Journal of Information Hiding and Multimedia Signal Processing ©2014 ISSN 2073-4212
Ubiquitous International Volume **5**, Number **2**, April **2014**

A Secure and Efficient t-out-of-n Oblivious Transfer Based on the Generalized Aryabhata Remainder Theorem

Yanjun Liu

School of Computer Science and Technology
Anhui University
No.111 Jiulong Rd., Hefei, 230601, China
Department of Computer Science and Information Engineering
Asia University
No.500, Lioufeng Rd., Wufeng, Taichung, 413, Taiwan
yjliu104@gmail.com

Chin-Chen Chang

Department of Information Engineering and Computer Science
Feng Chia University
No.100 Wenhwa Rd., Seatwen, Taichung, 407, Taiwan
Department of Computer Science and Information Engineering
Asia University
No.500, Lioufeng Rd., Wufeng, Taichung, 413, Taiwan
alan3c@gmail.com

Shih-Chang Chang

Department of Computer Science and Information Engineering
National Chung Cheng University
No.168, Sec.1, University Rd., Min-Hsiung Township, Chiayi, 621, Taiwan
chang.coby@gmail.com

Received January, 2013; revised November, 2013

Abstract. *The oblivious transfer (OT) protocol is a two-party protocol that is used extensively in applications of cryptography, such as wireless communication, video on demand, electronic contract signing, and secret exchanges. In a t-out-of-n OT protocol, the sender provides n messages from which the receiver can choose only t messages. After executing the t-out-of-n OT protocol, the sender does not know which t messages were received by the receiver, and the receiver gets no information other than the t messages that he or she received. Recently, Lee and Chang presented a novel t-out-of-n protocol based on the generalized Chinese remainder theorem (GCRT) that is more efficient than Wakaha and Ryota's protocol. Inspired by Lee and Chang's method, we propose a novel t-out-of-n protocol based on the generalized Aryabhata remainder theorem (GART). We proved that our proposed protocol can protect the secrecy of the message transfer phase by using BAN logic. Our analysis also indicated that the efficiency of our proposed protocol is higher than that of Lee and Chang's protocol.*
Keywords: Oblivious transfer (OT), Generalized Aryabhata remainder theorem (GART), BAN logic, Security, Efficiency

1. **Introduction.** In recent years, many applications of cryptography, such as wireless communication, video on demand, electronic contract signing, and secret exchanges, have been proposed in which the provider stores extensive information from which users can choose what they need. As a result, these applications must provide one essential functionality from the aspect of privacy, i.e., the user can obtain only the desired information from the provider, and the provider would not know which information the user selected. Oblivious transfer (OT) can be used to achieve this functionality, giving it an extremely important role in cryptography.

The concept of OT was first discussed by Rabin [21]. In 1981, Rabin proposed a two-party OT protocol in which two participants were involved, i.e., the sender, Alice, and the receiver, Bob. Alice transports a bit to Bob, and Bob has a 50% probability of obtaining the same bit and a 50% probability of obtaining nothing. Alice does not know which of the two options occurred. Rabin's OT protocol has been used extensively by cryptographic researchers around the world, and numerous OT variations of Rabin's idea have been proposed [2, 3, 13].

One well-known variation is the one-out-of-two OT protocol [13], denoted as OT_1^2 and which is executed as follows. Alice transmits two bits to Bob, and Bob has a 50% probability of receiving only one of the two bits. In addition, Alice does not know which bit Bob received. In 1986, Brassard et al. [3] proposed the 1-out-of-n OT protocol, denoted as OT_1^n, which is a natural extension of OT_1^2. In the OT_1^n protocol, Alice owns n messages, and Bob can receive only one of them. According to mutual secrecy, Alice has no idea which message has been received by Bob, and Bob cannot obtain any message except the desired one. Since Bob can get only one of n messages in OT_1^n, a more practical t-out-of-n OT protocol [2], called OT_t^n, is proposed, in which Alice owns n messages and Bob can reveal t out of them simultaneously. Furthermore, Alice cannot determine which messages Bob received, and Bob cannot obtain the other n-t messages.

Many publications related to OT_t^n protocols have been presented [1, 5, 6, 11-15, 18-20, 25] in which t calls of OT_1^n are required to accomplish the functionality of OT_t^n. However, such an approach is not efficient due to the high costs of computation and communication. In 2004, Wakaha and Ryota [24] proposed an improved OT_t^n protocol in which Bob can retrieve t out of n messages sent from Alice at the same time in every protocol run. Although their method guarantees two-party privacy, it is not practical because it lacks efficiency. Recently, Lee and Chang [17] proposed an efficient OT_t^n protocol based on the generalized Chinese remainder theorem (GCRT) [7, 8, 16]. They claimed that their OT_t^n scheme meets the basic requirements of a general OT_t^n protocol and also significantly reduces computation overhead of both the sender and the receiver.

In Lee and Chang's OT_t^n protocol, Alice uses the GCRT to compute an integer Y. Then, after interacting with Alice, Bob can successfully recover t messages that he chose by using the integer Y and some related information sent from Alice according to the GCRT. However, we found that some improvement can be made in Lee and Chang's OT_t^n protocol. Since the method of generating the integer Y dominates the computation cost of the sender, we can utilize a more efficient method to compute Y, thereby reducing the entire computation cost of an OT_t^n protocol. Therefore, inspired by Lee and Chang's method, we propose a secure and efficient OT_t^n protocol based on the generalized Aryabhata remainder theorem (GART) [10]. The main contributions of our proposed OT_t^n protocol are listed below.

