SCIS2000 The 2000 Symposium on Cryptography and Information Security
Okinawa, Japan, January 26-28, 2000

t Mixes by User's Selection

Weonkeun Huh * abcxyz@icu.ac.kr
Kwangjo Kim * kkj@icu.ac.kr

Abstract— The mix-net decrypts a list of messages without revealing the relationships of sizes, orders, times, and values between inputs and outputs. Using the (t,n)-threshold secret sharing, the mix-net assures high security and holds the robustness against some errors. For t mix servers to do the threshold decryption of the randomized messages, we must construct the quorum with t mixes in n ones. But the previous works of the threshold mix-net describe vaguely a way to construct a quorum. So, there is needed to select t mixes from n ones for a quorum. It is also necessary to manage the order of decryption for t mixes. In this paper, we propose that the user who sends his message to a mix-net selects the quorum respectively. If so, each mix-server is able to decrypt messages randomly instead of decrypting all messages sequentially. And our proposed mix-net holds the ${}_nC_t(t \leq n)$ cases of decryption, has the advantages that the load is distributed over n mix-servers, and can manage the errors on each mix-server.

Keywords: Mix-net, Threshold Secret Sharing, ElGamal Encryption, Randomization, Permutation.

1 Introduction

Electronic election may be defined as the protocol is constructed with the crypto algorithm ensuring the security and the physical polling booth minimizing the security assumption; therefore, we can implement the voting processes on the public network to ensure the properties of the election - privacy, completeness, fairness, verification, *etc.* So, the electronic election is superior to security, convenience, time, space, and mobility. On the other hand, if the secret key of an administrator is damaged or stolen, all processes of election will be stopped. Secret sharing scheme[1, 2] is likely to be suitable for providing against the errors on any mix-server.

The secret sharing scheme distributes the secret s to n participants using $t-1$ degree polynomial securely. Any t subsets from n participants will be able to recover the secret; any smaller subset than t will be unable to get any information about the secret because it is computationally infeasible to calculate s. So, this method improves the security as (t,n)-threshold scheme whenever a smaller subset of participants produce an error.

In electronic election protocol, the voter's privacy is the most important thing with security. Even though the vote of a voter is encrypted, the relationship between a voter and his vote can be revealed by traffic analysis on communication. In order to solve the traffic analysis problem, the mix-net decrypts a list of messages without revealing the relationships of sizes, orders, times and values between inputs and outputs. The mix-net is constituted with senders, receipients and mix-servers. The mix-server of mix-net plays an important role like an administrator in electronic election. Therefore as like the distribution of secret key of the administrator, we must prepare against the deception of some mix-servers using (t,n)-threshold secret sharing.

In this paper, we propose t mixes by user's selection as the construction of t mixes that randomizes, permutes and decrypts the messages on the anonymous channels with secret sharing. When the user sends the message to the mix-net, he sends the quorum data with t mixes. The mix-servers within the quorum data do the randomization and permutation of the message and then they modify the quorum data. And they send the quorum data to one of the remainder mix-servers of the quorum. The load is distributed by the result of the randomly order of randomization and permutation of the message. The fairness of the electronic election is improved because the construction of quorum by a user minimizes the dealer's participation. Also as the process of the mix-server's job is managed by quorum data, the message can be re-processed when the mix-server goes wrong.

The organization of this paper is as follows; we describe the anonymous channels in Section 2. In Section 3 we propose t mixes by user's selection. In Section 4 we discuss the proposed model. And we draw a conclusion in Section 5.

