Shuffle for Paillier's Encryption Scheme

Takao Onodera Keisuke Tanaka *

Dept. of Mathematical and Computing Science,
Tokyo Institute of Technology
2-12-1 Ookayama, Meguro-ku, Tokyo 152-8552, Japan
{onodera0, keisuke}@is.titech.ac.jp

June 1, 2004

Abstract

In this paper, we propose a zero-knowledge proof scheme of shuffle. Unlike the previous schemes [6, 9, 4], our scheme can be used as the shuffle of the elements encrypted by Paillier's encryption scheme, which has an additive homomorphic property in the message part. The ElGamal encryption scheme, used in the previous schemes [6, 9, 4], does not have this property.

Keywords: Paillier's encryption scheme, zero-knowledge proof, shuffle

1 Introduction

Mix-nets have important applications, e.g., voting. A mix-net is a secure channel that consists of servers to shuffle a number of elements such as encrypted ballots so that each output decryption cannot be linked to any of the input encryptions. To ensure the correctness of output, it is desirable to achieve the property of universal verifiability.

Several proof systems for proving shuffle were proposed. The schemes by Furukawa and Sako [6], Groth [9] and Furukawa [4] are especially efficient ones among them. The scheme of [6] is not zero-knowledge, but do not leak any information about the permutation. The others are zero-knowledge. The proof system in [6] is a scheme for the ElGamal encryption scheme and one in [9] is for the $E_{pk}(m_1 : r_1) \times E_{pk}(m_2 : r_2) = E_{pk}(m_1 + m_2 : r_1 + r_2)$ type public-key homomorphic encryption schemes, e.g., the modified ElGamal encryption scheme of the form $(g^r, g^m y^r)$. Several proof systems of shuffle and decryption, e.g., Furukawa et al. [5] and Furukawa [4], are also proposed. These proof systems are for the ElGamal encryption scheme. Both schemes are not zero-knowledge, but do not leak any information about the permutation. In particular, the scheme of [4] is more efficient than the schemes of [5], [6], and [9] if the protocols in [6, 9] are used as the schemes proving the correctness of both shuffle and decryption by using a technique of proving the correctness of decryption.

Additive homomorphic encryption schemes have been used in many cryptographic protocols, e.g., voting protocols, threshold encryption schemes, secret sharing schemes, commitment schemes, etc. As an instance of additive homomorphic encryption schemes, we have a variant of the ElGamal encryption scheme of the form $(g^r, g^m y^r)$. However, this scheme is not considered to be efficient because no trapdoor exists to determine m given $g^m \bmod p$. In voting schemes, we cannot apply the modified ElGamal scheme except

*Supported in part by NTT Information Sharing Platform Laboratories and Grant-in-Aid for Scientific Research, Ministry of Education, Culture, Sports, Science, and Technology, 14780190.

for very small scale elections that can be searched exhaustively. Other additive homomorphic encryption schemes were also proposed [8, 1, 3, 11]. However, these schemes have some shortcomings in the aspects of size of the message space or the efficiency for decryption. In [13], Paillier proposed a new probabilistic encryption scheme which has an additive homomorphic property, which is more efficient than the previous schemes.

In this paper, we propose a proof system of shuffle for Paillier's encryption scheme. Our proposed protocol is a variant of the Furukawa scheme [4]. However, our variant cannot be derived by straightforward adaptation of the Furukawa scheme, since the Furukawa scheme enjoys the property of the uniform distribution on $g^a \bmod p$ where g is a fixed generator, p a prime, and a a random value from Z_p^*. We cannot use this property for the protocol with Paillier's encryption scheme since it employs composite moduli.

In order to make the protocol zero-knowledge, we put a restriction on the parameters of Paillier's encryption scheme. In particular, we restrict the set of randomizers space (originally Z_n^*) to the set Q_n of quadratic residues modulo n, where n is an RSA-modulus which is a product of two primes. We can show that Paillier's encryption scheme is also IND-CPA in such a case.

In order to make the protocol more efficient, we employ a pseudo-random generator proposed by Goldreich and Rosen [7] of the form $g^a \bmod n$ where g is a fixed element, n a composite, and a a random value from Z_N, where $N = 2^{|n|/2}$.

Let us compare our scheme with the previous schemes in [6, 9]. The scheme in [6] is a proof system for the stndard ElGamal encryption scheme, and that in [9] is for $E_{pk}(m_1 : r_1) \times E_{pk}(m_2 : r_2) = E_{pk}(m_1 + m_2 : r_1 + r_2)$ type encryption schemes such as the modified ElGamal encryption scheme. Our scheme is a proof system for $E_{pk}(m_1 : r_1) \times E_{pk}(m_2 : r_2) = E_{pk}(m_1 + m_2 : r_1 r_2)$ type encryption schemes. Paillier's encryption scheme, which is efficient and additive homomorphic, is this type.

Our paper is organized as follows. In Section 2, we review Paillier's encryption scheme, the permutation matrices, and the ElGamal Shuffling proposed in [6, 4]. In Section 3, we give the idea of our protocol, then, describe our protocol. In Section 4, we discuss about security of our protocol. In particular, we show that our protocol is complete, sound, and honest verifier zero-knowledge. In Section 5, we analyze the efficiency of our protocol and compare it to those of the previous schemes. We conclude in Section 6.

2 Preliminaries

2.1 Paillier's Encryption Scheme

Paillier [13] proposed a new encryption scheme based on higher residuosity classes. It generalized the previous work by Okamoto and Uchiyama [12]. In [15], Damgård and Jurik proposed the generalization of Paillier's scheme. Then in [14], they proposed the extension of Paillier's. Paillier's encryption scheme is one-way under the assumption that computing n-th residuosity classes in $Z_{n^2}^*$ is hard, and IND-CPA under a much stronger decisional assumption: given $c \in Z_{n^2}^*$, it is hard to decide if c is an n-th residue or not.