(1) In the system set-up phase, the sender randomly chooses n positive, co-prime integers, a_1, a_2, ..., a_n, and an additional integer k and then computes an integer Y by the GART. In the message transfer phase, after interacting with the sender, the receiver

recovers t integers out of $\{a_i\}_{1\leq i\leq n}$, which can be combined with Y and k in the GART to successfully obtain the t messages that he or she wants to know.

(2) Our protocol can reduce computation cost of the sender since the GART utilized in our method has less time complexity than that of the GCRT, which is an important construction element in Lee and Chang's protocol.

(3) Our protocol meets the essential requirements of a general OT_t^n protocol, i.e., correctness, privacy of the receiver, and privacy of the sender.

(4) Our protocol can protect the secrecy of the message transfer phase by using BAN logic.

The rest of this paper is organized as follows. In Section 2, we briefly introduce some preliminaries that will be used in the design of our OT_t^n protocol. In Section 3, we propose our novel OT_t^n protocol on the basis of the GART. Section 4 presents analyses of the proposed protocol and the comparisons between the proposed protocol and Lee and Chang's mechanism. Finally, our conclusions are given in Section 5.

2. Preliminaries.

In this section, we briefly introduce some fundamental preliminaries that are important in the development of our protocol. First, we review the essential requirements of a general OT_t^n protocol and then describe the generalized Aryabhata remainder theorem (GART), which is a main building block of our novel OT_t^n protocol, which is presented in Section 3.

2.1. The essential requirements of a general t-out-of-n OT protocol.

In a general t-out-of-n protocol, two participants, the sender and the receiver, are involved. The sender possesses n messages, $\{m_1, m_2, ..., m_n\}$, and the receiver can obtain t of these messages simultaneously in a secure manner. Any OT_t^n protocol must satisfy the essential requirements stated below [9]:

Requirement 1. Correctness: If both the sender and the receiver follow the OT_t^n protocol, then the receiver can obtain the desired t messages after executing the protocol with the sender.

Requirement 2. Privacy of the receiver: The sender cannot determine which t messages were selected by the receiver after conducting the OT_t^n protocol with the receiver.

Requirement 3. Privacy of the sender: The receiver learns nothing other than these t messages after conducting the OT_t^n protocol with the sender.

2.2. Generalized Aryabhata remainder theorem.

In this subsection, we review the generalized Aryabhata remainder theorem (GART) and then give an example to show the computational process of the GART.

The generalized Aryabhata remainder theorem (GART) [10] is a significant extension of the Aryabhata remainder theorem (ART) [22] in which an extra modulus k is provided during the computational process. Let n positive integers, $a_1, a_2, ..., a_n$, form a moduli set, $\{a_1, a_2, ..., a_n\}$, where $\gcd(a_i, a_j) = 1$ for $i \neq j$, and assume that there are n positive integers, $y_1, y_2, ..., y_n$. A number Y can be represented as $\{y_1, y_2, ..., y_n\}$, satisfying $\mathrm{Max}\{y_i\}_{1\leq i\leq n} < k < \mathrm{Min}\{a_i\}_{1\leq i\leq n}$, where $y_i = \lfloor Y/a_i \rfloor \mathrm{mod} k$ for $i = 1, 2, ..., n$. According to the GART, the number Y can be computed from the parameters, $\{y_1, y_2, ..., y_n\}$, $\{a_1, a_2, ..., a_n\}$, and k by the following iterative algorithm:

Input: $(\{y_1, y_2, ..., y_n\}, \{a_1, a_2, ..., a_n\}, k)$

Output: X

1. $A_1 \leftarrow a_1, Y_1 \leftarrow y_1 \cdot a_1$.
2. for $i = 2$ to n do
3. $A_i \leftarrow A_{i-1} \cdot a_i$.

4. $Y_i \leftarrow k \cdot A_{i-1} \cdot ((\lceil (y_i \cdot a_i - Y_{i-1})/k \rceil \cdot (A_{i-1})^{-1}) \mathrm{mod} a_i) + Y_{i-1}$, where $(A_{i-1})^{-1} \mathrm{mod} a_i$ is the multiplicative inverse of A_{i-1} modulo a_i.
5. end for.
6. Return Y_n.

After executing this algorithm, we can get the unique solution Y_n in Z_{kA_n}, where $A_n = \prod_{i=1}^{n} a_i$. Example 2.1 illustrates the computation process of the GART.

Example 2.1. *Find a positive integer* $Y = \{y_1, y_2, y_3\} = \{1, 2, 4\}$ *with the moduli set* $\{a_1, a_2, a_3\} = \{7, 11, 13\}$ *and* $k = 5$ *by the GART.*

According to the GART, the computational process consists of the three steps shown below:

Step 1. $A_1 = a_1 = 7, Y_1 = y_1 \cdot a_1 = 1 \cdot 7 = 7$.

