A Provably Secure and Efficient Verifiable Shuffle based on a Variant of the Paillier Cryptosystem

Lan Nguyen
(University of Wollongong, Australia
ldn01@uow.edu.au)

Rei Safavi-Naini
(University of Wollongong, Australia
rei@uow.edu.au)

Kaoru Kurosawa
(Ibaraki University, Japan
kurosawa@cis.ibaraki.ac.jp)

Abstract: We propose a variant of the Paillier cryptosystem that improves efficiency in encryption, re-encryption and decryption while preserving the homomorphic property. We then use this variant to construct a new verifiable shuffle system and prove its security. We show that the new shuffle scheme has the least number of rounds and exponentiations compared to all known shuffle schemes. Finally, we show how to construct a publicly verifiable mix-net using the shuffle system.

Key Words: privacy, verifiable shuffles, mix-nets, Paillier's public-key system.
Category: E.3

1 Introduction

A *shuffle* takes an input list of ciphertexts and outputs a permuted and re-encrypted version of the input list. The main application (motivation for the study) of shuffles is to construct *mix-nets*, a cryptographic system introduced by Chaum [Chaum 1981] for providing anonymity and unlinkability in communication. Mix-nets are among the most widely used systems for providing communication privacy, and have found applications in anonymous email systems [Chaum 1981], Web browsing [Gabber et al. 1997], electronic voting [Park et al. 1993, Neff 2001, Jakobsson et al. 2002], location privacy for mobile networks [Kong and Hong 2003] and mobile IPs [Choi and Kim 2003], anonymous payment systems [Jakobsson and M'Raihi 1998, Choi and Kim 2003], secure multiparty computation [Jakobsson and Juels 2000] and privacy in advertisements [Juels 2001].

A mix-net consists of a number of mix-centres that collectively permute and decrypt the input list. Shuffles are used to implement mix-centres. A basic shuffle permutes its input list of ciphertexts through re-encryption. Mix-centres may also partially decrypt the list [Abe 1999], hence called *shuffle decryption.* Mix-nets that use shuffle decryption could be more efficient but in case one mix-centre fails, they require more effort to recover [Furukawa 2004].

The main security objective of a shuffle is to provide *unlinkability* of its input elements to output elements, and so effectively keeping the permutation secret. We refer to this property as *shuffle privacy.* A second important property of shuffles is *verifiability*: that is providing a proof that the output is correctly constructed. Verifiability of shuffles is used to provide *robustness* for mix-nets: that is ensuring that a mix-net works correctly even if a number of its mix-centres are malicious. If a shuffle's proof can be verified by any party, it allows the mix-net to provide *public verifiability*: that means the mix-net can prove its correct operation to any party. These are important properties of mix-nets and so verifiability of shuffles has received much attention. Shuffles must be efficient and the cost is measured in terms of computation and communication (number of rounds and communicated bits). Proving security properties of shuffles traditionally relied on proving the zero-knowledgeness of the underlying proof system.

Recently, a number of efficient constructions for verifiable shuffles have been proposed [Abe 1999, Abe and Hoshino 2001, Nguyen and Safavi-Naini 2003] [Ogata et al. 1997]. In Crypto'01, Furukawa and Sako [Furukawa and Sako 2001] gave a characterisation of permutation matrices in terms of two equations that could be efficiently proved, hence proposing an efficient verifiable shuffle with a 3-round proof system. However, the zero-knowledge property of the proof system remains an open problem. Furukawa et al. [Furukawa et al. 2002] noted a flaw in their original proof, proposed a new definition of security for shuffles and proved security of their system with respect to that definition. Neff later gave another efficient construction [Neff 2001], which was based on a generalisation of Chaum-Pedersen proof of knowledge of equality of discrete logarithms and the fact that a polynomial of degree n has at most n roots. An improved version of this proof system is given in [Neff 2003]. However, like the Furukawa-Sako scheme, the zero-knowledge property of the Neff proof system has not been correctly proved and still remains an open problem [Nguyen and Safavi-Naini 2004]. All these schemes use the El Gamal encryption system and their security relies on the discrete logarithm assumption. Based on Neff's method, Groth [Groth 2003] proposed a very efficient proof system that uses homomorphic commitments. The input ciphertexts in this scheme can be encrypted by any homomorphic cryptosystem. A recent direction in designing mix-nets has been to trade off some privacy or correctness for efficiency [Boneh and Golle 2002, Golle et al. 2002, Jakobsson et al. 2002].

We proposed a formal security model for shuffles [Nguyen et al. 2004] that provided a unified approach to the assessment of shuffle systems. The model rigorously defined the above two security properties with respect to an active adversary. In our approach, the definition of shuffle privacy is motivated by observing the similarity between a shuffle hiding the underlying permutation, and an encryption system hiding the input message. The definition of verifiability is based on the notions of completeness and soundness of the proof system that proves the output is correctly constructed. We also proposed a new efficient verifiable shuffle based on the Paillier encryption scheme [Paillier 1999]. The shuffle uses the Furukawa-Sako approach for characterisation of permutation matrices but has computations over a composite modulus. We prove security of our verifiable shuffle scheme in this model.

In this paper we build on our results in [Nguyen et al. 2004] by first proposing an efficient variant of the Paillier encryption system and using it to construct an efficient verifiable shuffle scheme. We then use the shuffle scheme to construct an efficient robust mix-net system with public verifiability. Similar to the original Paillier scheme, the variant encryption scheme provides semantic security against adaptive chosen plaintext attacks and homomorphism. However, it has more efficient encryption, re-encryption and decryption. The decryption method of our proposed variant is the same as the variant proposed in Paillier's original paper, however, as we will note in section 3.4, the original variant is insecure for his suggested selection of parameters and our proposed variant shows how this problem can be corrected. The efficiency of the verifiable shuffle based on this variant is not only due to the encryption system's efficiency but also the fact that it becomes possible (Theorem 10) to use smaller size exponents (challenges in the proof system) and so reduce the cost of each exponentiation.

Our proposed proof system provides the same round efficiency as the Nguyen et al. and Furukawa-Sako proof systems but it requires less exponentiations. Compared to Groth's protocol, it reduces the number of rounds to less than half and only requires slightly more exponentiations. Our proof system also requires less rounds and exponentiations compared to Neff's protocol. By using computation techniques described in [Menezes et al. 1997], such as the fixed-based comb method and the simultaneous multiple exponentiation algorithm, the computation cost of the shuffle can be reduced to $3.4n$ exponentiations while the same techniques for the Furukawa-Sako and Groth protocols give $4.8n$ and $3.5n$ exponentiations, respectively. Hence overall, the proposed shuffle has the least numbers of rounds and exponentiations. (Note that exponentiations in our case is in modulo N^2, which is more expensive than modulo p and so the number of bit operations in Groth's shuffle is smaller.) Also, similar to the Groth and Nguyen et al. schemes [Nguyen et al. 2004], our scheme does not require the message space to be prime (product of two primes instead).

The organization of the paper is as follows. In section 2, we recall some background on public-key encryption and shuffles. Section 3 shows our modification of the Paillier encryption scheme, its improvements on efficiency and the flaw in

Paillier's variant scheme. The next section gives a verifiable shuffle scheme based on our variant of the Paillier system, its security proofs and efficiency analysis. Section 5 constructs a robust mix-net with public verifiability from the verifiable shuffle and section 6 concludes the paper.

2 Background

2.1 Notations and Terminology

Let *lcm* and *gcd* stand for 'least common multiple' and 'greatest common divisor', respectively. For a set $\mathbf{S}$, $|\mathbf{S}|$ denotes the number of elements in the set and "$x \leftarrow \mathbf{S}$" denotes an element x uniformly chosen from $\mathbf{S}$. $\{Element|Conditions\}$ denotes the set of *Element*s satisfying the *Conditions*. An algorithm $\mathcal{A}$ can simply be viewed as a machine that takes as input a string x, performs some operations and outputs a string y. It is denoted by $y \leftarrow \mathcal{A}(x)$. Let PT denote *polynomial-time*, PPT denote *probabilistic* PT and DPT denote *deterministic* PT. Let "$\mathsf{Pr}[Predicate]$" denote the probability that *Predicate* is true. For a function $f : \mathbb{N} \rightarrow \mathbb{R}^+$, if for every positive number α, there exists a positive integer l_0 such that for every integer $l > l_0$, it holds that $f(l) < l^{-\alpha}$, then f is said to be *negligible*. A problem is said to be *computationally difficult* if for every PT algorithm, the probability that the PT algorithm can solve the problem is a negligible function.

2.2 Public-key Encryption Schemes

2.2.1 Syntax

A public-key encryption scheme consists of a *key generation* algorithm $\mathcal{G}$, an *encryption* algorithm $\mathcal{E}$ and a *decryption* algorithm $\mathcal{D}$. It is denoted by $(\mathcal{G}, \mathcal{E}, \mathcal{D})$.

- Key generation: The PPT algorithm $\mathcal{G}$ on input 1^l outputs (pk, sk) where pk is the public key, sk is the secret key and l is a security parameter. It is denoted by $(pk, sk) \leftarrow \mathcal{G}(1^l)$.
- Encryption: The PPT algorithm $\mathcal{E}$ takes as input the public key pk and a plaintext m and outputs a ciphertext c. It is denoted by $c \leftarrow \mathcal{E}(pk, m)$ or $c \leftarrow \mathcal{E}_{pk}(m)$.
- Decryption: The DPT algorithm $\mathcal{D}$ takes as input the secret key sk and a ciphertext c and outputs a plaintext such that if $c \leftarrow \mathcal{E}_{pk}(m)$ then $\mathcal{D}_{sk}(c) = m$, where $\mathcal{D}_{sk}(c)$ (or $\mathcal{D}(sk, c)$) denotes the output of $\mathcal{D}$ on input sk and c.

