Distributed Paillier Plaintext Equivalence Test

Pei-Yih Ting and Xiao-Wei Huang
(Corresponding author: Pei-yih Ting)

Department of Computer Science, National Taiwan Ocean University
2, Pei-Ning Road, Keelung 202, Taiwan, R.O.C. (Email: pyting@mail.ntou.edu.tw)

(Received Apr. 13, 2006; revised and accepted May 7, 2006 & Nov. 8, 2006)

Abstract

Consider the following scenario with $N+2$ parties, A, B, X_1, X_2, $\cdots$, X_N. Party A has a secret a, party B has a secret b, and party X_i has a secret x_i, for $i = 1, \cdots, N$. They want to know if $a = b$ without revealing any information about their secrets. We propose a distributed protocol for this problem based on the Paillier threshold homomorphic encryption scheme with a public broadcast channel. This protocol is suitable for voting which requires public verifiability. This protocol is secure and robust in an honest majority environment. We proved the security in a formal framework for secure multi-party computation.

Keywords: Homomorphic encryption, PET, plaintext equivalence test, secure multi-party computation protocol

1 Introduction

In this problem, there are $N+2$ parties, A, B, X_1, X_2, $\cdots$, X_N. Party A has a secret a, party B has a secret b, and party X_i has a secret x_i, for $i = 1, \cdots, N$. x_i is share of the private key of the underlying homomorphic encryption scheme. They want to know if $a = b$, under the constraint that A, B and X_i (for $i = 1, \cdots, N$) should not learn anything about other's secrets. The protocol to deal with this problem is known as the Plaintext Equivalence Test (PET) and is a specific instance of secure multiparty computation [10, 11]. We briefly sketch previous protocols.

In 2000, Jakobsson and Juels [10] presented a distributed plaintext equality test based on the ElGamal public-key system [5]. In their protocol, the equivalence of the plaintexts a and b corresponding to two ElGamal ciphertexts $E_{pk}(a) = (\alpha_a, \beta_a), E_{pk}(b) = (\alpha_b, \beta_b)$ is determined. A trusted third party C, who owns the secret key SK of the ElGamal scheme, picks a random number r, computes $(\zeta, \xi) = ((\alpha_a/\alpha_b)^r, (\beta_a/\beta_b)^r)$ and decrypts (ζ, ξ). C declares that $a = b$ if $D_{sk}((\zeta, \xi)) = 1$; C declares that $a \neq b$ otherwise, where $D_{sk}(\cdot)$ denotes the ElGamal decryption algorithm. Note that the party C can be a group of users sharing the decryption key and $D_{sk}(\cdot)$ can be executed in a distributed manner. Their protocol is secure unless more than t users are colluding. It is useful for applications such as voting or auction based on the ElGamal public-key system. However, a protocol of similar purpose is unavailable for applications based on the Paillier [15] public-key system.

In 2005, Li and Wu [11] proposed a co-operative private equality test based on the Paillier public-key system. There are three parties in their protocol: A, B, and C, in which A and B know a, b individually and C knows the decryption key SK. First, A and B jointly generate a random number r_{ab}. Second, A encrypts $r_{ab}a$ as $\xi = E_{pk}(r_{ab}a)$ and sends ξ to B through a secure channel. Then B encrypts $-r_{ab}b$ as $\eta = E_{pk}(-r_{ab}b)$, computes $\zeta = \eta\xi$ and sends ζ to C through a secure channel. Finally, C decrypts ζ and declares that $a = b$ if $D_{sk}(\zeta) = 0$, or $a \neq b$ otherwise. In the above, $E_{pk}(\cdot)$ and $D_{sk}(\cdot)$ denote the Paillier encryption and decryption algorithms, respectively. Li and Wu have proved that their protocol is secure under the assumptions of the passive adversary model and a secure channel. There are two problems with the above protocol. First, it is quite often that in a voting or an auction scenario, to disrupt the protocol or to manipulate the protocol results in a brute-force way is as interesting as to learn the intents of other players. For example, if C cheats, he can declare arbitrarily $a = b$ or $a \neq b$; if B is dishonest, he can send $E_{pk}(0)$ or $E_{pk}(1)$ to manipulate the result. It is essential to consider active adversaries in the design of a practical protocol. Second, the employment of homomorphic encryption schemes enables the design of a public verifiable voting application, the secure channel model used in Li's [11] protocol hinders the protocol from scrutiny of the public.

In this paper, we propose a protocol for distributed plaintext equivalence test based on the threshold Paillier public-key system with a public broadcast channel. The protocol is efficient and secure with honest majority assumption and is analyzed in a formal secure multi-party computation framework [7, Chap. 7].

