A Verifiable Secret Shuffle of the 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

January 26, 2004

Abstract

In this paper, we propose a zero-knowledge proof scheme of shuffle. Unlike the previous schemes [3, 6], our scheme can be used as the shuffle of the elements that are encrypted from Paillier's cryptosystem. The Paillier's encryption scheme has an additive homomorphic property. The ElGamal cryptosystem, used in the previous works [3, 6], does not have this property.

Keywords: Paillier's encryption, Zero-knowledge proof, Shuffle

1 Introduction

Mix-nets have important applications in the situation which require anonymity, 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 linked to any of the input encryptions.

Several zero-knowledge proof systems of shuffle were proposed. The schemes by Furukawa and Sako [3] and Groth [6] are especially efficient ones among them. The proof system in [3] is a public-coin honest verifier zero-knowledge scheme[1] for the ElGamal encryption and one in [6] is for the

*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.

[1]We say that a protocol is a public coin protocol when the proof of knowledge has a special form in that the verifier only sends uniformly chosen bits, and that a protocol has the honest verifier zero-knowledge property when the protocol does not reveal anything to the honest verifier, but it is not necessarily secure against a cheating verifier.

$E_{pk}(m_1 : r_1) \times E_{pk}(m_2 : r_2) = E_{pk}(m_1 + m_2 : r_1 + r_2)$ type public-key homomorphic encryptions, e.g., the modified ElGamal encryption of the form $(g^r, g^m y^r)$ instead of (g^r, my^r).

Additive homomorphic cryptosystems have been used in many cryptographic protocols, e.g., voting protocols, threshold cryptosystems, 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 cannot be considered to be efficient because no trapdoor exists to determine m given g^m mod p. In voting schemes, we cannot apply the modified ElGamal scheme except for very small scale elections that can be searched exhaustively. Other additive homomorphic encryption schemes were also proposed [5, 1, 2, 7]. However, these schemes have some shortcomings in the aspects of size of the message space or the efficiency for decryption.

In [9], Paillier[2] proposed a new probabilistic encryption scheme which has an additive homomorphic property by modifying the Okamoto-Uchiyama scheme [8]. The Paillier's encryption system is more efficient than the previous schemes. For this reason, we propose a proof system of shuffle for the Paillier's cryptosystem.

Let us compare our scheme with the previous proof schemes, [3, 6]. All of these are public-coin honest verifier zero-knowledge protocol. The scheme in [3] is a proof system for the standard ElGamal encryptions, and that in [6] 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 encryptions such as the modified ElGamal encryption. Then currently there is no 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 encryptions. However, the Paillier's encryption scheme, which is efficient and (additive) homomorphic scheme, is this type.

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

Next, we describe the communication and computation complexity. For the sake of comparison we consider the ElGamal cryptosystem 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 the case of [3] and [6], and the Paillier's cryptosystem based on a 1024-bit RSA modulus n in the case of our protocol. The prover uses about $8m$ exponentiations to make a proof, the verifier uses roughly $6m$ exponentiations to verify a proof, and the proofs themselves are of size about $6656m$ bits, to prove the correctness of an m-data shuffle. The protocol of [3] requires roughly $8m$ and $10m$ exponentiations for the prover and the verifier, respectively, and that of [6] requires roughly $6m$ and $6m$ exponentiations. Note that this may not be fair since the size of an ElGamal encryption is 1024-bit and one of Paillier's

[2] Afterwards,in [11],Damgård and Jurik proposed the generalization of the Paillier's scheme. Then in [10], they proposed the extension of the Paillier's.

encryption is 2048-bit. The proofs of [3] and [6] require the prover send roughly $5280m$ bits and $1184m$ bits, respectively.

2 Preparation

2.1 Paillier's Public-Key system

Paillier [9] proposed a new encryption scheme based on higher residuosity classes. It generalized the previous work by Okamoto and Uchiyama [8]. The Paillier's scheme was proven to be one-way under the assumption that computing n-residuosity classes in $Z^*_{n^2}$ is hard. Similarly the scheme can be proven to be semantically secure under a much stronger decisional assumption: given $c \in Z^*_{n^2}$, it is hard to decide if c is an N-residue or not.

This scheme has a homomorphic property: Let M_{pk} and R_{pk} be a message space and a randomizer space respectively. And denote by $E_{pk}(\cdot;\cdot)$ an encryption function, namely $E(m;r)$ 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 as follows. Refer to [9] for details and proofs which are not given here.

Let $n = pq$ be an RSA modulus, i.e., product of two large primes of roughly the same size. Consider the multiplicative group $Z^*_{n^2}$.

Denote by $\mathcal{E}_n$ the integer-valued function defined by

$$\mathcal{E}_n : Z_n \times Z^*_n \to Z^*_{n^2}$$
$$\mathcal{E}_n(x, r) = (1 + nx)r^n \bmod n^2$$

It can be shown that $\mathcal{E}_n$ is bijective, that is to say, 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 the Paillier's encryption system depends on the Decisional Composite Residuosity Assumption(DCRA) introduced by Paillier. Therefore we assume the following:

Assumption 1. *(The Decisional Composite Residuosity Assumption) Let $\mathcal{A}$ be any probabilistic polynomial time algorithm, and assume $\mathcal{A}$ gets n,x as input. Here $n = pq$ is an admissible RSA modulus of length k bits, and x is either random in $Z^*_{n^2}$ or it is 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 random 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.*

The Paillier's encryption scheme is as follows.

Key Generation Set $n = pq$. Let λ be the least common multiple of $p-1$ and $q-1$. Now 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 get the plaintext $m = D_{p,q}(c)$ as follows:

$$m = \frac{L(c^\lambda \bmod n^2)}{L(g^\lambda \bmod n^2)} \bmod n.$$

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

This scheme was proven to be both one-way and semantically secure[3] under the Assumption 1 [9].

2.2 Shuffling

We define a matrix (A_{ij}) to be a permutation matrix if it can be written as follows using some permutation function π.

