UNIVERSITY OF TARTU
Institute of Computer Science
Computer Science Curriculum

Janno Siim

Secure and Efficient Mix-Nets

Master's Thesis (30 ECTS)

Supervisor: Helger Lipmaa

Tartu 2016

Secure and Efficient Mix-Nets

Abstract: Mix-net is a system that can provide anonymity in a computer network. Mix-net takes as an input user's ciphertexts and outputs them in a shuffled order. Secure e-voting and variety of other applications can be built on top of mix-net architecture.

Major challenge of constructing mix-nets lies in efficiently proving that shuffling was done correctly. Mix-net cannot reveal the permutation because that would break the anonymity. One solution is to provide a zero-knowledge proof.

This thesis studies a zero-knowledge shuffle argument proposed by J. Furukawa in 2005.

Firstly we provide a more detailed and easily readable description of the shuffle and shuffle-decryption zero-knowledge protocols than in the original paper. Secondly we provide two new characterizations of a permutation matrix and two simple modifications of the shuffle protocol that reduce the computational complexity.

Keywords: mix-net, mix network, zero-knowledge, shuffle

CERCS: P170

Turvalised ja efektiivsed *mix-net*'id

Lühikokkuvõte:

Mix-net on süsteem, mis võimaldab saavutada anonüümsuse arvutite vahelises suhtluses. *Mix-net* võtab sisendiks kasutajate krüptogrammid ja väljastab krüptogrammid juhuslikult segatud järjekorras. *Mix-net*'id võimaldavad turvalise e-valimise ning paljude teiste anonüümsust vajavate rakenduste konstrueerimist.

Mix-net'ide ehitamisel on oluline võimalus veenduda, et segamine toimus korrektselt. Samas ei saa *mix-net* avaldada, kuidas segamine toimus, kuna sellega kaoks anonüümsus. Võimalik lahendus sellele probleemile on nullteadmusprotokolli kasutamine.

Antud magistritöös uuritakse J. Furukawa 2005. aastal välja pakutud nullteadmusprotokolli krüptogrammide segamise jaoks. Esiteks antakse detailne ja kergemini loetav kirjeldus Furukawa segamise ja segamis-dekrüpteerimise nullteadmusprotokollidest. Lisaks pakutakse välja kaks uut permutatsioonimaatriksi kirjeldust ning kaks lihtsat muudatust segamise protokollile, mis aitavad vähendada ajalist keerukust.

Võtmesõnad: *mix-net*, *mix-server*, nullteadmus, segamine

CERCS: P170

Contents

1 Introduction

A mix network (mix-net), first introduced in [Cha81], is a system that can provide anonymity in communication. A mix network contains several mix servers called mixers. Users send ciphertexts to the mix network and mixers take turns in shuffling the ciphertexts. Final mixer sends ciphertexts to their correct recipients. Shuffling can be done by permuting and rerandomizing the ciphertexts. Output ciphertexts will not be traceable to the original source if at least one mixer keeps the shuffle secret.

There is a wide range of applications that can use mix-nets. They are more suitable in situations where there is a large batch of ciphertexts that needs to be anonymized. Some of the examples are e-voting, anonymous e-mail and online surveys. Closely related is the concept of onion routing. In there, the anonymity comes from difficulty of observing a large network. Onion routing is more suitable in applications were few ciphertexts need to be anonymized at a given moment.

It is important that mixers shuffle correctly. It would be disastrous if, for example, in a voting application a mixer could copy, modify or insert new votes. A mixer cannot reveal the permutation or randomizers to prove correctness of the shuffle because it would de-anonymize the ciphertexts.

Correctness of a shuffle can be verified using a zero-knowledge proof. In a zero-knowledge proof, there are two parties: the prover and the verifier. The prover must convince the verifier that a certain statement holds. In the case of shuffling, the prover must convince the verifier that the output ciphertexts are a shuffle of the input ciphertexts and that it knows the corresponding permutation and randomizers.

In some applications, it might be useful if mixers also decrypt the ciphertexts. Then the private key of a mix-net would be shared between mixers. Users would encrypt messages with the mix-net's public key. In a mix-net, each mixer would do partial decryption together with shuffling. The output of the mix-net would be the anonymized plaintexts.

As the number of ciphertexts can be large, the efficiency is a mayor concern in a shuffle and shuffle-decryption protocols. Several different approaches like [Wik09] and [BG12] have been proposed for a zero-knowledge shuffle argument. In this thesis, we study shuffle and shuffle-decryption arguments proposed in [Fur05] by J. Furukawa.

Shuffle argument proposed by Furukawa is based on a characterization of a permutation matrix that is efficient to prove. Original argument, which is also presented in this thesis, is for ElGamal ciphertexts. Later, it has been generalized in [NSNK05] and [GL07] to a larger class of cryptosystems.

This thesis has two goals. Firstly we provide a more detailed and easily understandable description of the Furukawa shuffle and shuffle-decryption arguments compared to the original paper. Also, we describe the tools like Pedersen commitment and Schwartz-Zippel lemma that are needed in the protocol but were not described [Fur05].

Secondly we try to improve the Furukawa shuffle argument. We show two simple modifications that reduce the computation. We also propose two new characterizations of a permutation matrix. Further research is needed to see if these can be used to construct even more efficient shuffle arguments.

In section 2, we introduce notation and definitions used throughout the thesis. In section 3, we give a detailed description of the Furukawa shuffle argument and also propose some improvements and a new characterizations of a permutation matrix. In section 4, we show the construction of the Furukawa shuffle-decryption argument, which is based on the shuffle argument.

The author was supported by the European Union's Horizon 2020 research and innovation programme under grant agreement No 653497 (project PANORAMIX).

2 Preliminaries

Let $\mathbb{Z}$ denote the set of integers and $\mathbb{N} = \{1, 2, \dots\}$ is the set of natural numbers. Let $\mathbb{Z}_n$ be the set of integers modulo n. By $\mathbb{Z}_n^*$ we denote the set of all invertible elements of $\mathbb{Z}_n$. If p is a prime then $\mathbb{Z}_p$ is a field and $\mathbb{Z}_p^* = \mathbb{Z}_p \setminus \{0\}$ forms a multiplicative group.

By $x \leftarrow_r A$ we denote picking uniformly random element x from set A. Let $a, b \in \mathbb{Z}$ and $a \leq b$ then $[a..b] := \{c \in \mathbb{Z} : a \leq c \leq b\}$. We denote $(v_i)_{i=k}^{l} := (v_k, v_{k+1}, \dots, v_l)$ where $l \leq l$.

Definition 1. Let $\mathbb{F}$ be a field. Hadamard product is an operation $\circ : \mathbb{F}^{n \times m} \times \mathbb{F}^{n \times m} \to \mathbb{F}^{n \times m}$ such that for any $A = (a_{i,j}), B = (b_{i,j}) \in \mathbb{F}^{n \times m}$ where $n, m \in \mathbb{N}$ we have $A \circ B = (a_{i,j} \cdot b_{i,j})$.

We make use of the following lemma.

Lemma (Schwartz-Zippel). Let $p \in \mathbb{Z}_q[X_1, X_2, \dots, X_n]$ be a non-zero polynomial of degree d. Probability that $p(c_1, c_2, \dots, c_n) = 0$ for $c_1, c_2, \dots, c_n \leftarrow_r \mathbb{Z}_q$ is at most $\frac{d}{q-1}$.

In the proof of the protocol we use contrapositive of this lemma. We show that some unknown low-degree polynomial evaluates to 0 at a random point and conclude that with high probability it must be a zero polynomial i.e. the coefficients of the polynomial are 0.

Definition 2. Function $\varepsilon : \mathbb{N} \to \mathbb{R}$ is said to be negligible if for any $c > 0$ there exists $N \in \mathbb{N}$ such that

$$\forall n \geq N : \varepsilon(n) < \frac{1}{n^c}.$$

If outcome of an experiment happens with a negligible probability then the expected number of repetitions to make the event happen is superpolynomial. It is thus inefficient to make a negligible events happen.

Definition 3. Function $f : \mathbb{N} \to \mathbb{R}$ is said to be overwhelming if $1 - f$ is a negligible function.

2.1 ElGamal and Assumptions

Definition 4. [ElG84] ElGamal encryption scheme contains the following four algorithms:

Setup. Returns a suitable cyclic group G of order q and some generator g of that group.

Key generation. Picks a secret key $s \leftarrow_r \mathbb{Z}_q$. Public key is $h = g^s$.

Encryption. Let $m \in G$ be the message. Random value $r \leftarrow_r \mathbb{Z}_q$ is chosen. Encryption function E is defined as

$$E(m, r) := (g^r, m \cdot h^r) = (c_1, c_2).$$

Decryption. Let (c_1, c_2) be a ciphertext. Decryption function D is defined as

$$D(c_1, c_2) := \frac{c_2}{c_1^s} = \frac{m \cdot h^r}{(g^r)^s} = \frac{m \cdot h^r}{h^r} = m.$$

Let G be a cyclic group of prime order q with generator g.

Definition 5 (DDH Assumption). Let $x, y \leftarrow_r \mathbb{Z}_q$ and $\beta \leftarrow_r \{0,1\}$. If $\beta = 0$ then $z = xy$ and otherwise $z \leftarrow_r \mathbb{Z}_q$. We say that decisional Diffie-Hellman (DDH) assumption holds in group G if for any probabilistic polynomial-time adversary A the probability

$$\Pr[A(g, g^x, g^y, g^z) = \beta]$$

is negligible in $\log(q)$.

It can be shown that ElGamal encryption scheme is semantically secure if DDH assumption holds in group G.

Definition 6 (DL Assumption). We say that discrete logarithm (DL) assumption holds in group G if for any probabilistic polynomial-time adversary A the probability

$$\Pr[A(g, g^x) = x : x \leftarrow_r \mathbb{Z}_q]$$

is negligible in $\log(q)$.

It can be shown that if DDH assumption holds in group G then DL assumption holds in group G.

In the proof of the shuffle-decryption privacy we will make use of the extended DDH assumption.

Definition 7. Let $R_n^m := G^{(n+1)m}$. We denote elements of R_n^m as $\Theta_{m,n} := (\theta_{i,v})_{\substack{i \in [1..m] \\ v \in [0..n]}}$. Let

$$D_n^m := \{\Theta_{m,n} \in R_n^m \mid \forall v \in [1..n] \, \forall j \in [2..m] : \log_{\theta_{1,0}} \theta_{j,0} = \log_{\theta_{1,v}} \theta_{j,v}\}$$

and $\beta \leftarrow_r \{0,1\}$. If $\beta = 0$ then $\Theta_{m,n} \leftarrow_r D_n^m$, otherwise $\Theta_{m,n} \leftarrow_r R_n^m$. We say that DDH_n^m assumption holds in group G if for any probabilistic polynomial-time adversary A the probability

$$\Pr[A(\Theta_{m,n}) = \beta]$$

is negligible in $\log(q)$.

In DDH_1^2 assumption we have $\Theta_1^2 = (\theta_{1,0}, \theta_{1,1}, \theta_{2,0}, \theta_{2,1})$ such that if $\Theta_1^2 \in D_1^2$ then

$$\log_{\theta_{1,0}} \theta_{2,0} = \log_{\theta_{1,1}} \theta_{2,1} = \frac{\log_{\theta_{1,0}} \theta_{2,1}}{\log_{\theta_{1,0}} \theta_{1,1}}.$$

Thus

$$\log_{\theta_{1,0}} \theta_{2,0} \log_{\theta_{1,0}} \theta_{1,1} = \log_{\theta_{1,0}} \theta_{2,1}$$

and it can be seen that the DDH_1^2 is the DDH assumption.

In [FS01] it is shown that if DDH_n^m can be broken then DDH can be broken.

2.2 Permutation matrices

We denote by $\delta_{i,j}$ and $\delta_{i,j,k}$ the functions

$$\delta_{i,j} = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{otherwise} \end{cases}$$

and

$$\delta_{i,j,k} = \begin{cases} 1 & \text{if } i = j = k \\ 0 & \text{otherwise} \end{cases}.$$

Definition 8. Let q be a prime. A matrix $A = (A_{i,j}) \in \mathbb{Z}_q^{n \times n}$ is called a permutation matrix if for any $i, j \in [1..n]$ we have

$$A_{i,j} = \delta_{\pi(i),j}$$

where $\pi : [1..n] \to [1..n]$ is a permutation.

We prove two theorems (theorem 1 and 2) from [Fur05] that give alternative definitions for a permutation matrix. Furukawa uses these definitions in the shuffle and the shuffle-decryption protocols.

Lemma 1. Let q be a prime. If matrix $A \in \mathbb{Z}_q^{n \times n}$ is such that for any $i, j, k \in [1..n]$

$$\sum_{h=1}^{n} A_{h,i} A_{h,j} A_{h,k} = \delta_{i,j,k}$$

then any row and column of the matrix A has exactly one nonzero element.

Proof. We show first that matrix A is full rank.

Let us denote by A^i and A_i correspondingly the i-th column and row vector of A where $i \in [1..n]$.

To show that A is full rank we may show that vectors $A^1, \ldots, A^n$ are linearly independent. Let $\alpha_1, \ldots, \alpha_n \in \mathbb{Z}_q$ be such that

$$\sum_{j=1}^{n} \alpha_j A^j = \bar{0}$$

where $\bar{0}$ is a zero vector of length n.

Let us compute the following scalar product for $i \in [1..n]$

$$\langle \sum_{j=1}^{n} \alpha_j A^j, A^i \circ A^i \rangle = \sum_{j=1}^{n} \langle \alpha_j A^j, A^i \circ A^i \rangle = \sum_{j=1}^{n} \alpha_j \langle A^j, A^i \circ A^i \rangle$$

Let us look separately the part

$$\langle A^j, A^i \circ A^i \rangle = \sum_{h=1}^{n} A_{h,j} A_{h,i} A_{h,i} = \delta_{j,i,i} = \delta_{j,i}.$$

Therefore

$$\langle \sum_{j=1}^{n} \alpha_j A^j, A^i \circ A^i \rangle = \alpha_i.$$

We know in addition that $\langle \sum_{j=1}^{n} \alpha_j A^j, A^i \circ A^i \rangle = 0$ because $\sum_{j=1}^{n} \alpha_j A^j = 0$. Thus $\alpha_i = 0$. This argument works for any $i \in [1..n]$ which means that column vectors of A are linearly independent.

Let us show that for any $i, j \in [1..n]$, if $i \neq j$ then $A^i \circ A^j = \bar{0}$.

We compute

$$(A_{1,i} A_{1,j}) A_1 + \ldots + (A_{n,i} A_{n,j}) A_n = (\sum_{h=1}^{n} A_{h,i} A_{h,j} A_{h,1}, \ldots, \sum_{h=1}^{n} A_{h,i} A_{h,j} A_{h,n}) =$$

$$= (\delta_{i,j,1}, \ldots, \delta_{i,j,n}) = \bar{0}.$$

Because the row vectors of A are linearly independent, the coefficients must be zeros and thus $A^i \circ A^j = \bar{0}$.