This scheme has the additively homomorphic property in the message part: Let M_{pk} and R_{pk} be a message space and a randomizer space, respectively. Denote by $E_{pk}(\cdot : \cdot)$ an encryption function, namely $E_{pk}(m : r)$ is a ciphertext of plaintext m as randomized by r. Then, for all $m_1, m_2 \in Z_n$ and all $r_1, r_2 \in Z_n^*$, $E_{pk}(m_1 : r_1) \times E_{pk}(m_2 : r_2) = E_{pk}(m_1 + m_2 : r_1 r_2)$.

Now we describe Paillier's encryption scheme. Let $n = pq$ be an RSA modulus, i.e., a product of two large primes of roughly the same size. Consider the multiplicative group $Z_{n^2}^*$. We denote by $\mathcal{E}_n$ the integer-valued function defined by

$$\begin{aligned} Z_n \times Z_n^* &\mapsto Z_{n^2}^* \\ (x, r) &\mapsto (1 + nx) r^n \bmod n^2. \end{aligned}$$

It can be shown that $\mathcal{E}_n$ is bijective, i.e., for any element $c \in Z_{n^2}^*$ there exists an unique pair $(x, r) \in Z_n \times Z_n^*$ such that $c = (1 + nx)r^n \bmod n^2$.

The security of Paillier's encryption scheme depends on the Decisional Composite Residuosity Assumption.

Assumption 1. *(The Decisional Composite Residuosity Assumption) Let $\mathcal{A}$ be any probabilistic polynomial time algorithm, and assume $\mathcal{A}$ takes n, x as input, where n is a product of two large primes of roughly the same size, and x is either a random value in $Z_{n^2}^*$ or a random n-th power in $Z_{n^2}^*$. $\mathcal{A}$ outputs a bit b. Let $p_0(\mathcal{A}, k)$ be the probability that $b = 1$ if x is a random value in $Z_{n^2}^*$, and $p_1(\mathcal{A}, k)$ the probability that $b = 1$ if x is a random n-th power. Then $|p_0(\mathcal{A}, k) - p_1(\mathcal{A}, k)|$ is negligible in k.*

Paillier's encryption scheme is described as follows.

Key Generation Let $n = pq$ and λ be the least common multiple of $p - 1$ and $q - 1$. The public key is n while the secret key is p, q (or equivalently λ).

Encryption The plaintext set is Z_n. Given a plaintext m, choose a random $r \in Z_n^*$, and let the ciphertext be $E_n(m; r) = (1 + nm)r^n \bmod n^2$.

Decryption Given a ciphertext c, if $c = E_n(m, r)$, we obtain the plaintext $m = D_{p,q}(c)$ as follows:

$$m = \frac{L(c^\lambda \bmod n^2)}{L((1+n)^\lambda \bmod n^2)} \bmod n = \frac{L(c^\lambda \bmod n^2)}{\lambda} \bmod n,$$

where L is defined as the integer $L(u) = (u - 1)/n$.

For our protocol, we put a restriction on the parameters of Paillier's encryption scheme. In particular, we restrict the set of randomizers space (originally Z_n^*) to the set Q_n of quadratic residues modulo n, where n is an RSA-modulus which is a product of two primes. We can show that Paillier's encryption scheme is also IND-CPA in such a case. Assume that m_0 and m_1 are two known messages and c the ciphertext of either m_0 or m_1 with a random number from Q_n. Let $c' = cr^n \bmod n^2$, where r is an element of Z_n^*. Then, c' is the ciphertext of m_0 if and only if $c'(1 - nm_0) \bmod n^2$ is an n-th residue. Therefore, a successful chosen-plaintext attacker could decide the composite residuosity. This implies that even if we restrict the randomizer space to Q_n, Paillier's encryption scheme is still IND-CPA.

2.2 Permutation Matrices

Our proposed protocol is a variant of the Furukawa scheme [4]. This scheme employs permutation matrices. We define a matrix (A_{ij}) to be a permutation matrix if it can be written as follows using some permutation function π.

$$A_{ij} = \begin{cases} 1 & \text{if } \pi(i) = j \\ 0 & \text{otherwise.} \end{cases}$$

We also define δ_{ij} and δ_{ijl} $(i, j, l = 1, \ldots, m)$ to be, respectively,

$$\delta_{ij} = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{otherwise} \end{cases} \quad \text{and} \quad \delta_{ijl} = \begin{cases} 1 & \text{if } i = j = l \\ 0 & \text{otherwise.} \end{cases}$$

We use a following fact about a permutation matrix. See [6] for the proof.

Fact 1. *A matrix (A_{ij}) is a permutation matrix if and only if, for all i, j and l, the following two equations hold.*

$$\sum_{h=1}^{n} A_{hi}A_{hj} = \delta_{ij} \tag{1}$$

$$\sum_{h=1}^{n} A_{hi}A_{hj}A_{hk} = \delta_{ijl}. \tag{2}$$

2.3 The ElGamal Shuffling by Furukawa and Sako [6] and Furukawa [4]

A shuffle of ciphertexts $\{E_1, \ldots, E_k\}$ is a new set of ciphertexts $\{E'_1, \ldots, E'_k\}$ so that both sets of ciphertexts have the same plaintexts. That is, there exists a permutation π such that $D(E'_i) = D(E_{\pi^{-1}(i)})$ for all i, where D is the decryption algorithm.

