Bilinear Pairing-based Hybrid Mixnet with Anonymity Revocation

First Author Name[1], Second Author Name[1] and Third Author Name[2]
[1]*Institute of Problem Solving, XYZ University, My Street, MyTown, MyCountry*
[2]*Department of Computing, Main University, MySecondTown, MyCountry*
{*f_author, s_author*}*@ips.xyz.edu, t_author@dc.mu.edu*

Keywords: bilinear pairings, mix network, anonymity revocation, eligibility

Abstract: A hybrid mix is presented providing anonymity and eligibility verification of senders, the possibility of anonymous reply and anonymity revocation, that are usually required in practice. Furthermore the proposed mix is capable of processing messages with arbitrarily length. In the process of design we applied bilinear pairings due to their good properties. We compared the time and space complexity of Zhong's mix (Zhong, 2009) to our one, we achieved better efficiency. In the security evaluation we prove, that our mix is correct, provides anonymity and eligibility verification for senders.

1 INTRODUCTION AND PRELIMINARIES

In recent decades, the widespread use of public channels has led to the development of network-based services, where it is necessary to manage the critical, confidential or personal information. Since these channels can be easily eavesdropped, you need to pay attention to the transmitted information. Several cryptographic primitives are developed to ensure the protection of confidential information from unauthorized access. In some cases it may be important that the message can not be linked to the sender (for example in electronic voting systems, the voter and the vote can not be linked).

In 1981, Chaum (Chaum, 1981) proposed a cryptographic construction called *mix network* which can be used to hide senders' identity. In case of mixnets determining the identity of a sender, even if all messages transferred are given is a hard problem. Each mix server receives messages originating from multiple senders, permutes them, performs cryptographic operations (decryption, encryption or re-encryption) and sends them to the next server.

This design was the basis for many applications, especially in the field of electronic voting (Sako and Kilian, 1995; Michels and Horster, 1996; Neff, 2001; Jakobsson et al., 2002). Some further applications of mix networks: anonymous email (Parekh, 1996; Gulcu and Tsudik, 1996; Danezis et al., 2003), anonymous telecommunications (Pfitzmann et al., 1991; Jerichow et al., 1998), anonymous internet communications (Goldschlag et al., 1996; Syverson et al., 1997a) and location privacy (Federrath et al., 1996; Syverson et al., 1997b; Golle et al., 2002; Huang et al., 2006).

We propose a decryption/encryption-based mix, applying bilinear pairings. Let us review the definition of the admissible bilinear map (Boneh and Franklin, 2001).

Definition 1.1. *Let G_1 and G_2 be two groups of order q for some large prime q. A map $e : G_1 \times G_1 \to G_2$ is an admissible bilinear map if satisfies the following properties:*

1. *Bilinear: We say that a map $e : G_1 \times G_1 \to G_2$ is bilinear if $e(aP, bQ) = e(P,Q)^{ab}$ for all $P, Q \in G_1$ and all $a, b \in Z$.*
2. *Non-degenerate: The map does not send all pairs in $G_1 \times G_1$ to the identity in G_2. Since G_1, G_2 are groups of prime order, if P is a generator of G_1 then $e(P,P)$ is a generator of G_2.*
3. *Computable: There is an efficient algorithm to compute $e(P,Q)$ for any $P, Q \in G_1$.*

We should mention that bilinearity can be restated to for all $P, Q, R \in G_1$ $e(P+Q,R) = e(P,R)e(Q,R)$ and $e(P,Q+R) = e(P,Q)e(P,R)$. We can find G_1 and G_2 where these properties hold. The Weil and Tate pairings prove the existence of such constructions. Typically, G_1 is an elliptic-curve group and G_2 is a finite field.

Usually the security of cryptographic protocols applying bilinear maps is based on the problem of Bilinear Diffie-Hellman Problem.

Definition 1.2. *(Bilinear Diffie-Hellman Problem (BDHP) in G_1)*
Let $e : G_1 \times G_1 \to G_2$ be a bilinear map on (G_1, G_2). The problem is for every $a,b,c \in Z_q^*$, given P, aP, bP, cP, computing $e(P,P)^{abc}$.

BDHP is closely related to the Computational Diffie-Hellman Problem. For simplicity we give the definition in G_1.

Definition 1.3. *(Computational Diffie-Hellman Problem (CDHP) in G_1)*
The problem is for every $a,b \in Z_q^*$, given P, aP, bP, computing abP.

Hardness of the BDHP implies the hardness of the CDHP in both G_1 and G_2. Note, that the Decisional Diffie-Hellman Problem (DDHP) in G_1 can be efficiently solved.

Definition 1.4. *(Decisional Diffie-Hellman Problem (DDHP) in G_1)*
The problem is for every $a,b,c \in Z_q^*$, given P, aP, bP and cP, deciding whether $cP = abP$.

By computing $e(P,cP) = e(P,P)^c$ and $e(aP,bP) = e(P,P)^{ab}$, one can decide whether $cP = abP$ holds, since $e(P,P)^{ab} = e(P,P)^c$ if and only if $abP = cP$. G_1 is a Gap Diffie-Hellman (GDH) group, if the DDHP is easy and the CDHP is hard in G_1.