Because A is fullrank there are no zero columns or rows. Let $A_{s,i}$ be some nonzero element in s-th row . Then it must be the only nonzero element in s-th row because otherwise there would exist some other j-th column such that $A^i \circ A^j \neq \bar{0}$.

Because all the rows have exactly one nonzero element and there are no zero columns then every column has also exactly one nonzero element as well. □

Theorem 1. Let q be a prime. Matrix $A \in \mathbb{Z}_q^{n\times n}$ is a permutation matrix if and only if for any $i, j, k \in [1..n]$

$$\sum_{h=1}^{n} A_{h,i}A_{h,j}A_{h,k} = \delta_{i,j,k} \tag{1}$$

and

$$\sum_{h=1}^{n} A_{h,i}A_{h,j} = \delta_{i,j}. \tag{2}$$

Proof. $\Rightarrow$) Let $A \in \mathbb{Z}_q^{n\times n}$ be a permutation matrix for an arbitrary permutation $\pi : [1..n] \to [1..n]$. Let i, j, k be arbitrary column indices from the set $[1..n]$. Then

$$\sum_{h=1}^{n} A_{h,i}A_{h,j}A_{h,k} = \sum_{h=1}^{n} \delta_{\pi(h),i}\delta_{\pi(h),j}\delta_{\pi(h),k}$$

Expression $\delta_{\pi(h),i}\delta_{\pi(h),j}\delta_{\pi(h),k} = 1$ only if $\pi(h) = i = j = k$. Because π is a permutation we have that $\pi(h) = i$ for exactly one value of h, thus we may write

$$\sum_{h=1}^{n} A_{h,i}A_{h,j}A_{h,k} = \delta_{i,i}\delta_{i,j}\delta_{i,k} = 1 \cdot \delta_{i,j}\delta_{i,k} = \delta_{i,j,k}.$$

Proof for the second property is analogues

$$\sum_{h=1}^{n} A_{h,i}A_{h,j} = \sum_{h=1}^{n} \delta_{\pi(h),i}\delta_{\pi(h),j} = \delta_{i,i}\delta_{i,j} = \delta_{i,j}.$$

$\Leftarrow$) According to lemma 1 every row and column of matrix A contains exactly one nonzero element. Let us pick an arbitrary column index $i \in [1..n]$. From property (1) and (2) we get

$$\sum_{h=1}^{n} A_{h,i}^3 = 1$$

and

$$\sum_{h=1}^{n} A_{h,i}^2 = 1.$$

Because there is only one nonzero element then there must exist row index s such that $A_{s,i}^3 = 1$ and $A_{s,i}^2 = 1$. Then

$$A_{s,i} = \frac{A_{s,i}^3}{A_{s,i}^2} = 1.$$

Therefore A is a permutation matrix. $\square$

Theorem 2. Let q be a prime such that $q \bmod 3 = 2$. Matrix $A \in \mathbb{Z}_q^{n\times n}$ is a permutation matrix if and only if for any $i, j, k \in [1..n]$

$$\sum_{h=1}^{n} A_{h,i}A_{h,j}A_{h,k} = \delta_{i,j,k}.$$

Proof. $\Rightarrow$) Holds due to the theorem 1.

$\Leftarrow$) According to the lemma 1 every row and column of the matrix A has exactly one nonzero element. For any column index i there exists a row index s such that $A_{s,i}^3 = 1$ where $A_{s,i} \in \mathbb{Z}_q$.

Because $q \bmod 3 = 2$ we have that $q = 3k + 2$ for some integer k. Then

$$A_{s,i}^q = A_{s,i}^{3k+2} = (A_{s,i}^3)^k A_{s,i}^2 = A_{s,i}^2.$$

From Fermat's little theorem we also have that

$$A_{s,i}^q = A_{s,i}.$$

These facts together imply that

$$A_{s,i} = \frac{A_{s,i}^2}{A_{s,i}} = \frac{A_{s,i}^q}{A_{s,i}^q} = 1.$$

Therefore A is a permutation matrix. $\square$

2.3 Commitment schemes

Commitment scheme allows to commit to some value while keeping the value secret. Committer has the ability to reveal (open) the commitment at a later time. Commitment scheme should be **hiding** i.e. commitment should not reveal information about the committed value. Commitment scheme should be **binding** i.e. commitment can be opened only to the value that was committed.

Protocol described later makes use of the extended Pedersen vector commitment scheme. [Ped91]

Definition 9. Extended Pedersen commitment scheme for n elements consists of three algorithms:

Setup. Let G be a cyclic group of prime order q. Algorithm picks generators $g, f_1, f_2, \ldots, f_n \leftarrow_r G \setminus \{1\}$ and returns $ck \leftarrow (G, g, f_1, \ldots, f_n)$.

Commitment. Let $\bar{a} = (a_i)_{i=1}^n \in \mathbb{Z}_q^n$ and $r \leftarrow_r \mathbb{Z}_q$. Commitment to the vector $\bar{a}$ is

$$com_{ck}(\bar{a}, r) := g^r \prod_{i=1}^{n} f_i^{a_i} = C.$$

In the commitment phase C is published and in the revealing phase the opening $(\bar{a}, r)$ is published.

Verification. Let C be the commitment and $(\bar{a}, r)$ the corresponding opening. Commitment is correct if $com_{ck}(\bar{a}, r) = C$.

The extended Pedersen commitment is perfectly hiding because $g^r \prod_{i=1}^n f_i^{a_i}$ is a uniformly random element in G. It can be shown that the extended Pedersen commitment is computationally binding if the discrete logarithm assumption holds in group G.

2.4 Zero-knowledge Protocols

This section introduces basic notions related to zero-knowledge protocols (also called zero-knowledge proofs).

To model interactive communication we use interactive Turing machines. In the following (P, V) is a pair of probabilistic polynomial-time interactive Turing machines. Interactive pair of Turing machines is defined almost like a regular Turing machines but in addition Turing machines P and V share two input/output tapes. To one tape P can write and V can only read, to the other tape V can write and P can only read. Turing machines take turns in computing. Prover P starts the computation and verfier V ends it. Output of the pair (P, V) is the sequence of values communicated between P and V. A more detailed description can be seen in [GMR85].

We look at the situation where a prover P wants to convince a verifier V that a certain statement is true $(x \in L)$ using some auxiliary information ω. Depending if the verifier is convinced or not it outputs "accept" or "reject" accordingly.

We assume in the following that R is an MA relation i.e. there exists a probabilistic polynomial-time Turing machine A and a polynomial p such that

$$\forall x \in \{0,1\}^* \ \forall \omega \in \{0,1\}^{p(|x|)} : (x, \omega) \in R \Leftrightarrow \Pr[A(x, \omega) = 1] \geq 1 - 2^{-|x|}$$

$$\forall x \in \{0,1\}^* \ \forall \omega \in \{0,1\}^{p(|x|)} : (x, \omega) \notin R \Leftrightarrow \Pr[A(x, \omega) = 1] \leq 2^{-|x|}.$$

Corresponding language is denoted by $L_R := \{x : (\exists \omega \in \{0,1\}^*)[(x, \omega) \in R]\}$. Notably MA relations exist for all NP languages.

Definition 10. The pair (P, V) is called an interactive argument for language L_R if it satisfies the following conditions:

> **Completeness:** If $(x, \omega) \in R$ then given ω as a private input to P makes $(P(x, \omega), V(x))$ output "accept".
>
> **Soundness:** If $x \notin L$ then for any probabilistic polynomial-time prover P^* probability that the pair (P^*, V) outputs "accept" is negligible in the length of x.

In the case of shuffling it is not sufficient to prove that ciphertexts were shuffled. For example a malicious mixer could undo the suffling of the previous mixers by outputting the ciphertexts that were inputs to the first mixer. There exists some permutation and randomizers which would make it a legitimate shuffle. Therefore it is necessary that mixer also proves that it knows the permutation and the randomizer. A proof of knowledge is formalized in the following definition [BG93].

Definition 11. Let $\kappa : \{0,1\}^* \to [0,1]$. Interactive argument (P, V) is called a proof of knowledge for relation R with knowledge error κ if the following condition holds:

> **Knowledge soundness:** Let $\varepsilon_{P^*}(x)$ be the probability that prover P^* makes V accept. There exist a probabilistic Turing machine K called a knowledge extractor that gets a rewindable black-box oracle access to P^*. Extractor K satisfies the following condition: there exists $c > 0$ such that for any prover P^* and the corresponding probability function $\varepsilon_{P^*}(x)$ if $\varepsilon_{P^*}(x) > \kappa(x)$ then K outputs ω such that $(x, \omega) \in R$ in expected time at most
>
> $$\frac{|x|^c}{\varepsilon(x) - \kappa(x)}.$$

We can view a probabilistic Turing machine as a random variable. A machine M on some input x defines a distribution on the possible outputs.

Definition 12. Probabilistic Turing machines M_1 and M_2 are said to be perfectly indistinguishable if for any input $x \in \{0,1\}^*$ we have $M_1(x) = M_2(x)$. This is denoted as $M_1 \sim^p M_2$.

The following definition guarantees that the only information that the verifier learns is that $x \in L$.

Definition 13. An interactive pair (P, V) for language L_R is said to be zero-knowledge (ZK) if for any probabilistic polynomial-time verifier V^*, there exists a simulator S running in expected polynomial-time such that

$$(\forall x \in L)\ (\forall \sigma \in \{0,1\}^*)\ [S(x, \sigma) \sim^p (P(x), V^*(x, \sigma))].$$

Bitstring σ in the definition denotes some prior knowledges that verifier V^* might have. Definition implies that verifier V^* could produce the conversation by itself using the simulator S. Thus the conversation with prover P cannot leak any information.

Definition 14. A 3-message protocol (P, V) is said to be special honest-verifier zero-knowledge (SHVZK) if there exist a probabilistic polynomial-time simulator S such that for any $x \in L_R$:

1. For any challenge c simulator $S(c)$ outputs an accepting conversation (a, c, z).
2. For uniformly randomly chosen challenge c we have $S(c) \sim^p (P, V)$.

It is often easier to prove that a protocol is SHVZK rather than ZK. SHVZK assumes however that the challenge from the verifier is chosen uniformly randomly. This might not be true in the case of a malicious verifier.

There are standard constructions that allow to make 3-message SHVZK protocols into ZK protocols with one more round and additive overhead. [GMY06]

In practice it might be useful to turn a SHVZK protocol into a non-interactive ZK protocol instead. This can be done using the Fiat-Shamir heuristic. There the prover computes a challenge himself by using a hash function. It can be shown that Fiat-Shamir heuristic is secure in the random oracle model. [FS86]

3 Furukawa Shuffle Argument

This section contains a proof of knowledge argument for the ElGamal ciphertext shuffling proposed in [Fur05]. More specifically it is a 3-message proof of knowledge argument that is perfect special honest verifier zero-knowledge.

Let n be the number of ciphertexts. In this protocol prover and verifier must respectively do approximately $8n$ and $6n$ exponentiations. Communication is roughly $n \log p + 3n \log q$ bits where p and q are two large primes and $q \ll p$.

3.1 Description

3.1.1 Setup

Let p and q be two large primes such that $q|(p-1)$. Let G be an order q multiplicative subgroup of $\mathbb{Z}_p^*$ with generator g_0. According to the Cauchy's theorem such a subgroup always exists. Let $m_0 \in G$ be an ElGamal public key that was used for encrypting input ciphertexts $z_i = (g_i, m_i)$ for $i \in [1..n]$. The mix server picks a random permutation $\pi : [1..n] \to [1..n]$ and n randomizers $(A_{0,i})_{i=1}^n \in \mathbb{Z}_q^n$. The mix server computes output ciphertexts

$$z_i' = E(1, A_{0,i}) \circ z_{\pi^{-1}(i)} = (g_0^{A_{0,i}} g_{\pi^{-1}(i)}, m_0^{A_{0,i}} m_{\pi^{-1}(i)})$$

for $i \in [1..n]$. We denote $z_i' := (g_i', m_i')$.

Let $F := (f_i)_{i=-4}^n \leftarrow_r (G \setminus \{1\})^n$ be generators for the extended Pedersen commitment. Matrix $A \in \mathbb{Z}_q^{n \times n}$ is a permutation matrix corresponding to the permutation π. Then the following equation holds

$$z_i' = (g_i', m_i') = (g_0^{A_{0,i}} \prod_{j=1}^n g_j^{\delta_{\pi(j),i}}, m_0^{A_{0,i}} \prod_{j=1}^n m_j^{\delta_{\pi(j),i}}) = (\prod_{j=0}^n g_j^{A_{ji}}, \prod_{j=0}^n m_j^{A_{ji}}) \qquad (3)$$

for $i \in [1..n]$.

The mix server acting as a prover P must prove to a verifier V that it knows the permutation π and the randomizers $A_{0,i}$ for $i \in [1..n]$ without any leaking in other knowledge.

Both the prover P and the verifier V get as an input $pk \leftarrow (p, q, g_0, m_0, F, (z_i)_{i=1}^n, (z_i')_{i=1}^n)$. The prover P knows in addition A and $(A_{0,i})_{i=1}^n$.