In Section 2, we recall the cryptographic primitives such as Paillier cryptosystem, threshold Paillier decryption algorithm, and some zero-knowledge proofs. In Section 3, we recall the multi-party computation model and

give a formal secure definition to the protocol for PET. In Section 4, the complete protocol is described. Then, we present the formal security analysis in Section 5, a typical voting application in Section 6, and give some conclusions in Section 7.

2 Cryptographic Primitives

2.1 Paillier Cryptosystem

The Paillier Cryptosystem [15] is briefly summarized: let $n = pq$, where p and q are large primes. Let g be an element in $\mathbb{Z}^*_{n^2}$ with g's order being a non-zero multiple of n. (n, g) is the public key and $\lambda(n) = lcm(p-1, q-1)$ is the private key of the cryptosystem. The encryption and decryption algorithms are described as follows:

Encryption: for plaintext $m < n$
select a random $r \in \mathbb{Z}^*_n$
ciphertext $c = g^m r^n \bmod n^2$.

Decryption: for ciphertext $c < n^2$
plaintext $m = \frac{L(c^\lambda \bmod n^2)}{L(g^\lambda \bmod n^2)} \bmod n$
where $L(u) = (u-1)/n$ for $u = 1 \bmod n$.

The Paillier cryptosystem has the additive homomorphic property, which is essential to the design of the homomorphic tallying process of many e-voting protocols:

for all $m_1, m_2 \in \mathbb{Z}_n$ and $k \in \mathbb{N}$,

$$\begin{aligned} E_{pk}(m_1)E_{pk}(m_2) &= E_{pk}(m_1 + m_2) \\ E_{pk}(k \cdot m) &= E_{pk}(m)^k. \end{aligned}$$

The Paillier encryption is proved multi-message semantically secure under the assumption of decisional composite residue (DCRA) [15], which implies computational composite residue (CRA) and then implies RSA assumption with public encryption coefficient being n.

Definition 1. *multi-message indistinguishable encryption (G, $\bar{E}$, $\bar{D}$) [7, Def. 5.2.14]*

$\forall \bar{x} = (x_1, \ldots, x_{t(n)}), x_i \in \{0,1\}^{\ell(n)}$,
$\forall \bar{y} = (y_1, \ldots, y_{t(n)}), y_i \in \{0,1\}^{\ell(n)}$, *and*
$\forall$ *probabilistic polynomial time A,*
$|\Pr\{A(1^n, G(1^n), \bar{E}_{G(1^n)}(\bar{x})) = 1\} -$
$\Pr\{A(1^n, G(1^n), \bar{E}_{G(1^n)}(\bar{y})) = 1\}| < \frac{1}{p(n)}$.

2.2 Threshold Paillier Cryptosystem

In order to avoid possible frauds in which a party who knows the secret key decrypts an arbitrary ciphertext and violates the privacy of a participating party, we use the threshold Paillier cryptosystem [1, 3] to encrypt and decrypt.

The protocol includes the following players: a dealer and a set of N users X_i.

- **Initialization:** A user X_i gets a secret share SK_i of the private key SK corresponding to the public key PK through a trusted dealer. This trusted dealer can be replaced by a distributed key generation algorithm [4].
- **Encryption:** Any user can run the encryption algorithm described in Section 2.1 using PK.
- **Decryption:** A user X_i decrypts the ciphertext c using his share of the secret key SK_i to get the partial decryption c_i and forms a zero-knowledge proof (ZKP) of validity of the partial decryption. A set of more than t cooperating users, whose validity proofs are validated correctly, can recover the plaintext using a Lagrange-like combining protocol. The details will be integrated into the protocol and explained in Section 4.

2.3 Zero-knowledge Proof

ZKP for equal discrete logs: This ZKP follows [3] and is a generalization of [2] over $\mathbb{Z}^*_{n^2}$. The prover and verifier know the values $u, \tilde{u}, v, \tilde{v}, n$ and the prover knows the secret y such that $\tilde{u} = u^y \bmod n^2$ and $\tilde{v} = v^y \bmod n^2$. The prover wants to convince the verifier that the two discrete log $\mathrm{dlog}_u(\tilde{u})$ and $\mathrm{dlog}_v(\tilde{v})$ are equal without revealing the value y.

First, the prover chooses a random number $r \in \mathbb{Z}_n$ and commits $a = u^r \bmod n^2, b = v^r \bmod n^2$ to the verifier. Then, the verifier chooses a random challenge $e \in \mathbb{Z}_n$ and sends e to the prover. Finally, the prover sends the response $z = r + ey$ to the verifier and the verifier checks whether $u^z = a\tilde{u}^e \bmod n^2, v^z = b\tilde{v}^e \bmod n^2$. If both equations hold, then the verifier is convinced that $\mathrm{dlog}_u(\tilde{u}) = \mathrm{dlog}_v(\tilde{v})$.