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

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

$$\delta_{ij} = \begin{cases} 1 \bmod n & \text{if } i = j \\ 0 \bmod n & \text{otherwise} \end{cases} \quad \text{and} \quad \delta_{ijk} = \begin{cases} 1 \bmod n & \text{if } i = j = k \\ 0 \bmod n & \text{otherwise} \end{cases}$$

There is a following fact about a permutation matrix. Refer to [3] for the proof.

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

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

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

A shuffle of ciphertexts $E_1, \ldots, E_m$ is a new set of ciphertexts $E'_1, \ldots, E'_m$. We describe a shuffle $\{E'_i\}$ $(i = 1, \ldots, m)$ of ciphertexts $\{E_i\} \in Z^*_{n^2}$ by using random numbers $\{r'_i\} \in Z_N$ where $N = 2^{|n|/2}$ as follows:

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

[3]Semantically secure means that the system is IND-CPA.

2.3 Composite-DDH problem

We assume the following:

Assumption 2. *(composite-DDH assumption (C-DDH$_2^2$ assumption)) Let $\mathcal{A}$ be any probabilistic polynomial time algorithm, and assume $\mathcal{A}$ gets (n,g,$g^a \bmod n$,$g^b \bmod n$,y) as input. Here $n = pq$ is an admissible RSA modulus of length k bits, g is a element of Z_n^*. The values a , b and c are chosen uniformly random in Z_N where $N = 2^{|n|/2}$. The value y satisfies either $g^{ab} \bmod n$ or $g^c \bmod n$. $\mathcal{A}$ outputs a bit b. Let $p_0(\mathcal{A}, k)$ be the probability that $b = 1$ if $y = g^{ab} \bmod n$ and $p_1(\mathcal{A}, k)$ the probability that $b = 1$ if $y = g^c \bmod n$. Then $|p_0(\mathcal{A}, k) - p_1(\mathcal{A}, k)|$ is negligible in k.*

The reason why we choose a randomizer from not $Z_{\phi(n)/4}$ but Z_N where $N = 2^{|n|/2}$ is that the number $\phi(n)$ is not publicly known. Therefore we use the group Z_N where N is a sufficiently large value as a randomizer space. This is done to get a distribution of the elements g^r that is statistically close to the uniform distribution over the group generated by g. To get a difference in distribution smaller than $1/2^k$ the value $N = 2^{|n|+k}$ can be used. In [4], Goldreich and Rosen showed that $g^r \bmod n$ for $r \in \{0, \dots, 2^{|n|/2} - 1\}$ is a pseudo random generator. Therefore $N = 2^{|n|/2}$ is another possible choice and will result in elements g^r (for $r \in Z_N$) that cannot be distinguishes from an uniform element from the whole group generated by g.

Let the composite-DDH problem to be the decisional problem associated this assumption, i.e., given $(n, g, g^a \bmod n, g^b \bmod n, y)$, decide whether $g^{ab} \bmod n$ or $g^c \bmod n$.

Now we define other notations in the following.

Definition 2.1. *Define R_l^m to be the set of tuples of $l \times m$ elements:*
$I = (g, g^{a_{12}}, \dots, g^{a_{1m}}, g^{a_{21}}, \dots, g^{a_{2m}}, \dots, g^{a_{l1}}, \dots, g^{a_{lm}})$ *where* $g \in Z_n^*, a_{ij} \in Z_N (i = 1, \dots, l, j = 1, \dots, m)$.
Then define the subset D_l^m of R_l^m to be the set of tuples I satisfying $g^{a_{ij}} = g^{a_{i1}a_{1j}} \bmod n$ for all $i, j (i = 2, \dots, l, j = 2, \dots, m)$.

Definition 2.2. *(composite-DDH$_l^m$ problem) We define the problem of distinguishing instances uniformly chosen from R_l^m and those from D_l^m by composite-DDH$_l^m$ problem.*

Note that the composite-DDH problem can be denoted as the composite-DDH$_2^2$.

3 Outline of Main Protocol

3.1 Security for our protocol

The properties of sound and special honest verifier zero-knowledge can be proven under computational assumptions. The soundness can be proven the following proposition from the assumption of one-wayness of the Paillier's encryption scheme.

Proposition 3.1. *Assume that Paillier's public-key scheme is one-way. Then there is no polynomial-time algorithm which, on input a set of ciphertexts $E_1, \ldots, E_m \in Z^*_{n^2}$, output a non-trivial set* [4]*, $\alpha \in Z_N$ and $\{\alpha_i\} \in Z_N$ such that $g^{\alpha n} \prod_{j=1}^{m} E_j^{\alpha_j} = 1 \bmod n^2$.*

Proof:

Let us consider two algorithm $\mathcal{A}$ and $\mathcal{B}$. For $^\forall\alpha \in Z^*_N$, $\mathcal{A}$ is an algorithm which, on input a set made of n,$g \in Z^*_n$ and $\{E_i\}_{(i=1,\ldots,m)} \in Z^*_{n^2}$, outputs a non-trivial set $\{\alpha_1, \ldots, \alpha_n\}$ such that $\alpha^N \prod_{j=1}^{m} E_j^{\alpha_j} = 1 \bmod n^2$. And $\mathcal{B}$ is an algorithm which, on inputs a ciphertext $E \in Z^*_{n^2}$ and a public-key n of Paillier's scheme, outputs some value $x \in Z_n$ satisfying $E = (1 + nx)r^n \bmod n^2$ for $^\exists r \in Z^*_n$. We define these algorithm to output some correct answer with non-negligible probability.