Assuming that BDHP is hard, a one-round three-party key agreement protocol is constructed (Joux, 2000; Verheul, 2001). There are three participants, each possesses a secret value $a,b,c \in Z_q^*$, and publishes aP, bP, cP, respectively. With the given $e : G_1 \times G_1 \to G_2$ bilinear map the common secret key K can be easily calculated by $K = e(bP,cP)^a = e(aP,cP)^b = e(aP,bP)^c$.

2 RELATED WORK

Chaum's proposal called forth several designs of anonymous communication channels. One can read a nice survey by Sampigethaya in (Sampigethaya and Poovendran, 2006).

A hybrid mix is capable of handling arbitrarily long input messages. There is a hybrid mix proposed by Ohkubo and Abe (Ohkubo and Abe, 2000), that is based on the intractability of the Decision Diffie-Hellman problem and realizes length-flexibility, length-invariance and provable security (in terms of anonymity).

Subsequenlty proposed mix network schemes, known as public-key mixes, have focused on achieving robustness, typically through heavy reliance on public-key operations (Jakobsson, 1998; Markus and Ari, 1999; Desmedt and Kurosawa, 2000; Mitomo and Kurosawa, 2000). Jakobsson and Juels presented an optimally robust hybrid mix network (Jakobsson and Juels, 2001). They used MAC keys for providing the robustness and in their paper they described these properties of correctness, privacy, robustness and indistinguishability (Jakobsson and Juels, 2001).

In 2009, Zhong proposed an identity-based mix network (Zhong, 2009), which is based on bilinear maps. His construction is a re-encryption mix, that is suitable only for sending short messages, since it proceeds only asymmetric encryptions.

There are solutions that provide anonymous reply. In (Chaum, 1981) Chaum proposed untraceable return addresses which allow the receiver to send a reply message without knowing senders' identity. Another example is the Mixminion (Danezis et al., 2003) which is an anonymous remailer protocol and it supports Single-Use Reply Blocks (or SURBs) to allow anonymous recipients. In these schemes the sender recursively encrypts the return address block and sends it in the body of the message. These encryptions are necessary, even if the receiver does not intend to reply.

3 OUR HYBRID MIX

A hybrid mix uses both asymmetric and symmetric operations. It applies asymmetric cryptographic solutions for the key exchange and symmetric ones for encrypting plaintexts. Therefore a hybrid mix efficiently handles long messages as well as short ones. We designed a hybrid mix that is based on admissable bilinear maps due to their good properties. As far as we know, our construction is the first hybrid mixnet, which is based on bilinear maps.

Usually, in case of anonymous communication the receiver needs to know whether the sender is allowed to send a message, *i.e. is eligible for it*. One can think of an e-exam or an e-voting scheme. In both cases there are requirements for participating. In case of e-exams we should verify whether the students accomplished all the prerequisites of an exam, in case of e-voting whether the voters are citizens and have a clean record etc. should be checked. Usually eligibility is not provided for mix networks built in, cryptographers have to solve this problem.

There are situations, when the receiver needs to send messages back to the anonymous sender in a way that the sender remains anonymous. For example, in case of e-tender systems, many times the anonymous applicants have to make up supplements. These cases a mix network should provide *anonymous reply*. In our construction cryptographic operations are needed

only if the receiver sends messages back, hence we increased efficiency.

In order to prevent illegal activities *anonymity revocation* is needed. There are circumstances when the identity of the anonymous sender should be retrieved. We provide eligibility and anonymity revocation with the help of a registry authority and the mixnet. We take advantages of bilinear maps, since they provide great services for three party protocols. For anonymity revocation besides the encrypted identities, we also use commitment values. By verifying these values a receiver makes sure, that after the deadline senders' anonymity can be revoked. These values are based on user-specific, registry-specific and mixnet-specific elements. Bilinear maps make possible for the three participants to verify (share) these values easily.

Considering practical aspects of a mixnet, it is usually used in a situation, where besides sender's anonymity, eligibility verification, possibility to reply to an anonymous sender and anonymity revocation are also required. We have designed a mix network, that provides all these requirements built in.

3.1 Preparation

We use the following notation for the participants. We denote senders by S_i, where $i = 1,\ldots,n$, the publicly known receiver by R, the registry authority by $\mathcal{RA}$, mix servers by $\mathcal{M}_i$, where $i = 1,\ldots,N$ (the last mix server is the receiver, *i.e.* R=$\mathcal{M}_N$) and the bulletin board by $\beta\beta$. Our proposed protocol can be built on any G_1, G_2 groups, where G_1 is a Gap Diffie-Hellman group and G_2 is a multiplicative group. We assume, that Gap Diffie-Hellman problem is hard.