2 Anonymous channels

Chaum proposed the concept of mix-net using RSA cryptosystem[3]. Chaum's mix-net consists of sender, receipient, and mix-server. Using K_1(the public key of mix-server), A(the address of a receipient), K_a(the public key of a receipient), random number R_1 and R_0, a sender sends the message M that is encrypted like Eq.(1) to the mix-server. The mix-server sends the righthand side of Eq.(1) that is decrypted with its

* School of Engineering., Information and Communications Univ., 58-4 Hwaam-dong, Yusong-gu, Taejon, 305-348, Korea

secret key to a receipient. If the mix-server is honest, a relationship between the input and the output is hidden.

$$K_1(R_1, K_a(R_0, M), A) \rightarrow K_a(R_0, M), A \qquad (1)$$

Eq.(1) can be extended to n mix-servers model like $K_n(R_n, ..., K_2(R_2, K_1(R_1, M))...)$. However Chaum's mix-net has the following problems; the size of message is extended with proportional to the numbers of mixes and the process can be stopped by the error of a mix-server. Park *et al*[5](denoted as Park's mix-net) solved these problems with ElGamal cryptosystem[6]. In ElGamal cryptosystem, when a user's secret key is $s \epsilon Z_q^*$ his public key is $y = g^s$. Encryption of message m is $(x', y') = (g^r, y^r m)$ using random number r. Dividing y' by exponentiation with x' and s like that $y'/x'^s = (y^r m)/(g^r)^s = g^{sr}m/g^{rs} = m$, we can obtain the original message m from cipertext (x', y').

In Park's mix-net, X_i is the private key, $Y_i (1 \leq i \leq k)$ is the public key of the i^{th} mix-server, E_{B_i} is a receipient's public key, B_i is a receipient's address, g is a generator, and message m_i belongs to a sender i. In order to send a message to a receipient, a sender i selects a random number R, calculates (C_{0i}, C_{1i}) like the first equivalence of Eq.(2), and sends the encrypted message to the mix-net. The mix-servers decrypt the message like from the second line to the last of Eq.(2) using Eq.(3). The k^{th} mix-server publishes the result $B_i \circ E_{B_i}(m_i)$.

$$\begin{aligned}(C_{0i}, C_{1i}) &= (g^R, (B_i \circ E_{B_i}(m_i)) \times (Y_1 \ldots Y_k)^R) \\ &\rightarrow (g^{R_1}, (B_i \circ E_{B_i}(m_i)) \times (Y_2 \ldots Y_k)^{R_1}) \\ &\quad \vdots \qquad\qquad\qquad\qquad\qquad\qquad\qquad (2) \\ &\rightarrow (g^{R_{k-1}}, (B_i \circ E_{B_i}(m_i)) \times (Y_k)^{R_{k-1}}) \\ &\rightarrow B_i \circ E_{B_i}(m_i)\end{aligned}$$

$$\begin{aligned}&f_j(t, u, r) \equiv \\ &\quad (t \times g^r,\ u \times (Y_{j+1} \ldots Y_k)^r / t^{X_j}) \quad (1 \leq j \leq k-1), \\ &\qquad\qquad u/t^{X_k} \qquad\qquad\qquad\quad (j = k) \qquad (3)\end{aligned}$$

Park's mix-net keeps the same problem as Chaum's mix-net. That is, if one mix-server occurs an error, the decryption of the message is not possible. Also, the mix-net can be attacked with a malleability problem in ElGamal crytosystem. Sako and Kilian[9] proposed the receipt-free mix-type voting scheme on anonymous channels. But it has the problem of efficiency because a verifier must verify whether each mix-server does the cut-and-choose correctly. And as Michels and Horster[10] pointed out, there exists the possibility of revealing a user's anonymity.

In order to prevent the collusion or interference of any mix-servers, a secret sharing scheme is useful. Especially, when we construct a mix-net in electronic elections, secret sharing scheme is important because of improving its robustness and security. Abe's mix-net[7] and Jakobsson's mix-net[12] show the good examples about the threshold ElGamal adopted the secret sharing. The followings describe Abe's mix-net about the randomization and permutation, and the decryption using the secret sharing scheme.