3 Secure Multi-party Computation

3.1 Y-evaluation and PET

Y-evaluation is a general secure multi-party computation problem [12, 16], in which N parties, $X_1, X_2, \cdots, X_N$, want to compute $y = f(x_1, x_2, \cdots, x_N)$ without revealing any information about their secret inputs x_i. PET is a specific instance of the Y-evaluation problem.

In this paper, we use a version of threshold Paillier system [3] with its public key $pk \in PK$ and the shares of secret key $x_i \in SK$. Party A encrypts a as $\bar{a} = E_{pk}(a)$ and party B encrypts b as $\bar{b} = E_{pk}(b)$. The set-up assumptions are that the public key pk, $\bar{a}$, and $\bar{b}$ are known to all parties and the share x_i of the secret key is held by the party X_i. The function is computed jointly by N parties in the multi-party computation model and is defined by:

$$f : \mathcal{X} \times \mathcal{X} \times \cdots \times \mathcal{X} \to \{0,1\}$$

and

$$f(x_1, x_2, \cdots, x_N) = \begin{cases} 1, \text{ if } a \neq b \\ 0, \text{ if } a = b \end{cases}$$

where $\mathcal{X} = SK$.

Our goal is to design a protocol for the PET problem with the following properties:

1) Correctness: After the protocol, all parties will be convinced of the correctness of the result.

2) Privacy: No one in the protocol can learn more information about other party's secret except $a = b$ or not.

3) Robustness: No one can disrupt the protocol or manipulate the results under the threshold assumption.

3.2 Adversary Model

In the following, we consider the multi-party computation model with honest majority [7] where an adversary V controls some minority of parties to learn extra information, to disrupt the protocol or to manipulate the results of the protocol. Since every NP problem has a corresponding ZKP [8], we augment each secure protocol step with ZKPs to guarantee the detection of any deviation from the protocol. The remaining protocol messages are therefore valid as if the adversaries were passive. Our protocol only requires that the number of colluding parties is less than $t = \frac{N}{2}$.

We assume all parties communicate via a public broadcast channel such as a bulletin board system (BBS). Besides, we assume that the decisional composite residue problem [15] is computationally intractable.

3.3 Security Definition

In this section, we give a formal security definition for the Plaintext Equivalence Test protocol. The privacy of the protocol is defined in terms of the behavioral differences between the ideal and real world model [7, 11].

In the real model, all parties run the real protocol and the adversary attacks the protocol. An adversary V controls $t-1$ parties of $\{X_{\phi(i)}\}_{i=1,\cdots,t-1}$ where $\phi : \{1, \cdots, t-1\} \rightarrow \{1, \cdots, N\}$ is a one-to-one mapping. After the protocol, all parties output the value received and the adversary outputs arbitrary functions of their joint views.

In the ideal model, a trusted party K obtains secret inputs from all parties $\hat{X}_1, \hat{X}_2, \cdots, \hat{X}_N$, computes f privately and sends the result back to each party. An adversary $\hat{V}$ controls parties $\{\hat{X}_{\phi(i)}\}_{i=1,\cdots,t-1}$ with the same index set as in the real model. All parties output the value received from K and the adversary outputs arbitrary functions of their joint views.

Definition 2. *A Plaintext Equivalence Test protocol is secure with honest majority if for all real probabilistic polynomial-time adversary V, controlling some minority of the parties $\{X_{\phi(i)}\}_{i=1,\cdots,t-1}$ and $(t-1)$-inputs $\{x_{\phi(i)}\}_{i=1,\cdots,t-1}$, there exists an ideal probabilistic polynomial-time adversary $\hat{V}$, controlling the same set of parties and inputs, such that for all possible $x_1, x_2, \cdots, x_N$ the outputs of $(X_1, X_2, \cdots, X_N, V)$ in the real model and those of $(\hat{X}_1, \hat{X}_2, \cdots, \hat{X}_N, \hat{V})$ in the ideal model are computationally indistinguishable [9].*

4 Our Plaintext Equivalence Test Protocol

Assume there are N parties, $X_1, X_2, \cdots$, and X_N, participating this protocol, and external parties A and B having their secret inputs a, b, respectively. They want to know whether $a = b$ without revealing other information. For the sake of simplicity, we assume a trusted dealer $\mathcal{D}$ in the generation and distribution of the underlying threshold Paillier system. Note that the trusted party $\mathcal{D}$ can be removed by using distributed key generation protocols [14, 4]. Let threshold t be $\frac{N}{2}$. This protocol is secure if there are less than t colluding active adversaries. Figure 1 illustrates the procedure of the proposed protocol.