1. $\mathcal{RA}$ generates system parameters: groups G_1, G_2, bilinear map $e : G_1^2 \to G_2$, generator element P of G_1, hash functions $H_1 : \{0,1\}^* \to G_1$ and $H_2 : G_2 \to \{0,1\}^l$. All parameters are made public.
2. Furthermore $\mathcal{RA}$ creates a random secret value $\bar{s} \in Z_q^*$ and outputs $\bar{s}P$ public key.
3. Each $\mathcal{M}_i$ generates random, secret, composite value m_i and outputs

$$\prod_{k=1}^{i} m_k P.$$

   Finally R chooses random, secret m_N and calculates $\overline{m}P = \prod_{k=1}^{N} m_k P$. This value is used for generating commitment values.
4. Each $\mathcal{M}_i$ chooses random, secret values $x_i \in Z_q^*$, then calculates and publishes $PK_{\mathcal{M}_i} = x_i \prod_{j=1}^{i} m_j P$ and $\prod_{j=1}^{i} x_j m_j P$ values.
5. R also chooses $x_N \in Z_q^*$ and publishes $PK_R = x_N \prod_{j=1}^{N} m_j P = x_N \overline{m} P$ and $\prod_{j=1}^{N} x_j m_j P = \overline{xm} P$ where $\bar{x}$ is never calculated explicitly. R also outputs $x_N P$ public key for providing anonymous reply.

3.2 Registration

We consider the situation, when there are several senders and only one receiver. In practise, often there is only one receiver, one can think of an e-voting, e-tender or an e-survey scheme. Senders send messages anonymously to a (not anonymous) receiver. During registration $\mathcal{RA}$ verifies the eligibility of each sender and blindly authorizes their messages. We applied blind short signatures (Boldyreva, 2003) with a small modification.

1. Let us denote the message by msg, that sender S_i would like to send to receiver R. S_i generates an $u^{(i)} \in Z_q^*$ random value.
2. S_i authenticates himself to $\mathcal{RA}$ and asks for authorization by sending his identification number with bit length l and $H_1(msg) + u^{(i)}P$ on a *secret, authenticated channel*.
3. $\mathcal{RA}$ verifies whether S_i is eligible for sending messages to R. If S_i is eligible, then $\mathcal{RA}$ blindly signs $H_1(msg) + u^{(i)}P$ and sends $\bar{s}(H_1(msg) + u^{(i)}P)$ to S_i. $\mathcal{RA}$ also calculates a commitment value μ_i for verification purposes and ε_i, that is the sender's identity number encrypted.

$$\begin{aligned} \mu_i &= e(\overline{m}P, H_1(msg) + u^{(i)}P)^{\bar{s}} \\ \varepsilon_i &= S_i \otimes H_2(e(\overline{xm}P, H_1(msg) + u^{(i)}P)^{\bar{s}}) \end{aligned}$$

4. S_i calculates $\bar{s}H_1(msg)$ with the knowledge of $u^{(i)}\bar{s}P$ and generates $msg||\bar{s}H_1(msg)$, where $\bar{s}H_1(msg)$ is $\mathcal{RA}$'s signature on msg. S_i also verifies $e(\bar{s}H_1(msg), P) = e(\bar{s}P, H_1(msg))$.

$\mathcal{RA}$ makes (μ_i, ε_i) pairs public in a permuted order on $\beta\beta$.

3.3 Message submission and mixing

S_i generates a secret, random value: $a_{s_i} \in Z_q^*$ that is necessary for the reply and calculates the following plaintext:

$$p = msg||\bar{s}H_1(msg)||a_{s_i}P$$

S_i generates the following symmetric encryption keys:

$$\begin{aligned} K_j^{(i)} &= H_2(e(PK_{\mathcal{M}_j}, \bar{s}P)^{u^{(i)}}), \text{where } j = 1,\ldots,N-1 \\ K_R^{(i)} &= H_2(e(PK_R, \bar{s}P)^{u^{(i)}}) \end{aligned}$$

encrypts plaintext p:

$$M_1^{(i)} = Enc_{K_1^{(i)}}(Enc_{K_2^{(i)}}(\ldots Enc_{K_R^{(i)}}(p))),$$

randomly chooses $u_1^{(i)}, u_2^{(i)}$ such that $u^{(i)} = u_1^{(i)} \cdot u_2^{(i)}$ and sends $v_1^{(i)} = u_1^{(i)} P || w_1^{(i)} = u_2^{(i)} \bar{s} P || M_1^{(i)}$ to $\mathcal{M}_1$.

Mix server $\mathcal{M}_j$ receives three values concatenated $v_j^{(i)} || w_j^{(i)} || M_j^{(i)}$ for each S_i. Values $v_j^{(i)}$ and $w_j^{(i)}$ are necessary for symmetric key generation and the third one is the encrypted message. Each mix collects all messages from each S_i, where $i = 1, \ldots, n$, hence receives:

$$v_j^{(i)} = \prod_{k=1}^{j-1} a_k^{(i)} \cdot u_1^{(i)} P \text{ where } j = 2, \ldots, N-1$$

$$w_j^{(i)} = \prod_{k=1}^{j-1} b_k^{(i)} \cdot u_2^{(i)} \bar{s} P \text{ where } j = 2, \ldots, N-1.$$

$\mathcal{M}_j$ calculates $v_{j+1}^{(i)} = a_j^{(i)} \cdot v_j^{(i)}$, $w_{j+1}^{(i)} = b_j^{(i)} \cdot w_j^{(i)}$, where $a_j^{(i)}, b_j^{(i)} \in Z_q^*$ randomly chosen, such that $m_j = a_j^{(i)} \cdot b_j^{(i)}$, then gets randomized symmetric keys $K_j^{(i)} = H_2(e(v_{j+1}^{(i)}, w_{j+1}^{(i)})^{x_j})$. $\mathcal{M}_j$ decrypts ciphertexts $M_{j+1}^{(i)} = Dec_{K_j^{(i)}}(M_j^{(i)})$, then permutes the list of triplets $v_{j+1}^{(i)} || w_{j+1}^{(i)} || M_{j+1}^{(i)}$, where $i = 1, \ldots, n$ and outputs them to mix $\mathcal{M}_{j+1}$.