A public-key encryption scheme, such as the El Gamal and Paillier schemes, may have a *re-encryption* algorithm. Following the definition in [Wikstrom 2002], this means there is a PPT algorithm $\mathcal{R}$ that takes as input the public key pk and a ciphertext and outputs another ciphertext such that for every plaintext m and its ciphertexts c and c':

$$Pr[c' = \mathcal{R}_{pk}(c)] = Pr[c' = \mathcal{E}_{pk}(m)] \tag{1}$$

where $\mathcal{R}_{pk}(c)$ (or $\mathcal{R}(pk, c)$) denotes the output of $\mathcal{R}$ on input pk and c. A public-key encryption scheme with a re-encryption algorithm is denoted by $(\mathcal{G}, \mathcal{E}, \mathcal{D}, \mathcal{R})$.

2.2.2 Security

We briefly recall definitions and notions of security used in this paper and more details can be found in [Goldreich 2004]. There are two equivalent notions of security for encryption against *chosen plaintext attacks*, *semantic security* (SS-CPA) and *indistinguishability* (IND-CPA). A chosen plaintext attack means that the adversary can obtain ciphertexts corresponding to plaintexts that he adaptively chooses. Semantic security intuitively means that whatever the adversary is able to compute about the plaintext from a challenge ciphertext, can also be computed without the ciphertext. Indistinguishability means that it is computationally infeasible to distinguish encryptions of two plaintexts of the same length.

There are also two equivalent definitions of encryption security against *chosen ciphertext attacks*, *semantic security* (SS-CCA) and *indistinguishability* (IND-CCA). A chosen ciphertext attack means that the adversary can obtain plaintexts corresponding to ciphertexts that he adaptively chooses, even after the challenge ciphertext is given. Another type of security requirement is *non-malleability* which means that given a ciphertext, it is computationally infeasible to generate a different ciphertext such that the corresponding plaintexts are related in a known manner. It has been proved [Goldreich 2004] that non-malleability against chosen ciphertext attacks (NM-CCA) is equivalent to SS-CCA and IND-CCA.

2.3 Paillier Public-key System

Key generation: Let $N = pq$, where p and q are large primes, and $\lambda = lcm(p-1, q-1)$. The public key is $pk = N$ and the secret key is $sk = \lambda$. Hereafter, unless stated otherwise, we assume all modular computations are in modulo N^2.
Encryption: Plaintext $m \in \mathbb{Z}_N$ can be encrypted by choosing $r \leftarrow \mathbb{Z}_N^*$ and computing the ciphertext $e = r^N(1 + mN)$. (Paillier encryption is originally defined as $e = r^N g^m$, where $g \in \mathbb{Z}_{N^2}^*$ and its order in modulo N^2 is a non-zero multiple of N. For efficiency, we use $g = 1 + N$. Our results do not depend on this choice and are true for all values of g.)

Re-encryption: A Paillier ciphertext e for a plaintext m can be re-encrypted as $e' = r'^N e$ for the same plaintext m, where $r' \leftarrow \mathbb{Z}_N^*$. The re-encryption algorithm satisfies the condition (1) above.
Decryption: Ciphertext $e \in \mathbb{Z}_{N^2}^*$ can be decrypted as $m = L(e^\lambda \bmod N^2)/\lambda \bmod N$, where the function L takes its input from the set $\{u \in \mathbb{Z}_{N^2} | u = 1 \bmod N\}$ and is defined as $L(u) = (u-1)/N$.
Computational Composite Residuosity (CCR) Assumption: Suppose $z \leftarrow \mathbb{Z}_{N^2}^*$ is given, the Computational Composite Residuosity problem is to find $x \in \mathbb{Z}_N$ such that there exists $r \in \mathbb{Z}_N^*$ satisfying $z = r^N(1 + xN) \bmod N^2$. The CCR assumption states that the CCR problem is computationally difficult.
Decisional Composite Residuosity (DCR) Assumption: A number $z \in \mathbb{Z}_{N^2}^*$ is said to be a w^{th} *residue mod* N^2 if there exists a number $y \in \mathbb{Z}_{N^2}^*$ such that $z = y^w$. Let $\mathcal{W}_N$ denote the set of N^{th} residues modulo N^2. The Decisional Composite Residuosity problem is to distinguish between an element uniformly chosen from the set $\mathcal{W}_N$ and an element uniformly chosen from the set $\mathbb{Z}_{N^2}^*$. The DCR assumption states that the DCR problem is computationally difficult.
Security: Theorem 1 states security of the Paillier scheme and its proof can be found in [Paillier 1999].

Theorem 1. *The Paillier encryption scheme provides SS-CPA if and only if the DCR assumption holds.*

NM-CCA robust threshold encryption scheme: Using the twin-encryption paradigm [Naor and Yung 1990], the Shamir secret sharing scheme [Shamir 1979] and a simulation-sound proof of equality of plaintexts, Fouque and Pointcheval [Fouque and Pointcheval 2001] proposed an NM-CCA robust threshold encryption scheme based on the Paillier public-key system that is proved secure in the random oracle model.

2.4 Formal Model of Verifiable Shuffles

We proposed a formal model for verifiable shuffles [Nguyen et al. 2004]. The model defines a verifiable shuffle as a tuple of three elements: a public-key scheme with a re-encryption algorithm $\mathcal{RP}$, a PPT algorithm $\mathcal{S}$ for shuffling and a proof system $(\mathcal{P}, \mathcal{V})$. The shuffling algorithm takes a list of ciphertexts of the public-key scheme and outputs a permuted list of their re-encryptions. The proof system proves that the output is really a permutation of re-encryptions of the input ciphertexts.

The model also specifies two security requirements for verifiable shuffles, privacy and verifiability. Privacy requires an honest shuffle to protect its secret permutation whereas verifiability requires that any attempt by a malicious shuffle to produce an incorrect output must be detectable. The definition of privacy is based on the similarity between a shuffle hiding the permutation, and a ciphertext hiding the message. Adaptive attacks are modelled by an active adversary that uses *chosen permutation attacks (CPA$_S$)* (similar to chosen plaintext attacks) or *chosen transcript attacks (CTA$_S$)* (similar to chosen ciphertext attacks). For CPA$_S$, the adversary can obtain transcripts of the shuffle executions corresponding to permutations that the adversary adaptively chooses. For CTA$_S$, the adversary obtains permutations that correspond to valid shuffle transcripts that it adaptively chooses. The notions of privacy for shuffles are defined in line with semantic security and indistinguishability for encryption. *Semantic privacy* (SP) formalizes the intuition that whatever is computable about the permutation from a shuffle execution transcript must also be computable without the transcript. *Indistinguishability* (IND) for shuffles means that it is infeasible to distinguish transcripts of two shuffle executions that correspond to two permutations of the same size. It has been proved that these two notions of privacy are equivalent and can be interchangeably used [Nguyen et al. 2004].

The definition of verifiability mainly depends on the verifiable shuffle's proof system. The proof system proves that the shuffle's output is a permutation of re-encryptions of the input ciphertexts. The proof system should satisfy two conditions, completeness and soundness. The completeness condition states that if the output is truly a permutation of re-encryptions of the input, then the proof system accepts with overwhelming probability. The soundness condition means that if the proof system accepts with overwhelming probability, then the output is truly a permutation of re-encryptions of the input.

We will show that our proposed verifiable shuffle system achieves SP-CPA$_S$ and verifiability based on some computational assumptions.

2.5 Paillier-based Verifiable Shuffle

We proposed [Nguyen et al. 2004] an efficient verifiable shuffle scheme based on the Paillier public-key system and proved its security in the formal model above. Similar to the Furukawa-Sako scheme, a permutation is represented as a matrix (Definition 2) and the proof system proves validity of a set of equations derived from the matrix (Theorem 3). Computation over a composite modulus complicates the security proof and requires another theorem, Theorem 4.

Definition 2. A matrix $(A_{ij})_{n\times n}$ is a permutation matrix modulo l if it satisfies the following for some permutation π

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

Theorem 3. *([Nguyen et al. 2004]) A matrix $(A_{ij})_{n\times n}$ is a permutation matrix modulo N, where $N = pq$ with primes p and q, if for all i, j and k, $gcd(A_{ij}, N)$*

is different from p and q and both of the following equations hold:

$$\sum_{l=1}^{n} A_{li}A_{lj} = \begin{cases} 1 \ mod\ N \ if\ i = j \\ 0 \ mod\ N\ otherwise \end{cases} \tag{2}$$

$$\sum_{l=1}^{n} A_{li}A_{lj}A_{lk} = \begin{cases} 1 \ mod\ N \ if\ i = j = k \\ 0 \ mod\ N\ otherwise \end{cases} \tag{3}$$

Theorem 4. *([Nguyen et al. 2004]) For a set of vectors* $\mathbf{S}$*, let* $\langle \mathbf{S} \rangle_k$ *denote the vector space spanned by* $\mathbf{S}$ *over* $\mathbb{Z}_k$ *(so the coordinates of a vector in* $\langle \mathbf{S} \rangle_k$ *are in* $\mathbb{Z}_k$*). Consider a set of vectors* $S_n = \{(1, c_1, ..., c_n) \mid (c_1, ..., c_n \in \mathbb{Z}_N) \wedge (\nexists Q_n \subseteq S_n : |Q_n| = n + 1 \wedge \langle Q_n \rangle_p = \mathbb{Z}_p^{n+1} \wedge \langle Q_n \rangle_q = \mathbb{Z}_q^{n+1})\}$ *(that means* S_n *is the set of vectors* $(1, c_1, ..., c_n)$*, where* $c_1, ..., c_n \in \mathbb{Z}_N$ *and there does not exist any subset* $Q_n \subseteq S_n$ *of size* $n + 1$ *such that* Q_n *spans* $\mathbb{Z}_p^{n+1}$ *and* $\mathbb{Z}_q^{n+1}$*). Then* $|S_n| \leq (p + q)N^{n-1}$*.*

2.6 Robust Mix-nets

A mix-net that consists of a set of servers receives as input a list of ciphertexts. The servers collectively permute and decrypt the input list and the mix-net finally outputs a permuted list of the corresponding plaintexts. By keeping the permutation secret, the mix-net can hide the correspondence between input items and output items hence providing privacy for the originators and receivers of messages. Informally, a robust mix-net must satisfy the following properties:

- *privacy:* it is infeasible for an adversary to output a pair of an input item and the corresponding output item of an honest user with probability non-negligibly better than a random guess.
- *robustness:* the probability that the mix-net produces correct output is negligibly less than 1.