In [6, 4], a proof for shuffle of the ElGamal cryptosystem was proposed. The shuffling procedure for the ElGamal encryption scheme is defined as follows:
Consider the ElGamal cryptosystems, with public key (p, q, g, y) and secret key $x \in Z_q$ such that $y = g^x \bmod p$.[1] Given a set $\{E_i\} = \{(g_i, m_i)\}$ of k ciphertexts, where each $\{g_i\}$ and $\{m_i\}$ have the order q, a set $\{E'_i\} = \{(g'_i, m'_i)\}$ of shuffled ciphertexts are obtained by

$$(g'_i, m'_i) = (g^{r_i} \prod_{j=1}^{k} g_j^{A_{ji}}, y^{r_i} \prod_{j=1}^{k} m_j^{A_{ji}}) \bmod p,$$

where $\{r_i\}$ are the set of randomizers and (A_{ij}) is a permutation matrix.

3 The Idea and the Description of Our Protocol

3.1 The Idea of Our Protocol

Our protocol is a variant of the scheme proposed by Furukawa [6]. However, we cannot straightforwardly apply the same shuffling procedure as in [6, 4] to Paillier's encryption scheme. In [6, 4], the additively homomorphic property on randomizer plays an important role in the protocols. However, Paillier's encryption scheme does not have the additively homomorphic property on randomizer. To overcome this problem, we employ a generator of randomizer in order to gain a kind of the uniformly generating property on randomizer of Paillier's encryption scheme. However, there is no generators of $Z^*_{n^2}$.

To fix this problem, we consider the set Q_n of quadratic residues modulo n, where n is an RSA-modulus which is a product of safe primes. That is, we set $n = pq$ using two large primes of the equal length (e.g., 512 bits) p and q, where $p = 2p' + 1$, $q = 2q' + 1$, with p', q' themselves prime. If we let J_n denote the subgroup of elements $x \in Z^*_n$ with Jacobi symbol $(\frac{x}{n}) = 1$, then we have $Q_n \subset J_n \subset Z^*_n$. Moreover, Q_n is cyclic of order $p'q'$ and J_n is cyclic of order $2p'q'$. Generally speaking, we ensure that all group computations are done in Q_n and the corresponding exponent arithmetic in $Z_{p'q'}$. If we choose $g \in Q_n$ at random, we may assume that the g generates Q_n, since all elements in Q_n except for a negligible fraction are generators.

Our shuffling procedure is the following: For $i \in \{1, \cdots, k\}$, we compute E'_i by using $g \in Q_n$, a permutation matrix $(A_{ij})_{(i,j=1,\ldots,k)}$ and random numbers $\{A_{0i}\}_{(i=1,\ldots,k)} \in Z_N$, where $N = 2^{|n|/2}$, as follows:

$$E'_i = g^{A_{0i}n} \prod_{j=1}^{m} E_j^{A_{ji}} \bmod n^2. \tag{3}$$

[1] Assume that p and q are two primes such that $p = kq + 1$, where k is an integer, and g is an element that generates a subgroup G_q of order q in Z^*_p

Note that a randomizer A_{0i} does not affect the message of ciphertext E_i.

The reason why we choose a randomizer from not $Z_{(p-1)(q-1)/4}$ but Z_N is that the number $(p-1)(q-1)$ is not publicly known. Therefore we use the group Z_N, where N is sufficiently large, as the randomizer space. This is done to get a distribution of the elements g^r, where $g \in Z_n^*$, that is statistically close to the uniform distribution over the group generated by g. To make a difference in distribution smaller than $1/2^k$ the value $N = 2^{|n|+k}$ should be used. In [7], under the assumption that the factoring n is hard, Goldreich and Rosen showed that $g^r \bmod n$ for $r \in \{0, \ldots, 2^{|n|/2} - 1\}$ is a pseudo-random generator. Therefore $N = 2^{|n|/2}$ is a possible choice and will result in a element g^r (for $r \in Z_N$) that cannot be distinguished from a uniform element from the whole group generated by g. Note that solving the factoring problem is hard under the Decisional Composite Residuosity Assumption.

Our proof system consists of the following three proofs:

- a proof that given $\{E_i\}$ and $\{E'_i\}$, $\{E'_i\}$ can be expressed as Equation 3 using integers $\{A_{0i}\}$ and a matrix (A_{ij}) that satisfies Equation 1 in Section 2.2.
- a proof that given $\{E_i\}$ and $\{E'_i\}$, $\{E'_i\}$ can be expressed as Equation 3 using integers $\{A_{0i}\}$ and a matrix (A_{ij}) that satisfies Equation 2 in Section 2.2.
- a proof that a set $\{A_{0i}\}$ of integers and a matrix (A_{ij}) used in the above two proofs are identical.

For the first proof, the main idea is to issue $a_s = \sum_{j=1}^{k} A_{sj} c_j$ $(s = 0, \ldots, k)$ as a response to a set $\{c_i\}_{(i=1,\ldots,k)}$ of challenges and let the verifier check:

$$\sum_{j=1}^{k} a_j^2 = \sum_{j=1}^{k} c_j^2$$

$$g^{a_0 n} \prod_{i=1}^{k} E_i^{a_i} = \prod_{i=1}^{k} E_i'^{c_i} \bmod n^2.$$

However, this apparently leaks information on (A_{ij}), so we need to add randomizers and commitments, i.e., let $a_i = A_{i0} + \sum_{j=1}^{k} A_{ij} c_j$ using randomizers A_{i0}. The idea for the second proof is similar to this. A combination of the first and second proofs implies the third proof.