To prove the proposition, we show a probabilistic polynomial-time transformation of the algorithm $\mathcal{B}$ to the algorithm $\mathcal{A}$, namely we will construct the algorithm $\mathcal{B}$ by using the algorithm $\mathcal{A}$ as an oracle.

algorithm $\mathcal{B}$
oracle: algorithm $\mathcal{A}$
input: $E \in Z^*_{n^2}, n$
output: a plaintext x such that $E = (1 + xn)r^n \bmod n^2$ for some $r \in Z^*_n$
[**Step 1**] Generate values $(x_2, \ldots, x_m) \in Z_n$ and $(r_2, \ldots, r_m) \in Z^*_n$ at random.
Compute $(E_1, \ldots, E_m)$ according to $E_1 = E, E_2 = (1 + x_2 n)r_2^n, \ldots,$
$E_m = (1 + x_m n)r_m^n \bmod n^2$.
[**Step 2**] Generate a permutation π at random.
Then run the algorithm $\mathcal{A}$ on input $(E_{\pi(1)}, \ldots, E_{\pi(m)}, g \in Z^*_n, n)$.
[**Step 3**] Get an output $(\alpha, \alpha_1, \ldots, \alpha_m)$ from $\mathcal{A}$.
If it does not satisfy the equation $g^{\alpha n} \prod_{j=1}^{m} E_{\pi(j)}^{\alpha_j} = 1 \bmod n^2$ or if $\alpha_{\pi^{-1}(1)} = 0$, then go to Step1.
[**Step4**] Compute a plaintext x of E from following linear equation:

$$x\alpha_{\pi^{-1}(1)} + \sum_{j=2}^{m} x_j \alpha_{\pi^{-1}(j)} = 0 \bmod n.$$

Note that g is a element in Z^*_n. Therefore the pair $(x\alpha_{\pi^{-1}(1)} + \sum_{j=1}^{m} x_j \alpha_{\pi^{-1}(j)}$, $g^\alpha r^{\alpha_{\pi^{-1}(1)}} \prod_{j=1}^{m} r_j^{\alpha_{\pi^{-1}(j)}}) = (0, 1) \bmod n$ such that $(1 + (x\alpha_{\pi^{-1}(1)} + \sum_{j=1}^{m} m_j \alpha_{\pi^{-1}(j)})n) \times (g^\alpha r^{\alpha_{\pi^{-1}(1)}} \prod_{j=1}^{m} r_j^{\alpha_{\pi^{-1}(j)}})^n = (1 + 0 \cdot n) \cdot 1^n \bmod n^2$ because the Paillier's encryption function is bijective.

[4] If $(\alpha, \alpha_1, \ldots, \alpha_m) = (0, \ldots, 0)$, we call it a trivial set, otherwise a non-trivial set.

From this, if we cannot break Paillier's encryption scheme there is no polynomial-time algorithm that can find a non-trivial set $\{\alpha_i\} \in Z_N$ such that $\alpha^N \prod_{j=1}^{m} E_j^{\alpha_j} = 1 \bmod n^2$.
$\square$

The special honest verifier zero-knowledge can be proven form the following propositions using the composite-DDH assumption.

Proposition 3.2. *For any $l(\geq 2)$, if there is an probabilistic polynomial time algorithm to solve the composite-DDH$_l^2$ problem, then there is an probabilistic polynomial time algorithm to solve the composite-DDH$_2^2$ problem.*

Proof: To prove this proposition, we will construct a probabilistic polynomial-time algorithm A_2^2 to solve the composite-DDH$_2^2$ problem by using an algorithm A_l^2 that can solve the composite-DDH$_l^2$ problem as an oracle. We consider the following procedure:
for any $I_2^2 = (g, g^{a_{12}}, g^{a_{21}}, g^{a_{22}}) \in R_2^2$, choose uniformly $\{z_i\}$ and $\{w_i\}_{(i=1,\dots,l)}$ from Z_N and set $I_l^2 = (g, g^{a_{12}}, g^{a_{21}}, g^{a_{22}}, \dots, g^{a_{l1}}, g^{a_{l2}}) \in R_l^2$ where $g^{a_{i1}} = g^{z_i}(g^{a_{21}})^{w_i}, g^{a_{i2}} = (g^{a_{12}})^{z_i}(g^{a_{22}})^{w_i}$ for $i = 3, \dots, l$.

algorithm A_2^2
oracle: algorithm A_l^2
input: n, g, I_2^2
output: if $I_n^2 \in_U D_n^2$ then output $b = 1$, else output $b = 0$
[Step 1] Generate I_l^2 along the above procedure.
[Step 2] Run the algorithm A_l^2 on input (n, g, I_l^2).
[Step 3] Get an output b from A_l^2 and output b.

If $b = 1$, $I_2^2 \in_R D_l^2$, and if $b = 0$, $I_2^2 \in_R R_l^2$. $\square$

Proposition 3.3. *For any $l(\geq 2)$ and $m(\geq 2)$, if there is an probabilistic polynomial time algorithm to solve the composite-DDH$_l^m$ problem, then there is an probabilistic polynomial time algorithm to solve the composite-DDH$_l^2$ problem.*

Proof: The proof is by induction on l, that is, if the composite-DDH$_l^m$ problem is easy, then either the composite-DDH$_l^{m-1}$ problem or the composite-DDH$_l^2$ problem is easy.

To prove this, we define the subset M_l^m of R_l^m to be the set of tuples of $n \times m$ elements:
$I = (g, g^{a_{12}}, \dots, g^{a_{1m}}, g^{a_{21}}, \dots, g^{a_{2m}}, \dots, g^{a_{l1}}, \dots, g^{a_{lm}})$
where $g \in Z_n^*, a_{ij} \in Z_N (i = 1, \dots, l, j = 1, \dots, m)$ and satisfying $g^{a_{ij}} = g^{a_{i1}a_{j1}} \bmod n^2$.

We will prove that if the composite-DDH$_l^m$ problem is easy, then we can either distinguish between the instances chosen uniformly from R_l^m and M_l^m or the instances chosen uniformly from D_l^m and M_l^m. In the former case, if we can solve the composite-DDH$_l^m$ problem, it means that we can distinguish between the instances chosen uniformly from M_l^{m-1} and R_l^{m-1} because the distribution of the elements from M_l^m and R_l^m are same. Therefore the composite-DDH$_l^{m-1}$ problem is easy. In the latter case, to prove this claim, we consider the following procedure:
for any $I_l^2 = (g, g^{a_{12}}, g^{a_{21}}, g^{a_{22}}, \ldots, g^{a_{l1}}, g^{a_{l2}}) \in R_l^2$, choose randomly $\{z_i\}$ from Z_N for $i = 1, \ldots, m$ and set

$$I_l^m = (g, g^{Z_2}, g^{z_3}, \ldots, g^{z_{m-1}}, g^{a_{12}}, g^{a_{21}}, (g^{a_{21}})^{z_2}, (g^{a_{21}})^{z_3},$$
$$\ldots, (g^{a_{21}})^{z_{m-1}}, g^{a_{22}}, \ldots, g^{a_{l1}}, (g^{a_{l1}})^{z_2}, (g^{a_{l1}})^{z_3}, \ldots, (g^{a_{l1}})^{z_{m-1}}, g^{a_{l2}}) \in R_l^2.$$

If $I_l^2 \in_R D_l^2$, $I_l^m \in_R D_l^m$, and if $E_l^2 \in_R R_l^2$, $I_l^m \in_R R_l^m$.
□

3.2 Proof-1 and Proof-2

Our protocol can be constructed by the following three proofs.