Key generation:
$\mathcal{D}$ picks the primes p and q such that $p = 2p' + 1$ and $q = 2q' + 1$ where p' and q' are large primes. Let $n = pq$, $m = p'q'$, $g = 1+n$, and $\Delta = N!$. Then, $\mathcal{D}$ calculates d with the Chinese Remainder Theorem such that $d = 0 \bmod m$ and $d = 1 \bmod n$ and selects a polynomial $f(X) = \sum_{i=0}^{t} a_i X^i$ where $a_0 = d$ and $\{a_i\}_{i=1,\cdots,N}$ are random numbers in $\{0, \cdots, n(m-1)\}$. The secret share of d is $x_i = f(i)$ (for $i = 1, \cdots, N$) and the public key pk is (g, n). For the verification of each joint decryption, $\mathcal{D}$ picks a random number $v \in_R \mathbb{Z}^*_{n^2}$, computes $v_i = v^{\Delta x_i}$ for each X_i and publishes $v, \{v_i\}_{i=1,\cdots,N}$.

Setup: The external party A encrypts a as $\bar{a} = E_{pk}(a)$ and B encrypts b as $\bar{b} = E_{pk}(b)$ by Paillier encryption algorithm with the public key (g, n). The public key (g, n), $\bar{a}$ and $\bar{b}$ are known to all parties and the share x_i of the secret key is held by the party X_i.

Step 1: Each X_i secretly picks a random number $r_i \in \mathbb{Z}^*_n$, computes $\bar{c_i} = (\bar{a}/\bar{b})^{r_i} \bmod n^2$ and submits a proof-of-knowledge of the discrete log $\mathrm{dlog}_{(\bar{a}/\bar{b})}\bar{c_i}$ ZKP [17] as shown in Step 1 of Figure 1.

Step 2: All parties jointly determine the set $S = \{i | X_i$ submits the correct proof for $\bar{c_i}\}$ and calculate

$$c = \prod_{i \in S} \bar{c_i} \bmod n^2.$$

Due to the additive homomorphic property mentioned in Section 2.1, we know that c is the encryption of $(a - b) \cdot (\sum_{i \in S} r_i)$.

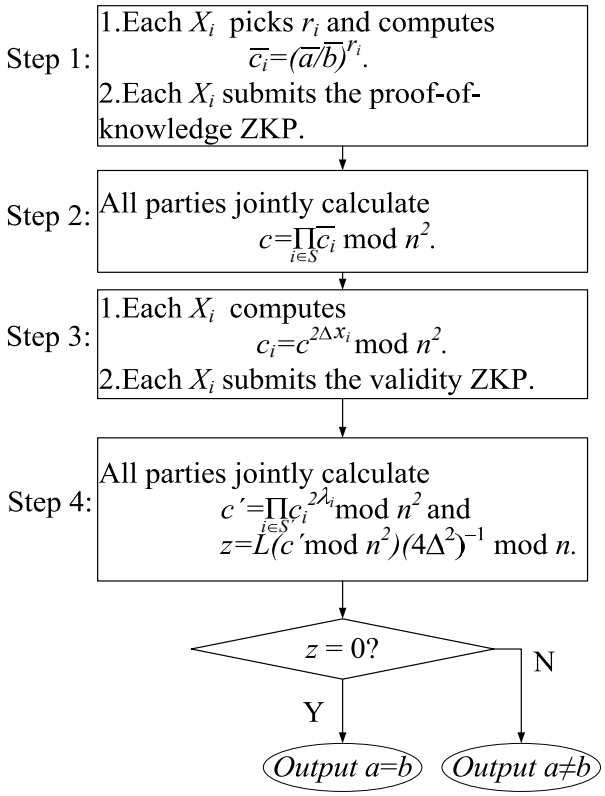

Figure 1: Illustrative diagram of the protocol

Step 3: Each party X_i calculates the partial decryption $c_i = c^{2\Delta x_i} \bmod n^2$ and publishes it as shown in Step 3 of Figure 1. Besides, X_i provides the ZKP for equal discrete logs as described in Section 2.3 to prove that he uses exactly the same secret x_i that is involved in the calculation of c_i and v_i, i.e. $\mathrm{dlog}_{c^4}(c_i^2) = \mathrm{dlog}_v v_i$.

Step 4: If there are more than t c_i's having valid ZKPs, each party determines the set $S' = \{i | X_i$ submits the valid proof for $c_i\}$ and calculates

$$c' = \prod_{i\in S'} c_i^{2\lambda_i} \bmod n^2,$$

where $\lambda_i = \Delta \prod_{i'\in S'\setminus i} \frac{-i'}{i-i'}$ are integers. The decrypted plaintext is $z = L(c' \bmod n^2)(4\Delta^2)^{-1} \bmod n$. If $z = 0$ then the protocol declares $a = b$; otherwise, declares $a \neq b$.