3.1.2 Protocol

$P\big(pk, A, (A_{0i})_{i=1}^{n}\big)$

$V(pk)$

Commitment:

$$A_{\nu 0}, A'_\nu \leftarrow_r \mathbb{Z}_q \qquad \nu \in [-4..n]$$

$$\left.\begin{aligned} A_{-1i} &\leftarrow_r \mathbb{Z}_q \\ A_{-2i} &= \sum_{j=1}^{n} 3A_{j0}^2 A_{ji} \\ A_{-3i} &= \sum_{j=1}^{n} 3A_{j0} A_{ji} \\ A_{-4i} &= \sum_{j=1}^{n} 2A_{j0} A_{ji} \end{aligned}\right\} \quad i \in [1..n]$$

$$f'_\mu = \prod_{\nu=-4}^{n} f_\nu^{A_{\nu\mu}} \qquad \mu \in [0..n] \tag{7}$$

$$\tilde{f}'_0 = \prod_{\nu=-4}^{n} f_\nu^{A'_\nu} \tag{8}$$

$$g'_0 = \prod_{\nu=0}^{n} g_\nu^{A_{\nu 0}} \qquad m'_0 = \prod_{\nu=0}^{n} m_\nu^{A_{\nu 0}}$$

$$\omega = \sum_{j=1}^{n} A_{j0}^3 - A_{-20} - A'_{-3}$$

$$\dot{\omega} = \sum_{j=1}^{n} A_{j0}^2 - A_{-40}$$

$$\xrightarrow{g'_0,\, m'_0,\, \tilde{f}'_0,\, (f'_\mu)_{\mu=0}^{n},\, \omega,\, \dot{\omega}}$$

Challenge:

$$c_i \leftarrow_r \mathbb{Z}_q \qquad i \in [1..n]$$

$$\xleftarrow{(c_i)_{i=1}^{n}}$$

Response:

$$\left.\begin{aligned} r_\nu &= \sum_{\mu=0}^{n} A_{\nu\mu} c_\mu \\ r'_\nu &= \sum_{i=1}^{n} A_{\nu i} c_i^2 + A'_\nu \end{aligned}\right\} \quad \nu \in [-4..n]$$

$$(c_0 = 1)$$

$$\xrightarrow{(r_\nu)_{\nu=-4}^{n},\, (r'_\nu)_{\nu=-4}^{n}}$$

Verification:

$$\alpha \leftarrow_r \mathbb{Z}_q$$

$$\prod_{\nu=-4}^{n} f_\nu^{r_\nu + \alpha r'_\nu} \overset{?}{=} f'_0 \left(\tilde{f}'_0\right)^{\alpha} \prod_{i=1}^{n} \left(f'_i\right)^{c_i + \alpha c_i^2} \tag{9}$$

$$\prod_{\nu=0}^{n} g_\nu^{r_\nu} \overset{?}{=} \prod_{\mu=0}^{n} \left(g'_\mu\right)^{c_\mu} \tag{10}$$

$$\prod_{\nu=0}^{n} m_\nu^{r_\nu} \overset{?}{=} \prod_{\mu=0}^{n} \left(m'_\mu\right)^{c_\mu} \tag{11}$$

$$\sum_{h=1}^{n} \left(r_h^3 - c_h^3\right) \overset{?}{=} r_{-2} + r'_{-3} + \omega \tag{12}$$

$$\sum_{h=1}^{n} \left(r_h^2 - c_h^2\right) \overset{?}{=} r_{-4} + \dot{\omega} \tag{13}$$

3.1.3 Overview

In this subsection we explain the general idea of the protocol.

Commitment

The protocol starts with the prover committing to a series of values.

Most important part of the commitment is the equation (7). This is an Extended Pedersen commitment of the columns of the permutation matrix, randomizers and also some additional values $(A_{-2i})_{i=1}^n, (A_{-3i})_{i=1}^n, (A_{-4i})_{i=1}^n$. The elements $(A_{-1i})_{i=1}^n$ are used to hide the commitment. Values g_0', m_0', $\tilde{f}_0'$ and f_0 are needed to get zero-knowledge. Values ω and $\dot{\omega}$ are used to cancel out unnecessary terms in the equation (12) and (13).

Challenge

The verifier sends n uniformly random and independent elements of $\mathbb{Z}_q$ to prover. This allows us to later use Schwartz-Zippel lemma.

Response

The prover responds with $r_v = A_{v0} + \sum_{\mu=1}^n A_{v\mu} c_\mu$ and $r_v' = A_v' + \sum_{i=1}^n A_{vi} c_i^2$ for $v \in [-4..n]$. These can be thought of as polynomials evaluated at random point $(c_i)_{i=1}^n$. Values A_{v0} and A_v' are used to hide the response.

Having a responses in the form of a linear equations gives an easy way to extract the permutation matrix A and randomizers $(A_{0i})_{i=1}^n$. For a fixed commitment, extractor can use rewinding and attains for each $v \in [-4..n]$ linear equations where $(A_{v\mu})_{\mu=0}^n$ are the unknowns. Having $n+1$ equations with linearly independent coefficients allows an extractor to solve the system and obtain $(A_{v\mu})_{\mu=0}^n$.

Verification

Equation (9) can be used to prove that prover computed responses $(r_v)_{v=-4}^n$ and $(r_v')_{v=-4}^n$ correctly.

Equations (12) and (13) can be used to show that respectively properties (1) and (2) hold for matrix A i.e. A is a permutation matrix. If we view $\sum_{h=1}^n r_h^3 - c_h^3$ and $\sum_{h=1}^n r_h^2 - c_h^2$ as polynomials evaluated at a random point $(c_i)_{i=1}^n$, it turns out that coefficients of those polynomials are such that using the Schwartz-Zippel lemma allows us to prove property (1) and (2). The constant term and some of the coefficients in those polynomials however cannot be 0 because they depend on randomly chosen values $(A_{v0})_{v=-4}^n$. Constant term can be cancelled with ω and $\dot{\omega}$ in the respective equation. Nonzero coefficients are removed with r_{-2}, r_{-3}' and r_{-4}. That is why the prover must commit to A_{-2i}, A_{-3i} and A_{-4i}.

Equations (10) and (11) show that the equation (3) holds i.e. ciphertexts were shuffled with the permutation matrix A and rerandomized with $(A_{0i})_{i=1}^n$.

3.2 Proof

Theorem 3. The shuffle protocol is complete.

Proof. Let us see that all the verification equations hold if the prover is honest.

(9) Let $\alpha \leftarrow_r \mathbb{Z}_q$, then

$$\prod_{v=-4}^{n} f_v^{r_v+\alpha r'_v} = \prod_{v=-4}^{n} f_v^{r_v} \cdot f_v^{\alpha r'_v} = \prod_{v=-4}^{n} f_v^{A_{v0}c_0} \cdot f_v^{\sum_{i=1}^{n} A_{vi}c_i} \cdot f_v^{\alpha A'_v} \cdot f_v^{\alpha \sum_{i=1}^{n} A_{vi}c_i^2} =$$

$$= f'_0 \cdot \tilde{f}'^{\alpha}_0 \prod_{v=-4}^{n} f_v^{\sum_{i=1}^{n} A_{vi}(c_i+\alpha c_i^2)} = f'_0 \cdot \tilde{f}'^{\alpha}_0 \prod_{v=-4}^{n} \prod_{i=1}^{n} f_v^{A_{vi}(c_i+\alpha c_i^2)} = f'_0 \cdot \tilde{f}'^{\alpha}_0 \prod_{i=1}^{n} (\prod_{v=-4}^{n} f_v^{A_{vi}})^{c_i+\alpha c_i^2} =$$

$$= f'_0 \cdot \tilde{f}'^{\alpha}_0 \prod_{i=1}^{n} f'_i{}^{c_i+\alpha c_i^2}.$$

(10)

$$\prod_{v=0}^{n} g_v^{r_v} = \prod_{v=0}^{n} g_v^{\sum_{\mu=0}^{n} A_{v\mu}c_\mu} = \prod_{v=0}^{n} \prod_{\mu=0}^{n} g_v^{A_{v\mu}c_\mu} = \prod_{\mu=0}^{n} (\prod_{v=0}^{n} g_v^{A_{v\mu}})^{c_\mu} = \prod_{\mu=0}^{n} (g'_\mu)^{c_\mu}$$

Last equality holds due to the equation (3).

(11) Analogous to the previous one.

(12) If the prover is honest then

$$r_h = \sum_{\mu=0}^{n} A_{h\mu}c_\mu = A_{h0} + \sum_{\mu=1}^{n} \delta_{\pi(h)\mu}c_\mu = A_{h0} + c_{\pi(h)}$$

for $v \in [1..n]$.

Then left-hand side of the equation (12) is

$$\sum_{h=1}^{n} r_h^3 - c_h^3 = \sum_{h=1}^{n} A_{h0}^3 + 3A_{h0}^2 c_{\pi(h)} + 3A_{h0}c_{\pi(h)}^2 + c_{\pi(h)}^3 - c_h^3 =$$

$$= \sum_{h=1}^{n} A_{h0}^3 + 3A_{h0}^2 c_{\pi(h)} + 3A_{h0}c_{\pi(h)}^2.$$

Right-hand side of the equation (12) is

$$r_{-2} + r'_{-3} + \omega = A_{-20} + \sum_{h=1}^{n} A_{-2h}c_h + \sum_{h=1}^{n} A_{-3h}c_h^2 + A'_{-3} + \omega =$$

$$= A_{-20} + \sum_{h=1}^{n}(\sum_{j=1}^{n} 3A_{j0}^2 A_{jh})c_h + \sum_{h=1}^{n}(\sum_{j=1}^{n} 3A_{j0}A_{jh})c_h^2 + A'_{-3} + \omega =$$

$$= \sum_{h=1}^{n} 3A_{h0}^2 c_{\pi(h)} + \sum_{h=1}^{n} 3A_{h0}c_{\pi(h)}^2 + A_{-20} + A'_{-3} + \sum_{h=1}^{n} A_{h0}^3 - A_{-20} - A'_{-3} =$$

$$= \sum_{h=1}^{n} A_{h0}^3 + 3A_{h0}^2 c_{\pi(h)} + 3A_{h0}c_{\pi(h)}^2$$

(13) Left-hand side of the equation (13) is

$$\sum_{h=1}^{n} r_h^2 - c_h^2 = \sum_{h=1}^{n} A_{h0}^2 + 2A_{h0}c_{\pi(h)} + c_{\pi(h)}^2 - c_h^2 = \sum_{h=1}^{n} A_{h0}^2 + 2A_{h0}c_{\pi(h)}.$$

Right-hand side of the equation (13) is

$$r_{-4} + \dot{\omega} = A_{-40} + \sum_{h=1}^{n} A_{-4h}c_h + \dot{\omega} = A_{-40} + \sum_{h=1}^{n}(\sum_{j=1}^{n} 2A_{j0}A_{jh})c_h + \dot{\omega} =$$

$$= A_{-40} + \sum_{h=1}^{n} 2A_{\mu 0}c_{\pi(h)} + \sum_{h=1}^{n} A_{h0}^2 - A_{-40} = \sum_{h=1}^{n} A_{h0}^2 + 2A_{\mu 0}c_{\pi(h)}.$$

All the verification equations hold so the protocol is complete. □

In the rest of the proof we show that this protocol has knowledge soundness under the DL assumption.

Lemma 2. If the equation (9) in the protocol holds then the probability that

$$\prod_{v=-4}^{n} f_v^{r_v} = \prod_{\mu=0}^{n} (f'_\mu)^{c_\mu} \tag{4}$$

and

$$\prod_{v=-4}^{n} f_v^{r'_v} = \tilde{f}'_0 \prod_{\mu=1}^{n} (f'_\mu)^{c_\mu^2} \tag{5}$$

hold is $1 - \frac{1}{q}$.

Proof. For $\alpha \leftarrow_r \mathbb{Z}_q$ we have

$$\prod_{v=-4}^{n} f_v^{r_v + \alpha r'_v} = f'_0(\tilde{f_0}')^{\alpha} \prod_{i=1}^{n} (f'_i)^{c_i + \alpha c_i^2}.$$

Let us divide both sides of the equation by $\prod_{\mu=0}^{n}(f'_\mu)^{c_\mu}$ and $\prod_{v=-4}^{n} f_v^{\alpha r'_v}$. We get

$$\frac{\prod_{v=-4}^{n} f_v^{r_v}}{\prod_{\mu=0}^{n}(f'_\mu)^{c_\mu}} = \left(\frac{\tilde{f_0}'\prod_{i=1}^{n}(f'_i)^{c_i^2}}{\prod_{v=-4}^{n} f_v^{r'_v}}\right)^{\alpha}$$

Let us denote $x := \frac{\prod_{v=-4}^{n} f_v^{r_v}}{\prod_{\mu=0}^{n}(f'_\mu)^{c_\mu}}$ and $y := \left(\frac{\tilde{f_0}'\prod_{i=1}^{n}(f'_i)^{c_i^2}}{\prod_{v=-4}^{n} f_v^{r'_v}}\right)^{\alpha}$. If $y = 1$ then $x = 1$ and equation $x = y^\alpha$ holds for all $\alpha \in \mathbb{Z}_q$. We may note also that if $x = 1$ then the equation (14) holds and if $y = 1$ then the equation (15) holds. If $y \neq 1$ then y is a generator of group G and there is exactly one value of $\alpha \in \mathbb{Z}_q$ such that $x = y^\alpha$. Thus the probability that (14) and (15) hold is $\frac{1}{q}$. Conversely with probability $1 - \frac{1}{q}$ the equations (14) and (15) hold. □

Lemma 3. If P^* can make V accept the protocol with probability $\varepsilon(pk) > 0$ then there exists an extractor K that can extract from P^* the values $(A_{v,\mu})_{v=-4}^{n}$ for $\mu \in [0..n]$ and $(A'_v)_{v=-4}^{n}$ in expected time $O(\frac{poly(n)}{\varepsilon(pk)})$. The extracted values satisfy the equations (7) and (8).

Proof. The extractor K works as follows:

1. Let $C = \emptyset$. Set a rewinding point after the commitment of P^*.
2. Pick $c_1, c_2, \ldots, c_n \leftarrow_r \mathbb{Z}_q$ and set $\bar{c} \leftarrow (1, c_1, c_2, \ldots, c_n)$.
3. If $C' \leftarrow C \cup \{\bar{c}\}$ is not linearly independent go to step 2.
4. Send $\bar{c}$ to P^* then the P^* sends a response. If the response satisfies the verification equations then sets $C \leftarrow C'$ and saves the response.
5. If $|C| \neq n + 1$ then rewind and go to step 2.
6. We have that $C = \{\bar{c}^{(1)}, \ldots, \bar{c}^{(n+1)}\}$. Let us denote $\bar{c}^{(i)} := (1, c_1^{(i)}, \ldots, c_n^{(i)})$ and corresponding responses by $(r_v^{(i)})_{v=-4}^{n}$ and $({r'_v}^{(i)})_{v=-4}^{n}$ for $i \in [1..n+1]$. It will solve for each $v \in [-4..n]$ a linear system

$$\begin{cases} r_v^{(1)} &= \sum_{\mu=0}^{n} A_{v\mu} c_\mu^{(1)} \\ r_v^{(2)} &= \sum_{\mu=0}^{n} A_{v\mu} c_\mu^{(2)} \\ \vdots \\ r_v^{(n+1)} &= \sum_{\mu=0}^{n} A_{v\mu} c_\mu^{(n+1)} \end{cases}$$

   to extract $(A_{v\mu})_{\mu=0}^{n}$. This system has exactly one solution because vectors in C are linearly independent.
7. The values $(A'_v)_{v=-4}^{n}$ can be extracted by computing

$$A'_v = {r'_v}^{(1)} - \sum_{i=1}^{n} A_{vi}(c_i^{(1)})^2$$

   for $v \in [-4..n]$.