It is also desirable for a robust mix-net to achieve

- *public verifiability:* that means the correctness of the mix-net's operation can be verified by any participant in the system.

3 Modified-Paillier Public-key System

In the Paillier encryption scheme, encryption and re-encryption requires an exponentiation to power N. The following modification of the Paillier scheme allows encryption and re-encryption operation to use exponentiation of a fixed base to a random power much smaller than N. Due to the fixed base, we can use "fixed-based comb method" [Menezes et al. 1997] which improves efficiency for multiple exponentiations where the base is fixed and the exponent varies. The scheme preserves the homomorphic property but requires a new assumption, the Decisional Fixed Base (DFB) assumption, which is stronger than the DCR assumption. It has an efficient decryption algorithm that uses the same technique as used in Paillier's efficient-decryption variant scheme in [Paillier 1999]. We will also show that the originally proposed parameter selection for Paillier's variant scheme makes it insecure and propose a parameter selection method that results in a secure system.

3.1 Description

Key generation: Let l_N and l_η be security parameters. Suppose p and q are distinct $l_N/2$-bit strong primes and p' and q' are distinct $l_\eta/2$-bit primes, such that p' is a divisor of $p-1$ but not a divisor of $q-1$ and q' is a divisor of $q-1$ but not a divisor of $p-1$. Suppose $N = pq$, $\eta = p'q'$, θ has order ηN in modulo N^2 and $\gamma = \theta^N$. The public key is $pk = (N, \theta, \gamma)$ and the secret key is $sk = \eta$.
Encryption: Plaintext $m \in \mathbb{Z}_N$ can be encrypted by choosing an $r \leftarrow \{0,1\}^{l_\eta}$ and computing the ciphertext $e = \gamma^r(1 + mN)$.
Re-encryption: A Modified-Paillier ciphertext e can be re-encrypted as another ciphertext $e' = e \times \gamma^{r'}$ of the same plaintext m, where $r' \leftarrow \{0,1\}^{l_\eta}$.
Decryption: Ciphertext $e \in \mathbb{Z}^*_{N^2}$ can be decrypted as $m = L(e^\eta \bmod N^2)/\eta \bmod N$, where the function L takes its input from the set $\{u \in \mathbb{Z}_{N^2} | u = 1 \bmod N\}$ and is defined as $L(u) = (u-1)/N$. This can be done very efficiently using the Chinese Remainder Theorem [Paillier 1999]. Note that a Modified-Paillier ciphertext is also a valid Paillier ciphertext, so the decryption can also be performed using $\lambda = lcm(p-1, q-1)$, as in the Paillier encryption scheme.

3.2 New complexity assumptions

Before proving security of the Modified-Paillier public-key system, we present new complexity assumptions underlying security of the Modified-Paillier public-key system.

Computational Fixed Base (CFB) Assumption: *Suppose N, θ and γ are generated as in the* key generation *algorithm. Let $\mathcal{C}_{N,\gamma}$ be the set $\{\gamma^r(1+xN) \in \mathbb{Z}^*_{N^2} \mid r \leftarrow \{0,1\}^{l_\eta}, x \leftarrow \mathbb{Z}_N\}$ (which is the set of Modified-Paillier ciphertexts). The Computational Fix Based problem is defined as follows: given (N, θ, γ) and $z \leftarrow \mathcal{C}_{N,\gamma}$, compute $x \in \mathbb{Z}_N$ such that there exists $r \in \{0,1\}^{l_\eta}$ satisfying $z = \gamma^r(1+xN) \bmod N^2$. The Computational Fix Based assumption states that the Computational Fix Based problem is computationally difficult.*

The relationship between the CFB assumption and the CCR assumption is stated in Lemma 5.

Lemma 5. *If the CFB assumption holds, then the CCR assumption holds.*

Proof. To prove that the CFB assumption leads to the CCR assumption, we show that if a PPT algorithm $\mathcal{A}$ can break the CCR assumption, then a PPT algorithm $\mathcal{B}$, which solves the CFB problem, can be constructed as follows. If $\mathcal{B}$ is given (N, θ, γ, z) where $z \leftarrow \mathcal{C}_{N,\gamma}$, $\mathcal{B}$ generates $r \leftarrow \mathbb{Z}_N^*$ and gives (N, zr^N) to $\mathcal{A}$.

We observe that if z is uniformly distributed in $\mathcal{C}_{N,\gamma}$, then zr^N is uniformly distributed in the set $\mathbb{Z}_{N^2}^*$. Therefore, if $\mathcal{A}$ can compute $x \in \mathbb{Z}_N$ such that there exists $r' \in \mathbb{Z}_N^*$ satisfying $zr^N = r'^N(1 + xN) \bmod N^2$, then $\mathcal{B}$ can compute $x \in \mathbb{Z}_N$ such that there exists $r” \in \{0,1\}^{l_\eta}$ satisfying $z = \gamma^{r”}(1 + xN) \bmod N^2$. In other words, if $\mathcal{A}$ can break the CCR assumption, then $\mathcal{B}$ can solve the CFB problem.

The semantic security of the Modified-Paillier public-key system relies on the Decisional Fixed Base Assumption, which is presented as follows.

Decisional Fixed Base (DFB) Assumption: *Suppose N, θ and γ are generated as in the* key generation *algorithm, and $\mathcal{C}_{N,\gamma}$ is defined as above. Let $\mathcal{C}_{N,\gamma}^0$ be the set $\{\gamma^r \mid r \leftarrow \{0,1\}^{l_\eta}\}$, which is a subset of $\mathcal{C}_{N,\gamma}$. The Decisional Fix Based problem is defined as follows: given (N, θ, γ), distinguish between a uniform distribution on the set $\mathcal{C}_{N,\gamma}^0$ and a uniform distribution on the set $\mathcal{C}_{N,\gamma}$. The Decisional Fix Based assumption states that the Decisional Fix Based problem is computationally difficult.*

The relationship between the DFB assumption and the DCR assumption is stated in Lemma 6.

Lemma 6. *If the DFB assumption holds, then the DCR assumption holds.*

Proof. To prove the lemma, we show that if a PPT algorithm $\mathcal{A}$ can break the DCR assumption, then a PPT algorithm $\mathcal{B}$, which solves the DFB problem, can be constructed as follows. To decide if a value z is uniformly chosen from $\mathcal{C}_{N,\gamma}$ or from $\mathcal{C}_{N,\gamma}^0$, $\mathcal{B}$ generates $r \leftarrow \mathbb{Z}_N^*$ and gives zr^N to $\mathcal{A}$.

We observe that if z is uniformly distributed in $\mathcal{C}_{N,\gamma}^0$, then zr^N is uniformly distributed in the set $\mathcal{S}_N$ of N-th residues modulo N^2; and if z is uniformly distributed in $\mathcal{C}_{N,\gamma}$, then zr^N is uniformly distributed in the set $\mathbb{Z}_{N^2}^*$. Therefore, if $\mathcal{A}$ can distinguish between a uniform distribution on the set $\mathcal{S}_N$ and a uniform distribution on the set $\mathbb{Z}_{N^2}^*$, then $\mathcal{B}$ can distinguish between a uniform distribution on the set $\mathcal{C}_{N,\gamma}^0$ and a uniform distribution on the set $\mathcal{C}_{N,\gamma}$. In other words, if $\mathcal{A}$ can break the DCR assumption, then $\mathcal{B}$ can solve the DFB problem.

3.3 Security

Security of the Modified-Paillier public-key system is stated in Theorem 7.

Theorem 7. *The Modified-Paillier encryption scheme has SS-CPA if and only if the DFB assumption holds.*

Proof. Assume that m_0 and m_1 are two known plaintexts and e is the Modified-Paillier ciphertext of either m_0 or m_1. Then e is the ciphertext of m_0 if and only if $e(1+m_0N)^{-1}$ is an exponentiation of γ. Therefore, if a party can distinguish an exponentiation of γ, he can break SS-CPA of the Modified-Paillier encryption scheme, and vice versa.

3.4 Parameter Selection in Paillier's Variant Scheme

Paillier proposed a decryption-efficient variant [Paillier 1999] of his public-key cryptosystem. In this variant, the public key includes g of order αN (modulo N^2). Paillier recommended the secret key α to be a prime. However, if α is a prime, then the knowledge of g allows factorization of $N = pq$ or finding the secret α, as shown in the following. Since g is of order αN (modulo N^2), α is a prime and $g^{\lambda N} = 1 \bmod N^2$ where $\lambda = lcm(p-1, q-1)$, α is a divisor of λ. This means that α must divide $p-1$ or $q-1$, or both. If it divides both $p-1$ and $q-1$, then it divides $N-1$ and can be recovered from factoring $N-1$. If α divides $p-1$ but not $q-1$, then let $h = g^{(N-1)N} \bmod N^2$. It can be seen that $h = 1 \bmod q$ and $h \neq 1 \bmod p$, so we can compute $q = gcd(h-1, N)$ and hence N can be factored.

The flaw can be fixed by choosing g of order ηN modulo N^2 instead, where η is computed as in our Modified-Paillier Public-key System.

Our proposed modification shares the decryption algorithm of this scheme but has a more efficient encryption algorithm because each encryption in our scheme costs only about one exponentiation of a fixed base to a random power much smaller than N. In Paillier's variant scheme, each encryption costs either one exponentiation to a power very much larger than N or two exponentiations.