3.4 Receiving the message

R receives $v_N^{(i)} || w_N^{(i)} || M_N^{(i)}$ from each S_i, calculates $K_R^{(i)} = H_2(e(v_N^{(i)}, m_N \cdot w_N^{(i)})^{x_N})$ and decrypts $p = Dec_{K_R^{(i)}}(M_N^{(i)})$. We repeat that $p = msg || \bar{s} H_1(msg) || a_{s_i} P$. R examines, whether the message p came from an eligible sender by verifying the signature of $\mathcal{RA}$. Hence R confirms whether:

$$e(\bar{s} H_1(msg), P) = e(\bar{s} P, H_1(msg))$$

R also checks whether the commitment value

$$\mu_i = e(v_N^{(i)}, m_N \cdot w_N^{(i)}) \cdot e(\bar{s} H_1(msg), \overline{m} P)$$

exists on $\beta\beta$. If R finds μ_i, then S_i sent the correct uP and $H_1(msg) + u^{(i)} P$ to the first mix and $\mathcal{RA}$. That means the sender's encrypted identity can be decrypted by the mix servers and R after the deadline. For eligible senders R stores:

$$\mu_i || msg || \bar{s} H_1(msg) || a_{s_i} P.$$

We will use μ_i for anonymity revocation and $a_{s_i} P$ to reply to the anonymous sender.

3.5 Anonymous reply

In case receiver R is willing to send a message t back to the anonymous sender S_i, then chooses a random value $r_{s_i} \in Z_q^*$ and calculates $\widehat{K_R^{(i)}} = H_2(e(r_{s_i} a_{s_i} P, r_{s_i}^{-1} \bar{s} P)^{x_N})$ symmetric key, and encrypts message t: $\widehat{M_1^{(i)}} = Enc_{\widehat{K_R^{(i)}}}(t)$. R sends $r_{s_i} a_{s_i} P || r_{s_i}^{-1} \bar{s} P || \widehat{M_1^{(i)}}$ for each sender to $\mathcal{M}_1$. Each mix server $\mathcal{M}_j$ after receiving $\widehat{v_j^{(i)}} || \widehat{w_j^{(i)}} || \widehat{M_j^{(i)}}$, where

$$\widehat{v_j^{(i)}} = \prod_{k=1}^{j-1} a_k^{(i)} \cdot r_{s_i} a_{s_i} P, \ \widehat{w_j^{(i)}} = \prod_{k=1}^{j-1} b_k^{(i)} \cdot r_{s_i}^{-1} \bar{s} P.$$

calculates

$$\widehat{v_{j+1}^{(i)}} = a_j^{(i)} \cdot \widehat{v_j^{(i)}}, \ \widehat{w_{j+1}^{(i)}} = b_j^{(i)} \cdot \widehat{w_j^{(i)}}$$

$$\widehat{K_j^{(i)}} = H_2(e(\widehat{v_{j+1}^{(i)}}, \widehat{w_{j+1}^{(i)}})^{x_j}), \ \widehat{M_{j+1}^{(i)}} = Enc_{\widehat{K_j^{(i)}}}(\widehat{M_j^{(i)}}).$$

Values $a_j^{(i)}$ and $b_j^{(i)}$ are chosen randomly such that $m_j = a_j^{(i)} \cdot b_j^{(i)}$, and sends $\widehat{v_{j+1}^{(i)}} || \widehat{w_{j+1}^{(i)}} || \widehat{M_{j+1}^{(i)}}$ to $\mathcal{M}_{j+1}$. Server $\mathcal{M}_{N-1}$ outputs all the calculated values with $H_1(\widehat{K_{N-1}^{(i)}})$ to $\beta\beta$.
S_i calculates keys

$$\widehat{K_j^{(i)}} = H_2(e(PK_{\mathcal{M}_j}, \bar{s} P)^{a_{s_i}}), \text{where } j = 1, \ldots, N-1$$

$$\widehat{K_R^{(i)}} = H_2(e(x_N P, \bar{s} P)^{a_{s_i}}).$$

and looks for $H_1(\widehat{K_{N-1}^{(i)}})$ on $\beta\beta$, accesses all data and decrypts the proper $\widehat{M_N^{(i)}}$ with the keys above. S_i gets $t = Dec_{\widehat{K_R^{(i)}}}(Dec_{\widehat{K_1^{(i)}}}(\ldots Dec_{\widehat{K_{N-1}^{(i)}}}(\widehat{M_N^{(i)}})))$ plaintext.

3.6 Anonymity revocation

There are several applications, when after a certain deadline the identity of the anonymous sender should be revealed. We could think of either an e-tender or an e-exam scheme. In general, anonymity revocation should be provided even if the sender is not willing to reveal his identity (e.g. an examinee does not want to get a bad grade).