Proof-1 is a proof that given $\{E_i\}$ and $\{E_i'\}$, $\{E_i'\}$ can be expressed as Equation (3) using integers $\{r_i'\}$ and a matrix that satisfies the first condition (1), **Proof-2** is a proof that given $\{E_i\}$ and $\{E_i'\}$, $\{E_i'\}$ can be expressed as Equation (3) using integers $\{r_i'\}$ and a matrix that satisfies the second condition (2), and **Proof-3** is a proof that integers $\{r_i'\}$ and the matrix used in the above two proofs are identical.

Adopting Furukawa and Sako [3], we give two proofs, **Proof-1** and **Proof-2**, that consist of the main protocol.

Finally, we will describe the main protocol as a combination **Proof-1** and **Proof-2**. Note that Proving **Proof-3** can be done by this combination.

3.3 Proof-1

Now we provide a protocol for Proof-1.

<u>**Proof-1**</u>

Common Input:$\{E_i\}, \{E_i'\} \in Z_{n^2}^*$, $g \in Z_n^*$, n and $N = 2^{|n|/2}$.

Private Input:$\{A_{ij}\}, \{r_i'\}$ such that $E_i' = g^{r'_i n} \prod_{j=1}^{m} E_j^{A_{ji}} \bmod n^2$.

1. Prover P chooses random values $\alpha, \{\alpha_i\}, \sigma \in_R Z_N (i = 1, \ldots, m)$ and computes the following:

$$
\begin{aligned}
w &= g^{\sigma} \bmod n \\
E' &= g^{\alpha n} \prod_{j=1}^{m} E_j^{\alpha_j} \bmod n^2 \qquad (4) \\
\tilde{w}_i &= g^{\sum_{j=1}^{m} 2\alpha_j A_{ji} + \sigma r'_i} \bmod n \qquad i = 1, \ldots, m \\
\tilde{w} &= g^{\sum_{j=1}^{m} \alpha_j^2 + \sigma\alpha} \bmod n.
\end{aligned}
$$

Then P sends $w, E', \{\tilde{w}_i\}, \tilde{w}$ to the verifier V.

2. V chooses random values $\{c_i\} \in Z_N (i = 1, \ldots, m)$ and sends them to P as a challenge.

3. P computes the following:

$$
\begin{aligned}
s_i &= \sum_{j=1}^{m} A_{ij} c_j + \alpha_i \bmod N \qquad i = 1, \ldots, m \\
s &= \sum_{j=1}^{m} r'_j c_j + \alpha \bmod N.
\end{aligned}
$$

And sends them to V.

4. V verifies the following:

$$
\begin{aligned}
E' \prod_{j=1}^{m} E_j'^{c_j} &= g^{sn} \prod_{j=1}^{m} E_j^{s_j} \bmod n^2 \qquad (5) \\
\tilde{w} \prod_{j=1}^{m} \tilde{w}_j{}^{c_j} &= w^s g^{\sum_{j=1}^{m}(s_j^2 - c_j^2)} \bmod n^2. \qquad (6)
\end{aligned}
$$

the security of Proof-1

Theorem 1. *The protocol is a public coin 3-move proof system for proving a correctness of shuffle. It is complete, sound and special honest verifier zero-knowledge.*

Proof: It is easy to see that the protocol is a 3-move public coin protocol. Completeness is clear and soundness follows from Theorem 2. Special honest verifier zero-knowledge follows from Theorem 3.
□

Theorem 2. *The protocol is sound, that is, if V accepts the protocol with a non-negligible probability, then P either knows both $\{r'_i\}$ and $\{A_{ij}\}$ satisfying Equation (1) and (3) or can generate a non-trivial set $\alpha, \{\alpha_i\} \in Z_N$ satisfying $g^{\alpha n} \prod_{j=1}^{m} E_j^{\alpha_j} = 1 \bmod n^2$.*

Proof: If V accepts the protocol with non-negligible probability, from Lemma 2 P knows randomizers $\{r'_i\}$ and a permutation matrix (A_{ij}) satisfying $E'_i = g^{r'_i n}\prod_{j=1}^{m} E_j^{A_{ji}} \bmod n^2$. And from Lemma 1 the matrix satisfies the first condition 1 or P can generate a non-trivial set $\alpha, \alpha_1, \ldots, \alpha_m$ satisfying $g^{\alpha}\prod_{j=1}^{m} E_i^{\alpha_i} = 1 \bmod n^2$ with overwhelming probability. But from Proposition 3.1, it is infeasible to compute such a set.

□

Lemma 1. *If V accepts the protocol with non-negligible probability, the condition* (1) *hold or P can generate a non-trivial set* $\alpha, \alpha_1, \ldots, \alpha_m$ *satisfying* $g^{\alpha}\prod_{j=1}^{m} E_i^{\alpha_i} = 1 \bmod n^2$ *with overwhelming probability.*

Proof: From Lemma 2, a prover who can succeed in the protocol with non-negligible probability knows the values $\{A_{ij}\}, \{r'_i\}, \alpha, \{\alpha_i\}$ satisfying Equations (3) and (4), and the answers $s, \{s_i\}$ satisfying the verifications. From Lemma 3, if P cannot compute a non-trivial set $\{\alpha, \alpha_1, \ldots, \alpha_m\}$, $s = \sum_{j=1}^{m} r'_j c_j + \alpha$ and $s_i = \sum_{j=1}^{m} A_{ij} c_j + \alpha_i \bmod N$ hold for all $i \in \{1, \ldots, m\}$. So if Equation (6) holds, the condition (1) holds from Lemma 4.