Randomization and permutation

The n mix-servers shares the secret x with Shamir's method[1] and the dealer publishes the encryption key $y = g^x$. The share x_i of each mix-server is published as $y_i = g^{x_i} (1 \leq i \leq n)$. The message list sent by users on a public board is $(C_{0,j}, G_{0,j})$ where $C_{0,j} = v_j y^{t_{0,j}}$ and $G_{0,j} = g^{t_{0,j}} (1 \leq j \leq L)$. v_j is a message and $t_{0,j}$ is a random number. A user proves the knowledge of $t_{0,j}$ to prevent the Pfitzmann's attack[8].

The mix-servers hide the relationship between the inputs and the outputs as they do the randomization and permutation like Eq.(4). They denote the set of all possible permutation on L messages by Π_L and $\pi_i \epsilon \Pi_L$.

$$(C_{i,j} := C_{i-1,\pi_i(j)} y^{t_{i,\pi_i(j)}}, G_{i,j} := G_{i-1,\pi_i(j)} g^{t_{i,\pi_i(j)}}) \\ (1 \leq i \leq n, 1 \leq j \leq L) \qquad (4)$$

Eq.(5) is a final result of randomizing and permuting the messages sequentially by n mix-servers. The $(C_{n,j}, G_{n,j})$ $(1 \leq j \leq L)$ of Eq.(5) that were hidden relationships between the inputs and the outputs satisfy the validity of randomization by the proof of knowledge.

$$(C_{n,j} := C_{0,\pi(j)} y^{\hat{t}_{n,j}}, G_{n,j} := G_{0,\pi(j)} g^{\hat{t}_{n,j}}) \\ (\hat{t}_{n,j} = \sum_{i=1}^{n} \hat{t}_{i,\hat{\pi}_i(j)}) \qquad (5)$$

Threshold ElGamal decryptions

After the randomization and permutation, the mix-servers in quorum $Q = \{1, \ldots, t\}$ incorporate in order to obtain the message v_j. A mix-server $i \ \epsilon \ Q$ decrypts $(C_{n,j}, G_{n,j})$ using $(\lambda_i = \prod_{1 \leq k \leq t, k \neq i} \frac{x_k}{x_k - x_i})$[11] like that

$$W_{i,j} := W_{i-1,j} {G_{n,j}}^{x_i \lambda_i} (W_{0,j} = 1)$$

and sends all $W_{i,j}$ to the next mix-server. The last mix-server publishes $W_{i,j}$ and then v_j can be obtained by calculating $C_{n,j}/W_{t,j}$. If all t mix-servers are honest, Eq.(6) can be obtained, and the validity can be proved by proof of knowledge.

$$W_{t,j} = \prod_{i=1}^{t} {G_{n,j}}^{x_i \lambda_i} = {G_{n,j}}^{\sum_{i=1}^{t} x_i \lambda_i} = {G_{n,j}}^{x} \qquad (6)$$

In Abe's mix-net[7] and Jakobsson's mix-net[12], all mix-servers work sequentially as shown in Fig.(1). Therefore the load is concentrated on a working mix-server

