http://dx.doi.org/10.7236/JIIBC.2014.14.1.245
JIIBC 2014-1-31

전자투표에서 익명성 보장을 위한 빠르고 안전한 방식

A Fast and Secure Method to Preserve Anonymity in Electronic Voting

양형규[*]

Hyung-Kyu Yang[*]

요 약 Mix 네트워크는 전자투표 시스템에서 익명성 보장을 위해서 중요한 역할을 담당하고 있으며 많은 mixnet 방식들이 현재까지 제안되고 있다. 그러데, 기존의 방식들은 안전한 mixing 동작들을 구현하기 위해서 복잡하고 비용부담이 있는 영지식 증명 방식을 사용하고 있다. 2010년도에 Sebé 등은 암호학적으로 안전한 해쉬 함수를 사용해서 효율적이고 비용 부담이 적은 mixnet 방식을 제안하였다. 본 논문에서 우리는 같은 가정하에서 Sebé의 방식보다 안전하고 효율적이고 빠른 방식을 제안한다.

Abstract Mix network plays a key role in electronic voting to preserve anonymity and lots of mixnet schemes have been proposed so far. However, they requires complex and costly zero-knowledge proofs to provide their correct mixing operations. In 2010, Sebé et al. proposed an efficient and lightweight mixnet scheme based on a cryptographic secure hash function instead of zero-knowledge proofs. In this paper, we present a more efficient and faster mixnet scheme than Sebé et al.'s scheme under the same assumption. Also, our scheme is secure.

Key Words : Zero-knowledge proof, e-Voting, Anonimity, Mixnet, hash function

Ⅰ. Introduction

Mixes are a means of untraceable communication based on a public key cryptosystem, as published by D.Chaum in 1981[1]. A mix-network or mixnet accepts as input a collection of ciphertexts, and outputs associated plaintexts(or ciphertexts) in a randomly permuted order. A well constructed mixnet makes it infeasible for an adversary to determine which plaintext output corresponds to which ciphertext input more efficiently than by guessing at random. Proposed by D.Chaum in 1981 as a technique for anonymous e-mail and e-voting, mixnet can be categorized into decryption mix-nets and re-encryption mix-nets[2, 3]. A mixnet consists of a sequence of servers, mixes. Each server receives a batch of input messages and produces as output the batch in permuted(mixed) order[1,4,5]. In e-voting, after [1], lots of e-voting schemes based on mixnet have been proposed so far, and they are called Mix-type voting schemes[4,6,7].

Mix networks are often called mix cascades or shuffle networks. In e-voting, encrypted votes are inputted to the mix network as an input batch. An observer should not be able to match the outputs with

[*]정회원, 강남대학교 컴퓨터미디어정보공학부

접수일자 : 2014년 1월 23일 , 수정완료 : 2014년 2월 7일
게재확정일자 : 2014년 2월 7일

Received: 23 January, 2014 / Revised: 7 February, 2014
Accepted: 7 February, 2014

[*]Corresponding Author: hkyang@kangnam.ac.kr
Dept of Computer Engineering, Kangnam University, Korea

corresponding inputs; this property provides anonymity(voter privacy) in an e-voting. To provide this property, each mix server should also output a proof that it has operated correctly. Otherwise, a dishonest server could replace a ballot with another one or simply do not shuffle at all without anybody noticing.

In [6], they stated that a robust mix network should:

- operate correctly: the output should correspond to a permutation of the input,
- provide privacy: an observer should not be able to determine which input element corresponds to a given output element(and vice versa) in any way better than guessing, and
- be robust: provide a proof or at least strong evidence that it has operated correctly. In addition, it is beneficial if any interested party is able to check the proof or evaluate the evidence; a property called publicverifiability.

Though a mix-net is based on a public cryptosystem, for e.g., ElGamal cryptosystem, direct implementation using RSA without composition with other functions(for e.g., destroying the multiplicative structure), can be broken by an active attack which is perfectly feasible in a typical mix-environment[8,9].