Note that the threshold Paillier cryptosystem used in this paper can be replaced by other variants of threshold Paillier cryptosystems such as [1].

5 Security Analysis

In this section, we analyze the security of the proposed protocol and prove the privacy according to the definition in Section 3.3.

Correctness: Consider the **Step4** in Section 4:

$$c = \prod_{i\in S} \bar{c_i} = \prod_{i\in S} (\frac{\bar{a}}{\bar{b}})^{r_i} = (\frac{\bar{a}}{\bar{b}})^{\sum_{i\in S} r_i} \bmod n^2.$$

Let r denote $\sum_{i\in S} r_i$. The probability that $n\lambda \mid r$ is clearly negligible if there is at least one party X_i choosing r_i randomly. By the additive homomorphic property, we know that c is the encryption of $(a-b)\cdot r$, i.e.

$$\begin{aligned} c &= (\frac{\bar{a}}{\bar{b}})^r = (\frac{g^a {r_a}^n}{g^b {r_b}^n})^r \\ &= (g^{r(a-b)}(r_a r_b^{-1})^{rn}) \bmod n^2. \end{aligned}$$

Then,

$$\begin{aligned} c' &= \prod_{i\in S'} c_i^{2\lambda_i} = \prod_{i\in S'} (c^{2\Delta x_i})^{2\lambda_i} \\ &= c^{4\Delta \sum_{i\in S'} x_i \Delta \prod_{i'\in S'\setminus i} \frac{-i'}{i-i'}} \\ &= c^{4\Delta^2 f(0)} = c^{4\Delta^2 d} \bmod n^2. \end{aligned}$$

Note that $c = g^{r(a-b)}(r_a r_b^{-1})^{rn}$, thus

$$c' = c^{4\Delta^2 d} = g^{4\Delta^2 dr(a-b)}((r_a r_b^{-1})^{4\Delta^2 drn}) \bmod n^2.$$

Since $d = 0 \bmod m$, $d = 1 \bmod n$, and $g = 1+n$, we have $(1+n)^{kn} = 1 \bmod n^2$ and $x^{2mn} = 1 \bmod n^2$. Thus,

$$c' = (1+n)^{4\Delta^2 r(a-b)} \bmod n^2,$$

and

$$\begin{aligned} z &= L(c' \bmod n^2)(4\Delta^2)^{-1} \bmod n \\ &= (\frac{((1+n)^{4\Delta^2 r(a-b)} \bmod n^2) - 1}{n}) \cdot \\ &\quad (4\Delta^2)^{-1} \bmod n \\ &= (\frac{1 + 4\Delta^2 r(a-b)n - 1}{n})(4\Delta^2)^{-1} \bmod n \\ &= 4\Delta^2 r(a-b)(4\Delta^2)^{-1} = r(a-b) \bmod n. \end{aligned}$$

Thus, $z = 0$ if $a = b$; $z \neq 0$ if $a \neq b$ and $r \neq 0 \bmod n\lambda$.

Privacy: Through the use of a threshold cryptosystem, no colluding group of size less than t can decrypt $\bar{a}$ and $\bar{b}$ to get the plaintexts a and b, respectively. Thus, the secrecy of a and b is protected by the secret sharing scheme and the DCRA computational intractability assumption of the underlying Paillier cryptosystem. Besides, the honest majority assumption implies that nobody knows the exact value of r, $z = r\cdot(a-b)$ tells nothing about a or b except whether a equals b.

Now we provide a formal security proof with respect to the privacy definition in the secure multi-party computation model and the security definition of a multi-message indistinguishable encryption. We consider N

participating parties $\{X_i\}_{i=1,\ldots,N}$ as well as a passive adversary V who controls $t-1$ parties, $\{X_i\}_{i\in\Phi}$ where $\Phi=\{\phi(i)\}_{i=1,\ldots,t-1}$. To simplify the following proof, we treat the private random value r_i as input, i.e. each player X_i inputs independently uniformly distributed x_i and r_i and V merely gathers information about private inputs of honest players $\{X_i\}_{i\in\bar{\Phi}}$ where $\bar{\Phi}$ denotes the complement set $\{1,\ldots,N\}\backslash\Phi$.

Suppose the protocol is not secure, by definition there exists a PPT adversary V in the real model such that for all possible PPT adversaries $\hat{V}$ in the ideal world, $(X_1(\cdot),\cdots,X_N(\cdot),V(\cdot))$ and $(\hat{X_1}(\cdot),\cdots,\hat{X_N}(\cdot),\hat{V}(\cdot))$ are computationally distinguishable for all possible inputs $x_1,\cdots,x_N$ and $r_1,\cdots,r_N$, where $X_i(\cdot)$ is the output of X_i, $V(\cdot)$ is the output of V, $\hat{X_i}(\cdot)$ is the output of $\hat{X_i}$, and $\hat{V}(\cdot)$ is the output of $\hat{V}$. Without deviating from the protocol, the output of any real player X_i can always be simulated in the ideal model by $\hat{X_i}$ while those adversarial behaviors are modelled by the output of V and $\hat{V}$. Thus, we focus on the outputs of the adversaries in both models.