Step 2.

$$A_2 = A_1 \cdot a_2 = 7 \cdot 11 = 77.$$
$$\begin{aligned} Y_2 &= k \cdot A_1 \cdot ((\lceil (y_2 \cdot a_2 - Y_1)/k \rceil \cdot (A_1)^{-1}) \mathrm{mod} a_2) + Y_1 \\ &= 5 \cdot 7((\lceil (2 \cdot 11 - 7)/5 \rceil \cdot 7^{-1}) \mathrm{mod} 11) + 7 \\ &= 5 \cdot 7 \cdot 2 + 7 = 77. \end{aligned}$$

Step 3.

$$A_3 = A_2 \cdot a_3 = 77 \cdot 13 = 1001.$$
$$\begin{aligned} Y_3 &= k \cdot A_2 \cdot ((\lceil (y_3 \cdot a_3 - Y_2)/k \rceil \cdot (A_2)^{-1}) \mathrm{mod} a_3) + Y_2 \\ &= 5 \cdot 77((\lceil (4 \cdot 13 - 77)/5 \rceil \cdot 77^{-1}) \mathrm{mod} 13) + 77 \\ &= 5 \cdot 77 \cdot 5 + 77 = 2002. \end{aligned}$$

Hence, Y_3 is the solution in Z_{5005}, and the validation of Y_3 can be conducted as follows:

$$y_1 = \lfloor Y_3/a_1 \rfloor \mathrm{mod} k = \lfloor 2002/7 \rfloor \mathrm{mod} 5 = 1,$$
$$y_2 = \lfloor Y_3/a_2 \rfloor \mathrm{mod} k = \lfloor 2002/11 \rfloor \mathrm{mod} 5 = 2,$$
$$\text{and } y_3 = \lfloor Y_3/a_3 \rfloor \mathrm{mod} k = \lfloor 2002/13 \rfloor \mathrm{mod} 5 = 4.$$

3. Our Proposed Protocol.

In this section, we describe the proposed OT_t^n protocol that is based on the GART. Our proposed protocol consists of two phases, i.e., the system set-up phase and the message transfer phase.

3.1. Notations.

First, we list the notations used throughout our protocol:

- Alice: the sender
- Bob: the receiver
- $N : N = p \cdot q$, where p and q are two large prime numbers
- (e, N): the public key of Alice and $\gcd(e, \phi(N)) = 1$
- d: the private key of Alice and $ed \equiv (e, 1(\mathrm{mod}\phi(N))$
- m_i: the message preserved by Alice for $i = 1, 2, ..., n$, where the values of m_i are positive integers
- a_i: a positive integer for $i = 1, 2, ..., n$, which satisfies $\gcd(a_i, a_j) = 1$ for $i \neq j$
- k: a positive integer that satisfies $\mathrm{Max}\{m_i\}_{1 \leq i \leq n} < k < \mathrm{Min}\{a_i\}_{1 \leq i \leq n}$
- ID_i: the identity of the message m_i for $i = 1, 2, ..., n$
- c_i: the value revealed by Alice on the bulletin board that enables Bob to obtain the desired messages, where $i = 1, 2, ..., n$
- m_{s_j}: the message required by Bob with the corresponding pair (ID_{s_j}, c_{s_j}) for $j = 1, 2, ..., t$, where $m_{s_j} \in \{m_1, m_2, ..., m_n\}$

In the next two subsections, we provide a detailed description of our proposed OT_t^n protocol.

3.2. **The system set-up phase.** In this phase, Bob sends a request for retrieving the demanded messages to Alice, and Alice generates and publishes the information that will be used in the message transfer phase on the bulletin board.

Step 1. Bob requests a t-message service from Alice.

Step 2. For all messages $\{m_1, m_2, ..., m_n\}$, Alice selects a positive integer k that satisfies $k > \text{Max}\{m_i\}_{1\le i\le n}$. Then, Alice generates n positive integers, $a_1, a_2, ..., a_n$, subject to the conditions that $\gcd(a_i, a_j) = 1$ for $i \neq j$ and $\text{Max}\{m_i\}_{1\le i\le n} < k < \text{Min}\{a_i\}_{1\le i\le n}$.

Step 3. Let Y denote an integer. Alice constructs the congruence system below:

$$\begin{aligned} \lfloor Y/a_1 \rfloor &\equiv m_1 (\text{mod} k), \\ \lfloor Y/a_2 \rfloor &\equiv m_2 (\text{mod} k), \\ &\vdots \\ \lfloor Y/a_n \rfloor &\equiv m_n (\text{mod} k). \end{aligned}$$

Then, Alice uses k, a_i, and m_i for $i = 1, 2, ..., n$ to calculate the number Y according to the GART that is described in Section 2.2.

Step 4. Alice uses her public key (e, N) to compute c_i for $i = 1, 2, ..., n$ by the following equations:

$$\begin{aligned} c_1 &= a_1^e (\text{mod} N), \\ c_2 &= a_2^e (\text{mod} N), \\ &\vdots \\ c_n &= a_n^e (\text{mod} N). \end{aligned}$$

Step 5. Alice makes Y, k, and $\{(ID_i, c_i)\}_{1\le i\le n}$ known publicly on the bulletin board.

3.3. **The message transfer phase.** After obtaining the public information on the bulletin board, Bob interacts with Alice and retrieves his demanded messages by the following method:

Step 1. Since Bob wants to retrieve t out of n messages, he must pick t pairs of (ID_{s_j}, c_{s_j}) for $j = 1, 2, ..., t$ on the bulletin board.