The prover and the verifier have strings as a common input. Since $\{E'_i\}$ are originally chosen by those who encrypted the messages, there is no control to assure that the prover does not know the relations among them. Therefore, we use random values $\tilde{E}_{-3}, \ldots, \tilde{E}_k$ as a common input.

3.2 The Description

Common Input : ciphertexts $E_1, \ldots, E_k, E'_1, \ldots, E'_k \in Z_{n^2}^*$, $g \in Q_n$, $n = pq$, where p and q are safe primes, $N = 2^{|n|/2}$, and $\tilde{E}_s \in_R Q_n$ $(s = -3, \ldots, k)$.
Prover's Input : A permutation matrix $(A_{ij})_{(i,j=1,\ldots,k)}$ and a set $\{A_{0i}\}_{(i=1,\ldots,k)} \in Z_N$ of randomizers such that $E'_i = g^{A_{0i} n} \prod_{j=1}^{k} E_j^{A_{ji}} \bmod n^2$ $(i = 1, \ldots, k)$.

Commitment: Let $S \triangleq \{-3, \ldots, k\}$. For $i = 1, \ldots, k$, the prover P chooses $\{A_{i0}\}$ from $\{0, \ldots, 2^{2k_2} - 1\}$, $A_{-30}, \ldots, A_{00}$, u_0, and $\tilde{u}_0$ from $\{0, \ldots, 2^{|N|+5k_2+|k|+2} - 1\}$, where k_2 is a secondary security parameter (e.g. 160 bits), and $\{\tilde{u}_i\}$ and $\{A_{-1i}\}$ from Z_N, randomly. Then computes the following:

$$A_{-2i} = \sum_{j=1}^{k} 3A_{j0}^2 A_{ji} \quad i = 1, \ldots, k$$

$$
\begin{aligned}
A_{-3i} &= \sum_{j=1}^{k} 2A_{j0}A_{ji} \quad i = 1, \ldots, k \\
g'_l &= g^{A_{-1l}} \prod_{j \in S \setminus \{-1\}} \tilde{E}_j^{A_{jl}} \bmod n \quad l = 0, \ldots, k && (4) \\
E'_0 &= g^{A_{00}n} \prod_{j=1}^{k} E_j^{A_{j0}} \bmod n^2 \\
u_i &= \sum_{j=1}^{k} 3A_{j0}A_{ji} \quad i = 1, \ldots, k \\
t_l &= \tilde{E}_0{}^{u_l} g^{\tilde{u}_l} \bmod n \quad l = 0, \ldots, k && (5) \\
w &= \sum_{j=1}^{k} A_{j0}^3 - A_{-20} - u_0 \\
\tilde{w} &= \sum_{j=1}^{k} A_{j0}^2 - A_{-30}.
\end{aligned}
$$

Then P sends $g'_0, \ldots, g'_k$, E'_0, $t_0, \ldots, t_k$, w, $\tilde{w}$ to the verifier V.
Challenge: V chooses $c_1, \ldots, c_k$ at random from $\{0, \ldots, 2^{k_2} - 1\}$ and sends them to P.
Answer: P computes the following:

$$
\begin{aligned}
a_s &= A_{s0} + \sum_{j=1}^{k} A_{sj}c_j \quad s = -3, \ldots, k \\
\tilde{u} &= \tilde{u}_0 + \sum_{j=1}^{k} \tilde{u}_j c_j^2 \\
u &= u_0 + \sum_{j=1}^{k} u_j c_j^2.
\end{aligned}
$$

Then P sends $a_{-3}, \ldots, a_k$, $\tilde{u}$, u to V.
Verification: V accepts the proof if the following equations hold:

$$
g^{a_{-1}} \prod_{i \in S \setminus \{-1\}} \tilde{E}_i^{a_i} \equiv g'_0 \prod_{i=1}^{k} g_i'^{c_i} \pmod{n} \tag{6}
$$

$$
g^{a_0 n} \prod_{i=1}^{k} E_i^{a_i} \equiv E'_0 \prod_{i=1}^{k} E_i'^{c_i} \pmod{n^2} \tag{7}
$$

$$
\tilde{E}_0^u g^{\tilde{u}} \equiv t_0 \prod_{i=1}^{k} t_i^{c_i^2} \pmod{n} \tag{8}
$$

$$
\sum_{i=1}^{k} (a_i^2 - c_i^2) = a_{-3} + \tilde{w} \tag{9}
$$

$$
\sum_{i=1}^{k} (a_i^3 - c_i^3) = a_{-2} + u + w. \tag{10}
$$

4 Correctness and Security

The protocol is a public-coin 3-move proof system of shuffle. It is complete, sound and special honest verifier zero-knowledge.

Theorem 2. *The protocol is complete, that is, if the prover knows $\{A_{0i}\}_{(i=1,\ldots,k)}$ and $(A_{ij})_{(i,j=1,\ldots,k)}$ satisfying Equations 1, 2 and 3, the verifier always accepts.*

Proof: It can be proved by straightforward calculation.

□

We will prove that our proposed protocol is sound and special honest verifier zero-knowledge under the Decisional Composite Residuosity Assumption. Note that Paillier's encryption scheme is IND-CPA under the Decisional Composite Residuosity Assumption. If the cryptosystem is IND-CPA, it is not possible for somebody else to see which permutation used in the shuffle. Therefore a cryptosystem for shuffle has to be IND-CPA.

For soundness, we will claim that if the verifier accepts the protocol, then either the prover knows the permutation and the set of randomizers or he knows a set a and $\{a_i\}_{(i=1,\ldots,k)}$ of integers satisfying $g^a \prod_{i=1}^{k} g_i^{a_i} \equiv 1 \pmod{n}$, where g, $\{g_i\} \in Q_n$, with overwhelming probability.