□

Lemma 2. *If V accepts the protocol with non-negligible probability, then P knows* $\{A_{ij}\}, \{r'_i\}, \{\alpha_i\}, \alpha$ *satisfying Equations* (3) *and* (4).

Proof: A prover who can succeed in the protocol with non-negligible probability can choose with overwhelming probability that $n+1$ linearly independent vectors $\mathbf{c} = (1, c_1, \ldots, c_n)$ made of the challenges. From them, we can get $\alpha, \{r'_i\}, \{\alpha_i\}$ and $\{A_{ij}\}$ satisfying following:

$$s = \sum_{j=1}^{m} r'_i c_j + \alpha \bmod N$$

$$s_i = \sum_{j=1}^{n} A_{ij} c_j + \alpha_i \bmod N.$$

These values $\alpha, \{r'_i\}, \{\alpha_i\}$ and $\{A_{ij}\}$ satisfies Equation (3) and (4) from the satisfiability of Equation (5).

□

Lemma 3. *Assume P knows* $\{A_{ij}\}, \{r'_i\}, \{\alpha_i\}, \alpha$ *satisfying Equations* (3) *and* (4), *and* $\{s_i\}$ *and* s *satisfying Equation* (5). *If either* $s \neq \alpha \prod_{j=1}^{n} {r'_j}^{c_j} \bmod N$ *or* $s_i \neq \sum_{j=1}^{n} A_{ij} c_j + \alpha_i \bmod N$ *for some* $i \in \{1, \ldots, n\}$ *hold with non-negligible probability, P can generate a non-trivial set* $\alpha, \alpha_1, \ldots, \alpha_m$ *satisfying* $g^{\alpha}\prod_{j=1}^{m} E_i^{\alpha_i} = 1 \bmod n^2$ *with overwhelming probability.*

Proof: The following gives a non-trivial set $\alpha, \alpha_1, \ldots, \alpha_m$ satisfying $g^{\alpha} \prod_{j=1}^{m} E_i^{\alpha_i} = 1 \bmod n^2$ with overwhelming probability.

$$g^{\alpha n + \sum_{j=1}^{m} r'_j n - sn} \prod_{i=1}^{m} E_i^{\alpha_i + \sum_{j=1}^{m} A_{ij} c_j - s_i} = 1 \bmod n^2.$$

□

Lemma 4. *If V accepts the protocol with non-negligible probability and* $s_i = \sum_{j=1}^{n} A_{ij} c_j + \alpha_i \bmod N$ *holds, the condition* (1) *holds.*

Proof: From lemma 3, if V accepts the protocol, the following holds because of the satisfiability of Equation (6).

$$\sum_{i=1}^{n} \sum_{j=1}^{n} (\sum_{h=1}^{n} A_{hi} A_{hj} - \delta_{ij}) c_i c_j = 0 \bmod N.$$

So the condition (1) hold with overwhelming probability.

□

Theorem 3. *The protocol is special honest verifier zero-knowledge, that is, there is a procedure that can take any challenge* $\{c_i\}$ *as input and produce a conversation indistinguishable from the space of all conversations between the honest prover and verifier in which* $\{c_i\}$ *is the challenge.*

Proof: And to prove the special honest verifier zero-knowledge for this protocol, we will construct the distinguisher $\mathcal{D}$ who can distinguish between a real transcript for the protocol and an output from the simulator. Then we will prove that this distinguisher can be used to solve the composite-DDH assumption.

We first describe the simulator $\mathcal{S}$ as the following.

The simulation

Simulator $\mathcal{S}$(input: $\{E_i\}, \{E'_i\}, g, n, N, c$)

1. $\mathcal{S}$ chooses random values $\sigma, \alpha, \{\alpha_i\}, n, s, \{s_i\} \in_R Z_N$.

2. $\mathcal{S}$ computes as following.

$$E' = g^{sn} \prod_{j=1}^{m} E_j^{s_j} {E'_j}^{-c_j} \bmod n^2$$

$$\tilde{w} = w^s g^{\sum_{j=1}^{m}(s_j^2 - c_j^2)} \times \prod_{j=1}^{m} \tilde{w_j}^{-c_j} \bmod n^2.$$

3. $\mathcal{S}$ outputs the simulated proof $(E', w, \{\tilde{w}_i\}, \tilde{w}, s, \{s_i\})$.

We then construct a distinguisher $\mathcal{D}'$ who can distinguish between the instances of D^2_{l+1} and R^2_{l+1}, and choose the instance I uniformly from either D^2_{l+1} and R^2_{l+1}.

Then $\mathcal{D}$ chooses randomly a matrix (A_{ij}) from the set of permutation matrixes, and computes $E'_i = g^{a_{(i+1)1}n} \prod_{j=1}^m E_j^{A_{ji}}$.

We describe a simulator $\mathcal{S}'$ using above values, I and $\{E'_i\}$ as the following.

Simulator $\mathcal{S}'$(input: $\{E_i\}, \{E'_i\}, g, n, N, c$)

1. $\mathcal{S}'$ chooses random values $s, \{s_i\} \in_R Z_N$.
2. $\mathcal{S}'$ computes as following.

$$\begin{aligned}
w &= g^{a_{12}} \bmod n \\
\alpha_i &= s_i - \sum_{j=1}^{m} A_{ij} c_j \bmod n \quad (i = 1, \ldots, m) \\
\tilde{w}_i &= g^{a_{(i+1)2}} \prod_{j=1}^{m} g^{2\alpha_j A_{ji}} \bmod n \quad (i = 1, \ldots, m) \\
E' &= g^{sn} \prod_{j=1}^{m} E_j^{s_j} {E'_j}^{-c_j} \bmod n^2 \\
\tilde{w} &= w^s g^{\sum_{j=1}^{m}(s_j^2 - c_j^2)} \times \prod_{j=1}^{m} \tilde{w_j}^{-c_j} \bmod n^2.
\end{aligned}$$

3. $\mathcal{S}'$ outputs the simulated proof $(E', w, \{\tilde{w}_i\}, \tilde{w}, s, \{s_i\})$.