The mixing of encrypted votes should be done verifiably, i.e., there must exist some method to prove that no manupulation, such as vote replacement has taken place during this procedure. This verification method has to accomplish two main properties:

- A dishonest mixer will be caught with high probability even if a single plaintext message gets modified.
- It does not break unlinkability.

In 2010, Sebé et al. proposed an efficient and simple mixnet scheme based on acryptographic hash function[14]. They pointed out that current mixing verification processes make use of complex zero-knowledge proofs, and thus the cost of such schemes becomes especially unaffordable when several mix servers compose the mixnet.

In this paper, we present more efficient and fast mixnet scheme than Sebé et al.'s scheme under the same assumption. This paper is structured as follows. Section II reviews related works, ElGamal cryptosystem, and Sebé et al.'s mixnet scheme. In Section III, we propose an improved mixnet scheme and analyze the security and efficiency of the scheme. Finally, Section IV concludes the paper.

II. Related works

1. Elgamal cryptosystem

The ElGamal cryptosystem is based on discrete logarithms. Let p be a large prime and g be a generator of Z_p^* both public parameters and p must be chosen such that $p-1$ has at least one large prime factor q, i.e. $(p-1)|q$. Suppose A wants to send a message $m$to B, then B selects its private key x and computes $y=g^x\,(\mathrm{mod}p)$, and then publishes g,y,p. A computes $(c,d)=(g^r,y^r m)$ where r is a random number, and sends (c,d) to B. B can decrypt (c,d) by computing $dc^{-x}=y^r mg^{-rx}(\mathrm{mod}p)$ yielding m.

ElGamal cryptosystem has two important properties, homomorphic and remasking(reencryption) properties. Homomorphic property means that decryption of production of two ciphertexts produces production of two plaintexts, i.e., multiplicative homomorphic property[10,11]. The other property, reencryption means that, given a ciphertext (c,d), one can reencrypt it by computing $(c',d')=(cg^{r'},dy^{r'})$ where r' is a random number.

2. Verifiable decryption of an Elgamal cipertext

Given a tuple(g,u,y,v), P can prove in zero-knowledge that, she knows a secret satisfying $g^x = y$ and $u^x = v$ using one-way hash function $h(\bullet)$ as follows:

a. P chooses a random number s and computes $(a,b) = (g^s, u^s)$.

b. P computes $w = h(a||b)$ and $r = s + wx$, and sends (a,b,r) to the verifier V.

c. V computes $w = h(a||b)$. and checks $g^r = ay^r$ and $u^r = bv^w$ hold.

Given an ElGamal ciphertext $(c,d) = (g^r, y^r m)$ where $y = g^x (\bmod p)$ and x is the server's private key, the server can prove m is the plaintext without revealing x by publishing proof of $(g,c,y,\frac{d}{m})$. We will denote $CP(g,u,y,v)$ (Chaum-Pederson's proof) the tuple (a,b,r) sent by the prover to the verifier.

3. Sebe *et al.*'s mixnet

In this section, we briefly review Sebé *et al.*'s mixing scheme.

(1) Notations

The notations used in this paper are as follows:

- TP: trusted party.
- x: TP's private key.
- y: TP's public key $(y = g^x)$.
- λ: the number of mix servers.
- ME_i: the i^{th} mix server $(1 \le i \le \lambda)$.
- n : the number of encrypted votes.
- P_j: the j^{th} voter.
- C_j: the j^{th} encrypted vote. $(C_j = (c_j, d_j), 1 \le j \le n)$
- $h(\bullet)$: a cryptographic secure hash function.
- π: a permutation function.

(2) Setup

TP selects its private key x and computes $y = g^x (\bmod p)$ TP also selects xE and computes $QE = xEP$ where P is a point of an elliptic curve E for e.g., [16, 17]. TP publishes g, y, p, QE, and P.

(3) Voting

P_i selects her vote v_i, and encrypts it using public key QE, yielding $V_i = E_{O_v}(v_i)$. And then she computes $h_i = h(v_i)$ and prepares her message $m_i = V_i || h_i || b_i$, where b_i is chosen so that m_i is a quadratic residue of Z_p^*.