In the real world, the adversary V can see the views of $t-1$ parties $\{X_i\}_{i\in\Phi}$. Thus, the information viewed by V is

$$\begin{aligned} view_V \triangleq \{ \quad & pk=(g,n), \bar{a}=E_{pk}(a), \bar{b}=E_{pk}(b), \\ & \{x_i\}_{i\in\Phi}, \{r_i\}_{i\in\Phi}, \\ & \{\bar{c_i}\}_{i=1,\cdots,N}, c, \{c_i\}_{i=1,\cdots,N}, c', z, \\ & f((x_1,r_1),\cdots,(x_N,r_N)) \text{ and} \\ & \text{all the zero-knowledge proofs} \quad \}. \end{aligned}$$

In the ideal world, the information viewed by $\hat{V}$ is

$$\begin{aligned} view_{\hat{V}} \triangleq \{ \quad & pk=(g,n), \bar{a}=E_{pk}(a), \bar{b}=E_{pk}(b), \\ & \{x_i\}_{i\in\Phi}, \{r_i\}_{i\in\Phi}, \\ & \text{and } f((x_1,r_1),\cdots,(x_N,r_N)) \quad \}. \end{aligned}$$

If the protocol is not secure, $\exists$ a PPT V, $\forall$ PPT $\hat{V}$, $\exists$ a PPT distinguisher D, $\exists$ a polynomial $p(n)$, for infinitely many n's

$$|\Pr\{D(V(view_V))=1\} - \Pr\{D(\hat{V}(view_{\hat{V}}))=1\}| \geqq \frac{1}{p(n)}.$$

Because $c=\prod_{i=1}^{N}\bar{c_i}$ is directly derived from $\{\bar{c_i}\}_{i=1,\ldots,N}$, z is derived from c', $\{\bar{c_i}\}_{i\in\Phi}$ are derived from $\{r_i\}_{i\in\Phi}$ as $(\bar{a}/\bar{b})^{r_i}$, $\{c_i\}_{i\in\Phi}$ are derived from $\{x_i\}_{i\in\Phi}$ as $c^{2\Delta x_i}$, and all ZKPs are zero knowledge, we can safely omit these terms from the inputs of V in the above inequality, i.e.,

$$\begin{aligned} |\Pr\{D(V(view_{\hat{V}}, \{E_{pk}(r_i\cdot(a-b))\}_{i\in\bar{\Phi}}, \\ \{E_{pk}(2\Delta\cdot x_i\cdot r\cdot(a-b))\}_{i\in\bar{\Phi}}, c'))=1\} \\ -\Pr\{D(\hat{V}(view_{\hat{V}}))=1\}| \geqq \frac{1}{p(n)}. \end{aligned}$$

Note that $z=r\cdot(a-b)=(\sum_{i=1}^{N}r_i)\cdot(a-b)$ is uniformly distributed provided that one r_{i^*} is uniformly distributed where $i^*\in\bar{\Phi}$. $z=L(c' \bmod n^2)\cdot(4\Delta^2)^{-1} \bmod n$ also implies that c' is uniformly distributed. Denote $\{E_{pk}(r_i\cdot(a-b))\}_{i\in\bar{\Phi}}$ and $\{E_{pk}(2\Delta\cdot x_i\cdot r\cdot(a-b))\}_{i\in\bar{\Phi}}$ in the $view_V$ as a ciphertext vector $\overline{E_{pk}(m)}$ and observe that these ciphertexts are independent because $\{r_i\}_{i\in\bar{\Phi}}$ and $\{x_i\}_{i\in\bar{\Phi}}$ are independent. For an arbitrary input set $\{x_i^{(0)}\}_{i=1,\ldots,N}$, $\{r_i^{(0)}\}_{i=1,\ldots,N}$, construct a ciphertext vector $\overline{E_{pk}(m_0)}$ as $(\{E_{pk}(r_i^{(0)}\cdot(a-b))\}_{i\in\bar{\Phi}}, \{E_{pk}(2\Delta\cdot x_i^{(0)}\cdot(\sum_{i=1}^{N}r_i^{(0)})\cdot(a-b))\}_{i\in\bar{\Phi}})$.