Step 2. Bob selects t random numbers, $r_1, r_2, ..., r_t$, and employs Alice's public key (e, N) to compute:

$$\begin{aligned} g_1 &= r_1^e \cdot c_{s_1} (\text{mod} N), \\ g_2 &= r_2^e \cdot c_{s_2} (\text{mod} N), \\ &\vdots \\ g_t &= r_t^e \cdot c_{s_t} (\text{mod} N). \end{aligned}$$

and then transmits $\{g_1, g_2, ..., g_t\}$ to Alice.

Step 3. After obtaining $\{g_1, g_2, ..., g_t\}$ sent by Bob, Alice calculates

$$\begin{aligned} w_1 &= g_1^d (\text{mod} N), \\ w_2 &= g_2^d (\text{mod} N), \\ &\vdots \\ w_t &= g_t^d (\text{mod} N). \end{aligned}$$

by using her private key d, and then delivers $\{w_1, w_2, ..., w_t\}$ to Bob.

Step 4. Upon getting $\{w_1, w_2, ..., w_t\}$ sent by Alice, Bob generates

$$\begin{aligned} a_1' &= r_1^{-1} \cdot w_1 (\text{mod} N), \\ a_2' &= r_2^{-1} \cdot w_2 (\text{mod} N), \\ &\vdots \\ a_t' &= r_t^{-1} \cdot w_t (\text{mod} N). \end{aligned}$$

Step 5. Bob uses Y, k, and $\{a_1', a_2', ..., a_t'\}$ to recover the t desired messages, $\{m_{s_1}, m_{s_2}, ..., m_{s_t}\}$, as follows:

$$m_{s_1} = \lfloor Y/a'_1 \rfloor (\mathrm{mod} k),$$
$$m_{s_2} = \lfloor Y/a'_2 \rfloor (\mathrm{mod} k),$$
$$\vdots$$
$$m_{s_t} = \lfloor Y/a'_t \rfloor (\mathrm{mod} k).$$

FIGURE 1 illustrates how our protocol performs.

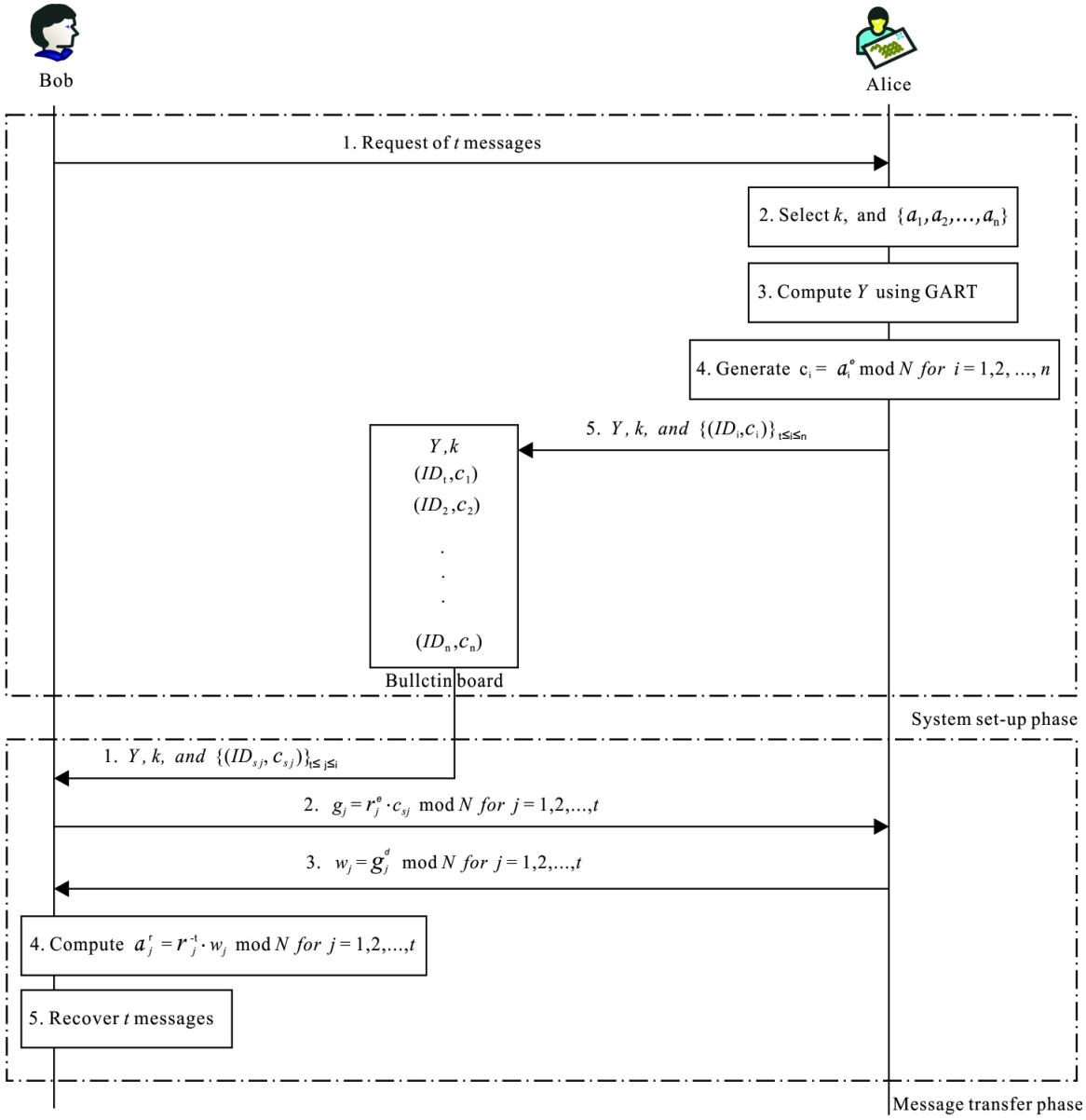

FIGURE 1. Flowchart of our proposed t-out-of-n OT protocol

4. **Discussion.** In this section, we prove that our proposed OT_t^n protocol meets the essential requirements of a general t-out-of-n protocol, provide the BAN analysis, and compare our protocol with Lee and Chang's protocol.