Theorem 3. *Assume the difficulty of computing discrete logarithms modulo composite numbers. There exists no polynomial time algorithms which take as input a composite number $n = pq$ (where p, q are primes) and g, $\{g_i\} \in Q_n$, and can compute a non-trivial set $\{a, a_1, \ldots, a_k\}$ satisfying $g^a \prod_{i=1}^{k} g_i^{a_i} \equiv 1 \pmod{n}$.*

This theorem can be proved analogously to [2].

We have the following proposition.

Proposition 4. *Let n be a composite number. If the discrete logarithm problem in Z_n^* can be solved in polynomial time, then n can be factored in expected polynomial time.*

In other words, the discrete logarithm problem in Z_n^* is at least as difficult as the problem of factoring n. See [10] for detail. The following proposition was proved in [13].

Proposition 5. *Let n be a composite number. If there exists a polynomial time algorithm which can solve the factoring problem, then there exists an algorithm which can solve the decisional composite residuosity problem in polynomial time.*

From Theorem 3, Propositions 4 and 5, we can get the following theorem.

Theorem 6. *Assume the Decisional Composite Residuosity Assumption. There exists no polynomial time algorithms which take as input a composite number $n = pq$ (where p, q are safe primes) and g, $\{g_i\} \in Q_n$, and can compute a non-trivial set $\{a, a_1, \ldots, a_k\}$ satisfying $g^a \prod_{i=1}^{k} g_i^{a_i} \equiv 1 \pmod{n}$.*

For soundness, we show several lemmas.

Lemma 7. *If the verifier accepts the protocol with probability p, then there is a polynomial time knowledge extractor K that can extract from the prover $\{A_{ij}\}_{(i=-3,\ldots,k,j=0,\ldots,k)}$ satisfying Equation 4 and $\{u_i, \tilde{u}_i\}_{(i=0,\ldots,k)}$ satisfying Equation 5 within an expected number of steps bounded by $\mathcal{O}(1/p)$.*

Proof: Define $\mathcal{C}_p$ as the space which is spanned by a set of vectors on the form $(1, c_1, \ldots, c_k)$ made up of the challenges to which the prover can compute responses $\{a_j\}_{(j=-3,\ldots,k)}$, u, and $\tilde{u}$ such that Equations 6 and 8 hold with the probability p. If $\dim(\mathcal{C}_p) = k+1$, K can choose $\mathcal{O}((k+1)/p)$ vectors of challenges that are uniformly distributed in $\mathcal{C}_p$ and feed them to the prover. Then K can extract from the prover $k+1$ responses that correspond to $(k+1)$ linearly independent vectors of challenges, from which K is able to extract $\{A_{ij}\}_{(i=-3,\ldots,k,j=0,\ldots,k)}$, $\{u_i\}_{(i=0,\ldots,k)}$, and $\{\tilde{u}_i\}_{(i=0,\ldots,k)}$ that satisfy the relation:

$$a_i = A_{i0} + \sum_{j=1}^{k} A_{ij} c_j \qquad i = -3, \ldots, k$$
$$u = u_0 + \sum_{i=1}^{k} u_i c_i^2$$
$$\tilde{u} = \tilde{u}_0 + \sum_{i=1}^{k} \tilde{u}_i c_i^2$$

for each vector of challenges and response. Such $\{A_{ij}\}_{(i=-3,\ldots,k,j=0,\ldots,k)}$, $\{u_i\}_{(i=0,\ldots,k)}$, and $\{\tilde{u}_i\}_{(i=0,\ldots,k)}$ satisfy Equations 4 and 5 with overwhelming probability.

□

Lemma 8. *Assume that there is a polynomial time knowledge extractor K that can extract from the prover $\{A_{ij}\}_{(i=-3,\ldots,k,j=0,\ldots,k)}$ satisfying Equation 4. If there is also a polynomial time knowledge extractor K' that can extract from the prover $\{a_i\}_{(i=-3,\ldots,k)}$ satisfying Equation 6, and $a_i \neq A_{i0} + \sum_{j=1}^{k} A_{ij} c_j$ for some $i \in \{-3, \ldots, k\}$, then K' can compute a set of non-trivial integers $\{s_i\}_{(i=-3,\ldots,k)}$ satisfying $g^{s_{-1}} \prod_{j \in S \setminus \{-1\}} \tilde{E}_j^{s_j} \equiv 1 \pmod{n}$ with overwhelming probability.*

Proof: From the following equation, we can compute a non-trivial set $\{s_i\}_{(i=-3,\ldots,k)}$:

$$g^{a_{-1} - (A_{-10} + \sum_{i=1}^{k} A_{-1i} c_i)} \prod_{j \in S \setminus \{-1\}} \tilde{E}_j^{a_j - (A_{j0} + \sum_{i=1}^{k} A_{ji} c_i)} \equiv 1 \pmod{n}.$$

□

Lemma 9. *Assume that there is a polynomial time knowledge extractor K that can extract from the prover $\{u_i, \tilde{u}_i\}_{(i=0,\ldots,k)}$ satisfying Equation 5. If there is also a polynomial time knowledge extractor K' that can extract from the prover $\{u, \tilde{u}\}$ satisfying Equation 8, and $u \neq u_0 + \sum_{i=1}^{k} u_i c_i^2$ or $\tilde{u} \neq \tilde{u}_0 + \sum_{i=1}^{k} \tilde{u}_i c_i^2$, then K' can compute a set of non-trivial integers $\{s, s'\}$ satisfying $\tilde{E}_0^s g^{s'} \equiv 1 \pmod{n}$ with overwhelming probability.*