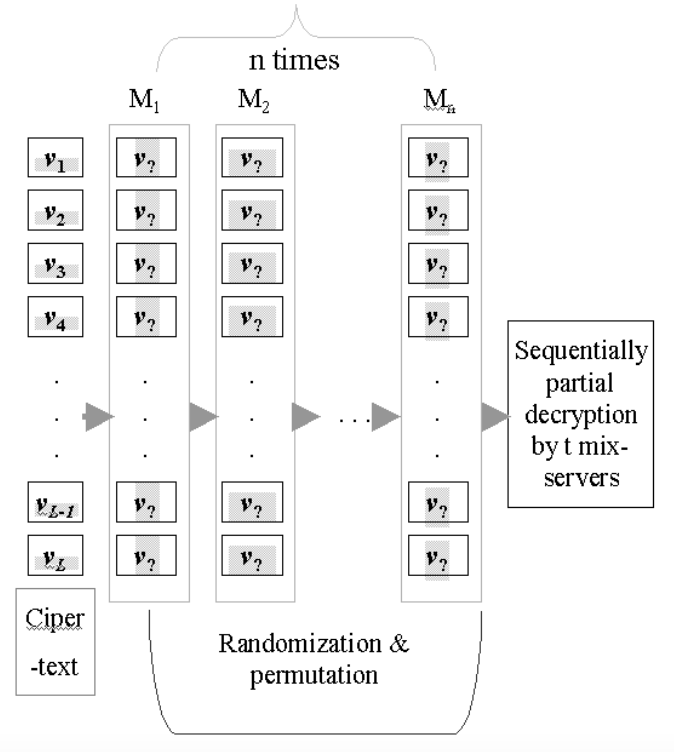

Figure 1: Sequential randomization and permutation

in a point of time because the next mix-server can work after one mix-server finishes working. And because the method of managing the error of mix-server is not suggested, the work of randomization or decryption may be done again when any error occurs in a mix-server. Furthermore, it is unclear that the quorum of mix-servers is constructed to decrypt the message list. In the next Section, we propose the improved idea that can distribute the loads of mix-servers and manage errors on them.

3 Proposed t Mixes by User's Selection

In (t, n)-threshold mix-net structure, the dealer who sets the system parameters is important. A dealer selects a secret polynomial to share a secret, and distributes shares to mix-servers using the polynomial. The public key corresponding to a secret is published (preparation phase). After the preparation phase, users encrypt their messages with the public key and send them to the mix-net(encryption and transmission phase). Mix-servers re-encrypt the messages with random number in order to conceal the relationships between the inputs and outputs(randomization phase). In order to decrypt the messages that were sent to the mix-net, first of all, we must select t mix-servers. Therefore, in the decryption phase the dealer intervenes in the mixing process again[7, 12, 13]. Finally, as the dealer almost participates in the processes, we must assume that the dealer is honest and trustful. However, if the dealer is dishonest, all processes may be stopped. And this reason lets us consider that the dealer participates in the only preparation phase. Our idea provides an option that a user selects t mixes randomly instead of the dealer's selection like Fig. (2). This method has several benefits in the followings:

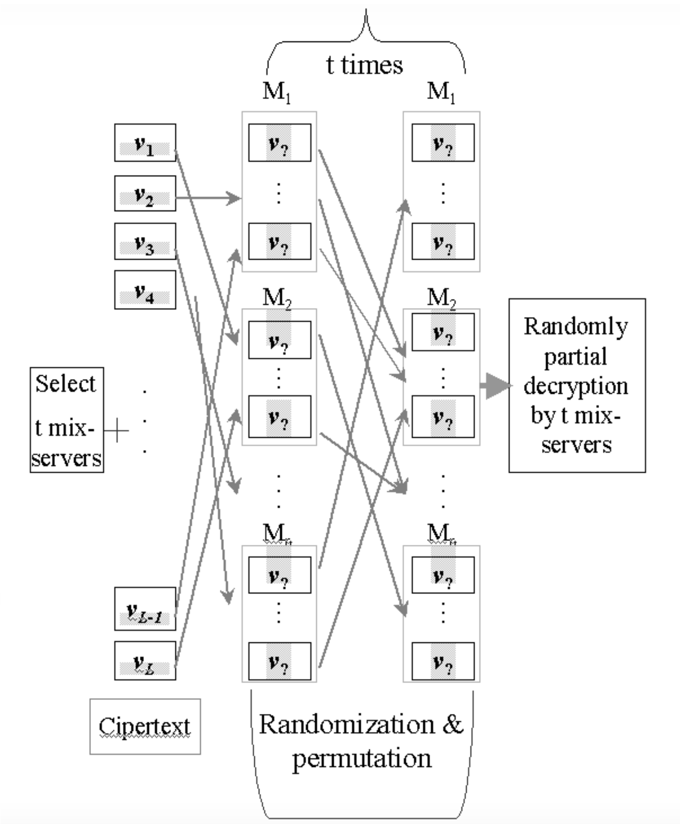

Figure 2: Random processing of t mixes by user's selection

- The injustice of the dealer can be localized or limited by minimizing the dealer's participation.
- There are ${}_nC_t (t \leq n)$ that the cases of processing the message are randomized
- We can construct the parallel mix-net that the load is distributed over the n mix-servers
- We can construct (t, n)-threshold mix-net by a user's selection