The indistinguishability

In the following, we prove that a distinguisher who can distinguish the output of the simulator $\mathcal{S}$ from a real transcript of the protocol can solve the composite-DDH$^2_{(l+1)}$, that is, We prove that simulator $\mathcal{S}'$ simulates the protocol when the instance $I \in_R D^2_{(l+1)}$ and $\mathcal{S}$ when the instance $I \in_R R^2_{(l+1)}$.

In the former case, we set $a_{12} = \sigma, a_{(i+1)1} = r_i$ $(i = 1, \ldots, m)$. then clearly the claim can be proved. In the latter case, we cannot distinguish the outputs of $\mathcal{S}'$ and $\mathcal{S}$ because $a_{(i+1)1}$ is chosen uniformly from Z_N.

□

3.4 Proof-2

We describe a zero-knowledge protocol for Proof-2 without proof. This proof can be proved analogously to the protocol for Proof-1.

Proof-2

Common Input:$\{E_i\}, \{E'_i\} \in Z^*_{n^2}$, $g \in Z^*_n$, n and $N = 2^{|n|/2}$.

Private Input:$\{A_{ij}\}, \{r'_i\}$ such that $E'_i = g^{r'_i n} \prod_{j=1}^m E_j^{A_{ji}} \bmod n^2$.

1. Prover P chooses random values $\tau, \rho, \alpha, \{\alpha_i\}, \lambda, \{\lambda_i\} \in_R Z_N (i = 1, \ldots, m)$ and computes the following:

$$
\begin{aligned}
t &= g^{\tau}, v = g^{\rho}, u = g^{\lambda}, u_i = g^{\lambda_i} \bmod n \qquad i = 1, \ldots, m \\
E' &= g^{\alpha n} \prod_{j=1}^{m} E_j^{\alpha_j} \bmod n^2 \\
\tilde{v} &= g^{\sum_{j=1}^m \alpha^3 + \tau\lambda + \rho\alpha} \bmod n \\
\tilde{t_i} &= g^{\sum_{j=1}^m 3\alpha_j A_{ji} + \tau\lambda_i} \bmod n \qquad i = 1, \ldots, m \\
\tilde{v_i} &= g^{\sum_{j=1}^m 3\alpha_j^2 A_{ji} + \rho r'_i} \bmod n \qquad i = 1, \ldots, m.
\end{aligned}
$$

Then P sends $t, v, u, \{u_i\}, E', \tilde{v}, \{\tilde{t_i}\}, \{\tilde{v_i}\}$ to the verifier V.

2. V chooses random values $\{c_i\} \in_R Z_N (i = 1, \ldots, m)$ and sends them to P as a challenge.

3. P computes the following:

$$
\begin{aligned}
s &= \sum_{j=1}^{m} r'_j c_j + \alpha \bmod N \\
s_i &= \sum_{j=1}^{m} A_{ij} c_j + \alpha_i \bmod N \qquad i = 1, \ldots, m.
\end{aligned}
$$

And sends them to V.

4. V verifies the following:

$$
\begin{aligned}
g^{sn} \prod_{j=1}^{m} E_j^{s_j} &= E' \prod_{j=1}^{m} E_j'^{c_j} \bmod n^2 \\
g^{\lambda'} &= u \prod_{j=1}^{m} u_j^{c_j^2} \bmod n^2 \\
t^{\lambda'} v^s g^{\sum_{j=1}^m (s_j^3 - c_j^3)} &= \tilde{v} \prod_{j=1}^{m} \tilde{t_j}^{c_j^2} \tilde{v_j}^{c_j} \bmod n^2.
\end{aligned}
$$

This protocol is a zero-knowledge protocol for proving a correct shuffle.

4 Our protocol

Protocol Proving a shuffle

Before going into our main protocol, we mention that there is a possibility that witnesses $\{r_i'\}$ leak out when we use only $\{E_i\}, \{E_i'\}$ as an input as the protocol for Proof-1 and Proof-2. Since $\{E_i'\}$ is 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 the values $\tilde{g}, \{\tilde{E}_i\}$ as an input in our protocol not to leak the witness. Let these values be chosen from the ciphertext space $Z_{n^2}^*$ in a way we can assure the relations among these values unknown. This way gives a zero-knowledge property from Proposition 3.1.

We provide a zero-knowledge protocol for proving a correct shuffle.

4.1 Description

<u>**Main Protocol**</u>

Common Input:$\{E_i\}, \{E_i'\} \in Z_{n^2}^*$, $\{\tilde{E}_i\} \in_R Z_{n^2}^*, \tilde{g} = g^a \bmod n$ such that $a \in_R Z_N$, g ,n and $N = 2^{|n|/2}$.

Private Input:$\{A_{ij}\}, \{r_i'\}$ such that $E_i' = g^{r_i' n} \prod_{j=1}^m E_j^{A_{ji}} \bmod n^2$.

1. Prover P chooses random values $\tau, \rho, \sigma, \alpha, \{\alpha_i\}, \lambda, \{\lambda_i\} \in_R Z_N(i = 1, \ldots, m)$ and computes the following:

$$t = g^\tau, v = g^\rho, w = g^\sigma, u = g^\lambda, u_i = g^{\lambda_i} \bmod n \qquad i = 1, \ldots, m$$

$$E' = g^{\alpha n} \prod_{j=1}^m E_j^{\alpha_j} \bmod n^2 \tag{7}$$

$$\tilde{w}_i = g^{\sum_{j=1}^m 2\alpha_j A_{ji} + \sigma r_i'} \bmod n$$

$$\tilde{w} = g^{\sum_{j=1}^m \alpha_j^2 + \sigma\alpha} \bmod n \qquad i = 1, \ldots, m$$

$$\tilde{v} = g^{\sum_{j=1}^m \alpha_j^3 + \tau\lambda + \rho\alpha} \bmod n$$

$$\tilde{t}_i = g^{\sum_{j=1}^m 3\alpha_j A_{ji} + \tau\lambda_i} \bmod n \qquad i = 1, \ldots, m$$

$$\tilde{v}_i = g^{\sum_{j=1}^m 3\alpha_j^2 A_{ji} + \rho r_i'} \bmod n \qquad i = 1, \ldots, m$$

$$\tilde{E}_i{}' = \tilde{g}^{r_i' n} \prod_{j=1}^m \tilde{E}_j{}^{A_{ji}} \bmod n^2 \qquad i = 1, \ldots, m \tag{8}$$

$$\tilde{E}' = \tilde{g}^{\alpha n} \prod_{j=1}^m \tilde{E}_j{}^{\alpha_j} \bmod n^2. \tag{9}$$

Then P sends $t, v, w, u, \{u_i\}, E', \{\tilde{w}_i\}, \tilde{w}, \tilde{v}$, $\{\tilde{t}_i\}, \{\tilde{v}_i\}, \{\tilde{E}_i{}'\}, \tilde{E}'$ to the verifier V.