Proof: From the following equation, we can compute a non-trivial set $\{s, s'\}$:

$$\tilde{E}_0^{u - (u_0 + \sum_{i=1}^{k} u_i c_i^2)} g^{\tilde{u} - (\tilde{u}_0 + \sum_{i=1}^{k} \tilde{u}_i c_i^2)} \equiv 1 \pmod{n}.$$

□

Lemma 10. *Assume that $\{A_{ij}\}_{(i=-3,\ldots,k,j=0,\ldots,k)}$ and $\tilde{w}$ are given. We also assume that we can generate $\{a_i\}_{(i=-3,\ldots,k)}$ satisfying $a_i = A_{i0} + \sum_{j=1}^{k} A_{ij} c_j$ for a set $\{c_i\}_{(i=1,\ldots,k)}$ of challenges. If the given set $\{A_{ij}\}_{(i=-3,\ldots,k,j=0,\ldots,k)}$ do not satisfy Equation 1, then Equation 9 will not hold with overwhelming probability.*

Proof: If Equation 9 holds, then

$$\sum_{i=1}^{k}\sum_{j=1}^{k}(\sum_{h=1}^{k} A_{hi}A_{hj} - \delta_{ij})c_i c_j + \sum_{i=1}^{k}(\sum_{j=1}^{k} 2A_{j0}A_{ji} - A_{-3i})c_i + \sum_{i=1}^{k} A_{i0}^2 - A_{-30} - \tilde{w} = 0.$$

The above equation must be satisfied with any choices of $c_1, \ldots, c_k$. Therefore, $\sum_{h=1}^{k} A_{hi}A_{hj} = \delta_{ij}$ with overwhelming probability.

□

Lemma 11. *Assume that $\{A_{ij}\}_{(i=-3,\ldots,k,j=0,\ldots,k)}$, $\{u_h\}_{(h=0,\ldots,k)}$, and w are given. We also assume that we can generate $\{a_i\}_{(i=-3,\ldots,k)}$ and u satisfying $a_i = A_{i0} + \sum_{j=1}^{k} A_{ij}c_j$ and $u = u_0 + \sum_{i=1}^{k} u_i c_i^2$ for a set $\{c_i\}_{(i=1,\ldots,k)}$ of challenges. If the given set $\{A_{ij}\}_{(i=-3,\ldots,k,j=0,\ldots,k)}$ does not satisfy Equation 2, then Equation 10 will not hold with overwhelming probability.*

Proof: If Equation 10 holds, then

$$\sum_{i=1}^{k}\sum_{j=1}^{k}\sum_{l=1}^{k}(\sum_{h=1}^{k} A_{hi}A_{hj}A_{hl} - \delta_{ijl})c_i c_j c_l + \sum_{i=1}^{k}\sum_{j=1}^{k}(\sum_{h=1}^{k} 3A_{h0}A_{hi}A_{hj} - u_i\delta_{ij})c_i c_j + \sum_{i=1}^{k}(\sum_{h=1}^{k} 3A_{h0}^2 A_{hi} - A_{-2i})c_i + \sum_{i=1}^{k} A_{i0}^3 - A_{-20} - w - u_0 = 0.$$

The above equation must be satisfied with any choices of $c_1, \ldots, c_k$. Therefore, $\sum_{h=1}^{k} A_{hi}A_{hj}A_{hl} = \delta_{ijl}$ with overwhelming probability.

□

Lemma 12. *Assume that $\{E_i\}_{(i=0,\ldots k)}$ and $\{A_{ij}\}_{(i,j=0,\ldots,k)}$ are given. We also assume that we can generate $\{a_i\}_{(i=0,\ldots,k)}$ satisfying $a_i = A_{i0} + \sum_{j=1}^{k} A_{ij}c_j$ for a set $\{c_i\}_{(i=1,\ldots,k)}$ of challenges. If Equation 3 does not hold, then Equation 7 does not hold with overwhelming probability.*

Proof: From the following equation, if Equation 3 does not hold, then Equation 7 does not hold with overwhelming probability:

$$\frac{g^{A_{00}n}\prod_{j=1}^{k} E_j^{A_{j0}}}{E'_0}\prod_{i=1}^{k}(\frac{g^{A_{0i}n}\prod_{j=1}^{k} E_j^{A_{ji}}}{E'_i})^{c_i} \equiv 1 \pmod{n^2}.$$

The above equation must be satisfied with any choices of $c_1, \ldots, c_k$. □

Theorem 13. *The protocol is sound, that is, if the verifier accepts with non-negligible probability, then the prover either knows both $\{A_{0i}\}_{(i=1,\ldots,k)}$ and $(A_{ij})_{(i,j=1,\ldots,k)}$ satisfying Equations 1, (2) and (3) or can break the Decisional Composite Residuosity Assumption.*