Next, P_i encrypts m_i using public key y, obtaining the ciphertext $C_i = (c_i, d_i)$ and publishes it with its digital signature $\sigma_{P_i}(C_j)$ to the bulletin board. All participants can check the validity of this digital signature.

(4) Vote mixing

After the poll is closed, performs the following steps.

a. TP computes

$$C_c = \prod_{j=1}^{n} C_j = (\prod_{j=1}^{n} c_{j,} \prod_{j=1}^{n} d_j)$$

b. The first mix server ME_1 generates s ElGamal ciphertexts of randomly chosen dummy message $\{\widehat{m_{1,1}}, \cdots, \widehat{m_{1,s}}\}$. Note that there are $n+s$ ciphertexts.

c. ME_1 publishes $\widehat{H_1} = h(\widehat{m}_{1,1}), \cdots, h(\widehat{m}_{1,s})$.

d. ME_1 reencrypts and permutes $C_{j_{1 \le i \le n+s}}$, by computing $C_j' = (c_j', d_j') = (c_{\pi_1(j)} g^{r_j'}, d_{\pi_1(j)} y^{r_j'})$, where r_j' is a random number and π is a random permutation, and publishe $C'_{j\,1 \le j \le n+s}$ to the bulletin board.

e. ME_1 outputs $C_j'{}_{1 \le j \le n+s}$ to the next mix server.

f. For $2 \le i \le \lambda$, the i^{th} mix server takes its input ciphertexts from the $(i-1)^{th}$ mix server, and performs the same operations.

g. At last, the last mix server publishes $\widehat{H}_j$ and $C_j'{}_{1 \le j \le n+s\lambda}$ to the bulletin board.

(5) Vote opening

After all mix servers finish their mixing, TP executes the following operations.

a. TP decrypts each ciphertext $C_j' = (c_j', d_j')$ with its private key x, and publishes the set $\widehat{m}_j'{}_{1 \le j \le n+s\lambda}$ on the bulletin board.

b. TP removes $\widehat{m}_j'$ from the set of decrypted messages satisfying $h(\widehat{m}_j') \in \widehat{H}$ where $1 \le i \le \lambda$. Note that this operation removes exactly $s\lambda$ messages.

c. TP decrypts $Cc = (c_c, d_c)$ and publishes m_c with its proof of correct decryption $CP(g, C_c, y, \dfrac{d_c}{m_c})$.

After this is done, all participants check:

- The correctness of m_c
- Check whether $m_c = \prod_{j=1}^{n} m_j'$ holds or not.
- Check whether $h_i' = h(V_i')$ holds or not using $m_j' = V_j' \| h_j' \| b_j'$.

If these checks are satisfied, TP decrypts $V_j' = E_{Q_E}(v_j')$ and publishes its plaintext v_j' with the corresponding proof of correct decryption.

(6) Security and efficiency

Sebé *et al.*claimed that their mixnet scheme is secure as long as at least one mix server honest, the overall permutation will not be known, and thus it satisfies anonymity. However, their scheme requires $n + s\lambda$ hash operations and $(n + s\lambda)s\lambda$ comparisons at most, and additional $2s\sum_{i=1}^{\lambda} i$ en/decryptions for dummies.

Ⅲ. Improved mixing scheme

In this section, we present an improved mixnet scheme.

1. Vote mixing

At first, The TP computes and publishes

$$C_c = (c_c, d_c) = \prod_{j=1}^{n} c_j = (\prod_{j=1}^{n} d_j, \prod_{j=1}^{n} d_j)$$

Suppose that there are λ mix servers, then the first mix server ME_1 performs the following operations.

a. ME_1 takes $C_{j 1 \le j \le n}$ as input batch, and computes a new list of permuted and reencrypted ciphertexts as $C_j' = (c_j', d_j') = (c_{\pi_1(j)} \bullet g^{r'_j}, d_{\pi_1(j)} \bullet y^{r_j'})$ where $1 \le j \le n$ and each r_j' is a random number.