Our solution determines senders' real identity with the help of the mixnet. Receiver R sends value $\mu_i^{x_N}$ to the first mix. Each server $\mathcal{M}_j$ power the received value to x_j and sends it to the next server. Finally, $\mu_i^{\bar{x}}$ is given. After $H_2(\mu_i^{\bar{x}}) =$

$H_2(e(\overline{xm}P, H_1(msg) + u^{(i)}P)^{\bar{s}})$ is calculated, identity number S_i is received with the help of ε_i.

We should mention, that if the sender is willing to participate in the revocation process, then the real identity can be determined without the mix servers in an easier and lower-cost way. If the sender provides the secret value $u^{(i)}$, $\mathcal{RA}$ can retrieve the sender's identity by calculating $H_2(e(\overline{xm}P, H_1(msg) + u^{(i)}P)^{\bar{s}}$.

4 Security evaluation

In this section we show that our mix provides security requirements of correctness, anonymity and eligibility.

4.1 Correctness

First we prove that our scheme is correct concerning the mix process, the anonymous reply and also the process of anonymity revocation.

Definition 4.1. *We call our mixnet correct, if for every plaintext calculated by the receiver there is a corresponding ciphertext in the input list of the mixnet. This means that every plaintext is a multiple decryption of a ciphertext, and no two plaintexts are the multiple decryptions of the same ciphertext.*
The following theorem states that our mixnet is correct.

Theorem 4.1. *The proposed mix protocol is operating correctly.*

Proof. Each sender S_i (where $i = 1, \ldots, n$) sends a triplet $(v_1^{(i)} = u_1^{(i)}P || w_1^{(i)} = u_2^{(i)}\bar{s}P || M_1^{(i)})$ to the first mix server $\mathcal{M}_1$. The third value $M_1^{(i)}$ is an N-times encryption of the plaintext p which contains the message *msg* of S_i. $\mathcal{M}_1$ receives n triplets and $\mathcal{M}_j$ (where $j = 2, \ldots, N-1$) receives a permutation of modified triplets from $\mathcal{M}_{j-1}$. The sender calculates the symmetric keys for secure communication with all mix server $\mathcal{M}_j$:

$$K_j^{(i)} = H_2(e(PK_{\mathcal{M}_j}, \bar{s}P)^{u^{(i)}}) = H_2(e(x_j \prod_{k=1}^{j} m_k P, \bar{s}P)^{u^{(i)}})$$

where $j = 1, \ldots, N-1$ and the mix server $\mathcal{M}_J$ (where $J = 1, \ldots, N-1$) calculates this symmetric key:

$$K_J^{(i)} = H_2(e(\prod_{k=1}^{J} a_k^{(i)} u_1^{(i)} P, \prod_{k=1}^{J} b_k^{(i)} u_2^{(i)} \bar{s}P)^{x_J})$$

Because of the bilinear property of mapping e the corresponding keys are the same if and only if $j = J$. (Note that $m_k = a_k^{(i)} \cdot b_k^{(i)}$ and $u^{(i)} = u_1^{(i)} \cdot u_2^{(i)}$.) The receiver R receives a set of the triplets from $\mathcal{M}_{N-1}$:

$$(v_N^{\sigma(i)} || w_N^{\sigma(i)} || M_N^{\sigma(i)})$$

where $\sigma(i)$ is the permutation of $i = 1, \ldots, n$ and gets:

$$(v_N^{\sigma(i)} || m_N \cdot w_N^{\sigma(i)} || M_N^{\sigma(i)})$$

The receiver in order to get the plaintexts does the following calculations for all $M_N^{(j)}$:

$$p'_j = Dec_{K_R^{(j)}}(M_N^{(j)}) = Dec_{K_R^{(j)}}(Enc_{K_R^{(i)}}(p_i))$$

where $j = 1, \ldots, n$, $i = \sigma(j)$ and

$$K_R^{(j)} = H_2(e(v_N^{(j)}, m_N \cdot w_N^{(j)})^{x_N})$$

$$= H_2(e(\prod_{k=1}^{N-1} a_k^{(j)} u_1^{(i)} P, m_N \prod_{k=1}^{N-1} b_k^{(j)} u_2^{(i)} \bar{s}P)^{x_N})$$

and the symmetric key for R calculated by the sender S_i:

$$K_R^{(i)} = H_2(e(PK_R, \bar{s}P)^u) = H_2(e(x_N \overline{m}P, \bar{s}P)^{u^{(i)}})$$

Thus using the bilinear property of mapping e the receiver able to get a plaintext if and only if $K_R^{(j)} = K_R^{(i)}$ and then the plaintext of p'_j is p_i.

The anonymous reply works similarly to the message submission. In this case the sender is R and the anonymous receiver is the sender S_i who sent the message *msg* that is stored with $a_{s_i}P$. In order to send the reply message S_i calculates $\widehat{H_1(K_{N-1}^{(i)})} = H_1(H_2(e(PK_{\mathcal{M}_{N-1}}, \bar{s}P)^{a_{s_i}})) = H_1(H_2(e(x_{N-1} \prod_{j=1}^{N-1} m_j P, \bar{s}P)^{a_{s_i}}))$ and searches this on $\beta\beta$. The list of messages contains values