[**Preparation Phase**]

- The dealer distributes a secret x using the method in [1] and publishes the encryption key $y = g^x$. The share x_i of each mix-server is published with $y_i = g^{x_i} (1 \leq i \leq n)$.
- Let the share $s_i = p(i)$ of each mix-server $M_i (1 \leq i \leq n)$ be the private key D_{M_i}, and the corresponding public key be E_{M_i}. Then the polynomial $p(x) = \sum_{i=0}^{n} \alpha_i x^i, (s = \alpha_0)$ and α is a generator.

[**Encryption and transmission Phase**]

- A user encrypts his message with the public key of the mix-net. That is, the message $(C_{0,j}, G_{0,j})$ of a user $A_j (1 \le j \le L)$ is that $C_{0,j} = v_j y^{t_{0,j}}$ and $G_{0,j} = g^{t_{0,j}}$. The user proves the knowledge of a random number $t_{0,j}$ in order to prevent Pfitzmann's attack.

- A user A_j selects t mix-servers(a quorum Q) that will decrypt his message. As the data structure that represents the t mix-servers has the size of n bits, it is constituted with the t numbers of 1s and $n-t$ numbers of 0s. In Table (1), the range of the data structure is that $2^{t-1} + \ldots + 2^0 \le \textit{the value of the data structure} \le 2^{n-1} + \ldots + 2^{n-t}$. For example, in the case of $(3, 4)$-threshold the value exists between 7(0111) and 14(1110).

- A user A_j concatenates the data structure and his signature for the data structure and a random number, and encrypts the data structure with the public key of a mix-server that it exists in the data structure as non-zero value(that is, the bit set by value 1). He combines the data structure and the encrypted message, and then sends it to the mix-server. The examples of the data structures in Table (1) are the followings("$\rightarrow$" means transmission).

$$
\begin{aligned}
v_1 :\ & E_{M_n}(Sign_{A_1}(101\ldots10)||(101\ldots10)||r_1) \\
& \rightarrow M_n \\
v_j :\ & E_{M_{n-1}}(Sign_{A_j}(010\ldots00)||(010\ldots00)||r_j) \\
& \rightarrow M_{n-1} \\
v_L :\ & E_{M_{n-1}}(Sign_{A_L}(110\ldots01)||(110\ldots01)||r_L) \\
& \rightarrow M_{n-1}
\end{aligned}
$$

Table 1: The binary expansion in the data structure of the message

Message	n^{th}	$n-1^{th}$	$n-2^{th}$	$\ldots i^{th} \ldots$	2^{nd}	1^{st}
v_1	1	0	1	$\ldots 0 \ldots 1 \ldots$	1	0
$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$
v_j	0	1	0	$\ldots 1 \ldots 0 \ldots$	0	0
$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$
v_L	1	1	0	$\ldots 0 \ldots 0 \ldots$	0	1

[**Randomization Phase**]

- The mix-servers(quorum Q) in the data structure randomize and permute the messages as the Eq.(7). Suppose that L messages are distributed over the n mix-servers randomly, all possible permutation in one mix-server is denoted by $\Pi_{L/n}$ and $\pi_i \,\epsilon\, \Pi_{L/n}$.

$$
(C_{i,j} := C_{i-1,\pi_i(j)} y^{t_{i,\pi_i(j)}},\ G_{i,j} := G_{i-1,\pi_i(j)} g^{t_{i,\pi_i(j)}})
$$
$$
(i \,\epsilon\, Q)(1 \le j \le L) \tag{7}
$$

- After the randomization and permutation, the mix-server sets the data structure with zero and signs it. And the mix-server concatenates the signed data structure with the original. Next, the mix-server selects the remainder mix with non-zero value, encrypts the concatenated message with the public key of the selected mix-server, and sends them to it.