4.1. **Analysis of the essential requirements.** We show that our proposed OT_t^n protocol achieves the essential requirements of a general t-out-of-n protocol below:

Requirement 1. Correctness: After Alice and Bob conduct the OT_t^n protocol, Bob can obtain the t demanded messages, $m_{s_j} \in \{m_1, m_2, ..., m_n\}$, for $j = 1, 2, ..., t$, where $\{m_1, m_2, ..., m_n\}$ are the messages preserved by Alice.

Assume that Alice and Bob cannot cheat each other. During the message transfer phase, Bob picks t pairs of (ID_{s_j}, c_{s_j}) for $j = 1, 2, ..., t$ on the bulletin board, where $c_{s_j} = a_{s_j}^e (\mathrm{mod} N)$. Then, Bob computes $g_j = r_j^e \cdot c_{s_j} (\mathrm{mod} N)$ for $j = 1, 2, ..., t$ and transmits g_j to Alice. Alice calculates $w_j = g_j^d (\mathrm{mod} N)$ for $j = 1, 2, ..., t$ and delivers w_j to Bob. Bob computes:

$$\begin{aligned} a'_j &= r_j^{-1} \cdot w_j \mathrm{mod} N = r_j^{-1} \cdot (g_j^d \mathrm{mod} N) \mathrm{mod} N \\ &= r_j^{-1} \cdot ((r_j^e \cdot c_{s_j} \mathrm{mod} N)^d \mathrm{mod} N) \mathrm{mod} N \\ &= r_j^{-1} \cdot ((r_j^e \cdot (a_{s_j}^e \mathrm{mod} N))^d \mathrm{mod} N) \mathrm{mod} N \\ &= (r_j^{-1} \cdot (r_j \cdot a_{s_j})^{ed}) \mathrm{mod} N = a_{s_j} \mathrm{mod} N. \end{aligned}$$

Consider that in the system set-up phase, Alice establishes the congruence system $m_{s_j} \equiv \lfloor Y / a_{s_j} \rfloor (\mathrm{mod} k)$ for $j = 1, 2, ..., t$, and then conveys Y that is computed by the GART to Bob. Since a'_j is equal to a_{s_j}, Bob can use Y, k, and a'_j to recover the messages that he wanted as shown below:

$$\begin{aligned} \lfloor Y / a'_1 \rfloor (\mathrm{mod} k) &= \lfloor Y / a_{s_1} \rfloor (\mathrm{mod} k) = m_{s_1}, \\ \lfloor Y / a'_2 \rfloor (\mathrm{mod} k) &= \lfloor Y / a_{s_2} \rfloor (\mathrm{mod} k) = m_{s_2}, \\ &\vdots \\ \lfloor Y / a'_t \rfloor (\mathrm{mod} k) &= \lfloor Y / a_{s_t} \rfloor (\mathrm{mod} k) = m_{s_t}. \end{aligned}$$

Therefore, the proposed protocol achieves this requirement.

Requirement 2. Privacy of the receiver: After Alice and Bob conduct the OT_t^n protocol, Alice does not know that Bob acquired $\{m_{s_1}, m_{s_2}, ..., m_{s_t}\}$ among n messages.

If Alice wants to know which messages were chosen by Bob, she must first obtain a_{s_j}, and then she can obtain $m_{s_j} = \lfloor Y / a_{s_j} \rfloor (\mathrm{mod} k)$ for $j = 1, 2, ..., t$. However, Alice's plan will fail according to our protocol. After selecting t pairs of (ID_{s_j}, c_{s_j}) for $j = 1, 2, ..., t$ on the bulletin board, Bob randomly chooses t random numbers r_1, r_2, ..., r_t and utilizes Alice's public key (e, N) to compute $g_j = r_j^e \cdot c_{s_j} (\mathrm{mod} N)$ for $j = 1, 2, ..., t$. Then, Alice can obtain g_j from Bob and calculate $w_j = g_j^d (\mathrm{mod} N)$ for $j = 1, 2, ..., t$.
Because

$$\begin{aligned} w_j &= g_j^d \mathrm{mod} N = (r_j^e \cdot c_{s_j} \mathrm{mod} N)^d \mathrm{mod} N \\ &= (r_j^e \cdot (a_{s_j}^e \mathrm{mod} N) \mathrm{mod} N)^d \mathrm{mod} N \\ &= r_j \cdot a_{s_j} \mathrm{mod} N. \end{aligned}$$

Alice cannot extract a_{s_j} from the combination of r_j and a_{s_j}. As a result, the second requirement is met by our proposed protocol.

Requirement 3. Privacy of the sender: After Alice and Bob conduct the OT_t^n protocol, Bob cannot get any messages other than his required messages, $\{m_{s_1}, m_{s_2}, ..., m_{s_t}\}$.