4 A Verifiable Shuffle based on the Modified-Paillier Public-key System

4.1 Description

We construct a verifiable shuffle scheme $(\mathcal{RP}, \mathcal{S}, (\mathcal{P}, \mathcal{V}))$, where the public-key encryption scheme with a re-encryption algorithm $\mathcal{RP}$ is our proposed Modified-Paillier scheme, $\mathcal{S}$ is a PPT algorithm for shuffling and $(\mathcal{P}, \mathcal{V})$ is a proof system for verifiability. Let the system public key be $pk = (N, \theta, \gamma)$, where $N = pq$ with primes p and q, and let the secret key be $sk = \eta$, as generated in the key generation algorithm of the Modified-Paillier public-key scheme. The shuffling algorithm $\mathcal{S}$ takes pk, a list of Modified-Paillier ciphertexts $e_1, ..., e_n \in \mathcal{C}_{N,\gamma}$

and a permutation π and outputs another list of Modified-Paillier ciphertexts $e'_1, ..., e'_n \in \mathcal{C}_{N,\gamma}$, where $\mathcal{C}_{N,\gamma}$ is defined in the definition of the CFB assumption. The proof system $(\mathcal{P}, \mathcal{V})$, which is described in the next subsection, must prove the existence of a permutation π and $r_1, ..., r_n \in \{0,1\}^{l_\eta}$ such that $e'_i = \gamma^{r_i} e_{\pi^{-1}(i)}$, $i = 1, ..., n$.

Outline of the proof system

The proof system is based on ideas underlying the Furukawa-Sako proof system [Furukawa and Sako 2001] and the Nguyen et al. proof system [Nguyen et al. 2004]. A permutation is also represented as a permutation matrix, which is defined in Definition 2. It also relies on Theorem 3, which states conditions of a permutation matrix modulo N.

Representing the permutation π used in the shuffle as a permutation matrix, the shuffle's proof system, which proves the correctness of the shuffle, must show the existence of a permutation matrix modulo N $(A_{ij})_{n \times n}$ and $\{r_i \in \{0,1\}^{l_\eta} | i = 1, ..., n\}$ satisfying the following relationship between input and output items:

$$e'_i = \gamma^{r_i} \prod_{j=1}^{n} e_j^{A_{ji}}, \ i = 1, ..., n \tag{4}$$

Theorem 3 states conditions to achieve a permutation matrix modulo N. Then the proof system needs to prove the existence of a matrix $(A_{ij})_{n \times n}$ and $\{r_i | i = 1, ..., n\}$ satisfying equation 4 and the conditions on the matrix, as stated in Theorem 3.

In the proof system, based on the CFB assumption, it is computationally difficult for the prover to compute p and q. Hence, for any matrix $(A_{ij})_{n \times n}$ the prover can generate, "$gcd(A_{ij}, N)$ is different from p and q". Therefore, based on Theorem 3, the proof system needs to prove the following statements:

- Given $\{e_i\}$ and $\{e'_i\}$, $\{e'_i\}$ can be expressed as equation (4) using $\{r_i\}$ and a matrix that satisfies equation (2). The part $\langle(\{\tilde{g_i}'\}, \tilde{g}', e', \{\dot{w_i}\}, \dot{w}), \{c_i\}, (\{s_i\}, \tilde{s}, s, v)\rangle$ of the proof system proves this relationship.
- Given $\{e_i\}$ and $\{e'_i\}$, $\{e'_i\}$ can be expressed as equation (4) using $\{r_i\}$ and a matrix that satisfies equation (3). The part $\langle(\{\tilde{g_i}'\}, \tilde{g}', e', \{\dot{t_i}\}, \{\dot{v_i}\}, \dot{v}), \{c_i\}, (\{s_i\}, \tilde{s}, s, u)\rangle$ of the proof system proves this relationship.
- The matrix and $\{r_i\}$ in the above two statements are the same. The same part $\langle(\{\tilde{g_i}'\}, \tilde{g}', e'), \{c_i\}, (\{s_i\}, \tilde{s}, s)\rangle$, which is used to show the above two relationships, proves this statement.

4.2 Proof System

The proof system $(\mathcal{P}, \mathcal{V})$ proves that the prover $\mathcal{P}$ knows a permutation π such that there exist $r_1, ..., r_n \in \{0,1\}^{l_\eta}$ satisfying $e'_i = \gamma^{r_i} e_{\pi^{-1}(i)}$. The input to the proof system is $N, \theta, \gamma, \{e_i\}, \{e'_i\},\ i = 1, ..., n$. Suppose there is a publicly known set, $\{\tilde{g_i}\}_{i=1}^{n}$, of elements uniformly generated from $\mathcal{C}_{N,\gamma}$. Choose $M \in \mathbb{Z}_N$ such that $(p+q)/M$ is negligible. Let the permutation π be represented by a permutation matrix modulo N $(A_{ij})_{n\times n}$. The protocol is as follows:

1. $\mathcal{P}$ generates: $\alpha_i \leftarrow \mathbb{Z}_N, \alpha, \tilde{r_i}, \tilde{\alpha}, \delta_i, \rho, \rho_i, \tau, \tau_i \leftarrow \{0,1\}^{l_\eta},\ i = 1, ..., n$

2. $\mathcal{P}$ computes in mod N^2:

$$\tilde{g_i}' = \gamma^{\tilde{r_i}} \prod_{j=1}^{n} \tilde{g_j}^{A_{ji}},\ i = 1, ..., n$$

$$\tilde{g}' = \gamma^{\tilde{\alpha}} \prod_{j=1}^{n} \tilde{g_j}^{\alpha_j}$$

$$e' = \gamma^{\alpha} \prod_{j=1}^{n} e_j^{\alpha_j}$$

$$\dot{t_i} = \gamma^{\delta_i}(1 + N\sum_{j=1}^{n} 3\alpha_j A_{ji}),\ i = 1, ..., n$$

$$\dot{v_i} = \gamma^{\rho_i}(1 + N\sum_{j=1}^{n} 3\alpha_j^2 A_{ji}),\ i = 1, ..., n$$

$$\dot{v} = \gamma^{\rho}(1 + N\sum_{j=1}^{n} \alpha_j^3)$$

$$\dot{w_i} = \gamma^{\tau_i}(1 + N\sum_{j=1}^{n} 2\alpha_j A_{ji}),\ i = 1, ..., n$$

$$\dot{w} = \gamma^{\tau}(1 + N\sum_{j=1}^{n} \alpha_j^2)$$

3. $\mathcal{P} \longrightarrow \mathcal{V}$: $\{\tilde{g_i}'\}, \tilde{g}', e', \{\dot{t_i}\}, \{\dot{v_i}\}, \dot{v}, \{\dot{w_i}\}, \dot{w},\ i = 1, ..., n$

4. $\mathcal{P} \longleftarrow \mathcal{V}$: challenges $\{c_i\}_{(i=1,...,n)}$, $c_i \leftarrow \mathbb{Z}_M$

5. $\mathcal{P} \longrightarrow \mathcal{V}$:

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

$$\tilde{s} = \theta^{\sum_{i=1}^{n} \tilde{r_i}c_i + \tilde{\alpha}} \prod_{i=1}^{n} \tilde{g_i}^{d_i} \bmod N$$

$$s = \theta^{\sum_{i=1}^{n} r_i c_i + \alpha} \prod_{i=1}^{n} e_i^{d_i} \bmod N$$

$$u = \theta^{\sum_{i=1}^{n} \rho_i c_i + \sum_{i=1}^{n} \delta_i c_i^2 + \rho} \bmod N$$
$$v = \theta^{\sum_{i=1}^{n} \tau_i c_i + \tau} \bmod N$$

where $d_i = (\sum_{j=1}^{n} A_{ij} c_j + \alpha_i - s_i)/N$, $i = 1, ..., n$ (so d_i can only be 0 or 1)

6. $\mathcal{V}$ verifies in mod N^2:

$$\tilde{s}^N \prod_{j=1}^{n} \tilde{g_j}^{s_j} = \tilde{g}' \prod_{j=1}^{n} \tilde{g_j}'^{c_j} \tag{5}$$

$$s^N \prod_{j=1}^{n} e_j^{s_j} = e' \prod_{j=1}^{n} e_j'^{c_j} \tag{6}$$

$$u^N (1 + N \sum_{j=1}^{n} (s_j^3 - c_j^3)) = \dot{v} \prod_{j=1}^{n} \dot{v_j}^{c_j} \dot{t_j}^{c_j^2} \tag{7}$$

$$v^N (1 + N \sum_{j=1}^{n} (s_j^2 - c_j^2)) = \dot{w} \prod_{j=1}^{n} \dot{w_j}^{c_j} \tag{8}$$

4.3 Security

The proposed shuffle provides SP-CPA$_S$ and Verifiability as defined in [Nguyen et al. 2004]. Proofs are based on security proofs of the verifiable shuffle scheme in [Nguyen et al. 2004] and given in Appendix A.

Theorem 8. *The shuffle achieves Verifiability if the CFB assumption holds and output and input consist of valid Modified-Paillier ciphertexts.*

Theorem 9. *The shuffle achieves SP-CPA$_S$ if the DFB assumption holds.*

Theorem 10, which is the generalization of Theorem 4, is used to prove that the proposed verifiable shuffle scheme provides Verifiability even if the challenges c_i, $i = 1, ..., n$ are chosen from $\mathbb{Z}_M$ instead of $\mathbb{Z}_N$.

Theorem 10. *Let U be a subset of $\mathbb{Z}_N$. Consider a set $S_n = \{(1, c_1, ..., c_n) \mid (c_1, ..., c_n \in U) \wedge (\nexists Q_n \subseteq S_n : |Q_n| = n + 1 \wedge \langle Q_n \rangle_p = \mathbb{Z}_p^{n+1} \wedge \langle Q_n \rangle_q = \mathbb{Z}_q^{n+1})\}$ (that means S_n is the set of vectors $(1, c_1, ..., c_n)$, where $c_1, ..., c_n \in U$ and there does not exist any subset $Q_n \subseteq S_n$ of size $n + 1$ such that Q_n spans $\mathbb{Z}_p^{n+1}$ and $\mathbb{Z}_q^{n+1}$). Then $|S_n| \leq (p + q)|U|^{n-1}$.*

Proof. This proof is the same as the proof of Theorem 4, except that '$\mathbb{Z}_N$' in the proof of Theorem 4 is replaced by 'U' in this proof. The proof is shown as follows.