- In Eq.(8), $(C_{Q,j},\ G_{Q,j})(1 \le Q \le n,\ 1 \le j \le L)$ is the last mix-server in Q finished randomization and permutation of the message.

$$
(C_{Q,j} := C_{0,\pi(j)} y^{\hat{t}_{n,j}},\ G_{Q,j} := G_{0,\pi(j)} g^{\hat{t}_{n,j}})
$$
$$
(\hat{t}_{n,j} = \sum_{i \epsilon Q} (\hat{t}_{i,\hat{\pi}_i(j)}) \tag{8}
$$

- The last mix-server publishes the data structure sent by a user.

[**Decryption Phase**]

- The encryption of $(C_{Q,j},\ G_{Q,j})$ is calculated by mix-servers in the data structure using their shares and $\lambda_i = \prod_{1 \le k \le t=Q, k \ne i} \frac{x_k}{x_k - x_i}$. After the calculation of $W_{i,j} := W_{i-1,j}\ {G_{Q,j}}^{x_i \lambda_i}$ $(W_{0,j} = 1)$, all $W_{i,j}$ are sent to the next mix-server. The last mix-server publishes $W_{t,j}$ in Eq.(9) and then we can obtain the v_j by calculating $C_{Q,j}/W_{t,j}$.

$$
W_{t,j} = \prod_{i=1}^{t(=Q)} {G_{Q,j}}^{x_i \lambda_i} = {G_{Q,j}}^{\sum_{i=1}^{t} x_i \lambda_i} = {G_{Q,j}}^{x} \tag{9}
$$

4 Analysis

The consideration point in the t mixes is that any mix-server may alter the data structure sent by a user. If the mix-server is honest, it selects a random number, randomizes and permutes a message, sets the bit by zero in the data structure, signs the data structure, and sends it to the next mix-server. If the mix-server is dishonest, it may alter the data structure sent by a user, or may not randomize and permute the message correctly. The followings are the countermeasures against these problems.

- The validity of the randomization and permutation can be proved by proof of knowledge in [7].

- The verification of the data structure can be verified by comparing the user's data structure with all the data structures of the mix-servers. This phase uses D_{M_i} $(1 \leq i \leq n)$ that distributed in the preparation phase.

The mix-server that received the message sent by a user merely sends the message to one of the t mix-servers after the randomization and permutation. That is, the selection of the next mix-server among the t mix-servers is the same like that a user selects the t mix-servers randomly. Next, we consider that when the obstacle occurs on the working mix-server what method is likely to exist. Commonly, as we become aware of the mix-server that happens to error on the network, we use the assumption that any error of a mix-server is known to the others.

[**The solving problems in normal mix -servers**]

- The normal mix-servers check their holding messages if they are related to the mix-server that an error was occurred on it; i.e., as the mix-servers check the bits in the data structures of their holding messages, they can judge if the abnormal mix-server executes the randomization and permutation on them.
- If the message was already executed by the abnormal mix-server, regardless of that, the job is continued.
- If a message contains the abnormal mix-server and if the abnormal mix-server does not execute the message yet, any mix-server holding the message now constitutes the new data structure as the following.

> [Constructing a new data structure] :
> The mix-server holding the message constructs the data structure like that it replaces the bit of an abnormal mix-server with one of the $n-t$ mix-servers. The holding mix-server encrypts the new data structure and the signature on it with the public key of the next mix-server, and then publishes them. The data structure sent by a user is also published.

[**The solving problems in abnormal mix-servers**]

- Because the others become aware of the mix-server that happens to error on the mix-net, it is possible to solve the error on a mix-server. The only triggering technique in database systems can connect between the front and the back mix-server related to the message. That is, if an error in the abnormal mix-server takes place before a mix-server sends the randomized and permuted message to the next mix-server, the message is returned to the previous mix-server owing to the triggering technique. Then, the previous normal mix-server can construct the new data structure like [**The solving problems in normal mix-servers**].