We assume that both Alice and Bob are honest. If Bob attempts to acquire the other $n - t$ messages, he must first obtain $a'_j = r_j^{-1} \cdot w_j (\mathrm{mod} N)$ for $t + 1 \leq j \leq n$ and then compute m_{s_j} by using $m_{s_j} = \lfloor Y / a'_j \rfloor (\mathrm{mod} k)$ for $t + 1 \leq j \leq n$. However, in our protocol, Alice uses her private key d to compute $w_j = g_j^d \mathrm{mod} N$ for $j = 1, 2, ..., t$. As a result, without knowing Alice's private key d, it is impossible for Bob to calculate w_j, where $t + 1 \leq j \leq n$. Hence, Bob is unable to get other $n - t$ messages and our protocol can guarantee Alice's privacy.

4.2. **BAN analysis.** This subsection uses BAN logic [4] to verify our OT_t^n protocol. According to the analytical procedures of BAN logic, each round of the protocol must be

transformed into the idealized form. Next, we briefly describe basic notations of BAN logic as follows:
$P \xleftrightarrow{K} Q$: P and Q may communicate with each other using the shared key K. The key K will never be discovered by any principal except P or Q.
$P \overset{X}{\Leftrightarrow} Q$: The formula X is a secret known only to P and Q. Only P and Q may use X to prove their identities to one another.
$\xmapsto{K} P$: P has K as a public key. The matching secret key (denoted as K^{-1}) will never be discovered by any principal except P.
$\{X\}_K$: This represents the formula X encrypted under the key K.
$\langle X \rangle_Y$: This represents the formula X combined with the formula Y.

In our protocol, two messages are used to protect the secrecy of our message transfer phase. These messages are shown in FIGURE 1. Here, we present Alice denoted as A and *Bob* denoted as B. Then, we idealize the protocol as follows:
Message 1. $B \rightarrow A : g_j = r_j^e \cdot c_{s_j} (\mathrm{mod} N)$ for $j = 1, 2, ..., t$.
Message 2. $A \rightarrow B : w_j = g_j^d (\mathrm{mod} N)$ for $j = 1, 2, ..., t$.

Before analyzing our protocol, we first make the following assumptions:
A.1 B **believes** $\overset{e}{\mapsto} A$.
A.2 B **believes** (A **controls** c_{s_j}).
A.3 B **believes fresh** (r_j).
A.4 B **believes** (A **controls** N).
A.5 A **believes** (B **controls fresh** (r_j)).
A.6 A **believes** c_{s_j}.
A.7 A **believes** N.
A.8 A **believes** $\overset{e}{\mapsto} A$.
A.9 B **believes** (A **controls** $e^{-1} = d$).

Then, we analyze the idealized form of our proposed protocol using the above assumptions and rules of BAN logic. Details of the logic proof are presented as follows.

A receives *Message* 1. The rule shows that
A **sees** $\{g_j = r_j^e \cdot c_{s_j} (\mathrm{mod} N)$ for $j = 1, 2, ..., t\}$. (Statement 1)

We break conjunctions and produce
A **sees** B **said** $r_j^e \cdot c_{s_j}$. (Statement 2)
and
A **sees** B **said** N. (Statement 3)

By *A 7* and Statement 3, we apply the nonce-verification rule to deduce
A **believes** N. (Statement 4)

By *A 6* and Statement 2, we apply the message-meaning rule to derive
A **believes** B **said** r_j^e. (Statement 5)

By *A 8* and Statement 5, the message-meaning rule applies and yields
A **believes** B **said** r_j. (Statement 6)

By *A 5* and Statement 6, we apply the nonce-verification rule to deduce
A **believes** r_j. (Statement 7)

Then, B receives *Message 2*. The annotation rule yields that
B **sees** $\{w_j = g_j^d \mathrm{mod} N$ for $j = 1, 2, ..., t\}$. (Statement 8)

We break conjunctions and produce as follows:
B **sees** A **said** g_j^d. (Statement 9)
and
B **sees** A **said** N. (Statement 10)

By *A 4* and Statement 10, we apply the nonce-verification rule to obtain
B **believes** N. (Statement 11)

By *A 9* and Statement 9, we apply the message-meaning rule to deduce
B **believes** A **said** g_j. (Statement 12)
By *Message* 1, the message-meaning rule applies and yields
B **believes** A **said** $r_j^e \cdot c_{s_j}$. (Statement 13)
By *A 2* and Statement 13, we apply the message-meaning rule to derive
B **believes** A **said** r_j^e. (Statement 14)
By *A 1* and Statement 14, the message-meaning rule applies and yields
B **believes** A **said** r_j. (Statement 15)
By *A 3* and Statement 15, we apply the nonce-verification rule to deduce
B **believes** r_j. (Statement 16)
Based on Statement 7 and Statement 16, we prove our proposed protocol can protect the secrecy of our message transfer phase.

4.3. **Comparison.** In this subsection, we compare the efficiency of our protocol and Lee and Chang's protocol. Before the comparison, we consider the generalized Chinese remainder theorem (GCRT) that was used in Lee and Chang's method.

