Fast Mixnets for Small Message Spaces ⋆

Extended abstract

René Peralta   Sheng Zhong

Department of Computer Science
Yale University

Abstract. We present new designs for implementing anonymous communication channels. Our techniques are based on implementation of Mixnets under certain restrictions on the number of messages and/or the size of the message space. The restrictions do not impact the applicability of our results in most real-world scenarios. They do, however, allow us to greatly improve on the efficiency of previous designs.

1 Introduction

Implementation of anonymous communication channels over the internet is important in various applications. Known techniques solve this problem in a general scenario where both the number of messages and the size of the message space is potentially very large. The complexity of the known solutions, as well as their computation and communication costs is partly a consequence of solving a very general problem. In many applications, however, either the message space or the number of messages is small. Such applications include electronic voting (where the size of the message space is small) and certain real-time financial transactions (such as a service provider communicating to a billing server the transactions of the last hour of business).

In this paper, we present fast and simple implementations of anonymous channels for small message spaces. Various indexing techniques can be used to reduce the problem of anonymizing a small number of messages to the small message space case. Our Mixnets are more efficient than previous designs.

2 Previous work

Among the earliest implementations of anonymous channels is Chaum's "Mixnet" [3]. The Mixnet construction is the basic technique used by most anonymous channel implementations today, including what we present in this paper. A Mixnet consists of a number z of "Mix servers". These are

⋆ Research partially supported by NSF grant CCR-0081823. Keywords: Anonymous communication channel; Mixnet; electronic voting; e-commerce.

network nodes each of which accepts a number $n > 1$ of messages at a time and re-routes them in random order. Cryptographic techniques are used to preserve confidentiality of messages and integrity of the messages in the face of a possibly corrupted Mix server.

The Mixnet by Park, Itoh and Kurosawa [7] is the basis for several later works. In their construction, the inputs are encrypted z times. That is, one time for each Mix server. For rerouting, each Mix server "peels" one layer of encryption, and then randomly reencrypts and permutes the messages. This Mixnet achieves anonymity against passively cheating servers, but is weak when there are actively cheating servers. Sako and Kilian's Mixnet [10] achieves anonymity against actively cheating servers. In this Mixnet, every Mix server must prove it has manipulated the messages according to the protocol. The proofs are zero-knowledge and are constructed using a standard cut-and-choose technique. Abe [1] was able to reduce the number of proofs by certifying the overall effect of the Mixnet, instead of every individual server's actions. The proofs of overall effects are implemented using secure multi-party computations among the servers. However, the basic proofs are based on cut-and-choose methods and are therefore quite expensive in computation and communication. Jakobsson proposed a Mixnet without cut-and-choose proofs [6]. His design is asymptotically the most efficient of all known Mixnets. For small-scale applications, Abe's second Mixnet design [2] can be more efficient than Jakobsson's.

A Mixnet is "universally verifiable" if its functionality can be verified by anyone on the net. Universal verifiability is achieved by posting proofs on a "bulletin board". Implementation of bulletin boards usually requires a broadcast mechanism. Broadcast is much more expensive than unicast. Therefore, designs and/or implementations of Mixnets often sacrifice universal verifiability for efficiency reasons. This is the case with the Naval Research Laboratory's "Onion" [9, 12]. The latter is probably the most well-known implementation of Mixnets.

3 A Fast Mixnet for 0/1 Messages

The most expensive component of most Mixnet protocols is the one in which a Mix server (or the set of all Mix servers) proves that the outputs are a permutation of the inputs. In order to make this step much more efficient, we make use of the fact that (when messages are restricted to one bit) the outputs of a Mix server are a permutation of the inputs if and only if the <u>sum</u> of both input and output messages is the same.

Let P and Q be (large) primes such that $P = 2Q + 1$. Assume g and h are two generators of the Quadratic Residue subgroup of Z_P^*. As in [2], the Mix servers share a private key k using a Verifiable Secret Sharing (VSS) scheme based on Shamir's technique [11]. In this application, $k = f(0)$ where f is a polynomial of degree $\tau - 1$ over the field Z_Q. The parameter τ is the number of Mix servers that need to pool their knowledge in order to determine the secret k. We will denote by $k_j = f(j)$ the share of k known to Mix server j. Each secret share has a public counterpart $K_j = g^{k_j} \% P$. The public key is $K' = g^k \% P$. The shares are distributed without need for a trusted third party using the protocol by Pedersen [8].