We show that the values $(A_{v\mu})_{v=-4}^{n}$ extracted in the step 6. satisfy the equation (7). Due to the lemma 2 we have that

$$\prod_{v=-4}^{n} f_v^{r_v^{(i)}} = \prod_{\mu=0}^{n} (f'_\mu)^{c_\mu^{(i)}}$$

for $i \in [1..n+1]$. Then

$$\prod_{v=-4}^{n} f_v^{\sum_{\mu=0}^{n} A_{v\mu} c_\mu^{(i)}} = \prod_{\mu=0}^{n} (f'_\mu)^{c_\mu^{(i)}}$$

$$\prod_{\mu=0}^{n} f_v^{\sum_{v=-4}^{n} A_{v\mu} c_\mu^{(i)}} = \prod_{\mu=0}^{n} (f'_\mu)^{c_\mu^{(i)}}$$

$$\prod_{\mu=0}^{n} \left(\frac{f_v^{\sum_{v=-4}^{n} A_{v\mu}}}{f'_\mu} \right)^{c_\mu^{(i)}} = 1.$$

Let us denote $x_\mu := \log_{g_0} \left(\frac{f_v^{\sum_{v=-4}^{n} A_{v\mu}}}{f'_\mu} \right)$ for $\mu \in [0..n]$. Then the previous equation can be written as

$$\prod_{\mu=0}^{n} g_0^{x_\mu c_\mu^{(i)}} = 1$$

$$g_0^{\sum_{\mu=0}^{n} x_\mu c_\mu^{(i)}} = g_0^0.$$

We get a homogeneous linear system of equations

$$\begin{cases} \sum_{\mu=0}^{n} x_\mu c_\mu^{(1)} = 0 \\ \sum_{\mu=0}^{n} x_\mu c_\mu^{(2)} = 0 \\ \vdots \\ \sum_{\mu=0}^{n} x_\mu c_\mu^{(n+1)} = 0 \end{cases}.$$

It has exactly one solution $x_1 = x_2 = \ldots = x_n = 0$. Thus

$$\frac{f_v^{\sum_{v=-4}^{n} A_{v\mu}}}{f'_\mu} = g^0 = 1 \Rightarrow f'_\mu = f_v^{\sum_{v=-4}^{n} A_{v\mu}} = \prod_{v=-4}^{n} f_v^{A_{v\mu}}$$

and the equation (7) holds.

We show that the values $(A_v)_{v=-4}^{n}$ extracted in the step 7. satisfies the equation (8). Using the lemma 2 again we have

$$\prod_{v=-4}^{n} f_v^{r_v'^{(1)}} = \tilde{f}_0' \prod_{\mu=1}^{n} (f_\mu')^{(c_\mu^{(1)})^2}.$$

From this equation we get

$$\tilde{f}_0' = \frac{\prod_{v=-4}^{n} f_v^{r_v'^{(1)}}}{\prod_{\mu=1}^{n} (f_\mu')^{(c_\mu^{(1)})^2}} = \frac{\prod_{v=-4}^{n} f_v^{\sum_{i=1}^{n} A_{vi}(c_i^{(1)})^2 + A_v'}}{\prod_{\mu=1}^{n} \prod_{v=-4}^{n} f_v^{A_{v\mu}(c_\mu^{(1)})^2}} = \prod_{v=-4}^{n} f_v^{A_v'}.$$

Results that the equation (8) holds.

We must show that this extractor works in expected time $O(\frac{poly(n)}{\varepsilon(pk)})$. The step 1 takes constant time an in the steps 2-4 we must analyse the number of iterations expected until we get $n+1$ linearly independent vectors for which P^* can produce an accepting response.

Suppose the algorithm has found $|C| = l \in [1..n]$ acceptable challenges so far. Let $A :=$ "$C \cup \{\bar{c}\}$ is linearly independent" and $B :=$ "$\bar{c}$ produces acceptable response". There are q^l vectors in $\mathbb{Z}_q^{n+1}$ that are linearly dependent with vectors in C. Of those vectors $\frac{1}{q}$ start with 1. Therefore $\Pr[\bar{A}] = \frac{q^l \frac{1}{q}}{q^n} = \frac{q^{l-1}}{q^n}$. It is also known that $\Pr[B] = \varepsilon(pk)$. Then

$$\Pr[A \wedge B] = 1 - \Pr[\bar{A} \vee \bar{B}] = 1 - \Pr[\bar{A}] - \Pr[\bar{B}] + \Pr[\bar{A} \wedge \bar{B}] =$$

$$= 1 - \frac{q^{l-1}}{q^n} - (1 - \varepsilon(pk)) + \Pr[\bar{A} \wedge \bar{B}] \geq \varepsilon(pk) - \frac{q^{l-1}}{q^n} \geq \varepsilon(pk) - \frac{q^{n-1}}{q^n} = \varepsilon(pk) - \frac{1}{q} \approx \varepsilon(pk).$$

If the probability of getting a suitable $\bar{c}$ in the step 2 is at least $\varepsilon(pk)$ then the expected number of times a new challenge has to be picked is $\frac{1}{\varepsilon(pk)}$. To get $n+1$ such challenges takes expected time of $\frac{n+1}{\varepsilon(pk)}$.

Solving a linear system of equation in the step 6, can be done in time $O(n^3)$. The step 7 takes linear amount of time. Altogether K does $O(\frac{\varepsilon(pk)(n^3+n)+n+1}{\varepsilon(pk)})$ expected number of steps. □

In the previous lemma we have used n linearly independent challenges to extract $(A_{v\mu})_{v=-4}^{n}$ for $\mu \in [0..n]$ and $(A_v')_{v=-4}^{n}$. For each of these challenges $\bar{c}$ it holds that $r_v = \sum_{\mu=0}^{n} A_{v\mu} c_\mu$ and $r_v' = \sum_{i=1}^{n} A_{vi} c_\mu^2 + A_v'$ for $v = [-4..n]$. It could be that prover P^* can also generate responses that are not of this form. Next lemma will show that this is computationally difficult.

Lemma 4. Suppose there exists a knowledge extractor K that can extract $(A_{v,\mu})_{v=-4}^{n}$ for $\mu \in [0..n]$ and $(A_v')_{v=-4}^{n}$ in the expected polynomial-time such that equations the (7) and (8) hold. Let $(r_v)_{v=-4}^{n}, (r_v)_{v=-4}^{n}$ be a response from the prover P^* that satisfies the equations (14) and (15). If the discrete logarithm assumption holds then with overwhelming probability

$$r_v = \sum_{\mu=0}^{n} A_{v\mu} c_\mu$$

and

$$r'_v = \sum_{i=1}^{n} A_{vi} c_i^2 + A'_v$$

for $v \in [-4..n]$.

Proof. Suppose there exists an expected polynomial-time extractor K' such that it can extract an accepting response where

$$r_v \neq \sum_{\mu=0}^{n} A_{v\mu} c_\mu$$

or

$$r'_v \neq \sum_{i=1}^{n} A_{vi} c_\mu^2 + A'_v$$

for some $v \in [-4..n]$. Due to equation 14 and (7) we have

$$\prod_{v=-4}^{n} f_v^{r_v} = \prod_{\mu=0}^{n} f_\mu'^{c} = \prod_{\mu=0}^{n} \prod_{v=-4}^{n} f_v^{A_{v\mu} c} = \prod_{v=-4}^{n} f_v^{\sum_{\mu=0}^{n} A_{v\mu} c}.$$

Probability that K' can extract such a response in polynomial-time is negligible because extended Pedersen commitment is computationally binding in a discrete logarithm group.

Analogous argument can be made about $(r'_v)_{v=-4}^{n}$. Due to the equation (15), (7) and (8) we have

$$\prod_{v=-4}^{n} f_v^{r'_v} = \tilde{f}'_0 \prod_{i=1}^{n} f_i'^{c_i^2} = \prod_{v=-4}^{n} f_v^{A'_v} \prod_{i=1}^{n} \prod_{v=-4}^{n} f_v^{A_{v\mu} c^2} = \prod_{v=-4}^{n} f_v^{A'_v + \sum_{i=1}^{n} A_{v\mu} c_\mu^2}.$$

This also means that K' would be able to generate two openings to the same commitment. $\square$

In the next two lemmas it will be shown that A is a permutation matrix. For these lemmas we define polynomials

$$R_v := \sum_{i=0}^{n} A_{vi} X_i \in \mathbb{Z}_q[X_1, X_2, \ldots X_n]$$

for $v \in \{-4, -2, 1, 2, \dots, n\}$ where $X_0 = 1$ and

$$R'_{-3} := \sum_{i=1}^{n} A_{-3i} X_i^2 + A'_{-3} \in \mathbb{Z}_q[X_1, X_2, \dots X_n].$$

Lemma 5. Let the elements $(A_{v,\mu})_{v=-4}^{n} \in \mathbb{Z}_q^{n+5}$ for $\mu \in [0..n]$, $(A'_v)_{v=-4}^{n} \in \mathbb{Z}_q^{n+5}$ and $\omega \in \mathbb{Z}_q$ be such that for some $(c_i)_{i=1}^{n} \in \mathbb{Z}_q^n$ values

$$r_v = \sum_{\mu=0}^{n} A_{v\mu} c_\mu$$

for $v \in \{-2, 1, 2, \dots, n\}$ and

$$r'_{-3} = \sum_{i=1}^{n} A_{-3i} c_i^2 + A'_{-3}$$

satisfy equation (12). Then with overwhelming probability $(A_{v,\mu})_{v=1}^{n}$ for $\mu \in [1..n]$ satisfies the equation (1).

Proof. Let us look at the polynomial $p = \sum_{j=1}^{n} (R_j^3 - X_j^3) - (R_{-2} + R'_{-3} + \omega) \in \mathbb{Z}_q[X_1, X_2, \dots X_n]$. Due to the equation (12) and the assumption of this lemma we have $p(c_1, \dots, c_n) = 0$. According to the Schwartz-Zippel lemma with overwhelming probability p is a zero polynomial.

Now let us express the polynomial p in a different form

$$p = \sum_{j=1}^{n} (\sum_{i=0}^{n} A_{ji}^3 X_i^3 + 3 \sum_{\substack{i \neq k \\ i,k \in [0..n]}} A_{ji}^2 X_i^2 A_{jk} X_k + \\ + 6 \sum_{0 \le i < k < l \le n} A_{ji} A_{jk} A_{jl} X_i X_k X_l - X_j^3) - (R_{-2} + R'_{-3} + \omega) =$$

$$= \sum_{i=0}^{n} \sum_{j=1}^{n} A_{ji}^3 X_i^3 - \sum_{j=1}^{n} X_j^3 + 3 \sum_{j=1}^{n} \sum_{\substack{i \neq k \\ i,k \in [0..n]}} A_{ji}^2 X_i^2 A_{jk} X_k + \\ + 6 \sum_{j=1}^{n} \sum_{0 \le i < k < l \le n} A_{ji} A_{jk} A_{jl} X_i X_k X_l - (R_{-2} + R'_{-3} + \omega). \quad (*)$$

Let us look the first and the second addend separately

$$\sum_{i=0}^{n} \sum_{j=1}^{n} A_{ji}^3 X_i^3 - \sum_{j=1}^{n} X_j^3 = \sum_{i=1}^{n} (\sum_{j=1}^{n} A_{ji}^3 - \delta_{iii}) X_i^3 + \sum_{j=1}^{n} A_{j0}^3.$$

The third addend in the equation (*) can be expressed as

$$3\sum_{j=1}^{n}\sum_{\substack{i\neq k\\ i,k\in[0..n]}} A_{ji}^2X_i^2A_{jk}X_k = 3\sum_{\substack{i\neq k\\ i,k\in[0..n]}}\sum_{j=1}^{n}A_{ji}^2A_{jk}X_i^2X_k =$$

$$= 3\sum_{\substack{i\neq k\\ i,k\in[1..n]}}(\sum_{j=1}^{n}A_{ji}^2A_{jk}-\delta_{iik})X_i^2X_k+\sum_{k=1}^{n}\sum_{j=1}^{n}3A_{j0}^2A_{jk}X_k+\sum_{i=1}^{n}\sum_{j=1}^{n}3A_{ji}^2A_{j0}X_i^2.$$

The fourth addend in (*) can be expressed as

$$6\sum_{j=1}^{n}\sum_{0\le i<k<l\le n}A_{ji}A_{jk}A_{jl}X_iX_kX_l = 6\sum_{1\le i<k<l\le n}\sum_{j=1}^{n}A_{ji}A_{jk}A_{jl}X_iX_kX_l+$$

$$+6\sum_{1\le k<l\le n}\sum_{j=1}^{n}A_{j0}A_{jk}A_{jl}X_kX_l =$$

$$= 6\sum_{1\le i<k<l\le n}(\sum_{j=1}^{n}A_{ji}A_{jk}A_{jl}-\delta_{ikl})X_iX_kX_l+3\sum_{1\le k<l\le n}\sum_{j=1}^{n}A_{j0}A_{jk}A_{jl}X_kX_l+$$

$$3\sum_{1\le l<k\le n}\sum_{j=1}^{n}A_{j0}A_{jk}A_{jl}X_kX_l =$$

$$= 6\sum_{1\le i<k<l\le n}(\sum_{j=1}^{n}A_{ji}A_{jk}A_{jl}-\delta_{ikl})X_iX_kX_l+\sum_{\substack{l\neq k\\ l,k\in[1..n]}}\sum_{j=1}^{n}3A_{j0}A_{jk}A_{jl}X_kX_l.$$

The final term in the equation (*) can be written as

$$-(R_{-2}+R'_{-3}+\omega) = -\sum_{i=1}^{n}A_{-2i}X_i - A_{-20}-\sum_{i=1}^{n}A_{-3i}X_i^2-A'_{-3}-\omega$$

Putting all of the expression above together and by renaming some of the variables we get

$$p=\sum_{i=1}^{n}(\sum_{h=1}^{n}A_{hi}^3-\delta_{iii})X_i^3+3\sum_{\substack{i\neq k\\ i,k\in[1..n]}}(\sum_{h=1}^{n}A_{hi}^2A_{hk}-\delta_{iik})X_i^2X_k+$$

$$+6\sum_{1\le i<k<l\le n}(\sum_{h=1}^{n}A_{hi}A_{hk}A_{hl}-\delta_{ikl})X_iX_kX_l+\sum_{l,k\in[1..n]}(\sum_{h=1}^{n}3A_{h0}A_{hk}A_{hl}-A_{-3k}\delta_{lk})X_kX_l+$$

$$+\sum_{k=1}^{n}(\sum_{h=1}^{n}3A_{h0}^2A_{hk}-A_{-2k})X_k+(\sum_{h=1}^{n}A_{h0}^3-A_{-20}-A'_{-3})-\omega=0.$$

Because this is a zero polynomial, coefficients of all the terms are zero. Therefore for any $i, k, l \in [1..n]$ we have

$$\sum_{h=1}^{n} A_{hi} A_{hk} A_{hl} = \delta_{ikl}.$$

Thus the equation (1) holds. In addition we have

$$\omega = \sum_{h=1}^{n} A_{h0}^3 - A_{-20} - A'_{-3}$$

$$A_{-2k} = \sum_{h=1}^{n} 3A_{h0}^2 A_{hk}$$

for any $k \in [1..n]$ and

$$A_{-3k}\delta_{lk} = \sum_{h=1}^{n} 3A_{h0} A_{hk} A_{hl}$$

for any $k, l \in [1..n]$. In particular for $l = k$ we have $A_{-3k} = \sum_{h=1}^{n} 3A_{h0} A_{hk}^2$. If A is a permutation matrix then $A_{-3k} = \sum_{h=1}^{n} 3A_{h0} A_{hk}$. □

Lemma 6. Let the elements $(A_{v,\mu})_{v=-4}^{n} \in \mathbb{Z}_q^{n+5}$ for $\mu \in [0..n]$ and $\dot{\omega} \in \mathbb{Z}_q$ be such that for some $(c_i)_{i=1}^{n} \in \mathbb{Z}_q^n$ values

$$r_v = \sum_{\mu=0}^{n} A_{v\mu} c_\mu$$

for $v \in \{-4, 1, 2, \ldots, n\}$ can satisfy the equation (13). Then with overwhelming probability $(A_{v,\mu})_{v=1}^{n}$ for $\mu \in [1..n]$ satisfies the equation (2).