$$\widehat{v_N^{(i)}} = \prod_{k=1}^{N-1} a_k^{(i)} r_{s_i} a_{s_i} P,$$

$$\widehat{w_N^{(i)}} = \prod_{k=1}^{N-1} b_k^{(i)} r_{s_i}^{-1} \bar{s}P,$$

$$\widehat{M_N^{(i)}} = \widehat{Enc_{K_{N-1}^{(i)}}}(\ldots(\widehat{Enc_{K_1^{(i)}}}(\widehat{Enc_{K_R^{(i)}}}(t))))$$

where $\widehat{K_j^{(i)}} = H_2(e(\prod_{k=1}^{j} a_k^{(i)} r_{s_i} a_{s_i} P, \prod_{k=1}^{j} b_k^{(i)} r_{s_i}^{-1} \bar{s}P)^{x_j})$ calculated by mix server $\mathcal{M}_j$. Due to the bilinear property of e these keys are the same keys as the sender S_i calculates for $\mathcal{M}_j$: $H_2(e(PK_{\mathcal{M}_j}, \bar{s}P)^{a_{s_i}}) = H_2(e(x_j \prod_{k=1}^{j} m_k P, \bar{s}P)^{a_{s_i}})$.

Furthermore R calculates the symmetric key:

$$\widehat{K_{R_1}} = H_2(e(r_{s_i} a_{s_i} P, r_{s_i}^{-1} \bar{s}P)^{x_N})$$

and S_i calculates the symmetric key:

$$\widehat{K_{R_2}} = H_2(e(x_N P, \bar{s}P)^{a_{s_i}})$$

Mapping e has bilinear property so

$\widehat{K_{R_1}} = H_2(e(r_{s_i} a_{s_i} P, r_{s_i}^{-1} \bar{s}P)^{x_N}) = H_2(e(x_N P, \bar{s}P)^{a_{s_i}}) = \widehat{K_{R_2}}$ holds.

Let us note that from the anonymous participants only S_i is able to calculate the necessary keys, since the secret value a_{s_i} is need.

□

4.2 Anonymity

We consider *static* adversary in a *semi-honest model*. A model is called semi-honest, if the dishonest users follow the protocol and also keep a record of all intermediate results. An adversary is static, if corrupted players are specified at the beginning of the protocol, they stay corrupted during the whole process and no new ones stand in with them. The adversary observes all public information and possesses all attacked players' secret information (*i.e.* keys, permutation).

The anonymity property of our system says that an adversary who has access to corrupt players' secret data and observes all the public information of the protocol including views of the registry and mix servers, input ciphertexts and the shuffled list of output messages, cannot tell which message was sent by which sender. We also assume, that there is at least one mix server and two senders that are not corrupted by the adversary, *i.e.* the secret permutation and secret keys are not revealed to the adversary, furthermore the registry and the receiver do not collude.
In order to give the proof, we assume, that the following problem in (G_1, G_2, e) is hard.

Definition 4.2. *For every* $r, r_1, r_2, r_3, r_4 \in Z_q^*$ *given* $P, rP \in G_1$ *and* (V_0, W_0), (V_1, W_1), $(r_1 V_b, r_2 W_b), (r_3 V_{\bar{b}}, r_4 W_{\bar{b}})$, *where* $r_1 r_2 = r_3 r_4 = r$, *the problem is to output* $b \in \{0,1\}$.

Let us review the Matching Find-Guess Problem (MFGP) (Fujisaki and Okamoto, 1999).

Definition 4.3. *Matching Find-Guess (MFG) Problem (Fujisaki and Okamoto, 1999)*
For every plaintexts x_0, x_1 *and for every symmetric keys* K_0, K_1 *given* $(Enc_{K_0}(x_0), Enc_{K_1}(x_1), x_b, x_{\bar{b}})$, *the problem is to output* $b \in \{0,1\}$.

Studying the proposed mix network, one can see, that the identity of a sender cannot be retrieved, since the attacker cannot connect messages of the input lists of $\mathcal{M}_i$ and $\mathcal{M}_{i+1}$. because the attacker is not able to solve the MFGP, or the BDHP or the problem given in Definition 4.2.

An adversary is able to connect messages of the input lists, if he can *calculate the secret symmetric key*, *i.e.* he can calculate $K_j^{(i)} = H_2(e(PK_{\mathcal{M}_j}, \bar{s}P)^{u^{(i)}}) = H_2(e(u_1^{(i)} P, u_2^{(i)} \bar{s}P)^{x_i \prod_{j=1}^{i} m_j})$, where $j = 1, \ldots, N$.

The adversary has access to the public key $PK_{\mathcal{M}_i} = x_i \prod_{j=1}^{i} m_j P$ and the messages

$$\begin{aligned} v_1^{(i)} &= u_1^{(i)} P, \\ w_1^{(i)} &= u_2^{(i)} \bar{s} P, \end{aligned}$$

that are sent by S_i. Since the BDHP is hard, the adversary is not able to calculate $K_j^{(i)} = H_2(e(u_1^{(i)} P, u_2^{(i)} \bar{s}P)^{x_i \prod_{j=1}^{i} m_j})$.

An adversary can connect messages of the input lists, if he is able to find a relationship between $M_j^{(i)}$ and $M_j^{(i+1)}$. Since the MFGP is hard, this is not possible.

The third way to match messages is to connect the pairs $(v_j^{(i)}, w_j^{(i)})$ and $(v_{j+1}^{(i)}, w_{j+1}^{(i)})$. Since the problem given in Definition 4.2 is hard, the adversary is not successful.