Proof: From Lemma 7, if the verifier accepts with non-negligible probability, then there exists a knowledge extractor K which can extract from the prover $\{A_{ij}\}_{(i=-3,\ldots,k,j=0,\ldots,k)}$, $\{u_i, \tilde{u}_i\}_{(i=0,\ldots,k)}$ satisfying Equations 4 and 5. Therefore, from Lemmas 8 and 9, K can either, (1) only generate a set $\{a_i\}_{(i=-3,\dots,k)}$, u satisfying:

$$a_i = A_{i0} + \sum_{j=1}^{k} A_{ij} c_j$$
$$u = u_0 + \sum_{i=1}^{k} u_i c_i^2$$

for a given set of challenges, or (2) generate a non-trivial set $\{s_i\}_{(i=-3,\dots,k)}$ satisfying $g^{s_{-1}} \prod_{j \in S \setminus \{-1\}} \tilde{E}_j^{s_j} \equiv 1 \pmod{n}$, or (3) a non-trivial set $\{s, s'\}$ satisfying $\tilde{E}_0^s g^{s'} \equiv 1 \pmod{n}$ with overwhelming probability. From Theorem 6, (2) and (3) will happen with negligible probability. From Lemmas 10 and 11, $\{A_{ij}\}_{(i,j=1,\dots,k)}$ by the extractor satisfy Equations 1 and 2. Therefore, this set $(A_{ij})_{(i,j=1,\dots,k)}$ constitute a permutation matrix. From Lemma 12, those $\{A_{ij}\}_{(i=0,\dots,k,j=1,\dots,k)}$ satisfy Equation 3 with overwhelming probability.
□

The special honest verifier zero-knowledge property of the proposed protocol is based on the following two fact. One is that g generates Q_n when n is a product of safe primes. The other is at the output of the function $f_{n,g}(x) = g^x \bmod n$ (where n is a composite number) is pseudo-random, even when the size of the input is restricted to be half of that of n if the factoring problem is difficult to solve.

Theorem 14. *The protocol is special honest verifier zero-knowledge, that is, there is a procedure that can take any challenge $\{c_i\}_{(i=1,\dots,k)}$ and produce a conversation indistinguishable from the space of all conversations between the honest prover and the verifier.*

Proof:
We first describe the simulator. Then, we claim that its output is indistinguishable from a real proof.
The simulation
We are given g, n, N, $E_1, \dots, E_k$, $E'_1, \dots, E'_k$, $\tilde{E}_{-3}, \dots, \tilde{E}_k$, $c_1, \dots, c_k$ as input, and wish to produce a simulated proof that is indistinguishable from a real proof.

We choose $a_1, \dots, a_k$ at random from $\{0, \dots, 2^{2k_2} - 1\}$, $a_{-3}, \dots, a_0$, u, and $\tilde{u}$ at random from $\{0, \dots, 2^{|N|+5k_2+|k|+2} - 1\}$, and $g'_1, \dots, g'_k$ and $t_1, \dots, t_k$ at random from Q_n. Then we set the other elements as follows:

$$g'_0 \equiv g^{a_{-1}} \prod_{i \in S \setminus \{-1\}} \tilde{E}_i^{a_i} \times \prod_{i=1}^{k} g_i'^{-c_i} \pmod{n}$$
$$E'_0 \equiv g^{a_0 n} \prod_{i=1}^{k} E_i^{a_i} \times \prod_{i=1}^{k} E_i'^{-c_i} \pmod{n^2}$$
$$t_0 \equiv \tilde{E}_0^u g^{\tilde{u}} \times \prod_{i=1}^{k} t_i^{-c_i^2} \pmod{n}$$
$$\tilde{w} = \sum_{i=1}^{k} (a_i^2 - c_i^2) - a_{-3}$$
$$w = \sum_{i=1}^{k} (a_i^3 - c_i^3) - a_{-2} - u.$$

Proof that the simulation works

In order to show that a simulated proof is indistinguishable from a real proof, we define the following sequence of experiments:

Exp$_1$: We carry out a real proof with challenges $c_1, \ldots, c_k$.

Exp$_2$: First we pick $a_1, \ldots, a_k$ at random from $\{0, \ldots, 2^{2k_2} - 1\}$, $a_{-3}, \ldots, a_0$, u, and $\tilde{u}$ at random from $\{0, \ldots, 2^{|N|+5k_2+|k|+2} - 1\}$. Then we carry out a real proof using these values.

Exp$_3$: We carry out Exp$_2$. We pick $g'_1, \ldots, g'_k$ and $t_1, \ldots, t_k$ at random from Q_n. Then we carry out a real proof using these values.

Exp$_4$: We carry out Exp$_3$. Then we set g'_0, E'_0, t_0, $\tilde{w}$, and w as we do in the simulation.

Exp$_5$: We make a simulated proof.

Exp$_1$ and Exp$_2$ are statistically indistinguishable, i.e., the statistical distance between the probability distribution on real proofs and that on simulated proofs is at most $1/2^{k_2}$ or $1/2^{|N|}$, since $\sum_{j=1}^{k} A_{ij}c_j$ is k_2-bit for $i = 1, \ldots, k$, when (A_{ij}) is a permutation matrix. Note that the length of the random numbers should be large enough, otherwise there exists a polynomial time algorithm which can distinguish a real proof from the output of the simulator.

Exp$_2$ and Exp$_3$ are computationally indistinguishable since the output of the function $f_{n,g}(x) = g^x \bmod n$ is pseudo-random and we can assume that g generates Q_n.

Exp$_3$ and Exp$_4$ are perfectly indistinguishable when g generates Q_n.

Exp$_4$ and Exp$_5$ are perfectly indistinguishable.

The above relations imply that Exp$_1$ and Exp$_5$ are indistinguishable, i.e., real proofs and simulated proofs are indistinguishable.

□

5 Efficiency

In this section, we analyze the efficiency of our protocol, and compare it to those of the protocols of Furukawa and Sako [6], Groth [9] and Furukawa [4].

For the round complexity, our scheme is 3-move, so are the schemes of [6] and [4]. However, the scheme of [9] is 7-move.

