), \Sigma_A, \Sigma_R)$ to the vendor, which is composed of three group elements only using our SME scheme. The vendor performs $N$ ciphertexts for the buyer, where each ciphertext has two group elements only.

Correctness. The remaining balance before the i-th transaction phase can be denoted by $CB_{i-1} = I_D - |G_{i-1}|$, where $|G_{i-1}|$ is the total payment in previous phases. If $CB_{i-1} \geq p_1 + p_2 + \cdots + p_i$, we have $|G_{i-1}| \leq I_D - (p_1 + p_2 + \cdots + p_i)$. In this case, the buyer can add at least $(p_1 + p_2 + \cdots + p_i)$ number of attributes into $G_{i-1}$ and the new attribute set $G_i$ still satisfies $|G_i| \leq I_D$. The selected items $M_j (j \in \{1, 2, \ldots, l_j\})$ will be successfully retrieved because $\text{Att}[M_j, p_j] \subseteq G_i$ holds. The balance is deducted by items’ price since

$$CB_{i-1} - CB_i = (I_D - |G_{i-1}|) - (I_D - |G_i|) = |G_i| - |G_{i-1}| = p_1 + p_2 + \cdots + p_i$$

Privacy. The information of all purchased items is stored in the attribute set $G_i$, but the buyer only sends the attribute token $P(G_i)$ to the vendor. When the subset membership encryption is secure under the provability model, we have the vendor knows nothing about the attribute set except $|P(G_i)| \leq I_D$. Our POT protocol therefore hides selected items, price and remaining balance from the vendor.

Security. The security of SME under the indistinguishability model implies that $P(G)$ must be generated correctly in order to decrypt items. If the buyer never paid for $M_j$ and the current secret state is $G_{i-1}$ but $p_j$ exceeds the remaining balance, we have $|G_{i-1}| > I_D - p_j$. When the security of provability of hidden subset relationship holds, i.e., $P(G_{i-1}) \subseteq P(G_i)$, we have

$$|P(G_i)| = |P(\text{Att}[M_j, p_j] \cup G_{i-1})| = p_j + |P(G_{i-1})| > p_j + I_D - p_j = I_D.$$

The security under the accountability model guarantees that the buyer cannot generate a valid $(P(G_i), I_D, \Sigma_A_i)$ with $\text{Att}[M_j, p_j] \subseteq G_i$. To retrieve item $(M_j, p_j)$, $\text{Att}[M_j, p_j]$ must be a subset of $G_i$ under the security model of subset relationship. Therefore, our protocol prevents the buyer from retrieving items whose cost are more than the remaining balance.

D. Restricted Priced Oblivious Transfer

In our POT protocol, for all $j = 1, 2, \ldots, N$, the item $M_j$ with price $p_j$ is encrypted by

$$\text{SME}[a_j, P(G), M_j]$$
where the attribute set $A_j$ is defined as $A_j = \text{Att}[M_j, p_j] = \{j|1, j|2, \cdots, j|p_j\}$. When the buyer has sufficient funds and adds $A_j$ into his/her purchase list $G$, he/she can retrieve $M_j$. This property fulfills the correctness definition of POT protocol that the buyer enables to buy any item as long as the remaining balance contains sufficient funds.

We observe that our POT protocol can be extended to a POT protocol with restrictions on items or prices. We call it restricted priced oblivious transfer (RPOT in short). In our RPOT protocol, the restriction is based on buyers’ purchase state and the state is associated with the indices and prices of purchased items. Let $\mathbb{M}^p = \{(M_1, p_1^1), (M_2, p_2^1), \cdots\}$ be a list of items and prices. There are two different scenarios for the vendor in defining the restriction.

1) A buyer cannot buy $M$ unless its purchase state contains all specified items in $\mathbb{M}^p$. In this scenario, the vendor sets $M$ as a particular good such that it is only on sale for those buyers who have purchased $\mathbb{M}^p$.

2) A buyer cannot buy $M$ with a reduced price unless its purchase state contains all specified items in $\mathbb{M}^p$.

In this scenario, the vendor sells item $M$ with an original price and with a reduced price at the same time. However, the reduced price is only available for those buyers who have purchased $\mathbb{M}^p$.

We give an example in Table III to further explain the restrictions on items or prices. In this table, the vendor sells items $M_1, M_3, M_4$ with prices $p_1, p_3, p_4$, respectively. The vendor sells $M_3$ with the restriction that a buyer must buy $(M_1, p_1)$ before buying $(M_2, p_2)$. The vendor also sells $M_4$ with a reduced price $p_4'$, but the reduced price is only available for those buyers who have purchased $(M_1, p_1)$ and $(M_3, p_3)$.