4.3 Eligibility

We assume a threat model, where senders during registration and message submission are in a controlled room, *i.e.* corrupt, eligible senders are not allowed to send messages (e.g. $msg, \bar{s}(H1(msg)), u^{(i)}$) to the adversary. This case, the adversary should choose values $(v_1^{(i)}, w_1^{(i)}, M_1^{(i)})$ such that, after the mix process the receiver could find $\mu_i = e(v_N^{(i)}, m_N \cdot w_N^{(i)}) \cdot e(\bar{s}H_1(msg), \overline{m}P) = e(\overline{m}P, H_1(msg) + u^{(i)}P)^{\bar{s}}$ on $\beta\beta$. Assuming that short signature generation without the knowledge of the secret key is hard, the attacker cannot calculate the correct triplet, hence cannot submit messages to the mix network to be successful.

5 Properties

We also examined the time and space complexity of our solution and compared it to the identity-based scheme proposed by Zhong (Zhong, 2009). His scheme is also based on bilinear pairings and implements a mix network, as ours. We denoted the operations as follows: additions in G_1 (ADD), scalar multiplications of elements in G_1 (SMU), multiplication in G_2 (MUL), bilinear maps (BMP) and divisions in G_2 (DIV). First we compare the number of that operations which provided by both systems: submitting, mixing and receiving messages.

The following tables contain the number of the calculations of the participants.

Sender (#n)	SMU	MUL	BMP	ADD
IB mix	2	1	1	1
Our mix	3	0	N	0

Mix servers (#$N-1$)	ADD	MUL	SMU	BMP
IB mix	n	n	$2n$	$2n$
Our mix	0	0	$3n$	n

Receiver	ADD	MUL	SMU	BMP	DIV
IB Mix	n	(n+N-1)	$2n$	$2n$	n
Our mix	0	0	$2n$	n	0

In our system the sender performs more calculations, since symmetric keys are generated for encrypting arbitrary long messages, that is not provided in (Zhong, 2009). Nevertheless, both the receiver's and the mix calculations are more efficient in our case. We should mention, that in case of cascade mixnets the first server starts its operation only if it receives enough number of messages. Basically, the efficiency of a mix depends on the computations made by the servers and the receivers.

Compare to Zhong's system we provide additional and optional services: anonymous reply, eligibility and revocation. The process and the cost of the *reply* are similar to the message submission.

In our solution the sender's *eligibility* is provided by the signature of the registration authority $\mathcal{RA}$ and the value μ_i on $\beta\beta$. The extra cost of eligibility verification is for the senders: 1 SMU, 2 BMP, 1 ADD and 1 ADDINV (additive inverse in G_1) calculations, for the registration authority: n SMU and $2n$ BMP calculations and for the receiver: $4n$ BMP calculations, n MUL and n SMU, (where n is the number of the senders thus the number of the messages).

We provide *anonymity revocation*, as well. If the senders are not cooperative, then the receiver R and the mix servers together retrieve the senders' identities, that costs n EXP (exponentiation in G_2) calculations for the receiver and for each server.

The overall space complexity of the communication is the following:

	κ bits security params
IB mix	$(3N-2)\kappa n$
Our mix	$3N\kappa n$

It is shown that our mix is capable of handling arbitrary long messages in an efficient way with $2\kappa n$ extra space.

6 Conclusion and Future work

In this paper we proposed a bilinear pairing-based hybrid mix and its security and efficiency evaluation. We proved, that the participants could send messages in an anonymous way and if it is necessary the real identity can be revoked after a certain deadline by the collaboration of the receiver and the mix servers.

Our mix also provides the possibility of eligibility verification, that is an important service in case of anonymous communication.

Furthermore, our mix network allows anonymous reply, in a way that the sender's identity still remains secret.

Finally, this is a hybrid mix, making possible of transmitting messages with arbitrary length. We could think of either an e-voting application where the messages are usually short, or an e-exam application where the messages could be short or long, as well.

In the future we plan to give a formal security evaluation for anonymity and eligibility and examine how our hybrid mix can be applied for one the most complicated applications, for an e-exam.

REFERENCES

Boldyreva, A. (2003). Threshold signatures, multisignatures and blind signatures based on the gap-diffie-hellman-group signature scheme. In *Proceedings of the 6th International Workshop on Theory and Practice in Public Key Cryptography: Public Key Cryptography*, PKC '03, pages 31–46, London, UK, UK. Springer-Verlag.

Boneh, D. and Franklin, M. K. (2001). Identity-based encryption from the weil pairing. In *Proceedings of the 21st Annual International Cryptology Conference on Advances in Cryptology*, CRYPTO '01, pages 213–229, London, UK, UK. Springer-Verlag.

Chaum, D. L. (1981). Untraceable electronic mail, return addresses, and digital pseudonyms. *Commun. ACM*, 24(2):84–90.

Danezis, G., Dingledine, R., Hopwood, D., and Mathewson, N. (2003). Mixminion: Design of a type iii anonymous remailer protocol. In *In Proceedings of the 2003 IEEE Symposium on Security and Privacy*, pages 2–15.

Desmedt, Y. and Kurosawa, K. (2000). How to break a practical mix and design a new one.