b. ME_1 selects a random number $k_1 (1 \le k_1 \le n)$ and removes $\overline{C_1} = C_{k^1}'$ from the list and computes $H_1 = h(C_{k_1}')$

c. ME_1 sends two lists $\overline{C_1}$ and $C_j'{}_{1 \le j \le n-1}$ to the next mix server and publishes H_1 to the public bulletin board.

d. The i^{th} mix server ME_i takes two input lists $\{\overline{C_1}, \cdots, \overline{C}_{i-1}\}$ and $C'_{j\ 1 \le j \le n-i+1}$ as input batch, and computes a new lists of permuted and reencrypted ciphertexts as
$C_j' = (c_j', d_j') = (c_{\pi_i(j)} \bullet g^{r'_j}, d_{\pi i(j)} \bullet y^{r_j'})$
where $1 \le j \le n-i+1$ and each r_j' is a random number.

e. ME_i picks a random number $k_i (1 \le k_i \le n-i+1)$ and removes $\overline{C_i} = C_k'$ from the new list and computes $H_i = h(C_k')$.

f. ME_i computes another list of permuted and reencrypted ciphertexts including $\overline{C}$ as
$\overline{C_j}' = (\overline{c_j}', \overline{d_j}') = (\overline{c_{\pi_i(j)}} \bullet g^{r_j''}, \overline{d_{\pi_i(j)}} \bullet y^{r_j''})$
where $1 \le j \le i$ and each r_j'' is a random number.

g. ME_i publishes H to the bulletin board and sends two lists $\{\overline{C_1}', \cdots, \overline{C_i}'\}$and $C'_{j\ 1 \le j \le n-1}$ to the $(i+1)^{th}$mix server.

h. Finally, the last mix server ME_λ outputs $\{\overline{C_1}', \cdots, \overline{C_\lambda}'\}$ and $C'_{j\ 1 \le j \le n-\lambda}$ and publishes the lists to the bulletin board, and at the end, the bulletin board will contain $H_1, \cdots, H_\lambda$ and two lists of ciphertexts $\{\overline{C_1}', \cdots, \overline{C_\lambda}'\}$and $C'_{j\ 1 \le j \le n-\lambda}$.

2. Vote opening

At the vote opening phase, TP and mix servers perform:

a. Each ME_1 publishes its ciphertext $\overline{C}$ and all participants verify $H_i = h(\overline{C_i})$.

b. TP computes
$\overline{C} = (\prod_{i=1}^{\lambda} \overline{c}_i, \prod_{i=1}^{\lambda} \overline{d}_i)$,
$\overline{C}' = \prod_{i=1}^{\lambda} \overline{C_i}' = (\prod_{i=1}^{\lambda} \overline{c}_i, \prod_{i=1}^{\lambda} \overline{d}_i)$ and decrypts $\overline{C}$ and $\overline{C}'$ with those proofs of correct decryptions yielding $\overline{m}$ and $\overline{m}'$respectively. All participants verify the proofs of correct decryptions and check $\overline{m} = \overline{m}'$.

c. TP computes
$C'_c = (c_c, d_c) = \prod_{j=1}^{n-\lambda} C_j' \bullet \prod_{\lambda j=1}^{\lambda} \overline{C_j}$
and obtains m'_c by decrypting C'_c and publishes m'_c with its proof of decryption
$CP(g, c_c', y, \frac{d'_c}{m_c'})$

d. TP decrypts Cc using its private key x and publishes m_c with its proof of correct decryption
$CP(g, c_c, y, \frac{d_c}{m_c})$

e. TP publishes $\overline{m_1'}, \cdots, \overline{m_\lambda}'$ and $m_j (1 \le j \le n-\lambda)$ by decrypting $\{\overline{C_1}', \cdots, \overline{C_i}'\}$ and $C'_{j\ 1 \le j \le n-\lambda}$

After these steps are done, all parties can check:

- The correctness of m_c
- Check whether $m_c = \prod_{j=1}^{\lambda} \overline{m_j}' \bullet \prod_{j=1}^{n-\lambda} m_j'$ holds or not.