For the computation complexity, we estimate the number of exponentiations used in each protocol when the number of shuffled ciphertexts is k, shown in Table 1. We ignore the constant factors. For fairness, we use s as the security parameter of the cryptosystems in Table 1, i.e. we assume the security parameter of the ElGamal encryption scheme to be $s = |p|$, and one of Paillier's encryption scheme to be $s = |n|$.

	size of modulus	Furukawa and Sako [6]	Groth [9]	Furukawa [4]	Ours
Prover	s	$8k$	$6k$	$8k$	$6k$
	$2s$	–	–	–	k
Verifier	s	$10k$	$6k$	$7k$	$3k$
	$2s$	–	–	–	$2k$

Table 1: The numbers of exponentiations in each protocol

For the communication complexity, we estimate the communicated bits which the prover sends in each protocol. We consider the ElGamal encryption scheme based on a 1024-bit prime p with the operations taking place in a group of order q, where q is a 160-bit prime so that $q|p-1$, in [6], [9], and [4], Paillier's encryption scheme based on a 1024-bit RSA modulus n in our protocol. We set $k_2 = 160$ and $|N| = 512$ in our protocol. The proofs of [6], [9], [4], and ours require the prover send roughly $5280k$ bits, $1184k$ bits, $1184k$ bits and, $2413k$ bits, respectively.

6 Concluding Remarks

In this paper, we have proposed a zero-knowledge proof scheme of shuffle. Unlike the previous schemes [6, 9, 4], our scheme can be used as the shuffle of the elements encrypted by Paillier's encryption scheme, which has an additive homomorphic property. The ElGamal encryption scheme, used in the previous schemes [6, 9, 4], does not have this property. In order to make the protocol zero-knowledge, we have put a restriction on the parameters of Paillier's encryption scheme. In particular, we have restricted the set of randomizers space (originally Z_n^*) to the set Q_n of quadratic residues modulo n, where n is an RSA-modulus which is a product of two primes. We can show that Paillier's encryption scheme is also IND-CPA in such a case. In order to make the protocol more efficient, we have employed a pseudo-random generator proposed by Goldreich and Rosen [7] of the form $g^a \bmod n$ where g is a fixed element, n a composite, and a a random value from Z_N, where $N = 2^{|n|/2}$.

References

[1] J. D. C. Benaloh. Verifiable Secret-Ballot Elections. PhD thesis, Yale University, 1987.

[2] S. Brands. An Efficient Off-line Electronic Cash System Based On The Representation Problem. CWI Technical Report CS-R9323, 1993.

[3] J. D. Cohen and M. Fischer. A Robust and Verifiable Cryptographically Secure Election Scheme. In *the 26th FOCS*. IEEE, 1985.

[4] J. Furukawa. Efficient, Verifiable Shuffle Decryption and Its Requirement of Unlinkability. In Robert Deng Feng Bao and Jianying Zhou, editors, *Public Key Cryptography - PKC 2004*, volume 2947 of *Lecture Notes in Conputer Science*, pages 319–332, Singapore, March 2004. Springer-Verlag.

[5] J. Furukawa, H. Miyauchi, K. Mori, S. Obana, and K. Sako. An Implementation of a Universally Verifiable Electronic Voting Scheme Based on Shuffling. In Matt Blaze, editor, *Financial Cryptography 2002*, volume 2357 of *Lecture Notes in Conputer Science*, pages 16–30, Southampton, Bermuda, March 2002. Springer-Verlag.

[6] J. Furukawa and K. Sako. An Efficient Scheme for Proving a Shuffle. In Joe Kilian, editor, *Advances in Cryptology – CRYPTO 2001*, volume 2139 of *Lecture Notes in Conputer Science*, pages 368–387, Santa Barbara, California, USA, August 2001. Springer-Verlag.

[7] O. Goldreich and V. Rosen. On the Security of Modular Exponentiation with Application to the Construction of Pseudorandom Generators. Cryptology ePrint Archive, record 2000/064. `http://eprint.iacr.org/`, December 2000.

[8] S. Goldwasser and S. Micali. Probabilistic Encryption. JCSS, 28, 2, 270-299, 1984.

[9] J. Groth. A Verifiable Secret Shuffle of Homomorphic Encryptions. In Yvo Desmedt, editor, *Public Key Cryptography - PKC 2003*, volume 2567 of *Lecture Notes in Conputer Science*, pages 145–160, Miami, Florida, USA, January 2003. Springer-Verlag.

[10] A. Manezes, C. van Oorschot, and S. Vanstone. Handbook of Applied Cryptography. page 114. CRC press.

[11] D. Naccache and J. Stern. A New Public-Key Cryptosystem Based on higher Residues. In *5th Symposium on Computer and Communications Security*, ACM, 1998.

[12] T. Okamoto and S. Uchiyama. A New Public Key Cryptosystem as Secure as Factoring. In Kaisa Nyberg, editor, *Advances in Cryptology – EUROCRYPT '98*, volume 1403 of *Lecture Notes in Conputer Science*, pages 308–318, Espoo, Finland, May 1998. Springer-Verlag.

[13] P. Paillier. Public-Key Cryptosystems Based on Composite Degree Residue Classes. In Jacques Stern, editor, *Advances in Cryptology – EUROCRYPT '99*, volume 1592 of *Lecture Notes in Conputer Science*, pages 223–238, Prague, Czech Republic, May 1999. Springer-Verlag.

[14] I. Damgård and M. Jurik. A Length-Flexible Threshold Cryptosystem with Applications. pages 350–564. the 8th Sustralasian Conference on Information Security and Privacy, Springer-Verlag.

[15] I. Dangård and M. Jurik. A Generalisation, a Simplification and Some Applications of Paillier's Probabilistic Public-Key System. In Kwangjo Kim, editor, *Public Key Cryptography*, volume 1992 of *Lecture Notes in Conputer Science*, pages 119–136, Cheju Island, Korea, February 2001. Springer-Verlag.