Let the messages be $m_1, m_2, \ldots, m_n$. We encode these with a variant of ElGamal encryption:

$$I_i = (h^{m_i} g^{kr_i} \% P, g^{r_i} \% P)$$

where $m_i = 0$ or 1 for $i = 1, 2, \ldots, n$ and "%" is the modular operator. After encryption, inputs, I_1, I_2, …, I_n, are sent, over a public channel, to Mix server 1.

When a Mix server receives the messages, it reencrypts and permutes them randomly, and then sends them to the next server. Let the inputs of Mix server j be $E_{j1}, E_{j2}, \ldots, E_{jn}$, and its outputs be $E'_{j1}, E'_{j2}, \ldots, E'_{jn}$. The server computes T_{ji}, a reencryption of E_{ji}, in the following manner:

$$E_{ji} = (M_{ji}, G_{ji}) = (h^{m_i} g^{kr_{ji}} \% P, g^{r_{ji}} \% P)$$

$$\Longrightarrow T_{ji} = (M_{ji}(g^k)^{r'_{ji}} \% P, G_{ji} g^{r'_{ji}} \% P) = (h^{m_i} g^{k(r_{ji}+r'_{ji})} \% P, g^{(r_{ji}+r'_{ji})} \% P)$$

where $r'_{j1}, r'_{j2}, \ldots, r'_{jn}$ are randomly selected. Then server j permutes the T_{ji}'s to get

$$E'_{ji} = (M'_{ji}, G'_{ji}) = T_{j,\rho(i)}$$

where ρ is a random permutation.

For universal verifiability, each server j only shows a simple proof of discrete logarithm equality: for

$$(M_j, G_j) = (\prod_{i=1}^{n} M_{ji} \% P, \prod_{i=1}^{n} G_{ji} \% P)$$

$$(M'_j, G'_j) = (\prod_{i=1}^{n} M'_{ji} \% P, \prod_{i=1}^{n} G'_{ji} \% P)$$

server j shows

$$\log_{g^k} \frac{M'_j}{M_j} \equiv \log_g \frac{G'_j}{G_j} \pmod{P}$$

using the following protocol (due to Chaum and Pedersen [4]):[A]

1. the server selects $R_1, R_2 \in Z_Q$ uniformly at random and publishes $g^{kR_1} \% P$ and $g^{R_2} \% P$;
2. the verifier generates a challenge $c \in Z_Q$ uniformly at random;
3. the server publishes $Y_1 = c \sum_{i=1}^{n} r_{ji} + R_1$ and $Y_2 = c \sum_{i=1}^{n} r_{ji} + R_2$;
4. the verifier checks

   $$g^{kR_1} (\frac{M'_j}{M_j})^c \equiv (g^k)^{Y_1} \pmod{P}$$

   and

   $$g^{R_2} (\frac{G'_j}{G_j})^c \equiv g^{Y_2} \pmod{P}$$

[A] A non-interactive implementation of this protocol is possible using standard cryptographic techniques to simulate a random oracle.

After the last Mix server, z, reencrypts and permutes the messages, τ Mix servers can jointly decrypt the messages using standard Lagrangian interpolation techniques (see [5, 2]). That is, a subset S of τ Mix servers can jointly compute

$$h^{m_i} = \frac{M'_{zi}}{(G'_{zi})^k} \;\%P = \frac{M'_{zi}}{\prod_{j\in S}(G'_{zi})^{k_j \prod_{u\in S-\{j\}} \frac{-u}{j-u}}} \;\%P.$$ [B]

Each server j is responsible for raising G'_{zi} to k_j. Note that h^{m_i} is produced without any of the servers learning k. To prove the decryption, each server j should show that it has computed $S_{ji} = (G'_{zi})^{k_j} \%P$ correctly. Since K_j is public, it simply proves

$$\log_{G_{zi}} S_{ji} \equiv \log_g K_j \pmod P$$

using the Chaum-Pedersen protocol presented above.

Let's call this Mixnet FM_1.

Claim. FM_1 is universally verifiable for $0/1$ messages if the number of messages n is less than Q.

Proof. First, we show: if the inputs of the Mixnet are encryptions of $0/1$, the outputs of the Mixnet are $0/1$'s, and Mix servers can generate the required proofs, then the outputs must be a permuted decryption of the inputs. Consider Mix server j. Its proof (published when it processes the messages, *before the decryption phase*) ensures

$$h^{\sum_{i=1}^n D_k(E_{ji})} \equiv h^{\sum_{i=1}^n D_k(E'_{ji})} \pmod P$$

where $D_k()$ is the decryption function. Equivalently, we have:

$$\sum_{i=1}^n D_k(E_{ji}) \equiv \sum_{i=1}^n D_k(E'_{ji}) \pmod Q.$$

Since $E_{j+1,1}, E_{j+1,2}, \ldots, E_{j+1,n}$ are exactly $E'_{j1}, E'_{j2}, \ldots, E'_{jn}$ (the input of server $j+1$ is the output of server j),