The theorem is proved by induction.

- $n = 1$: Consider a set of vectors $S_1 \subseteq \{(1, c)|c \in U\}$ satisfying $|S_1| > (p+q)$; and a vector $(1, c_1) \in S_1$. Consider a set $R_1 = \{(1, c_1 + kp \bmod N)|k \in \mathbb{Z}_q\} \cup \{(1, c_1 + kq \bmod N)|k \in \mathbb{Z}_p\}$. As $|R_1| = p + q - 1$, there exists $c'_1 \in U$ such that $(1, c'_1) \in S_1$ but $(1, c'_1) \notin R_1$. Then $Q_1 = \{(1, c_1), (1, c'_1)\}$ satisfies $(|Q_1| = 2) \wedge (\langle Q_1\rangle_p = \mathbb{Z}_p^2) \wedge (\langle Q_1\rangle_q = \mathbb{Z}_q^2)$.

- Suppose the theorem holds for n. We prove it is also true for $n + 1$. Let a set $S_{n+1} = \{(1, c_1, ..., c_{n+1})|(c_1, ..., c_{n+1} \in U) \wedge (\nexists Q_{n+1} \subseteq S_{n+1} : |Q_{n+1}| = n + 2 \wedge \langle Q_{n+1}\rangle_p = \mathbb{Z}_p^{n+2} \wedge \langle Q_{n+1}\rangle_q = \mathbb{Z}_q^{n+2})\}$. Consider $S'_n = \{(1, c_1, ..., c_n) \mid \exists c_{n+1} \in U : (1, c_1, ..., c_n, c_{n+1}) \in S_{n+1}\}$, there are two possibilities:
  1. If $\nexists Q'_n \subseteq S'_n : |Q'_n| = n + 1 \wedge \langle Q'_n\rangle_p = \mathbb{Z}_p^{n+1} \wedge \langle Q'_n\rangle_q = \mathbb{Z}_q^{n+1}$, then $|S'_n| \le (p + q)|U|^{n-1}$, as the theorem holds for n. So $|S_{n+1}| \le |S'_n||U| \le (p + q)|U|^n$.
  2. If $\exists Q'_n \subseteq S'_n : |Q'_n| = n + 1 \wedge \langle Q'_n\rangle_p = \mathbb{Z}_p^{n+1} \wedge \langle Q'_n\rangle_q = \mathbb{Z}_q^{n+1}$, select a set T of $n + 1$ vectors $(1, c_{i1}, ..., c_{i(n+1)}) \in S_{n+1}$, $i = 1, ..., n + 1$ such that $Q'_n = \{(1, c_{i1}, ..., c_{in})\}$

     Let $d = det \begin{pmatrix} 1 & c_{11} & ... & c_{1n} \\ .. & .. & .. & .. \\ 1 & c_{(n+1)1} & ... & c_{(n+1)n} \end{pmatrix} \bmod N$, then $gcd(d, N) = 1$, so $d^{-1} \bmod N$ exists.

     For each vector $x = (1, x_1, ..., x_{n+1}) \in S_{n+1}$ (including those in T), let

     $$d_x = det \begin{pmatrix} 1 & c_{11} & ... & c_{1(n+1)} \\ .. & .. & .. & .. \\ 1 & c_{(n+1)1} & ... & c_{(n+1)(n+1)} \\ 1 & x_1 & ... & x_{n+1} \end{pmatrix} = dx_{n+1} - F(x_1, ..., x_n) \bmod N$$

     for some function F. The conditions of S_{n+1} lead to either $d_x = 0 \bmod p$ or $d_x = 0 \bmod q$.

     Suppose $d_x = 0 \bmod p$, then $x_{n+1} = d^{-1}F(x_1, ..., x_n) \bmod p$, so the number of possible vectors $x = (1, x_1, ..., x_{n+1})$ is no more than $q|U|^n$. Similarly for the case $d_x = 0 \bmod q$, the number of possible vectors $x = (1, x_1, ..., x_{n+1})$ is no more than $p|U|^n$ and so $|S_{n+1}| \le (p + q)|U|^n$.

4.4 Efficiency

Theorem 10 allows c_i, $i = 1, ..., n$ to be chosen in $\mathbb{Z}_M$, which is much smaller than $\mathbb{Z}_N$ as required in the original verifiable shuffle scheme in [Nguyen et al. 2004]. This reduces the cost of exponentiations to the exponents c_i, $i = 1, ..., n$. Following [Furukawa and Sako 2001] and using computation techniques such as the fixed-based comb method and the simultaneous multiple exponentiation algorithm [Menezes et al. 1997], the number of exponentiations can be substantially reduced. The following table summarizes the result of efficiency comparison between some of the most well-known shuffle schemes.

Verifiable Shuffles	Number of (No.) Exponentiations	No. Exponentiations, efficient techniques	No. Rounds
Neff	$23n$	$6.3n$	7
Furukawa-Sako	$18n$	$4.8n$	3
Groth	$12n$	$3.5n$	7
Modified-Paillier	$13n$	$3.4n$	3

Table 1: Efficiency Comparison of Verifiable Shuffle schemes

5 A Robust Mix-net based on the Modified-Paillier Public-key System

5.1 Overview

The main motivation for analysing and constructing verifiable shuffles is to construct *robust mix-nets* that consist of the following polynomially bounded participants. *Users* send ciphertexts to the mix-net. Each *mix-server* (also mix-centre) is implemented as a verifiable shuffle. It takes as input a list of ciphertexts and outputs a permuted list of the re-encrypted ciphertexts to the next mix server. *Decryption servers* collaboratively decrypt the list of ciphertexts output by the last mix-server. A *verifier* verifies correctness of the mix-net operation. All communication is assumed accessible by all mix-centres.

Inputs to a mix-net must be encrypted by an NM-CCA encryption scheme [Jakobsson 1998]. Otherwise, an adversary can trace an input ciphertext ci by creating another input ciphertext ci' whose plaintext is related to ci's plaintext in a known manner and checking the mix-net's output for plaintexts that satisfy the relationship. An example of this attack is shown in [Pfitzmann 1994] against the mix-net in [Park et al. 1993]. It is also desirable to distribute the decryption ability, so that a minimum number of decryption servers, the *threshold*, is needed to decrypt the ciphertexts. The decryption process should also be *robust* that means the corrupted decryption servers should not be able to prevent uncorrupted ones from correctly decrypting the ciphertexts. In short, an *NM-CCA robust threshold* encryption scheme is required.

5.2 Model

(Set up) There are t mix servers, $S_1, \cdots, S_t$, one or more decryption servers and a verifier $\mathcal{V}$. Each mix server is implemented by a verifiable shuffle that shuffles its input list and proves the correctness of its operation. If the proof succeeds, the shuffle's output will be used as the input to the next shuffle. Otherwise, the next shuffle uses the previous shuffle's input.

The shuffle uses an NM-CCA robust threshold version of the Modified-Paillier encryption scheme, which is constructed similar to the NM-CCA robust threshold version of the Paillier encryption scheme
[Fouque and Pointcheval 2001]. A ciphertext encrypted using this scheme has the form (e, aux), where e is the normal Modified-Paillier ciphertext and aux allows the ciphertext to be non-malleable.

The secret key $sk = \lambda$, where $\lambda = lcm(p-1, q-1)$, is shared among the decryption servers. Note that the secret key is λ instead of η as in the Modified-Paillier public-key system. The reason will be explained in the security analysis of the mix-net.

(Operations)

1. Users send the first mix-server ciphertexts encrypted by the public key of the mix-net (the NM-CCA robust threshold version of the Modified-Paillier encryption scheme). An input ciphertext (e, aux) needs to pass non-malleability test by the verifier before sub-ciphertext e is taken to the first mix-server. Suppose $L_0 = (c_1, \cdots, c_n)$ is a list of those sub-ciphertexts taken to the first mix-server.

2. Each S_i in turn computes a randomly permuted and re-encrypted list $L_i = (a'_{\tau_i(1)}, \cdots, a'_{\tau_i(n)})$ from $L_{i-1} = (a_1, \cdots, a_n)$, where a'_i is a re-encryption of a_i and τ_i is a secretly chosen random permutation on $\{1, \cdots, n\}$, and then outputs L_i. S_i runs a proof system $(\mathcal{P}, \mathcal{V})$ to prove that L_i is a permutation of re-encryptions of elements in L_{i-1}.
In case the proof does not succeed, S_i is excluded from the mix-net operation. If $i \neq t$, the mix-centre S_{i+1} that receives the output of the corrupted mix-centre S_i, will instead use S_i's input list as its input, effectively disregarding S_i. If $i = t$, S_i's input list will be sent to the decryption servers.

3. The decryption servers jointly decrypt ciphertexts, which are sent from the mix-centres, in a robust way and output a list of messages $L_{out} = (m_{\phi(1)}, \cdots, m_{\phi(n)})$, where $\phi = \tau_t \cdots \tau_1$ and m_i is a plaintext of c_i.

5.3 Security

As a formal security model for mix-nets has not been well defined, we can only give an informal statement on our mix-net's security.

Robustness and public verifiability of the mix-net result from verifiability of its shuffles (mix-centres). However, as stated in Theorem 8, verifiability of the shuffle depend on the condition that its output and input consist of only valid Modified-Paillier ciphertexts. In case this condition does not hold, the output and input ciphertexts are still valid Paillier ciphertexts. And we show that the shuffle's proof system still proves that the output is a permutation of Paillier re-encryptions of its input ciphertexts. This is stated in Theorem 11 and its proof is actually one part of the proof for Theorem 8 that can be found in Appendix A. Theorem 11 implies robustness, and as any honest party can be the verifier, the mix-net achieves public verifiability.