If these checks are satisfied, TP decrypts $V_j' = E_{Q_E}(v_j')(1 \le j \le n-\lambda)$ and $\overline{V_j'} = E_{Q_E}(\overline{v}_j')(1 \le j \le \lambda)$, and publishes its plaintext v_j' and $\overline{v}_j'$ with the corresponding proof of correct decryption.

3. Security and efficiency

As long as at least one mix server is honest, then the overall permutation will not be known and the probability of not detecting that a honest mix server has been bypassed is n^{-1} because the honest server selects one ciphertext randomly among n ciphertexts.

표 1. 제안한 방식과 Sebé 방식 비교 결과
Table 1. Comparison results

Operations	Sebé's scheme	Our proposal
Permutation	λ	λ
Hash	$n+2s\lambda$	λ
Comparison	$(n+s\lambda)s\lambda$	none
En/Decryptions	$2s\sum_{i=1}^{\lambda} i$	none
Security	$(\frac{s}{n})^s$	$\frac{1}{n}$

* : For dummies.

The proposed mixnet scheme is more efficient than Sebé *et al.*'sscheme because the proposed scheme requires fewer hash operations and no comparisons at all. Comparison results are shown in Table 1.

Ⅳ. Conclusion

In this paper, an efficient and simple mixnet scheme has been proposed. It requires fewer hash operations and no comparison operations at all, and thus is more efficient than Sebé *et al.*'s mixnet scheme under the same assumptions.

References

[1] D. Chaum: Untraceable Electronic Mail, Return Addresses, and Digital Pseudonyms, Comm. of the ACM, vol. 24, no. 2, pp. 84 - 88, 1981

[2] A. C. Neff, A Verifiable Secret Shuffle and Its Application to E-Voting, Proc. of the 8th ACM Conference on Computers and Communications Security, November 6-8, Philadelphia, U, 2001.

[3] P. Golle, M. Jakobsson, A. Juels, and P. Syverson: Universal Re-encryption for Mixnets, CT-RSA 2004 San Francisco, USA, pp. 23-27, 2004.

[4] K. Sako and J. Kilian: Receipt-Free Mix-Type Voting Scheme-A Practical Solution to the Implementation of A Voting Booth, Proc. of Advances in Cryptology (Eurocrypt'95), pp. 21-25, 1995

[5] M. Jakobsson and A. Juels, Millimix: Mixing in small batches, DIMACS Technical Report, pp. 99 - 33, 1999.

[6] M. Jakobsson, A. Juels, and R. L. Rivest: Making Mix Nets Robust for Electronic Voting by Randomized Partial Checking, Proc. of the 11th USENIX Security Symposium, pp. 230-239, 2002

[7] M. Jakobsson: A Practical Mix, Proc. of Advances in Cryptology (Eurocrypt'98), pp. 242-251, 1998

[8] M. Abe: Mix-networks on permutation networks, Proc. of Asiacrypt'99, pp. 270-279, 1999

[9] B. fitzmann and A. fitzmann: How to Break the Direct RSA-Implementation of Mixes, Proc. of Advances in Cryptology(Eurocrypt'89), pp. 310-319, 1989

[10] M. Jeong, S. Shin: Remote System User Authentication Scheme using Smartcards, Journal of the Korea Academia-Industrial cooperation Society, v.10, no.3, pp. 81-89, 2009.

[11] K. Min, J. Kang: Rights to Control Information and Related Security Technologies on the CyberSpac, Journal of the institute of internet, Broadcasting and Communication(IIBC), v.10, no 2, pp. 135-141, 2010

※ 본 연구는 강남대학교 교내연구비 지원 연구임.

양 형 규(정회원)

- 1995년 2월 : 성균관대학교 석사
- 1995년 2월 : 성균관대학교 정보공학과 공학박사
- 1995년 ~ 현재 : 강남대학교 컴퓨터미디어정보공학부 교수
- 1984년 12월 ~ 1990년 2월 : 삼성전자 컴퓨터부문 선임연구원

<주관심분야 : 정보보안, 네트워크 보안, DRM>