$$\sum_{i=1}^n D_k(E_{ji}) \equiv \sum_{i=1}^n D_k(E_{j+1,i}) \pmod Q.$$

Because this holds for every $j \quad (1 \le j < z)$, we can safely conclude

$$\sum_{i=1}^n D_k(E'_{zi}) \equiv \sum_{i=1}^n D_k(E_{zi}) \equiv \sum_{i=1}^n D_k(E_{1i}) \equiv \sum_{i=1}^n D_k(I_i) \equiv \sum_{i=1}^n m_i \pmod Q.$$

Since

$$\sum_{i=1}^n D_k(E'_{zi}) \le n < Q$$

and

$$\sum_{i=1}^n m_i \le n < Q,$$

[B] For easy reference, we provide a proof of this interpolation formula in the technical appendix to this paper.

we can reduce the above to

$$\sum_{i=1}^{n} D_k(E'_{zi}) = \sum_{i=1}^{n} m_i.$$

As the decryption phase is known to be universally verifiable (see [2]), the sum of the outputs of the Mixnet must be the same as the sum of the decryptions of the original inputs. On the other hand, a sequence of n 0/1's is a permutation of another sequence of n 0/1's if and only if the two sequences have the same sums. Therefore the outputs of the Mixnet must be a permuted decryption of the original inputs.

Second, consider the case where some outputs are not 0/1's. The unexpected outputs warn the public that somebody is cheating. Therefore the cheating is detected and some measure can be taken against the attack.

Finally, consider the case where some inputs are illegal (not encryptions of 0/1), although all outputs are 0/1's. Then we can safely argue that there exists a subset of outputs such that it is a permuted decryption of the set of legal inputs. We simply don't care how the Mixnet deals with the illegal inputs. It's enough to ensure all legal inputs are processed correctly. □

FM_1 achieves anonymity for honest users when an adversary controls at most $\tau - 1$ mix servers. The extra communication cost necessary to achieve universal verifiability of permuted reencryptions is z broadcast messages, assuming non-interactive proofs are used.

For the verifiability of the decryption, every server involved still needs n extra broadcast messages. So the overall communication cost is $n\tau + z$ broadcast messages. This is significantly less communication than that of other Mixnets.

4 Extensions to larger message spaces

The design of FM_1 is based on the fact that two binary strings are permutations of each other if and only if their Hamming weight is the same. Our technique can be generalized to handle strings of size n over an alphabet of size L. Our "alphabet" is the set of integers $\Omega = \{0, 1, n+1, n^2+n+1, \ldots, n^{L-2}+\ldots+1\}$.Note the cyclotomic polynomial of degree i has the more succinct representation $\frac{n^{i+1}-1}{n-1}$. Each message is encoded as an integer in Ω. It is a simple observation that two strings of n messages are permutations of each other if and only if the sum of the messages is the same for both strings. Therefore the same techniques as in section 3 can be used to implement a Mixnet FM_2 capable of handling n messages from a space of size L. However, the parameters n, L, Q must satisfy

$$Q > \frac{n^{L-1} - 1}{n - 1} \sim n^{L-2}.$$

For Q of length 512 bits or more, this allows for n, L large enough for most applications.

Our last Mixnet, FM_3 encodes the messages using distinct primes. The encryption formula is redefined in such a way that unique factorization allows us to verify permuted reencryptions. Define a function $\vartheta(x)$ as "the $(x+1)$th quadratic residue prime mod P". Then we can encode message m_i as

$$I_i = E_{g^k \% P}(m_i, r_i) = (\vartheta(m_i) g^{kr_i} \% P, g^{r_i} \% P).$$

Everything else remains the same as FM_1. The proof for a permuted reencryption ensures that

$$\prod_{i=1}^{n} \vartheta(m_i) \% P$$

is preserved.

The L^{th} quadratic residue modulo P is approximately $2L \ln L$. [C] Thus, for this Mixnet to be able to handle n messages from a message space of size L, we must have

$$P > (2L \ln L)^n.$$

As in FM_2, our third Mixnet is capable of handling n, L large enough for most applications. Note, however, that the bound on the modulus P is exponential in L for FM_2 and exponential in n for FM_3. Therefore both designs should be considered when the size of the modulus is constrained by choice or design.

As was the case for FM_1, the Mixnets in this section can be shown to achieve anonymity for honest users when an adversary controls at most $\tau - 1$ Mix servers. The communication and computation costs are similar to FM_1 and thus significantly less than that of previous Mixnets.