2. V chooses random values $\{c_i\} \in_R Z_N (i = 1, \ldots, m)$ and sends them to P as a challenge.

3. P computes the following:

$$s = \sum_{j=1}^{m} r'_j c_j + \alpha \bmod N$$
$$s_i = \sum_{j=1}^{m} A_{ij} c_j + \alpha_i \bmod N \qquad i = 1, \ldots, m$$
$$\lambda' = \sum_{j=1}^{m} \lambda_j c_j^2 + \lambda \bmod N.$$

   And sends them to V.

4. V verifies the following:

$$g^{\tilde{s}n} \prod_{j=1}^{m} \tilde{E}_j^{s_j} = \tilde{E}' \prod_{j=1}^{m} \tilde{E}_j'^{c_j} \bmod n^2 \tag{10}$$
$$g^{sn} \prod_{j=1}^{m} E_j^{s_j} = E' \prod_{j=1}^{m} E_j'^{c_j} \bmod n^2 \tag{11}$$
$$g^{\lambda'} = u \prod_{j=1}^{m} u_j^{c_j^2} \bmod n^2 \tag{12}$$
$$t^{\lambda'} v^s g^{\sum_{j=1}^{m}(s_j^3 - c_j^3)} = \tilde{v} \prod_{j=1}^{m} \tilde{t}_j^{c_j^2} \tilde{v}_j^{c_j} \bmod n^2 \tag{13}$$
$$w^s g^{\sum_{j=1}^{m}(s_j^2 - c_j^2)} = \tilde{w} \prod_{j=1}^{m} \tilde{w}_j^{c_j} \bmod n^2. \tag{14}$$

4.2 The Security of our protocol

Theorem 4. *The protocol is a public coin 3-move proof system. It is complete, sound and special honest verifier zero-knowledge.*

Proof: It is easy to see that the protocol is a 3-move public coin protocol. Completeness is clear, and soundness follows from Theorem 5. Special honest verifier zero-knowledge follows from Theorem 6.
□

Theorem 5. *The protocol is sound, that is, if V accepts the protocol with a non-negligible probability, then P either knows both $\{r'_i\}$ and $\{A_{ij}\}$ satisfying Equation (1), (2) and (3) or can generate a non-trivial set $\alpha, \{\alpha_i\} \in Z_N$ satisfying $g^{\alpha n} \prod_{j=1}^{m} E_j^{\alpha_j} = 1 \bmod n^2$.*

Proof: We assume V accepts the protocol with a non-negligible probability. From Proposition 3.1, the probability of generating a non-trivial a non-trivial set $\alpha, \{\alpha_i\}$ satisfying $\tilde{g}^{\alpha} \prod_{j=1}^{m} \tilde{E_j}^{\alpha_j} = 1 \bmod n^2$ is negligible. Therefore, from the satisfiability of Equation (10), P knows $\{A_{ij}\}, \{r'_i\}, \{\alpha_i\}, \alpha$ satisfying Equations (8) and (9) and both $s = \sum_{j=1}^{m} R'_j c_j + \alpha \bmod N$ and $s_i = \sum_{j=1}^{m} A_{ij} c_j + \alpha$ hold similarly to Lemma 2 and 3. And similarly to Lemma 4, from the satisfiability of Equation (13) and (14), the following hold:

$$\sum_{i=1}^{m} \sum_{j=1}^{m} (\sum_{h=1}^{m} A_{hi} A_{hj} - \delta_{ij}) c_i c_j = 0 \bmod n$$

$$\sum_{i=1}^{m} \sum_{j=1}^{m} \sum_{k=1}^{m} (\sum_{h=1}^{n} A_{hi} A_{hj} A_{hk} - \delta_{ijk}) c_i c_j c_k = 0 \bmod n.$$

Therefore the conditions (1) and (2) hold with overwhelming probability.

And the following equation holds when s and $\{s_i\}$ is substituted for Equation (11).

$$\frac{g^{\alpha n} \prod_{j=1}^{m} E_j^{\alpha_j}}{E'} \times \prod_{j=1}^{m} (\frac{r'_j n \prod_{i=1}^{m} E_i^{A_{ij}}}{E'_j})^{c_j} = 1 \bmod n^2$$

From this, Equations (3) and (7) are hold with overwhelming probability for all $i \in \{1, \ldots, m\}$.

□

Theorem 6. *The protocol is special honest verifier zero-knowledge.*

And to prove the special honest verifier zero-knowledge for our main protocol, we will construct the distinguisher $\mathcal{D}$ who can distinguish between a real transcript for the protocol and an output from the simulator. Then we will prove that this distinguisher can be used to solve the composite-DDH assumption.

We first describe the simulator $\mathcal{S}$ as the following.