Because $\hat{V}$ is an arbitrary PPT algorithm, we can pick an adversary $\hat{V'}$ that takes a uniformly distributed random number $\hat{c'}$ and the above vector $\overline{E_{pk}(m_0)}$ as extra inputs. The output of $\hat{V'}$ is denoted as $\hat{V'}(view_{\hat{V}}, \overline{E_{pk}(m_0)}, \hat{c'})$. Thus, $\exists$ a PPT V, $\exists$ a PPT distinguisher D, $\exists$ a polynomial $p(n)$, for infinitely many n's

$$\begin{aligned} |\ \Pr\{D(V(view_{\hat{V}}, \overline{E_{pk}(m)}, c')=1\} - \\ \Pr\{D(\hat{V'}(view_{\hat{V}}, \overline{E_{pk}(m_0)}, \hat{c'}))=1\}\ | \geqq \frac{1}{p(n)}, \end{aligned}$$

where the first probability is over all possible m, i.e. averaged over all possible r_i and x_i. Thus, there exists a ciphertext vector $\overline{E_{pk}(m_1)}=(\{E_{pk}(r_i^{(1)}\cdot(a-b))\}_{i\in\bar{\Phi}}, \{E_{pk}(2\Delta\cdot x_i^{(1)}\cdot(\sum_{i=1}^{N}r_i^{(1)})\cdot(a-b))\}_{i\in\bar{\Phi}})$ such that

$$\begin{aligned} |\ \Pr\{D(V(view_{\hat{V}}, \overline{E_{pk}(m_1)}, c')=1\} - \\ \Pr\{D(\hat{V'}(view_{\hat{V}}, \overline{E_{pk}(m_0)}, \hat{c'}))=1\}\ | \geqq \frac{1}{p(n)}. \end{aligned}$$

Therefore, we concluded that the underlying Paillier cryptosystem is not a multi-message indistinguishable encryption and obtained the contradiction.

Robustness: We consider active adversaries who want to disrupt the protocol. In our protocol, each X_i provides ZKPs to prove that his operations are conforming to the protocol. Thus, the protocol can distinguish dishonest players from honest ones. If V inputs the incorrect $\{x_i\}_{i\in\Phi}$ into the protocol, V cannot provide the correct ZKPs described in **Step 3** of the protocol. Thus, the protocol can ignore the invalid values and go on with the valid ones in an honest majority environment.

6 Applications

In this section, we give a comprehensive example to show the applicability of the proposed plaintext equivalence test protocol in voting.

Assume the ciphertext T is the encryption of the tally of the votes for the candidate *Alice* and there are k voters together with a voting center VC. VC and the voters want to know that whether *Alice* wins majority supports but do *not* want to reveal the tally to anyone in order to eliminate possible voting strategies. VC can use the proposed protocol as following:

Step 1: First, VC prepares a $(k+1)$-by-2 conversion table S where the first column is the possible tally enumeration from the set the set $\{0, 1, 2, \cdots, k\}$ and the second column is the unary representation of the first column. It is shown as Table 1.

Table 1: The conversion table S

tally	unary representation
0	$(0, 0, \cdots, 0)$
1	$(1, 0, \cdots, 0)$
2	$(1, 1, \cdots, 0)$
.	.
.	.
$k-1$	$(1, 1, \cdots, 1, 0)$
k	$(1, 1, \cdots, 1)$

Step 2: VC encrypts all entries in S, uses a verifiable mix-net [13, 6] to shuffle the conversion table, and obtains the shuffled conversion table $\tilde{S}$ as shown in Table 2.

Table 2: The shuffled conversion table $\tilde{S}$

tally ciphertext	ciphertext vector of the unary representation
$E_{pk}(3)$	$(E_{pk}(1), E_{pk}(1), \cdots, E_{pk}(0))$
$E_{pk}(k-1)$	$E_{pk}(1), E_{pk}(1), \cdots, E_{pk}(1))$
$E_{pk}(0)$	$(E_{pk}(0), E_{pk}(0), \cdots, E_{pk}(0))$
.	.
.	.
$E_{pk}(1)$	$(E_{pk}(1), E_{pk}(0), \cdots, E_{pk}(0))$
$E_{pk}(2)$	$(E_{pk}(1), E_{pk}(1), \cdots, E_{pk}(0))$

Step 3: VC uses the proposed plaintext equivalence test protocol to find the index i such that the decryption of T and the decryption of the ciphertext of the i-th row of the first column of $\tilde{S}$ are equal.

Step 4: VC decrypts the $k/2$-th element of the i-th row of the second column of $\tilde{S}$. If it is 1, then VC declares that *Alice* wins majority supports. Otherwise, VC declares that *Alice* has only minority supports.

Through the use of the proposed plaintext equivalence test protocol and the verifiable shuffle schemes, we can keep even tally information secret in an e-voting scheme.