Theorem 11. *Assuming the CFB assumption holds, if the proof system of a mix-centre accepts with non-negligible probability, then its output is a permutation of Paillier re-encryptions of its input ciphertexts.*

The mix-net's privacy relies on its shuffles' SP-CPA$_S$ and the following assumption, which states the indistinguishability between a Paillier ciphertext and a Modified-Paillier ciphertext.

Decisional Paillier Ciphertext (DPC) Assumption: *Suppose N, θ and γ are generated and $\mathcal{C}_{N,\gamma}$ is defined as in the CFB assumption's definition. The Decisional Paillier Ciphertext problem is defined as follows: given (N, θ, γ), distinguish between a uniform distribution on the set $\mathcal{C}_{N,\gamma}$ and a uniform distribution on the set $\mathbb{Z}^*_{N^2}$. The Decisional Paillier Ciphertext assumption states that the Decisional Paillier Ciphertext problem is computationally difficult.*

Finally, an informal statement on our mix-net's security is as follows. The mix-net system provides robustness and public verifiability under the CFB assumption. The mix-net system provides privacy under the DFB and DPC assumptions.

5.4 NM-CCA robust threshold encryption scheme

To improve efficiency of the NM-CCA robust threshold encryption scheme, instead of using the twin-encryption paradigm as in
[Fouque and Pointcheval 2001], we may combine SS-CPA encryption and proof of knowledge to provide NM-CCA. However, there is no known scheme that combines the Paillier encryption scheme and proof of knowledge to provide NM-CCA (even in the random oracle model). A combination of the El Gamal encryption scheme and the Schnorr proof system has been proved to provide NM-CCA in the random oracle model but the proof requires either another strong assumption [Tsiounis and Yung 1998] or is in the generic model
[Schnorr and Jakobsson 2000]. Abe [Abe 2004] showed an approach of combining encryption and proof of knowledge to achieve NM-CCA, but the construction

does not preserve the homomorphic property which is essential for applications to mix-nets. (For robustness and the threshold property, the same method in [Fouque and Pointcheval 2001] can be used.)

The following alteration of the Paillier and Modified-Paillier schemes combines encryption with proof of knowledge and maintains the homomorphic property. Let $\mathcal{E}^{pa}$ and $\mathcal{E}^{mo}$ denote the Paillier and Modified-Paillier encryption algorithms, respectively. For the Paillier scheme, the ciphertext of a message $m \in \mathbb{Z}_N$ is (e, c, s, r_s), where $e = \mathcal{E}^{pa}(r, m)$, $c = \mathcal{H}(e, \mathcal{E}^{pa}(r_w, w))$, $s = mc + w \bmod N$, $r_s = r^c r_w \bmod N$; $\mathcal{H}$ is a hash function $\mathcal{H} : \{0,1\}^* \rightarrow \mathbb{Z}_N$ and $w \leftarrow \mathbb{Z}_N$, $r, r_w \leftarrow \mathbb{Z}_N^*$. Similar for the Modified-Paillier scheme, a ciphertext of message $m \in \mathbb{Z}_N$ is (e, c, s, r_s), where $g = \mathcal{E}^{mo}(r, m)$, $c = \mathcal{H}(e, \mathcal{E}^{mo}(r_w, w))$, $s = mc + w \bmod N$, $r_s = rc + r_w$ and $w \leftarrow \mathbb{Z}_N$, $r \leftarrow \{0,1\}^{l_\eta}$, $r_w \leftarrow \mathbb{Z}$. Validity of a ciphertext can be verified by checking whether $c \overset{?}{=} \mathcal{H}(e, \mathcal{E}^{pa}(r_s, s)/e^c)$ (or $c \overset{?}{=} \mathcal{H}(e, \mathcal{E}^{mo}(r_s, s)/e^c)$ respectively). Intuitively, e is the normal ciphertext and c, s and r_s show that the ciphertext has been encrypted by someone with the knowledge of r and m. However, proving that these combinations provide NM-CCA remains a challenge.

6 Conclusion

In this paper, we proposed a variant of the Paillier encryption scheme that reduces computation costs of encryption, re-encryption and decryption while still preserving the homomorphic property. We then presented a verifiable shuffle system based on the Modified-Paillier public-key encryption system, proved its security and compared its performance with other efficient shuffle systems. We finally used the shuffle to construct a robust mix-net. An interesting future direction is to construct a "Universal Re-encryption" scheme for Mix-nets [Golle et al. 2004], based on the Modified-Paillier system.

Acknowledgements: We would like to thank an anonymous reviewer for ACNS '04 for useful comments that help us figure out the flaw in Paillier's variant scheme.

References

[Abe 1999] Abe, M.: "Mix-networks on permutation networks"; Proc. ASIACRYPT '99, Springer-Verlag, LNCS 1716 (1999), 258-273.

[Abe and Hoshino 2001] Abe, M., Hoshino, F.: "Remarks on Mix-Network Based on Permutation Networks"; PKC'01, Springer-Verlag, LNCS (2001), 317-324.

[Abe 2004] Abe, M.: "Combining Encryption and Proof of Knowledge in the Random Oracle Model"; Computer Journal, 47, 1 (2004). To appear.

[Boneh and Golle 2002] Boneh, D., Golle, P.: "Almost Entirely Correct Mixing with Application to Voting". Proc. ACM CCS'02, ACM Press (2002).

[Brands 1993] Brands, S.: “An efficient off-line electronic cash system based on the representation problem”; CWI Technical Report CS-R9323, (1993).

[Chaum 1981] Chaum, D.: “Untraceable electronic mail, return addresses, and digital pseudonyms”; Communications of the ACM, 24, 2 (1981), 84-88.

[Choi and Kim 2003] Choi, S. and Kim, K.: “Authentication and Payment Protocol Preserving Location Privacy in Mobile IP”; GLOBECOM’03, San Francisco (2003).

[Desmedt and Kurosawa 2000] Desmedt, Y., Kurosawa. K.: “How to break a practical mix and design a new one”; Proc. EUROCRYPT ’00, Springer-Verlag, LNCS 1807 (2000), 557-572.

[Fouque and Pointcheval 2001] Fouque, P., Pointcheval D.: “Threshold Cryptosystems Secure against Chosen-Ciphertext Attacks”; Proc. ASIACRYPT’01, Springer-Verlag, LNCS 2248 (2001), 351-368.

[Furukawa and Sako 2001] Furukawa, J., Sako K.: “An Efficient Scheme for Proving a Shuffle”; Proc. CRYPTO’01, Springer-Verlag, LNCS 2139 (2001), 368-389.

[Furukawa et al. 2002] Furukawa, J., Miyauchi, H., Mori, K., Obana, S., Sako, K.: “An Implementation of a Universally Verifiable Electronic Voting Scheme based on Shuffling”; Financial Cryptography’02 (2002).

[Furukawa 2004] Furukawa, J.: “Efficient, Verifiable Shuffle Decryption and Its Requirement of Unlinkability”; PKC’04.

[Gabber et al. 1997] Gabber, E., Gibbons, P., Matias, Y., Mayer, A. “How to make personalized Web browsing simple, secure, and anonymous”; Financial Cryptography’97 (1997), 17-31.

[Goldreich 2001] Goldreich, O.: “Foundations of Cryptography Basic Tools”; Cambridge University Press (2001).

[Goldreich 2004] Goldreich, O.: “Foundations of Cryptography, Basic Applications”; Cambridge University Press (2004).

[Golle et al. 2004] Golle, P., Jakobsson, M., Juels, A., Syverson, P.: “Universal Reencryption for Mixnets”; Proc. RSA Conference Cryptographers’ Track ’04, Springer-Verlag, LNCS 2964 (2004), 163-178.

[Golle et al. 2002] Golle, P., Zhong, S., Boneh, D., Jakobsson, M., Juels, A.: “Optimistic Mixing for Exit-Polls”; Proc. ASIACRYPT’02, Springer-Verlag, LNCS 2501 (2002), 451-465.

[Groth 2003] Groth, J.: “A Verifiable Secret Shuffle of Homomorphic Encryptions”; Proc. PKC’03, Springer-Verlag, LNCS 2567 (2003), 145-160.

[Jakobsson 1998] Jakobsson, M.: “A practical mix”; Proc. EUROCRYPT’98, Springer-Verlag, LNCS 1403 (1998), 448-461.

[Jakobsson and M’Raihi 1998] Jakobsson, M., M’Raihi, D.: “Mix-based electronic payments”; Proc. SAC ’98, Springer-Verlag, LNCS 1505 (1998), 057-473.

[Jakobsson 1999] Jakobsson, M.: “Flash mixing”; Proc. PODC ’99, ACM (1999), 83-89.

[Jakobsson and Juels 1999] Jakobsson, M., Juels, A.: “Millimix: Mixing in small batches”; DIMACS Technical Report 99-33 (1999).

[Jakobsson and Juels 2000] Jakobsson, M., Juels, A.: “Mix and match: Secure function evaluation via ciphertexts”; Proc. ASIACRYPT’00, Springer-Verlag, LNCS 1976 (2000), 162-177.

[Jakobsson and Juels 2001] Jakobsson, M., Juels, A. “An Optimally Robust Hybrid Mix Network”, Proc. PODC ’01, ACM (2001).

[Jakobsson et al. 2002] Jakobsson, M., Juels, A., Rivest, R.: “Making Mix Nets Robust For Electronic Voting By Randomized Partial Checking”; USENIX Security ’02 (2002).

[Juels 2001] Juels, A.: “Targeted advertising and privacy too”; Proc. RSA-CT’01 (2001).

[Kong and Hong 2003] Kong, J.,Hong, X.: “ANODR: ANonymous On Demand Routing with Untraceable Routes for Mobile Ad-hoc Networks”; MobiHoc’03, ACM (2003), 291-302.

[Menezes et al. 1997] Menezes, A., van Oorschot, P., Vanstone, S.: "Handbook of Applied Cryptography"; CRC Press (1996).

[Mitomo and Kurosawa 2000] Mitomo, M., Kurosawa, K.: "Attack for flash mix"; Proc. ASIACRYPT'00, Springer-Verlag, LNCS 1976 (2000), 192-204.