The simulation

Simulator $\mathcal{S}$(input: $\{E_i\}, \{E'_i\}, \{\tilde{E}_i\}, \tilde{g}, g, n, N, c$)

1. $\mathcal{S}$ chooses random values $s, \{s_i\}, \lambda', \tau, \rho, \sigma, \{\lambda_i\}, \{a_i\}, \{b_i\}, \{d_i\}, \{e_i\} \in_R Z_N$

2. $\mathcal{S}$ computes as following.

$$t = g^{\tau}, v = g^{\rho}, w = g^{\sigma}, u_i = g^{\lambda_i}, \tilde{w_i} = g^{a_i}, \tilde{t_i} = g^{b_i}, \tilde{v_i} = g^{d_i} \bmod n$$

$$\tilde{E_i}' = g^{e_i}, \quad \tilde{E}' = \tilde{g}^{sn} \prod_{j=1}^{m} \tilde{E_j}^{s_j} \tilde{E_j}'^{-c_j} \mod n^2$$

$$E' = g^{sn} \prod_{j=1}^{m} E_j^{s_j} E_j'^{-c_j}, \quad \tilde{w} = w^s g^{\sum_{j=1}^{m}(s_j^2 - c_j^2)} \times \prod_{j=1}^{m} \tilde{w_j}^{-c_j} \mod n^2$$

$$u = g^{\lambda'} \prod_{j=1}^{m} u_j^{-c_j}, \quad \tilde{v} = t^{\lambda'} v^s g^{\sum_{j=1}^{m}(s_j^3 - c_j^3)} \times \prod_{j=1}^{m} \tilde{v_j}^{-c_j} \tilde{t_j}^{-c_j^2} \mod n^2.$$

3. $\mathcal{S}$ outputs the simulated proof $(t,v,w,u,\{u_i\}, E',\{\tilde{w_i}\},\tilde{w},\tilde{v}, \{\tilde{t_i}\},\{\tilde{v_i}\},\{\tilde{E_i}'\}, \tilde{E}',s,\{s_i\},\lambda')$.

We then construct a distinguisher $\mathcal{D}'$ who can distinguish between the instances of D_{l+1}^5 and R_{l+1}^5, and choose the instance I uniformly from either D_{l+1}^5 and R_{l+1}^5.

Then $\mathcal{D}$ chooses randomly a matrix (A_{ij}) from the set of permutation matrixes, and computes $g^{\tilde{a_{12}}} \bmod n, E_i' = g^{a_{(i+1)1}n} \prod_{j=1}^{m} E_j^{A_{ji}}$.

We describe a simulator $\mathcal{S}'$ using above values, I, $g^{\tilde{a_{12}}}$ and $\{E_i'\}$ as the following.

<u>Simulator $\mathcal{S}'$</u>(input: $\{E_i\}, \{E_i'\}, \{\tilde{E_i}\}, \tilde{g}, g, n, N, c$)

1. $\mathcal{S}'$ chooses random values $s, \{s_i\}, \lambda', \{\beta_i\} \in_R Z_N$.

2. $\mathcal{S}'$ computes as following.

$$t = g^{a_{13}}, v = g^{a_{14}}, w = g^{a_{15}}, u_i = g^{a_{(i+1)1}\beta_i} \bmod n$$

$$\tilde{E_i}' = g^{a_{(i+1)2}n} \prod_{j=1}^{m} \tilde{E_j}^{\alpha_j}, \quad \tilde{E}' = \tilde{g}^{sn} \prod_{j=1}^{m} \tilde{E_j}^{s_j} \tilde{E_j}'^{-c_j} \mod n^2$$

$$E' = g^{sn} \prod_{j=1}^{m} E_j^{s_j} E_j'^{-c_j}, \quad \tilde{w} = w^s g^{\sum_{j=1}^{m}(s_j^2 - c_j^2)} \times \prod_{j=1}^{m} \tilde{w_j}^{-c_j} \mod n^2$$

$$u = g^{\lambda'} \prod_{j=1}^{m} u_j^{-c_j}, \quad \tilde{v} = t^{\lambda'} v^s g^{\sum_{j=1}^{m}(s_j^3 - c_j^3)} \times \prod_{j=1}^{m} \tilde{v_j}^{-c_j} \tilde{t_j}^{-c_j^2} \mod n^2$$

$$\tilde{u_i} = g^{\sum_{j=1}^{m} 2\alpha_j A_{ji} + a_{(a+1)5}}, \tilde{t_i} = g^{\sum_{j=1}^{m} 3\alpha_j A_{ji} + a_{(a+1)3}\beta_i} \mod n^2$$

$$\tilde{v_i} = g^{\sum_{j=1}^{m} 3\alpha_j^2 A_{ji} + a_{(a+1)4}} \mod n^2.$$

3. $\mathcal{S}'$ outputs the simulated proof $(t,v,w,u,\{u_i\},E',\{\tilde{w_i}\},\tilde{w},\tilde{v}, \{\tilde{t_i}\},\{\tilde{v_i}\},\{\tilde{E_i}'\}, \tilde{E}',s,\{s_i\},\lambda')$.

The indistinguishability

In the following, we prove that a distinguisher who can distinguish the output of the simulator $\mathcal{S}$ from a real transcript of the protocol can solve the composite-DDH$^5_{(l+1)}$, that is, We prove that simulator $\mathcal{S}'$ simulates the protocol when the instance $I \in_R D^5_{(l+1)}$ and $\mathcal{S}$ when the instance $I \in_R D^5_{(l+1)}$.

In the former case, we set $\tilde{g}^{a_{12}}, \tau = a_{13}, \rho = a_{14}, \sigma = a_{15}, r_i = a_{(i+1)1}, \lambda_i = \beta_i r_{i+1}$. Then clearly the claim can be proved. In the latter case, we cannot distinguish the outputs of $\mathcal{S}'$ and $\mathcal{S}$ because the randomizers are chosen uniformly from Z_N.

$\square$

4.3 Efficiency

In this section, we compare the efficiency of our protocol to the protocol of [3] and [6]. We assume the security parameter of the ElGamal cryptosystem to be $k = |p|$, and one of the Paillier's cryptosystem to be $k = |n|$. We set $N = 2^{|n|/2}$ We consider the case of shuffling m ciphertexts. We compare them by the number of exponentiations used in each protocol. We use the well-known "Square-and-Multiply algorithm" for modular exponentiation. In the protocols of [3], [6], and ours, the prover uses $8kml^3$, $6kml^3$ and $28kml^3$, and the verifier uses $10kml^3$, $6kml^3$ and $80kml^3$ for computing exponentiations, respectively. In the protocols of [3], [6], and ours, the prover sends $6km$ bits, $2km$ bits and $7km$ bits, respectively.

References

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

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

[3] 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.

[4] 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.

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

[6] 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.

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

[8] 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.

[9] 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.

[10] 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.

[11] 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.