Proof. Proof is analogous to the proof of the previous lemma. Let us look at the polynomial

$$p = \sum_{h=1}^{n} (R_h^2 - X_h^2) - (R_{-4} + \dot{\omega}) \in \mathbb{Z}_q[X_1, X_2, \ldots, X_n].$$

As in the previous lemma, according to the Schwartz-Zippel lemma $p = 0$.

Polynomial p can be expressed as

$$p = \sum_{h=1}^{n} (R_h^2 - X_h^2) - (R_{-4} + \dot{\omega}) = \sum_{h=1}^{n} (\sum_{i=0}^{n} A_{hi}^2 X_i^2 + 2 \sum_{0 \le i < j \le n} A_{hi} A_{hj} X_i X_j - X_h^2) - (R_{-4} + \dot{\omega}) =$$

$$= \sum_{i=1}^{n}(\sum_{h=1}^{n} A_{hi}^2 - \delta_{ii})X_i^2 + 2\sum_{1\le i<j\le n}\sum_{h=1}^{n} A_{hi}A_{hj}X_iX_j + \sum_{j=1}^{n}\sum_{h=1}^{n} 2A_{h0}A_{hj}X_j + \sum_{h=1}^{n} A_{h0}^2 - (R_{-4} + \dot{\omega}) =$$

$$= \sum_{i=1}^{n}(\sum_{h=1}^{n} A_{hi}^2 - \delta_{ii})X_i^2 + 2\sum_{1\le i<j\le n}(\sum_{h=1}^{n} A_{hi}A_{hj} - \delta_{ij})X_iX_j + \sum_{j=1}^{n}(\sum_{h=1}^{n} 2A_{h0}A_{hj} - A_{-4j})X_j +$$
$$+ (\sum_{h=1}^{n} A_{h0}^2 - A_{-40} - \dot{\omega}).$$

All the coefficient in this polynomial are zero, thus for any $i, j \in [1..n]$ we have

$$\sum_{h=1}^{n} A_{hi}A_{hj} = \delta_{i,j}.$$

Therefore property 2 holds. In addition

$$\dot{\omega} = \sum_{h=1}^{n} A_{h0}^2 - A_{-40}$$

and

$$A_{-4j} = \sum_{h=1}^{n} 2A_{h0}A_{hj}$$

for $j \in [1..n]$. $\square$

Lemma 7. Let $((g_v, m_v))_{v=0}^{n}$ and $(A_{v,\mu})_{v=-4}^{n} \in \mathbb{Z}_q^{n+5}$ for $\mu \in [0..n]$ be some fixed values. If for some $(c_i)_{i=1}^{n} \in \mathbb{Z}_q^n$ values

$$r_v = \sum_{\mu=0}^{n} A_{v\mu}c_\mu$$

for $v \in [0..n]$ satisfy the equations (10) and (11) then with overwhelming probability

$$g'_\mu = \prod_{v=0}^{n} g_v^{A_{v\mu}} \tag{A}$$

$$m'_\mu = \prod_{v=0}^{n} m_v^{A_{v\mu}} \tag{B}$$

for $\mu \in [0..n]$.

Proof. The equation (10) holds if and only if

$$1 = \frac{\prod_{v=0}^{n} g_v^{r_v}}{\prod_{\mu=0}^{n} (g'_\mu)^{c_\mu}} = \prod_{\mu=0}^{n} \left(\frac{\prod_{v=0}^{n} g_v^{A_{v\mu}}}{g'_\mu} \right)^{c_\mu}.$$

Let us denote $x_\mu := \log_{g_0} \frac{\prod_{v=0}^{n} g_v^{A_{v\mu}}}{g'_\mu}$ for $\mu \in [0..n]$. Then

$$g^0 = \prod_{\mu=0}^{n} g_0^{x_\mu c_\mu} = g_0^{\sum_{\mu=0}^{n} x_\mu c_\mu}.$$

Thus

$$\sum_{\mu=0}^{n} x_\mu c_\mu = 0.$$

According to the Schwartz-Zippel lemma with overwhelming probability $x_0 = x_1 = \ldots = x_n = 0$ so the equation (12) holds. The equation (B) holds with similar argument for the equation (11). $\square$

In the proof of the next theorem we will combine all the lemmas of this section to show that the shuffle protocol has knowledge soundness.

Theorem 4. If the DL assumption holds then the shuffle protocol has knowledge soundness.

Proof. Suppose there exists a prover P^* that can make the verifier V accept with probability $\varepsilon(pk) > 0$.

We construct a knowledge extractor K' that can extract in the expected time $O(poly(n)/\varepsilon(pk))$ the permutation matrix and the randomizers that were used in shuffling process.

The prover commits values g'_0, m'_0, $\tilde{f}'_0$, $(f'_\mu)_{\mu=0}^{n}$, ω and $\dot{\omega}$. We set a rewinding point for P^* after the commitment.

According to the lemma 3 there exists an extractor algorithm K that can in an expected number of steps $O(\frac{poly(n)}{\varepsilon(pk)})$ extract $(A_{v,\mu})_{v=-4}^{n} \in \mathbb{Z}_q^{n+5}$ for $\mu \in [0..n]$ and $(A'_v)_{v=-4}^{n}$ that satisfy the equations (7) and (8). We apply extractor K to the prover P^*.

In the $1/\varepsilon(pk)$ expected number of steps we can extract an accepting response $(r_v)_{v=-4}^{n}$, $(r'_v)_{v=-4}^{n}$ for a uniformly random challenge $(c_i)_{i=1}^{n} \leftarrow_r \mathbb{Z}_q^n$. According to the lemma 2 the reponse satisfies equations (14) and (15). If the discrete logarithm assumption holds then according to the lemma 4 with overwhelming probability

$$r_v = \sum_{\mu=0}^{n} A_{v\mu} c_\mu$$

$$r'_v = \sum_{i=1}^{n} A_{vi} c_i^2 + A'_v$$

for $v \in [-4..n]$.

According to the lemma 5 and 6 with overwhelming probability $A := (A_{v\mu})$ for $v, \mu \in [1..n]$ satisfies the equations (1) and (2). Due to the theorem 1 the matrix A is a permutation matrix.

According to the lemma 7 the matrix A and the vector$(A_{0\mu})_{\mu=0}^{n}$ satisfy the equation (3). Thus extractor K' has extracted a permutation matrix A and randomizers $(A_{0\mu})_{\mu=0}^{n}$ used for shuffling.

$\square$

Theorem 5. The shuffle protocol is special honest-verifier zero-knowledge.

Proof. Let $(c_i)_{i=1}^{n} \in \mathbb{Z}_q^n$ be an arbitrary challenge. We first show that using this challenge it is possible to create an accepting view in polynomial-time.

We pick uniformly randomly $(r_v)_{v=-4}^{n}, (r'_v)_{v=-4}^{n} \leftarrow_r \mathbb{Z}_q^{n+5}$ and $(f'_i)_{i=1}^{n} \leftarrow_r G^n$. We generate values $\tilde{f}'_0, f'_0, g'_0, m'_0, \omega$ and $\dot{\omega}$ such that the verification equations hold. We take

$$g'_0 = \frac{\prod_{v=0}^{n} g_v^{r_v}}{\prod_{\mu=1}^{n} (g'_\mu)^{c_\mu}} \qquad m'_0 = \frac{\prod_{v=0}^{n} m_v^{r_v}}{\prod_{\mu=1}^{n} (m'_\mu)^{c_\mu}}$$

$$\omega = \sum_{j=1}^{n} (r_j^3 - c_j^3) - r_{-2} - r'_{-3} \qquad \dot{\omega} = \sum_{j=1}^{n} (r_j^2 - c_j^2) - r_{-4}.$$

and as a result the equations (10), (11), (12) and (13) hold. We pick $\tilde{f}'_0$ and f'_0 such that the equations (14) and (15) hold i.e.

$$f'_0 = \frac{\prod_{v=-4}^{n} f_v^{r_v}}{\prod_{\mu=1}^{n} (f'_\mu)^{c_\mu}} \qquad \tilde{f}_0{}' = \frac{\prod_{v=-4}^{n} f_v^{r'_v}}{\prod_{\mu=1}^{n} (f'_\mu)^{c_\mu^2}}.$$

As it is shown in the proof of the lemma 2, the equation (9) holds if and only if

$$\frac{\prod_{v=-4}^{n} f_v^{r_v}}{\prod_{\mu=0}^{n} (f'_\mu)^{c_\mu}} = \left(\frac{\tilde{f}_0{}' \prod_{i=1}^{n} (f'_i)^{c_i^2}}{\prod_{v=-4}^{n} f_v^{r'_v}} \right)^{\alpha}.$$

Previous equation holds thus also equation (9) holds. We have generated an accepting view for challenge $(c_i)_{i=1}^{n}$.

Secondly we show that for uniformly random challenge $(c_i)_{i=1}^{n}$, simulated conversation has exactly the same distribution as the conversation between honest prover and honest verifier.

Values $(r_v)_{v=-4}^{n}$ and $(r'_v)_{i=-4}^{n}$ are in the real protocol uniformly random and independent elements in $\mathbb{Z}_q$ because they are hidden respectively by $(A_{v0})_{v=-4}^{n}$ and $(A'_v)_{v=-4}^{n}$. Values $(f'_\mu)_{\mu=1}^{n}$ are uniformly random and independent elements in G in the real conversation because $(A_{-1i})_{i=1}^{n}$ hides them.

Values g'_0, m'_0, $\tilde{f}'_0$, f'_0, ω and $\dot{\omega}$ depend on $(r_v)_{v=-4}^n$, $(r'_v)_{i=-4}^n$ and $(f'_\mu)_{\mu=1}^n$ the same way in the real conversation and the simulated conversation. Thus they are identically distributed.

$\square$

3.3 Ideas for Improvement

One of the goals of this thesis is to see whether the Furukawa shuffle argument can be improved. We suggest two simple variations on the shuffle protocol and then propose two new approaches how to characterize a permutation matrix.

Let us analyse the complexity of the current protocol. In the equation (7) prover does no exponentiations for $f_1^{A_{1\mu}}, \dots, f_n^{A_{n\mu}}$ as the exponent is 0 or 1 and one exponentiation for each of $f_{-4}^{A_{-4\mu}}, f_{-3}^{A_{-3\mu}}, f_{-2}^{A_{-2\mu}}, f_{-1}^{A_{-1\mu}}$ and $f_0^{A_{0\mu}}$ for $\mu \in [0..n]$. That constitutes for roughly $5n$ exponentiation. First three of them are needed to cancel out redundant terms in the equations (12) and (13). Using different characterization of a permutation matrix or different verification equations could reduce the exponentiations. Computing values g_0, m_0, $\tilde{f}_0$ takes roughly $3n$ exponentiations. Value $\tilde{f}_0$ is needed to verify that $(r'_v)_{v=-4}^n$ is correctly computed. Values $(r'_v)_{v=-4}^n$ are sent because the equation (12) needs r'_{-3} to cancel out one quadratic term. Again it would beneficial if the verification equations would not have as many terms that need to be canceled.

Altogether the prover has to do $8n$ exponentiations. The verifier does $2n$ exponentiations in each of the equations (9), (10) and (11). In commitment phase the prover has to send roughly n elements of G. In challenge and response phase $3n$ elements of $\mathbb{Z}_q$ are sent. The communication complexity will be $n \log p + 3n \log q$ bits.

New Characterizations

Let us notice the following fact. Let $A = (A_{i,j}) \in \mathbb{Z}_q^{n \times n}$. If we interpret the property $\sum_{h=1}^n A_{hi} A_{hj} = \delta_{i,j}$ for $i, j \in [1..n]$ as a matrix product then we have $A^T A = I$. From linear algebra we know that if a square matrix is left invertible then it is also right invertible and those inverses are the same. Then we will $A^T A = A A^T = I$. This is a well-known class of orthogonal matrices.

This fact was also noticed in [GL07]. In [GL07] they modify Furukawa shuffle argument using the following theorem.

Theorem 6. Let $A = (A_{i,j})$ be an $n \times n$ integer matrix. If $A^T A = I$ and for $i \in [1..n]$

$$\sum_{h=1}^n A_{hi} = 1$$

then A is a permutation matrix.

Proof. Because $A^T A = I$ we have

$$\sum_{h=1}^{n} A_{hi}^2 = 1$$

for $i \in [1..n]$. This is only possible if the i-th column has exactly one nonzero element that is either 1 or -1. Property $\sum_{h=1}^{n} A_{hi} = 1$ guarantees that the nonzero element is 1. There cannot be any zero rows because the matrix A is regular. Then A is a permutation matrix. □

Problem with this property is that we need to use integer commitment schemes which are much less efficient. Over the field $\mathbb{Z}_q$ this characterization does not hold.

Next we will present two new characterizations.

Theorem 7. Let q be a prime. Matrix $A \in \mathbb{Z}_q^{n\times n}$ is a permutation matrix if and only if for any $i, j, k \in [1..n]$

$$\sum_{h=1}^{n} A_{h,i}A_{h,j}A_{h,k} = \delta_{i,j,k}$$

and

$$\sum_{h=1}^{n} A_{h,i} = 1.$$

Proof. According to the lemma 1 each row and column of the matrix A has exactly one nonzero element. The second property gives that the nonzero elements are equal to 1. Thus A is a permutation matrix. □

Both in [Fur05] and [GL07] it seems to have gone unnoticed that the equation (1) in the theorem 1 can be made weaker.

Theorem 8. Let $n \le q$ and $A \in \mathbb{Z}_q^{n\times n}$. Then A is a permutation matrix if and only if A is orthogonal and for any $i, j \in [1..n]$

$$\sum_{h=1}^{n} A_{h,i}A_{h,j}^2 = \delta_{i,j}. \tag{*}$$

Proof. $\Rightarrow$) Follows from theorem 1.