Federrath, H., Jerichow, A., and Pfitzmann, A. (1996). Mixes in mobile communication systems: Location management with privacy. In *Proceedings of the First International Workshop on Information Hiding*, pages 121–135, London, UK, UK. Springer-Verlag.

Fujisaki, E. and Okamoto, T. (1999). Secure integration of asymmetric and symmetric encryption schemes. In *Proceedings of the 19th Annual International Cryptology Conference on Advances in Cryptology*, CRYPTO '99, pages 537–554, London, UK, UK. Springer-Verlag.

Goldschlag, D. M., Reed, M. G., and Syverson, P. F. (1996). Hiding routing information. In *in Information Hiding*, pages 137–150. Springer-Verlag.

Golle, P., Jakobsson, M., Juels, A., and Syverson, P. (2002). Universal re-encryption for mixnets. In *IN PROCEEDINGS OF THE 2004 RSA CONFERENCE, CRYPTOGRAPHERS TRACK*, pages 163–178. Springer-Verlag.

Gulcu, C. and Tsudik, G. (1996). Mixing email with babel. In *Symposium on Network and Distributed System Security*, pages 2–16.

Huang, L., Yamane, H., Matsuura, K., and Sezaki, K. (2006). Silent cascade: Enhancing location privacy without communication qos degradation. In Clark, J. A., Paige, R. F., Polack, F., and Brooke, P. J., editors, *SPC*, volume 3934 of *Lecture Notes in Computer Science*, pages 165–180. Springer.

Jakobsson, M. (1998). A practical mix. In *Advances in Cryptology - EUROCRYPT '98, International Conference on the Theory and Application of Cryptographic Techniques, Espoo, Finland, May 31 - June 4, 1998, Proceeding*, pages 448–461.

Jakobsson, M. and Juels, A. (2001). An optimally robust hybrid mix network. PODC'01.

Jakobsson, M., Juels, A., and Rivest, R. L. (2002). Making mix nets robust for electronic voting by randomized partial checking. In *Proceedings of the 11th USENIX Security Symposium*, pages 339–353, Berkeley, CA, USA. USENIX Association.

Jerichow, A., Mller, J., Pfitzmann, A., Pfitzmann, B., and Waidner, M. (1998). Real-time mixes: a bandwidth-efficient anonymity protocol. *IEEE Journal on Selected Areas in Communications*, pages 495–509.

Joux, A. (2000). A one round protocol for tripartite diffie-hellman. In *Proceedings of the 4th International Symposium on Algorithmic Number Theory*, ANTS-IV, pages 385–394, London, UK, UK. Springer-Verlag.

Markus, J. and Ari, J. (1999). Millimix: Mixing in small batches. Technical report.

Michels, M. and Horster, P. (1996). Some remarks on a receipt-free and universally verifiable mix-type voting scheme. In Kim, K. and Matsumoto, T., editors, *ASIACRYPT*, volume 1163 of *Lecture Notes in Computer Science*, pages 125–132. Springer.

Mitomo, M. and Kurosawa, K. (2000). Attack for flash mix. In *In Advances in Cryptology - ASIACRYPT 2000, LNCS*, pages 192–204. Springer-Verlag.

Neff, C. A. (2001). A verifiable secret shuffle and its application to e-voting. In *Proceedings of the 8th ACM Conference on Computer and Communications Security*, CCS '01, pages 116–125, New York, NY, USA. ACM.

Ohkubo, M. and Abe, M. (2000). A length-invariant hybrid mix. In Okamoto, T., editor, *ASIACRYPT*, volume 1976 of *Lecture Notes in Computer Science*, pages 178–191. Springer.

Parekh, S. (1996). Prospects for remailers. *First Monday*, 1(2).

Pfitzmann, A., Pfitzmann, B., and Waidner, M. (1991). Isdn-mixes: Untraceable communication with very small bandwidth overhead. In *In Proceedings of the GI/ITG Conference on Communication in Distributed Systems*, pages 451–463. Springer-Verlag.

Sako, K. and Kilian, J. (1995). Receipt-free mix-type voting scheme: A practical solution to the implementation of a voting booth. In *Proceedings of the 14th Annual International Conference on Theory and Application of Cryptographic Techniques*, EUROCRYPT'95, pages 393–403, Berlin, Heidelberg. Springer-Verlag.

Sampigethaya, K. and Poovendran, R. (2006). A survey on mix networks and their secure applications. *Proceedings of the IEEE*, 94(12):2142–2181.

Syverson, P. F., Goldschlag, D. M., and Reed, M. G. (1997a). Anonymous connections and onion routing. In *Proceedings of the 1997 IEEE Symposium on Security and Privacy*, SP '97, pages 44–, Washington, DC, USA. IEEE Computer Society.

Syverson, P. F., Goldschlag, D. M., and Reed, M. G. (1997b). Protocols using anonymous connections: Mobile applications. In *in Security Protocols: Fifth International Workshop*, pages 13–23. Springer-Verlag.

Verheul, E. R. (2001). Evidence that xtr is more secure than supersingular elliptic curve cryptosystems. In *J. Cryptology*, pages 195–210. Springer-Verlag.

Zhong, S. (2009). Identity-based mix: Anonymous communications without public key certificates. *Computers & Electrical Engineering*, (5):705–711.