The GCRT can be described as follows. Let n positive integers, a_1, a_2, ..., a_n, form a moduli set, $\{a_1, a_2, ..., a_n\}$, where $\gcd(a_i, a_j) = 1$ for $i \neq j$. Assume that there are n positive integers, y_1, y_2, ..., y_n, and an additional modulus, k, which satisfies $\text{Max}\{y_i\}_{1\leq i\leq n} < k < \text{Min}\{a_i\}_{1\leq i\leq n}$. A congruence system can be constructed as shown below:

$$\begin{aligned} \lfloor Y/a_1 \rfloor &\equiv y_1 (\text{mod} k), \\ \lfloor Y/a_2 \rfloor &\equiv y_2 (\text{mod} k), \\ &\vdots \\ \lfloor Y/a_n \rfloor &\equiv y_n (\text{mod} k). \end{aligned}$$

Therefore, the unique solution Y of this congruence system that satisfies $0 \leq Y < k\prod_{i=1}^{n} a_i$ can be computed by the following equation: $Y = \sum_{i=1}^{n} a_i^{'} \cdot a_i^{''} \cdot b_i (\text{mod} kA)$, where $A = \prod_{i=1}^{n} a_i$, $a_i^{'} = k \cdot A/a_i$, $a_i^{'} \cdot a_i^{''} \equiv k (\text{mod} k \cdot a_i)$, and $b_i = \lceil y_i \cdot a_i/k \rceil$.

Notice that the GCRT and the GART both use n positive co-prime integers, a_1, a_2, ..., a_n, n positive integers, y_1, y_2, ..., y_n, and an integer k satisfying $\text{Max}\{y_i\}_{1\leq i\leq n} < k < \text{Min}\{a_i\}_{1\leq i\leq n}$, to construct a congruence system of the integer Y. The difference between the GCRT and the GART lies only in the approach used to compute Y. In the GCRT, a modular operation with a large number, kA, must be computed to obtain the unique solution Y, which is a time-consuming operation. Unlike the GCRT, the GART decomposes this time-consuming operation into several iterations, each of which computes a modular operation with a smaller number, thus making it more efficient than the GCRT [10, 22].

In the OT_t^n protocol, $\{y_1, y_2, ..., y_n\} = \{m_1, m_2, ..., m_n\}$ are the messages possessed by the sender, Alice. Now, we show that our protocol is more efficient than Lee and Chang's protocol. Since both protocols have the same communication cost due to their enabling the receiver to obtain t messages simultaneously in one protocol run instead of repeating an OT_t^n protocol t times, we only concentrate on the comparison of their computation cost. Let T_{exp} denote the time required to execute one modulo exponential operation. Although the computation cost of the receiver of our protocol is tT_{exp}, the same as Lee and Chang's protocol, where t is the number of messages that Bob requires to obtain, we demonstrate that our method requires less computation cost for the sender. Since the method that is used to generate the number Y in the system set-up phase dominates the computation cost of the sender, we now compare the time complexity of the GCRT and the GART.

In the system set-up phase, our protocol replaces the GCRT with the GART to compute the number Y. In the following, we determined that the GART has lower time complexity than that of the GCRT. According to the GCRT, $Y = \sum_{i=1}^{n} a_i^{'} \cdot a_i^{''} \cdot b_i (\mathrm{mod} kA)$, where $A = \prod_{i=1}^{n} a_i$, $a_i^{'} = k \cdot A/a_i$, $a_i^{'} \cdot a_i^{''} \equiv k (\mathrm{mod} k \cdot a_i)$, and $b_i = \lceil y_i \cdot a_i / k \rceil$. Here, $k \cdot A$ and $a_i^{'} \cdot a_i^{''}$ can be pre-computed. Hence, there are $2n$ multiplications, n divisions, (n-1) additions, and one modular operation. Assuming that a_i is allocated h digits, the multiplication/division and addition of two moduli require h^2 and h bit operations, respectively. Moreover, an h-bit modular operation requires h^2 bit operations. Therefore, the computation cost of the GCRT is about $2n \cdot h^2 + n \cdot h^2 + (n-1) \cdot h + ((n+1) \cdot h)^2$ bit operations, where $(n+1) \cdot h$ is the number of digits in kA. Thus, the time complexity of the GCRT is $O(n^2 h^2)$.

According to the GART, $Y_i = k \cdot A_{i-1} \cdot ((\lceil (m_i \cdot a_i - Y_{i-1})/k \rceil \cdot (A_{i-1})^{-1}) \mathrm{mod} a_i) + Y_{i-1}$, where $A_1 = a_1, Y_1 = m_1 \cdot a_1$, and $A_i = A_{i-1} \cdot a_i$. The equation above must execute (n-1) rounds.

Here, $k \cdot A_{i-1} \cdot ((A_{i-1})^{-1} \mathrm{mod} a_i)$ can be pre-computed. So, there are two multiplications, one subtraction, one division, one addition, and one modular operation in every round. Assume that k is allocated h digits and that the division and subtraction of two moduli require h^2 and h bit operations, respectively. As a result, after performing (n-1) rounds, the computation cost of the GART is about $(n-1) \cdot (2h^2 + h + h^2 + h + h^2)$ bit operations. Therefore, the time complexity is $O(nh^2)$. This analysis shows that the GART requires lower time complexity than that of the GCRT, which implies that our proposed protocol can decrease the sender's computation cost.

5. **Conclusions.** In this paper, we proposed a secure and efficient OT_t^n protocol based on the generalized Aryabhata remainder theorem (GART). Our protocol meets the essential requirements of a general OT_t^n protocol, i.e., correctness, privacy of the receiver, and privacy of the sender. We also determined that our protocol reduces the sender's computation cost due to the low time complexity of the GART that is used in our method. In addition, a BAN analysis was given to prove that our protocol can protect the secrecy of the message transfer phase.