$\Leftarrow$) If we look the equation (*) as a matrix multiplication then $A^T(A \circ A) = I$. Multiplying by A from left gives

$$A \circ A = A.$$

Then

$$A_{i,j}^2 = A_{i,j} \Rightarrow A_{i,j}(A_{i,j} - 1) = 0$$

and thus $A_{i,j} \in \{0, 1\}$ for any $i, j \in [1..n]$. Considering that the matrix A is both orthogonal and boolean matrix we have

$$1 = \sum_{h=1}^{n} A_{hi}^2 = \sum_{h=1}^{n} A_{hi}$$

for $i \in [1..n]$. As $n \leq q$ we have that $\sum_{h=1}^{n} A_{hi}$ cannot wrap around and so the i-th column has one 1 and all the other elements are 0. The matrix A is regular therefore there cannot be any zero rows. Then A is a permutation matrix. $\square$

It is not clear if this weaker property allows to construct a more efficient shuffle protocol.

Variation 1

According to the theorem 2, matrix $A \in \mathbb{Z}_q^{n \times n}$ is a permutation if the equation (1) holds and $q \equiv 2 \pmod 3$. In [Fur05] this property is used for the shuffle-decryption protocol but not for the shuffle protocol. The article does not explain why this is not used for the shuffle protocol.

If we take $q \equiv 2 \pmod 3$ then we do not need the equation (13). Then r_{-4} is not needed, it means that in all of the equations where we previously had indices $v \in [-4..n]$ we now have $v \in [-3..n]$. In the equation (7) we do not need to compute $f_{-4}^{A_{-4\mu}}$ for $\mu \in [0..n]$. Then prover's computation is $7n$ exponentiations instead of $8n$ exponentiations. Soundness and SHVZK proofs are almost indentical.

Variation 2

If we do not add the extra restriction on q, it is still possible to reduce the prover's computation by n exponentiations.

We remove the values related to r_{-4} as in variation 1. We change how random values A_{v0} used for hiding r_v, are picked. For $v \in [-3..n-1]$ we pick $A_{v0} \leftarrow_r \mathbb{Z}_q$ as before but $A_{n0} = -\sum_{h=1}^{n-1} A_{h0}$.

We add the verification equation $\sum_{h=1}^{n} r_h - c_h = 0$. Let us see that modified protocol is sound.

$$\sum_{h=1}^{n} r_h - c_h = \sum_{h=1}^{n}\sum_{\mu=0}^{n} A_{h\mu}c_\mu - \sum_{h=1}^{n} c_h = \sum_{h=1}^{n} A_{h0} + \sum_{\mu=1}^{n}\sum_{h=1}^{n} A_{h\mu}c_\mu - \sum_{h=1}^{n} c_h =$$

$$= \sum_{\mu=1}^{n}(\sum_{h=1}^{n} A_{h\mu} - 1)c_\mu$$

According to the Schwartz-Zippel lemma $\sum_{h=1}^{n} A_{h\mu} = 1$ with overwhelming probability for $\mu \in [1..n]$.

Using the theorem 7 gives that A is a permutation matrix. Rest of the soundness proof is the same as before.

The modified protocol is SHVZK. Difference with simulation in the theorem 5 is the generation of r_v. Simulator picks randomly $r_v \leftarrow_r \mathbb{Z}_q$ for $v \in [-3..n-1]$. The value r_n is generated as

$$r_n = \sum_{h=1}^{n} c_h - \sum_{h=1}^{n-1} r_h.$$

It is clear that the new verification equation holds. Also $(r_v)_{v=1}^{n}$ has the same distribution as for the honest P and V.

4 Furukawa Shuffle-Decryption Argument

In practice mix-network should work as follows, users send messages that are encrypted with mix-networks public key of the first mixer. Output from the last mixer should be decrypted ciphertexts in randomized order.

Giving private key to one particular mixer imposes a vulnerability. Instead private key should be secret shared between the mixers and each mixer does partial decryption with it's share.

In order to prove the correctness of shuffling, each mixer should also prove correctness of decryption. Shuffling and decryption proofs can be done separately. It might however be more efficient to prove them together.

We present shuffle-decryption protocol from [Fur05]. It combines argument of shuffling in previous section and decryption argument that will be presented in the next subsection. Some of the computation in those arguments overlaps and so the combined protocol is more efficient then performing two arguments separately.

This protocol is not HVZK because it is impossible to simulate shuffled state of ciphertexts. It can be shown that shuffle-decryption satisfies weaker privacy requirement which is called complete permutation hiding (CPH). Weaker security definition allows us to reduce computation complexity by removing some of the random values.

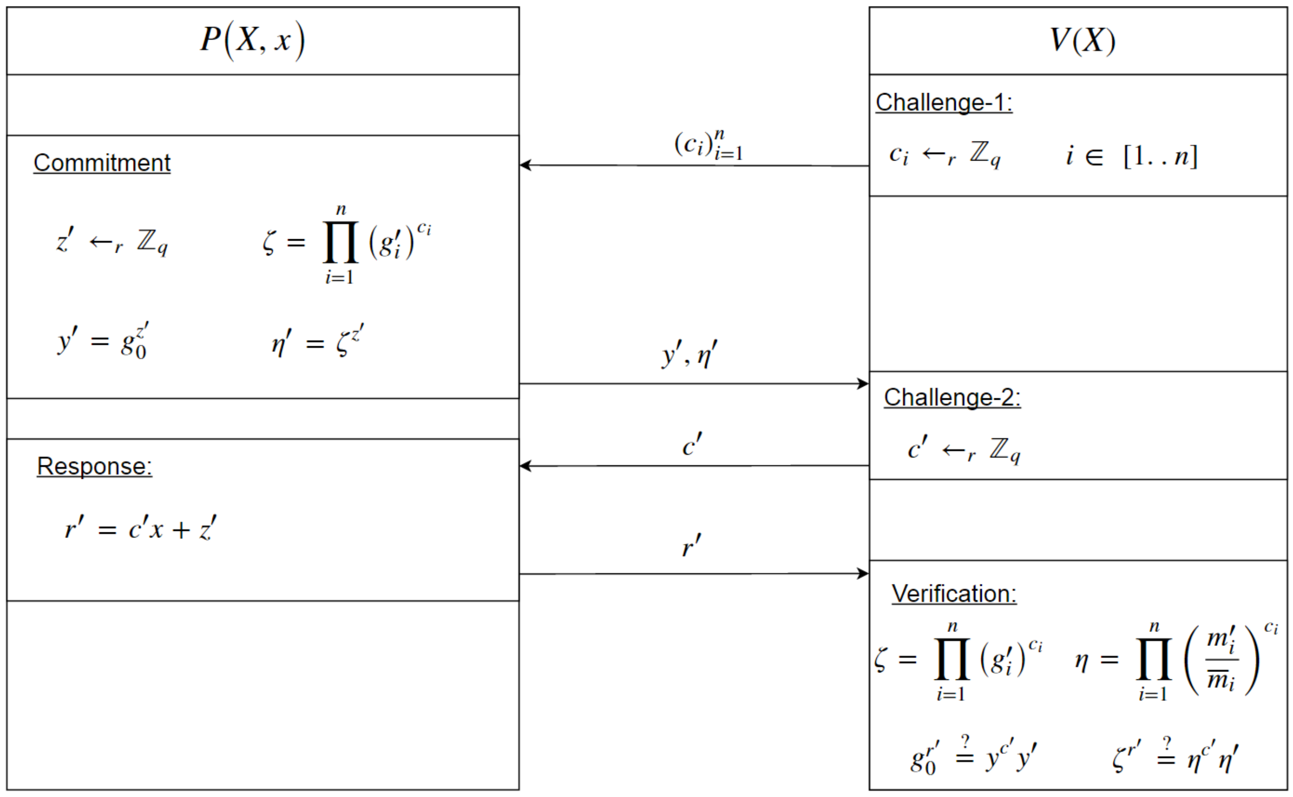
4.1 Decryption Protocol

We describe the decryption argument that we later combine with the shuffle argument. Decryption argument is based on the Schnorr's protocol [Sch91].

Let p and q be large primes such that $q|p-1$. Let G be an order q subgroup of a multiplicative group $\mathbb{Z}_q^*$ and g_0 is a generator of G. Let x be an ElGamal private key and $y = g_0^x$ the corresponding public key. We have ElGamal ciphertexts $(g_i', \bar{m}_i)_{i=1}^{n}$ and

$(g_i', m_i')_{i=1}^n$. Prover must convince verifier that (g_i', m_i') is decryption of $(g_i', \bar{m}_i)$ without revealing anything about the private key x.

We define $X := (p, q, g_0, y, (g_i', \bar{m}_i)_{i=1}^n, (g_i', m_i')_{i=1}^n)$ where the sequence X is a common input to the prover and the verifier. The prover knows in addition the private key x.


$P(X, x)$
$V(X)$
Challenge-1:
$c_i \leftarrow_r \mathbb{Z}_q \quad i \in [1..n]$
$(c_i)_{i=1}^n$
Commitment
$z' \leftarrow_r \mathbb{Z}_q$
$\zeta = \prod_{i=1}^n (g_i')^{c_i}$
$y' = g_0^{z'}$
$\eta' = \zeta^{z'}$
y', η'
Challenge-2:
$c' \leftarrow_r \mathbb{Z}_q$
c'
Response:
$r' = c'x + z'$
r'
Verification:
$\zeta = \prod_{i=1}^n (g_i')^{c_i} \quad \eta = \prod_{i=1}^n \left(\frac{m_i'}{\bar{m}_i}\right)^{c_i}$
$g_0^{r'} \overset{?}{=} y^{c'} y' \quad \zeta^{r'} \overset{?}{=} \eta^{c'} \eta'$

Theorem 9. Decryption protocol is complete, sound and HVZK.

Proof. It is easy to verify by substituting the correct values to the verification equation that this protocol is complete.

Next we will show that the presented protocol is sound. The extractor sends $(c_i)_{i=1}^n$ to the prover. The prover responds with sending y', η' to the extractor. The extractor sets a rewinding point and rewinds the prover until it gets two accepting responses r_1' and r_2' such that corresponding challenges c_1' and c_2' are not equal. This can be done in expected polynomial-time if the prover is successful with non-negligible probability.

Then we have

$$g_0^{r_1'} = y^{c_1'} y'$$
$$g_0^{r_2'} = y^{c_2'} y'.$$

Dividing one the mentioned equations by the other gives

$$g_0^{r_1' - r_2'} = y^{c_1' - c_2'} \Rightarrow y = g_0^{\frac{r_1' - r_2'}{c_1' - c_2'}}.$$

Thus we have extracted $x = \frac{r'_1 - r'_2}{c'_1 - c'_2}$.

Similarly from the equations $\zeta^{r'_1} = \eta^{c'_1}\eta'$ and $\zeta^{r'_2} = \eta^{c'_2}\eta'$ we can get

$$\eta = \zeta^{\frac{r'_1 - r'_2}{c'_1 - c'_2}} = \zeta^x.$$

Therefore

$$\prod_{i=1}^{n} \left(\frac{m'_i}{\bar{m}_i}\right)^{c_i} = \prod_{i=1}^{n} (g_i'^x)^{c_i}$$

$$\prod_{i=1}^{n} \left(\frac{m'_i/\bar{m}_i}{g_i'^x}\right)^{c_i} = 1.$$

Using Schwartz-Zippel similarly to lemma 7, we get

$$g_i'^x = \frac{m'_i}{\bar{m}_i}$$

for $i \in [1..n]$. It follows that the protocol is sound.

Finally we show that the protocol is HVZK. We pick uniformly randomly $c_1, c_2, \ldots, c_n, c', r' \leftarrow_r \mathbb{Z}_q$. The values y' and η' are picked such that the verification equations would hold i.e.

$$y' = \frac{g_0^{r'}}{y^{c'}} \qquad\qquad \eta' = \frac{\zeta^{r'}}{\eta^{c'}}.$$

As they depend on the values r' and c' the same way as in the real conversation between the prover and the verifier, we get that the simulated messages have the same distribution as in the honest conversation. $\square$

4.2 Description

Let p and q be primes such that $q | p-1$ and $q \equiv 2 \pmod 3$. Let G be an order q subgroup of $\mathbb{Z}_p^*$, g_0 be a generator of G and $m_0 \in G$ be an ElGamal public key. Element $y = g_0^x \in G$ is mixer's personal public key and $x \in \mathbb{Z}_q$ is the corresponding private key.

Let $(g_i, m_i)_{i=1}^n$ and $(g'_i, m'_i)_{i=1}^n$ be the ElGamal ciphertexts. The mixer picks randomly $(A_{0,i})_{i=1}^n \leftarrow_r \mathbb{Z}_q^n$ and a permutation matrix $A = (A_{i,j}) \in \mathbb{Z}_q^{n\times n}$. The mixer must prove that the ciphertexts were both shuffled and partially decrypted with the private key x i.e.

$$(g'_i, m'_i) = (g_0^{A_{0,i}} \prod_{j=1}^{n} g_j^{A_{ji}}, g_i'^{-x} \prod_{j=1}^{n} m_j^{A_{ji}})$$

for $i \in [1..n]$.

Because the shuffle-decryption protocol satisfies a weaker definition of privacy, we may omit f_{-1} from the shuffle protocol. Using $q \equiv 2 \pmod 3$ allows us to omit f_{-4} as well because the equation (13) is not needed anymore. Renaming elements f_i gives that we need $F_n := (f_i)_{i=-2}^{n} \leftarrow_r (G \setminus \{1\})^{n+3}$ for the extended Pedersen commitment.

We denote $X_\kappa = (p, q, y, g_0, m_0, F_n, (g_i, m_i)_{i=1}^{n}, (g'_i, m'_i)_{i=1}^{n})$ and $W_\kappa = (x, (A_{i,j})_{i,j \in [1..n]}, (A_{0i})_{i=1}^{n})$. Here κ is a security parameter. Length of the primes p and q is polynomial in κ. X_κ is the common input to the prover and the verifier and W_κ is the witness for the prover.