7 Conclusion

We propose an efficient plaintext equivalence test protocol based on the threshold Paillier cryptosystem. By using a threshold system and zero-knowledge proofs, the protocol archives its security goals and resists possible attacks from practical active adversaries. We analyze the protocol under secure multi-party protocol formulation and give a formal proof about its privacy. The protocol communicates through a practical public broadcast channel without demanding a secure channel as in [11]. It is suitable for public verifiable secure applications such as votings or auctions to determine the equality of secrets without revealing any other information.

References

[1] O. Baudron, P. A. Fouque, D. Pointcheval, G. Poupard, and J. Stern, "Practical multi-candidate election system," in *Proceedings of the ACM Conference on Principles on Distributed Computing*, pp. 274-283, Philadelphia, USA, 2001.

[2] D. Chaum and T. Pedersen, "Wallet databases with observers," in *Advances in Cryptology (Crypto'92)*, LNCS 740, pp. 89-105, 1992.

[3] I. Damgård, M. Jurik, and J. B. Nielsen, "A generalization of paillier's public-key system with applications to electronic voting," in*Proceedings of Public Key Cryptography 2001*, LNCS 1992, pp. 119-136, 2001.

[4] I. Damgård and M. Koprowski, "Practical threshold RSA signatures without a trusted dealer," in *Advances in Cryptology (Eurocrypt'01)*, LNCS 2045, pp. 152-165, 2001.

[5] T. ElGamal, "A public-key cryptosystem and signature scheme based on discrete logarithms", *IEEE Transactions on Information Theory*, vol. IT-31, pp. 469-472, 1985.

[6] J. Furukawa and K. Sako, "An efficient scheme for proving a shuffle," in *Advances in Cryptology (Crypto'01)*, LNCS 2139, pp. 368-387, 2001.

[7] O. Goldreich, *Foundations of Cryptography: Volume II Basic Applications*, Cambridge University Press, 2004.

[8] O. Goldreich, S. Micali, and A. Wigderson, "Proofs that yield nothing but their validity or all languages in NP have zero-knowledge proof systems," *Journal of the ACM*, vol. 38, no. 1, pp. 691-729, 1991.

[9] S. Goldwasser and S. Micali. "Probabilistic encryption," *Journal of Computer and System Science*, vol. 28, no. 2, pp. 270-299, 1984.

[10] M. Jakobsson and A. Juels, "Mix and match: Secure function evaluation via ciphertexts," in *Advances in Cryptology (Asiacrypt'00)*, LNCS 1976, pp. 162-177, 2000.
[11] R. Li and C. K. Wu, "Co-operative private equality test," *International Journal of Network Security*, vol. 1, no. 3, pp. 149-153, 2005.
[12] S. Micali and P. Rogaway, "Secure computation," in *Advances in Cryptology (Crypto'91)*, LNCS 576, pp 392-404. 1991.
[13] L. Nguyen, R. Safavi-Naini, and K. Kurosawa, "Verifiable shuffles: A formal model and a paillier-based efficient construction with provable security," in *Proceedings ACNS'04*, LNCS 3089, pp. 61-75, 2004.
[14] M. Ben-Or, S. Goldwasser, and A. Wigderson, "Completeness theorems for non-cryptographic fault tolerant distributed computation," in *Proceedings of 20th Annual ACM Symposium on Theory of Computing (STOC)*, pp. 1-10, 1988.
[15] P. Paillier, "Public-key cryptosystems based on composite degree residuosity classes," in *Advances in Cryptology (Eurocrypt'99)*, LNCS 1592, pp. 223-238, 1999.
[16] A. C. Yao, "Protocols for Secure Computation," in *Proceedings of the 23th FOCS*, pp. 160-164, 1982.
[17] C. P. Schnorr, "Efficient Identification and Signatures for Smart Cards," in *Advances in Cryptology (Crypto'89)*, LNCS 435, pp. 235-251, Springer-Verlag, Berlin, 1990.

Pei-yih Ting born in 1963, received his BS and MS degrees of EE in 1985 and 1987 at NTU, and PHD degrees of ECE at UCSB in 1992. He was with Telecommunication Laboratories, Chung-Hua Telecomm during the years of 1992-1996 as an associated researcher. He is currently an associate professor in the department of Computer Science and Engineering, National Taiwan Ocean University. Dr. Ting has been involved in projects of speech synthesis/recognition, secure web services, and trusted time sources. His current research interests are in the area of provable security features and applications of encryption/signature schemes as well as secure multi-party computations.

Xiao-Wei Huang was born on January 28, 1983. He received the B.S. in Applied Mathematics from National Chiayi University, Chiayi, Taiwan, ROC, in 2005. He is currently a M.S. student of the department of Computer Science, National Taiwan Ocean University, Keelung, Taiwan, ROC. His current research interests are in the area of cryptography and computation.