Generally, the restriction says a buyer cannot purchase $(M, p)$ unless its purchase state contains all specified items in $\mathbb{M}^p$. Using an SME scheme, it is quite easy for the vendor to set the restriction $\mathbb{M}^p$ on $(M, p)$. The vendor simply encrypts $M$ by computing $\text{SME}[\mathbb{A}, \mathbb{P}(G), M]$, where $\mathbb{A}$ contains all attributes computed from $\mathbb{M}^p$ and $(M, p)$. That is, $\mathbb{A} = \text{Att}[\mathbb{M}^p] \cup \text{Att}[M, p]$. According to the properties of SME, we have the buyer’s purchase state $G$ must fulfill the restriction (i.e., $\text{Att}[\mathbb{M}^p] \subseteq G$) if he/she wants to retrieve the item $M$ by paying with price $p$.

<table>
<thead>
<tr>
<th>Index</th>
<th>Item &amp; Price</th>
<th>Restriction</th>
<th>Encryption</th>
<th>Attribute Set</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$(M_1, p_1)$</td>
<td>–</td>
<td>SME$[\mathbb{A}_1, \mathbb{P}(G), M_1]$</td>
<td>$A_1 = \text{Att}[M_1, p_1]$</td>
</tr>
<tr>
<td>2</td>
<td>$(M_2, p_2)$</td>
<td>$(M_1, p_1)$</td>
<td>SME$[\mathbb{A}_2, \mathbb{P}(G), M_2]$</td>
<td>$A_2 = \text{Att}[M_1, p_1] \cup \text{Att}[M_2, p_2]$</td>
</tr>
<tr>
<td>3</td>
<td>$(M_3, p_3)$</td>
<td>–</td>
<td>SME$[\mathbb{A}_3, \mathbb{P}(G), M_3]$</td>
<td>$A_3 = \text{Att}[M_3, p_3]$</td>
</tr>
<tr>
<td>4</td>
<td>$(M_4, p_4)$</td>
<td>$(M_1, p_1), (M_3, p_3)$</td>
<td>SME$[\mathbb{A}_4, \mathbb{P}(G), M_4]$</td>
<td>$A_4 = \text{Att}[M_1, p_1] \cup \text{Att}[M_3, p_3] \cup \text{Att}[M_4, p_4]$</td>
</tr>
</tbody>
</table>

V. CONCLUSION

We introduced the notion of subset membership encryption (SME) with accountability and provability of hidden subset relationship. A very efficient SME scheme with constant-size attribute tokens and ciphertexts was also proposed. We present several efficient applications that are based on SME. The first application described in this paper is a $k$-out-of-$n$ oblivious transfer. It takes two rounds and the receiver is only required to send two group elements to the sender. The second application is a priced oblivious transfer. Our protocol allows a buyer to select any number of items in each transaction phase. The scheme is very compact and efficient without requiring any zero-knowledge proof or homomorphic encryption. We also introduced and proposed the notion of a restricted priced oblivious transfer, which allows a vendor to set necessary restrictions on items or prices in priced oblivious transfer. Our applications have enriched the notion of oblivious transfer to be more flexible and practical.

REFERENCES


Fuchun Guo received his B.S. and M.S. degrees from Fujian Normal University, China in 2005 and 2008 respectively, and the PhD degree from the University of Wollongong, Australia in 2013. He is currently an associate research fellow at the School of Computer Science and Software Engineering, University of Wollongong. His primary research interest is the public-key cryptography; in particular, protocols, encryption and signature schemes, and security proof.

Yi Mu (SM’03) received the Ph.D. degree from Australian National University in 1994. He is currently a Professor with the University of Wollongong. Prior to joining the University of Wollongong, he was a Senior Lecturer with the Department of Computing, Macquarie University. He has been with the University of Wollongong since 2003. His current research interest includes cryptography, network security, access control, and computer security. He was involved in the areas of quantum cryptography, quantum computers, atomic computations, and quantum optics. He is the Editor-in-Chief of the International Journal of Applied Cryptography and serves as an Associate Editor or Guest Editor for many international journals. He has served in program committees for a number of international security conferences, including ACM CCS, ACM AsiacCS, ESORICS, ACISP, CANS, EuroPKI, ICICS, ICISC, ProvSec, and ISPEC. He is a member of the IACR.

Willy Susilo (SM’01) received the Ph.D. degree in computer science from the University of Wollongong, Wollongong, Australia. He is a Professor with the School of Computer Science and Software Engineering and the Director of Centre for Computer and Information Security Research, University of Wollongong. He currently holds the prestigious ARC Future Fellow awarded by the Australian Research Council. His main research interests include cryptography and information security. His main contribution is in the area of digital signature schemes. He has served as a program committee member in dozens of international conferences. He has published numerous publications in the area of digital signature schemes and encryption schemes.