$P(X_\kappa, W_\kappa)$ | $V(X_\kappa)$

Commitment-1:

$$A_{v0}, A'_v \leftarrow_r \mathbb{Z}_q \qquad v \in [-2..n]$$

$$\left.\begin{aligned} A_{-1i} &= \sum_{j=1}^{n} 3A_{j0}A_{ji} \\ A_{-2i} &= \sum_{j=1}^{n} 3A_{j0}^2 A_{ji} \end{aligned}\right\} i \in [1..n]$$

$$f'_\mu = \prod_{v=-2}^{n} f_v^{A_{v\mu}} \qquad \mu \in [0..n]$$

$$\widetilde{f}'_0 = \prod_{v=-2}^{n} f_v^{A'_v}$$

$$g'_0 = \prod_{v=0}^{n} g_v^{A_{v0}} \qquad m'_0 = \prod_{v=0}^{n} m_v^{A_{v0}}$$

$$\omega = \sum_{j=1}^{n} A_{j0}^3 - A_{-20} - A'_{-1}$$

$$\xrightarrow{g'_0,\, m'_0,\, \widetilde{f}'_0,\, (f'_\mu)_{\mu=0}^{n},\, \omega}$$

Challenge-1:

$$c_i \leftarrow_r \mathbb{Z}_q \qquad i \in [1..n]$$

$$\xleftarrow{(c_i)_{i=1}^{n}}$$

Response-1:

$$\left.\begin{aligned} r_v &= \sum_{\mu=0}^{n} A_{v\mu} c_\mu \\ r'_v &= \sum_{i=1}^{n} A_{vi} c_i^2 + A'_v \end{aligned}\right\} \begin{aligned} &v \in [-2..n] \\ &(c_0 = 1) \end{aligned}$$

$$\xrightarrow{(r_v)_{v=-2}^{n},\, (r'_v)_{v=-2}^{n}}$$

Commitment-2

$$\zeta = \prod_{i=1}^{n} (g'_i)^{c_i} \qquad \beta \leftarrow_r \mathbb{Z}_q$$

$$\eta = \zeta^{x'} \qquad \eta' = \zeta^{\beta} \qquad y' = g_0^{\beta}$$

$$\xrightarrow{\eta,\, \eta',\, y'}$$

Challenge-2:

$$c' \leftarrow_r \mathbb{Z}_q$$

$$\xleftarrow{c'}$$

Response-2:

$$r' = c'x' + \beta$$

$$\xrightarrow{r'}$$

Verification:

$$\alpha \leftarrow_r \mathbb{Z}_q \qquad \zeta = \prod_{i=1}^{n} (g'_i)^{c_i}$$

$$\prod_{v=-2}^{n} f_v^{r_v + \alpha r'_v} \overset{?}{=} f'_0 \left(\widetilde{f}'_0\right)^{\alpha} \prod_{i=1}^{n} (f'_i)^{c_i + \alpha c_i^2} \tag{16}$$

$$\prod_{v=0}^{n} g_v^{r_v} \overset{?}{=} \zeta g'_0 \tag{17}$$

$$\prod_{v=0}^{n} m_v^{r_v} \overset{?}{=} \eta \prod_{\mu=0}^{n} (m'_\mu)^{c_\mu} \tag{18}$$

$$\sum_{h=1}^{n} \left(r_h^3 - c_h^3\right) \overset{?}{=} r_{-2} + r'_{-1} + \omega \tag{19}$$

$$g_0^{r'} \overset{?}{=} y^{c'} y' \quad (20) \qquad \zeta^{r'} \overset{?}{=} \eta^{c'} \eta' \quad (21)$$

4.3 Soundness

Theorem 10. The shuffle-decryption protocol is complete.

Proof. Similar to checking completeness of the shuffle protocol and the decryption protocol. □

Soundness proof is also very similar to the soundness proof of the shuffle protocol.

Lemma 8. If the equation (16) holds then the probability that

$$\prod_{v=-2}^{n} f_v^{r_v} = \prod_{\mu=0}^{n} (f'_\mu)^{c_\mu} \tag{14}$$

and

$$\prod_{v=-2}^{n} f_v^{r'_v} = \tilde{f}'_0 \prod_{\mu=1}^{n} (f'_\mu)^{c_\mu^2} \tag{15}$$

hold is $1 - \frac{1}{q}$.

Proof. Similar to the proof of lemma 2. □

Lemma 9. If P^* can make V accept the protocol with probability $\varepsilon(\kappa) > 0$ then there exists an extractor K that can extract from P^* the values $(A_{v,\mu})_{v=-2}^{n}$ for $\mu \in [0..n]$ and $(A'_v)_{v=-2}^{n}$ in expected time $O(\frac{poly(n)}{\varepsilon(\kappa)})$. The extracted values satisfy equations

$$f'_\mu = \prod_{v=-2}^{n} f_v^{A_{v\mu}} \tag{22}$$

$$\tilde{f}'_0 = \prod_{v=-2}^{n} f_v^{A'_v} \tag{23}$$

for $\mu \in [0..n]$.

Proof. Similar to the proof of lemma 3. □

Lemma 10. Suppose there exists a knowledge extractor K that can extract $(A_{v,\mu})_{v=-2}^{n}$ for $\mu \in [0..n]$ and $(A'_v)_{v=-2}^{n}$ in expected polynomial-time such that the equations (22) and (23) hold. Let $(r_v)_{v=-2}^{n}, (r_v)_{v=-2}^{n}$ be a response from the prover P^* that satisfies the equations (14) and (15). If the discrete logarithm assumption holds then with overwhelming probability we have

$$r_v = \sum_{\mu=0}^{n} A_{v\mu} c_\mu$$

and

$$r'_v = \sum_{i=1}^{n} A_{vi} c_i^2 + A'_v$$

for $v \in [-2..n]$.

Proof. Similar to the proof of lemma 4. $\square$

Lemma 11. Let the elements $(A_{v,\mu})_{v=-2}^{n} \in \mathbb{Z}_q^{n+3}$ for $\mu \in [0..n]$, $(A'_v)_{v=-2}^{n} \in \mathbb{Z}_q^{n+3}$ and $\omega \in \mathbb{Z}_q$ be such that for some $(c_i)_{i=1}^{n} \in \mathbb{Z}_q^n$ values

$$r_v = \sum_{\mu=0}^{n} A_{v\mu} c_\mu$$

for $v \in \{-2, 1, 2, \ldots, n\}$ and

$$r'_{-2} = \sum_{i=1}^{n} A_{-2i} c_i^2 + A'_{-2}$$

satisfy the equation (19). Then with overwhelming probability $(A_{v,\mu})_{v=1}^{n}$ for $\mu \in [1..n]$ satisfies the equation $\sum_{h=1}^{n} A_{hi} A_{hj} A_{hk} = \delta_{i,j,k}$ for any $i, j, k \in [1..n]$.

Proof. Similar to the proof of lemma 5. $\square$

Lemma 12. Let $((g_v, m_v))_{v=0}^{n}$ and $(A_{v,\mu})_{v=-2}^{n} \in \mathbb{Z}_q^{n+3}$ for $\mu \in [0..n]$ be some fixed values. If for some $(c_i)_{i=1}^{n} \in \mathbb{Z}_q^n$ values

$$r_v = \sum_{\mu=0}^{n} A_{v\mu} c_\mu$$

for $v \in [0..n]$ satisfy the equation (17) then with overwhelming probability

$$g'_\mu = \prod_{v=0}^{n} g_v^{A_{v\mu}}$$

for $\mu \in [0..n]$.

Proof. Similar to the proof of lemma 7. $\square$

Lemma 13. Suppose the prover P^* can make the verifier V accept with non-negligible probability. Then there exists an extractor $\bar{K}$ that can extract in expected polynomial-time x' such that $y = g_0^{x'}$.

Proof. Similar to the extractor in the theorem 9. $\square$

Lemma 14. Let $y = g_0^{x'}$. If the equations (20) and (21) hold then with overwhelming probability

$$\eta = \zeta^{x'}.$$

Proof. If (20) holds then

$$g_0^{r'} = y^{c'} y' = g_0^{x'c'} y' \Rightarrow y' = g_0^{r' - x'c'} =: g_0^{\beta}.$$

Then $r' = x'c' + \beta$ where β is some constant independent of c'.

Because the equation (21) holds we have

$$\eta^{c'} \eta' = \zeta^{r'} = \zeta^{c'x' + \beta}.$$

This can be expressed as

$$\left(\frac{\eta}{\zeta^{x'}}\right)^{c'} \cdot \frac{\eta'}{\zeta^{\beta}} = 1.$$

Using the Schwartz-Zippel similarly as in the lemma 3 gives that with overwhelming probability $\eta = \zeta^{x'}$. □

Lemma 15. Assume $(g_v, m_v)_{v=0}^n$, $(g'_v)_{v=1}^n$, $(A_{v\mu})_{v,\mu \in [0..n]}$ and x' are given. For a given $(c_i)_{i=1}^n$ values

$$r_v = \sum_{\mu=0}^{n} A_{v\mu} c_\mu$$

$$\eta = (\prod_{j=1}^{n} (g'_j)^{c_j})^{x'}$$

for $v \in [0..n]$ are generated. If the equation (18) holds then with overwhelming probability

$$m'_i = g_i'^{-x'} \prod_{v=0}^{n} m_v^{A_{vi}}$$

for $i \in [0..n]$.

Proof. The equation (18) holds, thus

$$\prod_{v=0}^{n} m_v^{r_v} = \eta \prod_{\mu=0}^{n} (m'_\mu)^{c_\mu} = m'_0 \prod_{\mu=1}^{n} (g_\mu'^{x'} m'_\mu)^{c_\mu}.$$

Considering that $r_v = \sum_{\mu=0}^{n} A_{v\mu} c_\mu$, we get

$$\prod_{v=0}^{n} m_v^{r_v} = \prod_{v=0}^{n} m_v^{\sum_{\mu=0}^{n} A_{v\mu} c_\mu} = \prod_{\mu=0}^{n} (\prod_{v=0}^{n} m_v^{A_{v\mu}})^{c_\mu}.$$

Dividing the two previous equations gives

$$\frac{m'_0}{\prod_{v=0}^{n} m_v^{A_{v0}}} \cdot \prod_{\mu=1}^{n} \left(\frac{g_\mu'^{x'} m'_\mu}{\prod_{v=0}^{n} m_v^{A_{v\mu}}} \right)^{c_\mu} = 1.$$

Using the Schwartz-Zippel lemma similarly as in the lemma 7 gives us that with overwhelming probability for $\mu \in [1..n]$

$$g_\mu'^{x'} m'_\mu = \prod_{v=0}^{n} m_v^{A_{v\mu}}.$$

Then also

$$m'_\mu = g_\mu'^{-x'} \prod_{v=0}^{n} m_v^{A_{v\mu}}.$$

□

Theorem 11. If the DL assumption holds then the shuffle-decryption protocol has knowledge soundness.

Proof. Suppose there exist a prover P^* that can make a verifier V accept with probability $p > 0$.

According to the lemma 9 there exist an extractor K that can extract $(A_{v\mu})_{v=-2}^{n}$ for $\mu \in [1..n]$ and $(A'_v)_{v=-2}^{n}$ in expected polynomial-time such that the equations (22) and (23) hold.

According to the lemma 10 if the discrete logarithm assumption holds then response from the P^* is with overwhelming probability

$$r_v = \sum_{\mu=0}^{n} A_{v\mu} c_\mu$$

$$r'_v = \sum_{i=1}^{n} A_{vi} c_i^2 + A'_v.$$

According to the lemma 11 and the theorem 2 $A = (A_{i,j})_{i,j \in [1..n]}$ is a permutation matrix. According to the lemma 12 $(g_i)_{i=1}^{n}$ were shuffled with the permutation matrix A.

Lemma 13 allows to extract the private key x' efficiently. Lemma 14 says that with overwhelming probability η is computed correctly. Then according to the lemma 15 $(m_i)_{i=1}^{n}$ is shuffled correctly. □

4.4 Privacy

It is impossible to simulate η without knowing x'. Thus we cannot prove that the shuffle-decryption protocol is HVZK. In [Fur05] they define a weaker security definition called complete permutation hiding (CPH) and show that the shuffle-decryption protocol has that property.

We consider the privacy of a mixer in a scenario where potentially all the users and all the other mixers collaborate.

Let $I_\kappa := (1^\kappa, p, q, \bar{x}, (M_i)_{i=1}^n)$ where $\bar{x}$ is the sum of all the other mixer's private keys and M_i are the messages that users wish to send. Let U be a probabilistic polynomial-time Turing machine that produces $((g_i, m_i))_{i=1}^n$. We assume that it is possible to extract randomness $\bar{r_i}$ from U such that $g_i = g_0^{\bar{r_i}}$. By $enc(U)$ we denote the encoding of the Turing machine U. We define an algorithm Gen for generating statement-witness pair (X_κ, W_κ).

Algorithm 1: $Gen(I_\kappa, enc(U))$

```
    input  : $I_\kappa, enc(U)$
    output: $X_\kappa, W_\kappa$

    $g_0 \leftarrow_r G$;
    $x' \leftarrow_r \mathbb{Z}_q$;
    $(s_i)_{i=1}^n \leftarrow_r \mathbb{Z}_q^n$;
    picks uniformly randomly a permutation $\pi$;
    $((g_i, m_i))_{i=1}^n = U(I_n, g_0, y)$;
    $((g_i', m_i'))_{i=1}^n = ((g_0^{s_i} g_{\pi^{-1}(i)}, g_i'^{-x'} m_0^{s_i} m_{\pi^{-1}(i)}))_{i=1}^n$;
    $F_n := (f_i)_{i=-2}^n \leftarrow_r (G \setminus \{1\})^{n+3}$;
    $X_\kappa = (p, q, y, g_0, m_0, F_n, ((g_i, m_i))_{i=1}^n, ((g_i', m_i'))_{i=1}^n)$;
    $W_\kappa = (\pi, (s_i)_{i=1}^n, x')$;
```

By $View_V^P$ we denote everything that verifier sees in the interaction: X_κ, messages V sends and receives from P and the random tape of V.

Definition 15. We say that a shuffle-decryption argument (P, V, Gen) has complete permutation hiding if

$$\begin{aligned} &\exists E'^{E}\ \forall E\ \forall H\ \forall f\ \forall U\ \forall c > 0\ \exists N\ \forall n > N\ \forall I_\kappa : \\ &\Pr[E(View_V^P, H(I_\kappa, enc(U), X_\kappa, \pi)) = f(\pi)] \\ &< \Pr[E'(X_\kappa, H(I_\kappa, enc(U), X_\kappa, \pi)) = f(\pi)] + \frac{1}{n^c} \end{aligned}$$

and

$$\exists K\ \forall U\ \forall I_\kappa\ View_V^P \sim^p K(I_\kappa, g_0, y, enc(U), \pi)$$

where E', E, f, U and K are probabilistic polynomial-time Turing machines and π is a permutation. For both predicates $Gen(I_\kappa, enc(U))$ is used to generate X_κ and W_κ.

The Turing machine E denotes an adversary. The Turing machine H denotes external information from other mixers and users and the partial information about π. The Turing machine f denotes information that the adversary E is trying to learn about the permutation π. The Turing machine K allows H to produce other interactions between P and V. These correspond to the previous interactions that an adversary might have seen. Definition says that given the same external information, there is a probabilistic Turing machine E' that can learn with overwhelming probability as much about π as any adversary E that sees the interaction between P and V. Thus with overwhelming probability the interaction does not reveal anything about the permutation.

We define a simulator S.