[Naor and Yung 1990] Naor, M., Yung, M.: "Public-Key Cryptosystems Provably Secure against Chosen Ciphertexts Attacks"; Proc. STOC'90, ACM Press (1990), 427-437.

[Neff 2001] Neff, A.: "A verifiable secret shuffle and its application to e-voting"; Proc. ACM CCS '01, ACM Press (2001), 116-125.

[Neff 2003] Neff, A.: "Verifiable Mixing (Shuffling) of ElGamal Pairs"; (2003) appeared as electronic version, http://www.votehere.org/vhti/documentation/egshuf.pdf.

[Nguyen and Safavi-Naini 2003] Nguyen, L., Safavi-Naini, R.: "Breaking and Mending Resilient Mix-nets"; Proc. PET'03, Springer-Verlag, LNCS 2760 (2003), 66-80.

[Nguyen et al. 2004] Nguyen, L., Safavi-Naini, R., Kurosawa, K.: "Verifiable Shuffles: A Formal Model and a Paillier-based Efficient Construction with Provable Security"; Proc. ACNS'04 (Second Conference of Applied Cryptography and Network Security), Springer-Verlag, LNCS (2004). To appear.

[Nguyen and Safavi-Naini 2004] Nguyen, L., Safavi-Naini, R.: "An Efficient Verifiable Shuffle with Perfect Zero-knowledge Proof System"; Cryptographic Algorithms and their Uses (Eracom 2004). To appear.

[Ogata et al. 1997] Ogata, W., Kurosawa, K., Sako, K., Takatani, K.: "Fault tolerant anonymous channel"; Proc. ICICS'97, Springer-Verlag, LNCS 1334 (1997), 440-444.

[Ohkubo and Abe 2000] Ohkubo, M., Abe, M.: "A length-invariant hybrid mix"; Proc. ASIACRYPT'00, Springer-Verlag, LNCS 1976 (2000), 178-191.

[Paillier 1999] Paillier, P.: "Public-Key Cryptosystems Based on Composite Degree Residuosity Classes"; Proc. EUROCRYPT'99, Springer-Verlag, LNCS 1592 (1999).

[Park et al. 1993] Park, C., Itoh, K., Kurosawa, K.: "Efficient anonymous channel and all/nothing election scheme"; Proc. EUROCRYPT '93, Springer-Verlag, LNCS 765 (1993), 248-259.

[Pfitzmann 1994] Pfitzmann, B.: "Breaking an Efficient Anonymous Channel"; Proc. EUROCRYPT '94, Springer-Verlag, LNCS 950 (1995), 332-340.

[Schnorr and Jakobsson 2000] Schnorr, P., Jakobsson, M.: "Security of signed El Gamal encryption"; Proc. ASIACRYPT'00, Springer-Verlag, LNCS 1976 (2000), 73-89.

[Shamir 1979] Shamir, A.: "How to Share a Secret"; Communications of the ACM, 22 (1979), 612-613.

[Tsiounis and Yung 1998] Tsiounis, Y., Yung, M.: "On the security of El Gamal based encryption"; Proc. PKC'98, Springer-Verlag, LNCS 1431 (1998), 117-134.

[Wikstrom 2002] Wikstrom, D.: "The security of a mix-center based on a semantically secure cryptosystem"; Proc. INDOCRYPT'02, Springer-Verlag, LNCS 2551 (2002), 368-381.

[Wikstrom 2003] Wikstrom, D.: "Five Practical Attacks for "Optimistic Mixing for Exit-Polls""; Proc. SAC'03, Springer-Verlag, LNCS (2003).

A Security Proofs

A.1 Proof of Theorem 8 for Verifiability

In the proof system, based on the CFB assumption, it is computationally difficult for the prover to compute p and q. Hence, for any matrix $(A_{ij})_{n\times n}$ the prover can generate, $gcd(A_{ij}, N)$ is different from p and q. Therefore, based on Theorem 3, the objective of the proof system can be re-stated as follows. The common input to the proof system includes $N, \theta, \gamma, \{e_i\}, \{e'_i\},\ i = 1, ..., n$. The auxiliary input to the prover $\mathcal{P}$ includes permutation π and $r_1, ..., r_n \in \{0,1\}^{l_\eta}$ satisfying $e'_i = \gamma^{r_i} e_{\pi^{-1}(i)}$ and does not include the secret key $sk = \eta$. The proof system $(\mathcal{P}, \mathcal{V})$ proves that $\mathcal{P}$ knows a matrix $(A_{ij})_{n\times n}$ such that equations (2) and (3) hold and there exist $r_1, ..., r_n \in \{0,1\}^{l_\eta}$ satisfying

$$e'_i = \gamma^{r_i} \prod_{j=1}^{n} e_j^{A_{ji}},\ i = 1, ..., n \tag{9}$$

Based on the definition of Verifiability, Theorem 8 can be concluded from Theorem 12 and Theorem 13, which state the Completeness and Soundness properties of the proof system. We also need Theorem 10 to prove Theorem 13. Theorems 12 and 13 are presented and proved as follows.

Theorem 12. *(Completeness) If $\mathcal{P}$ knows a matrix (A_{ij}) such that there exist $r_1, ..., r_n \in \{0,1\}^{l_\eta}$ satisfying equations (2), (3) and (9), and $\mathcal{P}$ also performs correctly in the protocol, then $\mathcal{V}$ always accepts.*

Proof. Suppose $\mathcal{P}$ knows a matrix (A_{ij}) such that there exist $r_1, ..., r_n \in \{0,1\}^{l_\eta}$ satisfying equations (2), (3) and (9); and $\{\tilde{g_i}'\}$, $\tilde{g}'$, g', $\{\dot{t_i}\}$, $\{\dot{v_i}\}$, $\dot{v}$, $\{\dot{w_i}\}$, $\dot{w}$, $\{c_i\}$, $\{s_i\}$, $\tilde{s}$, s, u, v for $i = 1, ..., n$ are generated as specified in the protocol. Then the verifier outputs accept, as the following equations hold.

- $\tilde{s}^N \prod_{j=1}^n \tilde{g_j}^{s_j} = (\theta^{\sum_{i=1}^n \tilde{r_i} c_i + \tilde{\alpha}} \prod_{i=1}^n \tilde{g_i}^{d_i})^N \prod_{j=1}^n \tilde{g_j}^{\sum_{i=1}^n A_{ji} c_i + \alpha_j}$ $= (\gamma^{\tilde{\alpha}} \prod_{j=1}^n \tilde{g_j}^{\alpha_j}) \prod_{i=1}^n (\gamma^{\tilde{r_i}} \prod_{j=1}^n \tilde{g_j}^{A_{ji}})^{c_i} = \tilde{g}' \prod_{i=1}^n \tilde{g_i}'^{c_i}$.
- $s^N \prod_{j=1}^n e_j^{s_j} = (\theta^{\sum_{i=1}^n r_i c_i + \alpha} \prod_{i=1}^n e_i^{d_i})^N \prod_{j=1}^n e_j^{\sum_{i=1}^n A_{ji} c_i + \alpha_j}$ $= (\gamma^{\alpha} \prod_{j=1}^n e_j^{\alpha_j}) \prod_{i=1}^n (\gamma^{r_i} \prod_{j=1}^n e_j^{A_{ji}})^{c_i} = g' \prod_{i=1}^n e_i'^{c_i}$.
- $u^N (1 + N \sum_{j=1}^n (s_j^3 - c_j^3)) = (\theta^{\sum_{i=1}^n \rho_i c_i + \sum_{i=1}^n \delta_i c_i^2 + \rho})^N (1+$ $N \sum_{j=1}^n ((\sum_{i=1}^n A_{ji} c_i + \alpha_j)^3 - c_j^3)) = \gamma^\rho (1 + N \sum_{j=1}^n \alpha_j^3) \prod_{i=1}^n (\gamma^{\rho_i} (1+$ $N \sum_{j=1}^n 3\alpha_j^2 A_{ji}))^{c_i} \prod_{i=1}^n (\gamma^{\delta_i} (1 + N \sum_{j=1}^n 3\alpha_j A_{ji}))^{c_i^2} = \dot{v} \prod_{i=1}^n \dot{v_i}^{c_i} \dot{t_i}^{c_i^2}$.
- $v^N (1 + N \sum_{j=1}^n (s_j^2 - c_j^2)) = (\theta^{\sum_{i=1}^n \tau_i c_i + \tau})^N (1 + N \sum_{j=1}^n ((\sum_{i=1}^n A_{ji} c_i + \alpha_j)^2 -$ $c_j^2)) = \gamma^\tau (1 + N \sum_{j=1}^n \alpha_j^2) \prod_{i=1}^n (\gamma^{\tau_i} (1 + N \sum_{j=1}^n 2\alpha_j A_{ji}))^{c_i} = \dot{w} \prod_{i=1}^n \dot{w_i}^{c_i}$.

Theorem 13. *(Soundness) Suppose the output and input of the shuffle consist of valid Modified-Paillier ciphertexts. Under the CFB assumption, if $\mathcal{V}$ accepts with non-negligible probability, then $\mathcal{P}$ knows a matrix (A_{ij}) such that there exist $r_1, ..., r_n \in \{0,1\}^{l_\eta}$ satisfying equations (2), (3) and (9).*

Proof. We first prove Theorem 11. It can be proved in the same way as the Soundness proof for Verifiability of the Nguyen et al. shuffle scheme in [Nguyen et al. 2004] (in the proof of Theorem 5 in that paper), except that the

CFB assumption and Theorem 10 (in the case $U = \mathbb{Z}_M$) are used instead of the CCR assumption and Theorem 4, respectively. Intuitively, the similarity is due to the fact that the Nguyen et al. proof system and this proof system have the same messages, which include a commitment $(\{\tilde{g}_i{}'\}, \tilde{g}', e', \{\dot{t_i}\}, \{\dot{v_i}\}, \dot{v}, \{\dot{w_i}\}, \dot{w})$, a challenge $\{c_i\}$ and a response $(\{s_i\}, \tilde{s}, s, u, v)$. And in both proof systems, the verifier needs to check the same equations, i.e. equations (5), (6), (7) and (8).