Table (2) shows that we compare the mix-net in [7, 12] with the proposed t mixes by user's selection. Our proposed mix-net is concentrated on the load balancing and the management of errors through the user's selection of t mixes. We use n bits data structure to represent the information of user's selection.

Table 2: The comparison of the previous mix-net with our proposed mix-net

Mix-net	previous ones	proposed one
Subject	a dealer	a user
dealer's intervention	preparation and decryption phase	preparation phase
message randomization	one	${}_nC_t$
message decryption	sequential processing by quorum	random processing by t mixes(quorum)
working type	sequential	random
working load	concentrating one mix-server	distributing n mix-servers
managing an error	not mentioned	using n bits data structure and triggering technique

5 Conclusions

As the secret sharing is a useful method to improve the security, it is used to share the secret key in ElGamal cryptosystem and to construct the anonymous channels. We propose that t mixes by user's selection expresses clearly the method for forming the quorum to re-construct the secret from n mix-servers. Our proposed mix-net has the goal that it distributes the load of randomizing and permuting the message and it manages an error on any mix-server. The t mixes by user's selection uses n bits data structure to manages t mix-servers. And as referencing this data structure, it is possible to send the message to any mix-server existing in the data structure randomly. Also, when an error takes place into any mix-server, we can change the abnormal mix-server into one of the $n-t$ mix-servers to continue the works.

References

[1] A. Shamir, "How to share a secret", Communications of the ACM, 22(11): pp. 612-613, 1979.

[2] G.R. Blakley, "Safeguarding cryptographic keys", In Proceedings of the National Computer Confer-

ence 1979, Vol. 48 of AFIPS Conference Proceedings, pp. 313-317, 1979.

[3] D. L. Chaum, "Untraceable Electronic Mail, Return Addresses, and Digital Pseudonyms", Communications of the ACM, Vol. 24, No.2, pp. 84-88, 1981.

[4] T. P. Pedersen, "A Threshold Cryptosystem Without A Trusted Party", Advances in Cryptology -EuroCrypt 91, LNCS Vol.547, pp. 522-526, Springer-Verlag, 1991.

[5] C. Park, K. Itoh, K. Kurosawa, "Efficient Anonymous Channel and All/Nothing Election Scheme", Advances in Cryptology -Eurocrypt 93, LNCS Vol.765, pp. 248-259, Springer-Verlag, 1993.

[6] T. ElGamal, "A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms", IEEE Transactions on Information Theory, IT-31(4):469 472, July 1985.

[7] M. Abe, "Universally Verifiable Mix-net with Verification Work Independent of the Number of Mix-servers", Advances in Cryptology-Eurocrypt 98, LNCS Vol. 1403, pp.437-447, Springer-Verlag, 1998.

[8] B. Pfitzmann, "Breaking an Efficient Anonymous channel", Advances in Cryptology - Eurocrypt 94, LNCS Vol.950, pp. 332-340, Springer-Verlag, 1994.

[9] K.Sako and J.Killian, "Receipt-free Mix type voting scheme - a practical solution to the implementation of a voting booth", Advances in Cryptology -Eurocrypt95, LNCS Vol.921, pp.393-403, Springer-Verlag, 1995.

[10] M. Michels, P. Horster, "Some Remarks on A Receipt free and Universally Verifiable Mix-Type Voting Scheme", Advances in Cryptology-Asiacrypt 96, LNCS Vol. 1163, pp. 125-132, Springer-Verlag, 1996.

[11] A.J. Menezes, P.C. van Oorschot, S.A. Vanstone, "Handbook of Applied Cryptography", CRC Press, September 1996.

[12] M. Jakobsson, "A Practical Mix", Advances in Cryptology-Eurocrypt 98, LNCS Vol. 1403, pp. 449-461, Springer-Verlag, 1998.

[13] R. Cramer, R. Gennaro, and B. Schoenmakers, "A Secure and Optimally Efficient Multi-Authority Election Scheme", Advances in Cryptology-Eurocrypt'97, LNCS Vol. 1233, pp.103-118, Springer-Verlag, 1997.