Algorithm 2: $S(X_\kappa)$

```
**input** : $X_\kappa$
**output:** $(f'_i)_{i=1}^n, \tilde{f}'_0, g'_0, m'_0, \omega, (c_i)_{i=1}^n, (r_v)_{v=-2}^n, (r'_v)_{v=-2}^n, \eta, \eta', y', c', r'$
$(c_i)_{i=1}^n \leftarrow_r \mathbb{Z}_q^n$;
$(r_v)_{v=-2}^n, (r'_v)_{v=-2}^n \leftarrow_r \mathbb{Z}_q^{n+3}$;
$r', c' \leftarrow_r \mathbb{Z}_q$;
$(f'_i)_{i=1}^n \leftarrow_r G^n$;
$\eta \leftarrow_r G$;
$f'_0 = \prod_{v=-2}^n f_v^{r_v} \prod_{i=1}^n f_i'^{-c_i}$;
$\tilde{f}'_0 = \prod_{v=-2}^n f_v^{r'_v} \prod_{i=1}^n f_i'^{-c_i^2}$;
$g'_0 = \prod_{v=0}^n g_v^{r_v} \prod_{i=1}^n g_i'^{-c_i}$;
$m'_0 = \eta^{-1} \prod_{v=0}^n m_v^{r_v} \prod_{i=1}^n m_i'^{-c_i}$;
$\omega = \sum_{j=1}^n (r_j^3 - c_j^3) - r_{-2} - r'_{-1}$;
$y' = g_0^{r'} y^{-c'}$;
$\eta' = \zeta^{r'} \eta^{-c'}$
```

Proof strategy is as follows. We show that if the shuffle-decryption is not CPH then the simulation can be distinguished from the real view. Then we show that distinguishing is equivalent to breaking DDH assumption. So the shuffle-decryption is CPH.

We define an algorithm M that generates a view from $\Theta_n^3 \in R_n^3$.

Algorithm 3: $M(I_\kappa, enc(U), \Theta_{3n})$

```
input  : $I_\kappa, enc(U), \Theta_{3n}$
output: $X_\kappa, (f'_i)_{i=0}^n, \tilde{f_0}', g'_0, m'_0, \omega, (c_i)_{i=1}^n, (r_v)_{v=-2}^n, (r'_v)_{v=-2}^n, \eta, \eta', y', c', r', A$
$f_{-2}, f_{-1}, f_1, \dots, f_n \leftarrow_r G \setminus \{1\}$;
$f_0 = \theta_{3,0}$;
$g_0 = \theta_{1,0}$;
$F_n = (f_i)_{i=-2}^n$;
$((g_i, m_i))_{i=1}^n = U(I_n, g_0, y, F_n)$;
extract $\bar{r}_i$ from $U$ such that $g_i = g_0^{\bar{r}_i}$ for $i \in [1..n]$;
$(c_i)_{i=1}^n \leftarrow_r \mathbb{Z}_q^n$;
$(r_v)_{v=-2}^n, (r'_v)_{v=-2}^n \leftarrow_r \mathbb{Z}_q^{n+3}$;
$r', c' \leftarrow_r \mathbb{Z}_q$;
picks a random permutation matrix $A = (A_{i,j})_{i,j \in [1..n]}$;
$m_0 = y g_0^{\bar{x}}$;
$(g'_i)_{i=1}^n = (\theta_{1,i} \prod_{j=1}^n g_j^{A_{ji}})_{i=1}^n$;
$(m'_i)_{i=1}^n = (y^{-\sum_{j=1}^n \bar{r}_j A_{ji}} \theta_{1,i}^{\bar{x}} \prod_{j=1}^n m_j^{A_{ji}})_{i=1}^n$;
$(A_{i,0})_{i=1}^n = (r_i - \sum_{j=1}^n A_{i,j} c_j)_{i=1}^n$;
$(A_{-1,i})_{i=1}^n = (\sum_{j=1}^n 3A_{j,0} A_{j,i})_{i=1}^n$;
$(A_{-2,i})_{i=1}^n = (\sum_{j=1}^n 3A_{j,0}^2 A_{j,i})_{i=1}^n$;
$\zeta = \prod_{i=1}^n g_i'^{c_i}$;
$\eta = y^{\sum_{i,j \in [1..n]} \bar{r}_j A_{j,i} c_i} \prod_{i=1}^n \theta_{2,i}^{c_i}$;
$\eta' = \zeta^{r'} \eta^{-c'}$;
$g'_0 = \zeta^{-1} \prod_{v=0}^n g_v^{r_v}$;
$m'_0 = \eta^{-1} \prod_{v=0}^n m_v^{r_v} \prod_{i=1}^n m_i'^{-c_i}$;
$y' = g_0^{r'} y^{-c'}$;
$(f'_i)_{i=1}^n = (f_{-2}^{A_{-2i}} f_{-1}^{A_{-1i}} \theta_{3,i} \prod_{j=1}^n f_j^{A_{j,i}})_{i=1}^n$;
$f'_0 = \prod_{v=-2}^n f_v^{r_v} \prod_{i=1}^n f_i'^{-c_i}$;
$\tilde{f}'_0 = \prod_{v=-2}^n f_v^{r'_v} \prod_{i=1}^n f_i'^{-c_i^2}$;
$\omega = \sum_{j=1}^n (r_j^3 - c_j^3) - r_{-2} - r'_{-1}$;
$X_\kappa = (p, q, y, g_0, m_0, F_n, ((g_i, m_i))_{i=1}^n, ((g'_i, m'_i))_{i=1}^n)$;
```

Lemma 16. Let $X_\kappa, W_\kappa = Gen(I_\kappa, enc(U))$. Let $View_V^P$ be the distribution of the verifier's view respect to X_κ, W_κ and $enc(U)$. If $\Theta_{3n} \leftarrow_r D_n^3$ then $View_V^P \sim^p M(I_\kappa, enc(U), \Theta_{3n})$.

Proof. It is clear that the only difference in the distributions comes from the values in Θ_{3n}.

For $i \in [1..n]$ we have

$$\log_{\theta_{2,0}} \theta_{2,i} = \frac{\log_{\theta_{1,0}} \theta_{2,i}}{\log_{\theta_{1,0}} \theta_{2,0}} = \frac{\log_{\theta_{1,0}} \theta_{2,i}}{\log_{\theta_{1,i}} \theta_{2,i}} = \frac{\log_{\theta_{1,0}} \theta_{2,i} \log_{\theta_{1,0}} \theta_{1,i}}{\log_{\theta_{1,0}} \theta_{2,i}} = \log_{\theta_{1,0}} \theta_{1,i}.$$

Similarly we can show that $\log_{\theta_{3,0}} \theta_{3,i} = \log_{\theta_{1,0}} \theta_{1,i}$.

Then $\log_{\theta_{1,0}} \theta_{1,i} = \log_{\theta_{2,0}} \theta_{2,i} = \log_{\theta_{3,0}} \theta_{3,i}$. We have that $\theta_{1,0}, \theta_{2,0}, \theta_{3,0}$ correspond respectively to g_0, y, f_0 and $\theta_{1,i}, \theta_{2,i}, \theta_{3,i}$ correspond respectively to $g_0^{A_{0i}}, y^{A_{0i}}, f_0^{A_{0i}}$.

Elements of Θ_{3n} are distributed the same way as the corresponding elements in the real protocol because we saw that for each $i \in [1..n]$ elements $\theta_{1,i}$, $\theta_{2,i}$ and $\theta_{3,i}$ have the same exponent. $\square$

Lemma 17. *Let $X_\kappa, W_\kappa = Gen(I_\kappa, enc(U))$ and $View^* = S(X_\kappa)$. If $\Theta_{3n} \leftarrow_r R_n^3$ then $View^* \sim^p M(I_\kappa, enc(U), \Theta_{3n})$.*

Proof. If $\Theta_{3n} \leftarrow_r R_n^3$ then the elements of Θ_{3n} are uniformly random in G. From the definition of the algorithm M it is clear that then f'_i and η are uniformly random and independent elements from G exactly as the simulator S generates. All the other elements are generated the same way by S and M. It follows that the distributions are the same. $\square$

Lemma 18. There exists a probabilistic polynomial-time Turing machine K such that

$$K(I_\kappa, g_0, y, enc(U), \pi) \sim^p View_V^P.$$

Proof. We assumed that it is possible to efficiently extract from U values $\bar{r}_i$ such that $g_i = g_0^{\bar{r}_i}$. Then it is possible to express $g_i'^{x'}$ as follows

$$g_i'^{x'} = (g_0^{A_{0,i}} \prod_{j=1}^n g_j^{A_{ji}})^{x'} = (g_0^{A_{0,i}} \prod_{j=1}^n g_0^{\bar{r_j} A_{ji}})^{x'} = y^{A_{0,i} + \sum_{j=1}^n \bar{r_j} A_{j,i}}.$$

This allows to simulate m'_i and η without the knowledge of x'

$$m'_i = g_i'^{-x'} \prod_{v=0}^n m_v^{A_{vi}} = y^{-A_{0,i} - \sum_{j=1}^n \bar{r_j} A_{j,i}} \prod_{v=0}^n m_v^{A_{vi}}$$

$$\eta = \prod_{i=1}^n (g_i'^{x'})^{c_i} = \prod_{i=1}^n (y^{A_{0,i} + \sum_{j=1}^n \bar{r_j} A_{j,i}})^{c_i}.$$

Rest of the values can be simulated as before. $\square$

Theorem 12. *If the DDH assumption holds then the shuffle-decryption protocol is CPH.*

Proof. Suppose the contrary. Then according to the definition of CPH

$$\forall E'^E \; \exists E \; \exists H \; \exists f \; \exists U \; \exists c > 0 \; \forall N \; \exists n > N \; \exists I_\kappa :$$
$$\Pr[E(View_V^P, H(I_\kappa, enc(U), X_\kappa, \pi)) = f(\pi)]$$
$$\geq \Pr[E'(X_\kappa, H(I_\kappa, enc(U), X_\kappa, \pi)) = f(\pi)] + \frac{1}{n^c}.$$

Let us pick E'^E such that it first generates $View_V^{P*}$ using simulator S and then gives it as an input to the algorithm E. Then

$$\exists H \; \exists f \; \exists U \; \exists c > 0 \; \forall N \; \exists n > N \; \exists I_\kappa :$$
$$\Pr[E(View_V^P, H(I_\kappa, enc(U), X_\kappa, \pi)) = f(\pi)]$$
$$\geq \Pr[E(View_V^{P*}, H(I_\kappa, enc(U), X_\kappa, \pi)) = f(\pi)] + \frac{1}{n^c}.$$

This allows us to break the DDH_n^3 assumption. Given Θ_{3n} we run the algorithm M on some I_κ and $enc(U)$. Machine M returns among other values X_κ and a permutation matrix A. Let π be the permutation corresponding to A.

From the lemmas 16 and 17 we know that if $\Theta_{3n} \leftarrow_r D_n^3$ then $View_V^P \sim^p M(I_\kappa, enc(U), \Theta_{3n})$ and if $\Theta_{3n} \leftarrow_r R_n^3$ then $View_V^{P*} \sim^p M(I_\kappa, enc(U), \Theta_{3n})$.

If $E(View_V^{P*}, H(I_\kappa, enc(U), X_\kappa, \pi)) = f(\pi)$ then with non-negligible probability $\Theta_{3n} \in D_n^3$. Therefore we can distinguish between the cases $\Theta_{3n} \in D_n^3$ and $\Theta_{3n} \in R_n^3$ with non-negligible probability.

The lemma 18 shows that it is possible to construct the simulator needed in the CPH definition. □

5 Conclusion

In this thesis, we gave a detailed description of the shuffle and the shuffle-decryption arguments from [Fur05].

We also showed two simple variations on the original shuffle protocol that allow to reduce the computation by n exponentiations. First variation is a very simple modification based on constraining the group size q. This idea is alluded in the original paper but for some reason they present a less efficient protocol that needs to check if A is orthogonal.

Second variation does not constrain q but uses another property to guarantee that A is a permutation matrix. Still, we see that computation is reduced by n exponentiation. Conclusion from these two variations is that if we use the property $\sum_{h=1}^n A_{hi}A_{hj}A_{hk} = \delta_{i,j,k}$, then there is no reason to verify if A is orthogonal. There are more efficient alternatives for verifying orthogonality which also guarantee that A is a permutation matrix.

We also showed that if A is orthogonal, then it is sufficient to check $\sum_{h=1}^n A_{hi}^2 A_{hj} = \delta_{i,j}$, instead of $\sum_{h=1}^n A_{hi}A_{hj}A_{hk} = \delta_{i,j,k}$ to get that A is a permutation matrix. Further research is needed to see, if this characterization can be used to get a more efficient protocol.

In the Furukawa shuffle protocol, half of the exponentiations on the prover's side come from the necessity of cancelling redundant terms in two of the equations. Having a better characterization of a permutation matrix or better verification equations for proving that A is a permutation matrix seem to have potential in reducing some of that computation. This seems to be the most promising direction for further improvement.

References

[BG93] M. Bellare, O. Goldreich. On Defining Proofs of Knowledge. Proceedings of the 12th Annual International Cryptology: 390-420, 1993.

[BG12] S. Bayer, J. Groth. Efficient Zero-Knowledge Argument for Correctness of a Shuffle. EUROCRYPT'12, Proceedings of the 31st Annual international conference on Theory and Applications of Cryptographic Techniques: 263-280, 2012.

[Cha81] D. Chaum. Untraceable electronic mail, return addresses and digital pseudonyms. Communications of the ACM, vol 24(2): 84-88, 1981.

[ElG84] T. ElGamal. A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms. CRYPTO, LNCS vol 196: 10-18, 1984.

[FS86] A. Fiat, A. Shamir. How to Prove Yourself: Practical Solutions to Identification and Signature Problems. In Proceedings on Advances in Cryptology: 186-194, 1986.

[FS01] J. Furukawa, K. Sako. An Efficient Scheme for Proving a Shuffle. In Proceeding of CRYPTO '01: 368-387, 2001.

[Fur05] J. Furukawa. Efficient and Verifiable Shuffling and Shuffle-Decryption. IEICE Transactions on Fundamentals of Electronics Communications and Computer Sciences, vol 88-A(1): 172-188, 2005.

[GL07] J. Groth, S. Lu. Verifiable Shuffle of Large Size Ciphertexts. 10th International Conference on Practice and Theory in Public-Key Cryptography: 377-392, 2007.

[GMY06] J. Garay, P. MacKenzie, K. Yang. Strengthening Zero-Knowledge Protocols Using Signatures. J. Cryptology, vol 19(2): 169-209, 2006.

[GMR85] S. Goldwasser, S. Micali, C. Rackoff. The Knowledge Complexity of Interactive Proof-Systems. Proceedings of the Seventeenth Annual ACM Symposium on Theory of Computing: 291-304, 1985.

[NSNK05] L. Nguyen, R. Safavi-Naini, K. Kurosawa. A Provably Secure and Efficient Verifiable Shuffle Based on a Variant of the Paillier Cryptosystem. Journal of Universal Computer Science, 11(6): 986-1010, 2005.

[Ped91] Pedersen, T. Non-Interactive and Information-Theoretic Secure Verifiable Secret Sharing. CRYPTO 1991. LNCS, vol. 576: 129-140, 1992.

[Sch91] C. P. Schnorr. Efficient Signature Generation by Smart Cards. Journal of Cryptology, vol 4(3): 161-174, 1991.

[Wik09] D. Wikström. A Commitment-Consistent Proof of a Shuffle. ACISP, vol. 5594: 407-421, 2009.