We have $\{e_i\}$ and $\{e'_i\}$ are valid Modified-Paillier ciphertexts. Thus, if $\{e'_i\}$ is a permutation of Paillier re-encryptions of $\{e_i\}$, then there exist $r_1, ..., r_n \in \{0,1\}^{l_\eta}$ satisfying $e'_i = \gamma^{r_i} \prod_{j=1}^{n} e_j^{A_{ji}}$, $i = 1, ..., n$ (which is equation (9)). Therefore, Theorem 13 has been proved.

A.2 Proof of Theorem 9 for Privacy

As SP-CPA$_S$ and IND-CPA$_S$ are equivalent [Nguyen et al. 2004], proving Theorem 9 is equivalent to proving Theorem 16 below. We need Definition 14 and Lemma 15 to prove Theorem 16.

Definition 14. Let R'_m be the set of m-element tuples where all elements are in $\mathcal{C}_{N,\gamma}$ and let $D'_m \subset R'_m$ be the set of m-element tuples where all elements are in $\mathcal{C}^0_{N,\gamma}$. The DFB$_m$ problem is defined as the problem of distinguishing instances uniformly chosen from R'_m and those uniformly chosen from D'_m. The DFB$_m$ assumption states that the DFB$_m$ problem is computationally difficult.

Lemma 15. *For any $m \geq 1$, the DFB$_m$ assumption holds if the DFB assumption holds.*

Proof. We prove the lemma by induction. We prove that if either the DFB assumption holds or the DFB$_{m-1}$ assumption holds, then the DFB$_m$ assumption holds.

We define the subset M'_m of R'_m to be the set of tuples $I = (x_1, ..., x_m)$ such that $x_1, ..., x_{m-1} \in \mathcal{C}^0_{N,\gamma}$ and $x_m \in \mathcal{C}_{N,\gamma}$. Hence, D'_m is a subset of M'_m.

If the DFB$_m$ problem is easy, then we can either distinguish between instances chosen uniformly from R'_m and M'_m or distinguish between instances chosen uniformly from M'_m and D'_m. In the former case, it means the DFB$_{m-1}$ problem is easy. In the following, we show that in the latter case, the DFB problem is easy.

For any $I_1 = (x) \in R'_1$, we generate a tuple $I_m \in R'_m$ as
$I_m = (\gamma^{r_1}, \gamma^{r_2}, ..., \gamma^{r_{m-1}}, x)$ where $r_i \leftarrow \{0,1\}^{l_\eta}$. If I_1 is chosen uniformly from D'_1, then I_m is distributed uniformly in D'_m. And if I_1 is chosen uniformly from R'_1, then I_m is distributed uniformly in M'_m. Therefore, if D'_m and M'_m are distinguishable, then the DFB problem is easy.

Theorem 16. *The shuffle provides IND-CPA$_S$ if the DFB assumption holds.*

Proof. The reader can refer to [Nguyen et al. 2004] for definitions of IND-CPA$_S$ and explanations of the notations in this proof.

Suppose there is a publicly known set, $\{\tilde{g_i}\}_{i=1}^n$, of elements uniformly generated from $\mathcal{C}_{N,\gamma}$. And suppose the challenge template includes two permutations $\pi_{(1)}, \pi_{(2)} \in T_n$, a list of ciphertexts $L_{in} = (e_1, ..., e_n)$, the list of corresponding plaintexts $L_{in}^{(p)}$ and the corresponding probabilistic inputs $C_{E_{pk}}^{L_{in}^{(p)}, L_{in}}$. The actual challenge $o^{\pi_{(k)}}$, which is randomly generated by using $\pi_{(k)}$ ($k = 1$ or 2) and is given to the adversary, includes L_{in}, $L_{in}^{(p)}$, $C_{E_{pk}}^{L_{in}^{(p)}, L_{in}}$, a list of re-encrypted ciphertexts $L_{out} = (e'_1, ..., e'_n)$ and $View_{\mathcal{V}}^{\mathcal{P}}(pk, L_{in}, L_{out}) = (\{\tilde{g_i}\}, \{\tilde{g_i}'\}, \tilde{g}', g', \{\dot{t_i}\}, \{\dot{v_i}\}, \{\dot{w_i}\}, \dot{v}, \dot{w}, \{c_i\}, \{s_i\}, \tilde{s}, s, u, v)$ Let $\mathcal{O}^{\pi_{(k)}}$ be the set of all possible $o^{\pi_{(k)}}$.

Let $\mathcal{O}_g$ be the set of all tuples o_g, each of which includes L_{in}, $L_{in}^{(p)}$, $C_{E_{pk}}^{L_{in}^{(p)}, L_{in}}$, a list of random Modified-Paillier ciphertexts $L_{out} = (e'_1, ..., e'_n)$, the set $\{\tilde{g_i}\}$, a tuple $(\{\tilde{g_i}'\}_{i=1}^n, \{\dot{t_i}\}_{i=1}^n, \{\dot{v_i}\}_{i=1}^n, \{\dot{w_i}\}_{i=1}^n)$ of randomly generated elements of $\mathcal{C}_{N,\gamma}$, a set $\{c_i\}_{i=1}^n$ of randomly generated elements of $\mathbb{Z}_M$, a set $\{s_i\}_{i=1}^n$ of randomly generated elements of $\mathbb{Z}_N$, a tuple $(\tilde{s}, s, u, v)$ of randomly generated elements of $\mathbb{Z}_N^*$ and $\tilde{g}', g', \dot{v}, \dot{w}$ satisfying:

$$\tilde{g}' = \gamma^{\tilde{s}} \prod_{j=1}^{n} \tilde{g_j}^{s_j} \tilde{g_j}'^{-c_j} \tag{10}$$

$$g' = \gamma^{s} \prod_{j=1}^{n} e_j^{s_j} e_j'^{-c_j} \tag{11}$$

$$\dot{v} = \gamma^{u} (1 + N \sum_{j=1}^{n} (s_j^3 - c_j^3)) \prod_{j=1}^{n} \dot{v_j}^{-c_j} \dot{t_j}^{-c_j^2} \tag{12}$$

$$\dot{w} = \gamma^{v} (1 + N \sum_{j=1}^{n} (s_j^2 - c_j^2)) \prod_{j=1}^{n} \dot{w_j}^{-c_j} \tag{13}$$

We first prove that if the DFB$_{5n}$ assumption holds, then the actual challenge $o^{\pi_{(k)}}$ uniformly chosen from $\mathcal{O}^{\pi_{(k)}}$ is computationally indistinguishable from a tuple o_g uniformly chosen from $\mathcal{O}_g$.

We show that from an element

$$I = (h_1, .., h_n, \tilde{h_1}, .., \tilde{h_n}, \overline{t_1}, .., \overline{t_n}, \overline{v_1}, .., \overline{v_n}, \overline{w_1}, .., \overline{w_n})$$

of D'_{5n} or R'_{5n} (see Definition 14), we can generate a random element o_r of $\mathcal{O}^{\pi_{(1)}}$ or $\mathcal{O}_g$ as follows. Choose $\{c_i\}_{i=1}^n$ uniformly from $\mathbb{Z}_M$, $\{s_i\}_{i=1}^n$ uniformly from $\mathbb{Z}_N$ and $\tilde{s}, s, u, v$ uniformly from $\mathbb{Z}_N^*$. Compute

$$\alpha_i = s_i - c_{\pi_{(1)}(i)} \bmod N,\ i = 1, ..., n$$
$$e'_i = h_i e_{\pi_{(1)}^{-1}(i)},\ i = 1, ..., n$$

$$
\begin{aligned}
\tilde{g_i}' &= \tilde{h_i}\tilde{g}_{\pi_{(1)}^{-1}(i)},\ i=1,...,n \\
\dot{t_i} &= \overline{t_i}(1+N3\alpha_{\pi_{(1)}^{-1}(i)}),\ i=1,...,n \\
\dot{v_i} &= \overline{v_i}(1+N3\alpha^2_{\pi_{(1)}^{-1}(i)}),\ i=1,...,n \\
\dot{w_i} &= \overline{w_i}(1+N2\alpha_{\pi_{(1)}^{-1}(i)}),\ i=1,...,n
\end{aligned}
$$

And compute $\tilde{g}', g', \dot{v}, \dot{w}$ as in equations (10), (11), (12) and (13). We have o_r=$(L_{in}, L_{in}^{(p)}, C_{E_{pk}}^{L_{in}^{(p)},L_{in}}, (e'_1, ..., e'_n), (\{\tilde{g_i}\}, \{\tilde{g_i}'\}, \tilde{g}', g', \{\dot{t_i}\}, \{\dot{v_i}\}, \{\dot{w_i}\}, \dot{v}, \dot{w}, \{c_i\}, \{s_i\}, \tilde{s}, s, u, v))$.

Then $o_r \in \mathcal{O}^{\pi_{(1)}}$ if and only if $I \in D'_{5n}$, and $o_r \in \mathcal{O}_g$ if and only if $I \in R'_{5n}$. So, if the DFB$_{5n}$ assumption holds, then a random element of $\mathcal{O}^{\pi_{(1)}}$ is computationally indistinguishable from a random element of $\mathcal{O}_g$, and so is from a random element of $\mathcal{O}^{\pi_{(2)}}$.

Therefore, if the DFB$_{5n}$ assumption holds, then a challenge generated from $\pi_{(1)}$, which is a random element of $\mathcal{O}^{\pi_{(1)}}$, is computationally indistinguishable from a challenge generated from $\pi_{(2)}$, which is a random element of $\mathcal{O}^{\pi_{(2)}}$ (as both are computationally indistinguishable from a random element of $\mathcal{O}_g$). Based on Lemma 15, if the DFB assumption holds, then the shuffle achieves IND-CPA$_S$.