Acknowledgment. This research was supported in part by the National Nature Science Foundation of China (grant number: 61202228) and the College Natural Science Key Project of Anhui Province of China (grant number: KJ2012A008).

REFERENCES

[1] B. Aiello, Y. Ishai, and O. Reingold, Priced oblivious transfer: how to sell digital goods, *Proc. of EUROCRYPT*, LNCS 2045, pp. 119-135, 2001.

[2] M. Bellare, and S. Micali, Non-interactive oblivious transfer and application, *Proc. of CRYPTO*, LNCS 435, pp. 547-557, 1989.

[3] G. Brassard, C. Crepeau, and J. M. Robert, Information theoretic reductions among disclosure problems, *Proc. of the 27th Annual Symposium on Foundations of Computer Science*, pp. 168-173, 1986.

[4] M. Burrows, M. Abadi, and R. Needham, A logic of authentication, *ACM Trans. Computer Systems*, vol. 8, no. 1, pp. 18-36, 1990.

[5] C. Cachin, On the foundations of oblivious transfer, *Proc. of EUROCRYPT*, LNCS 1403, pp. 361-374, 1998.

[6] G. D. Crescenzo, T. MalkinOn, and R. Ostrovsky, Single database private information retrieval implies oblivious transfer, *Proc. of EUROCRYPT*, LNCS 1807, pp. 122-138, 2000.

[7] C. C. Chang, and Y. P. Lai, A fast modular square computing method based on the generalized Chinese remainder theorem for prime module, *Journal of Applied Mathematics and Computation*, vol. 161, no. 1, pp. 181-194, 2005.

[8] C. C. Chang, and H. C. Lee, A new generalized group-oriented cryptoscheme without trusted centers, *IEEE Journal on Selected Areas in Communications*, vol. 11, no. 5, pp. 725-729, 2006.

[9] C. C. Chang, and J. S. Lee, Robust t-out-of-n oblivious transfer mechanism based on CRT, *Journal of Network and Computer Applications*, vol. 32, no. 1, pp. 226-235, 2009.
[10] C. C. Chang, J. S. Yeh, and J. H. Yang, Generalized Aryabhata remainder theorem, *International Journal of Innovative Computing, Information and Control*, vol. 6, no. 4, pp. 1865-1871, 2010.
[11] Y. Z. Ding, Oblivious transfer in the bounded storage model, *Proc. of CRYPTO*, LNCS 2139, pp. 155-170, 2001.
[12] Y. Dodis, and S. Micali, Lower bounds for oblivious transfer reductions, *Proc. of EUROCRYPT*, LNCS 1592, pp. 42-55, 1999.
[13] S. Even, O. Goldreich, and A. Lempel, A randomized protocol for signing contracts, *Communications of the ACM*, vol. 28, no. 6, pp. 637-647, 1985.
[14] J. A. Garay, and P. D. MacKenzie, Concurrent oblivious transfer, *Proc. of the 41st Annual Symposium on Foundations of Computer Science*, pp. 314-324, 2000.
[15] J. Kilian, Founding cryptography on oblivious transfer, *Proc. of the 20th annual ACM symposium on Theory of computing*, pp. 20-31, 1988.
[16] Y. P. Lai, and C. C. Chang, Parallel computational algorithms for generalized Chinese remainder theorem, *Journal of Computers and Electrical Engineering*, vol. 29, no. 8, pp. 801-811, 2003.
[17] J. S. Lee, and C. C. Chang, Choosing t-out-of-n secrets by oblivious transfer, *Journal of INFORMATION & SECURITY*, vol. 18, no. 8, pp. 67-84, 2006.
[18] N. Y. Lee, and T. Hwang, On the security of fair blind signature scheme using oblivious transfer, *Journal of Computer Communications*, vol. 22, no. 3, pp. 287-290, 1999.
[19] M. Naor, and B. Pinkas, Efficient oblivious transfer protocols, *Proc. of the 12th annual ACM-SIAM symposium on Discrete algorithms*, pp. 448-457, 2001.
[20] M. Naor, and B. Pinkas, Oblivious transfer and polynomial evaluation, *Proc. of the 31st annual ACM symposium on Theory of computing*, pp. 245-254, 1999.
[21] M. O. Rabin, How to exchange secrets by oblivious transfer, *Technical Report TR-81*, Aiken Computation Laboratory, Harvard University, USA, 1981.
[22] T. R. N. Rao, and C. H. Yang, Aryabhata remainder theorem: relevance to public-key Cryptoalgorithms, *Journal of Circuits, Systems and Signal Processing*, vol. 25, no. 1, pp. 1-5, 2004.
[23] W. G. Tzeng, Efficient 1-out-of-n oblivious transfer protocols with universally usable parameter, *IEEE Trans. Computers*, vol. 53, no. 2, pp. 232-240, 2004.
[24] O. Wakaha, and S. Ryota, k-out-of-n oblivious transfer without random oracles, *Journal of IEICE Trans Fundam Electron Commun Comput Sci*, vol. E87-A, no. 1, pp. 147-151, 2004.
[25] Q. H. Wu, J. H. Zhang, and Y. M. Wang, Practical t-out-n oblivious transfer and its applications, *Journal of Information and Communications Security*, LNCS 2836, pp. 226-237, 2003.