References

1. M. Abe. Universally verifiable mix-net with verification work independent of the number of mix-servers. In *Advances in Cryptology - Proceedings of EUROCRYPT 98*, volume 1403 of *Lecture Notes in Computer Science*, pages 437–447. Springer-Verlag, 1998.
2. M. Abe. Mix-networks on permutation networks. In *Advances in Cryptology - Proceedings of ASIACRYPT 99*, volume 1706 of *Lecture Notes in Computer Science*, pages 258–273. Springer-Verlag, 1999.
3. D. Chaum. Untraceable electronic mail, return address and digital pseudonyms. *Communications of the ACM*, 24(2):84–88, 1981.
4. D. Chaum and T. Pedersen. Wallet databases with observers. In *Advances in Cryptology - Proceedings of CRYPTO 92*, volume 740 of *Lecture Notes in Computer Science*, pages 89–105, 1993.
5. Y. Desmedt and Y. Frankel. Threshold cryptosystems. In *Advances in Cryptology - Proceedings of CRYPTO 89*, volume 435 of *Lecture Notes in Computer Science*, pages 307–315. Springer-Verlag, 1990.
6. M. Jakobsson. A practical Mix. In *Advances in Cryptology - Proceedings of EUROCRYPT 98*, volume 1403 of *Lecture Notes in Computer Science*, pages 448–461. Springer-Verlag, 1998.

[C] This can be slightly improved by choosing P appropriately, i.e. so that many small primes are quadratic residues modulo P.

7. C. Park, K. Itoh, and K. Kurosawa. Efficient anonymous channel and all/nothing election scheme. In *Advances in Cryptology - Proceedings of EUROCRYPT 93*, volume 765 of *LNCS*, pages 248–259. SV, 1993.
8. T. Pedersen. A threshold cryptosystem without a trusted third party. In *Advances in Cryptology - Proceedings of EUROCRYPT 91*, volume 547, pages 522–526, 1991.
9. M. Reed, P. Syverson, and D. Goldschlag. Anonymous connections and Onion routing. *IEEE Journal on Selected Areas in Communications: Special Issue on Copyright and Privacy Protection*, 16(4):482–494, 1998.
10. K. Sako and J. Kilian. Receipt-free Mix-type voting schemes- a practical solution to the implementation of a voting booth. In *Advances in Cryptology - Proceedings of EUROCRYPT 95*, volume 921 of *Lecture Notes in Computer Science*, pages 393–403. Springer-Verlag, 1995.
11. A. Shamir. How to share a secret. *Communications of the ACM*, 22(11):612–613, 1979.
12. P. Syverson, M. Reed, and D. Goldschlag. Onion routing access configurations. In *DISCEX2000:Proceedings of the DARPA information survivability conference and exposition*, pages 34–40. IEEE CS Press, 2000.

technical appendix

For the sake of completeness, we provide a proof for the interpolation formula

$$h^{m_i} = \frac{M'_{zi}}{(G'_{zi})^k} \, \%P = \frac{M'_{zi}}{\prod_{j \in S} (G'_{zi})^{k_j \prod_{u \in S-\{j\}} \frac{-u}{j-u}}} \, \%P.$$

That is, we need to show

$$k \equiv \sum_{j \in S} k_j \prod_{u \in S - \{j\}} \frac{-u}{j-u} \bmod Q$$

where

- $P = 2Q + 1$ (P and Q prime);
- $f(x)$ is a polynomial of degree $\tau - 1$ over Z_Q;
- $k = f(0)$;
- $k_i = f(i)$ for $i = 1, \ldots, z$;
- S is any subset of $\{1, \ldots, z\}$ of cardinality τ.

Let

$$g(x) = \sum_{j \in S} k_j \prod_{u \in S - \{j\}} \frac{x-u}{j-u} \bmod Q.$$

We claim, $f = g$ over Z_Q. Since Z_Q is a field, and g is clearly of degree at most $\tau - 1$, it is enough to show that f and g share τ points. Since $\tau = |S|$ it is enough to show that $f(r)$ and $g(r)$ are equal for each $r \in S$.

Consider

$$
\begin{aligned}
g(r) &= \sum_{j \in S} k_j \prod_{u \in S-\{j\}} \frac{r-u}{j-u} \\
&= \sum_{j \in S-\{r\}} k_j \prod_{u \in S-\{j\}} \frac{r-u}{j-u} + k_r \prod_{u \in S-\{r\}} \frac{r-u}{r-u} \\
&= k_r \\
&= f(r);
\end{aligned}
$$

since

$$\prod_{u \in S-\{j\}} \frac{r-u}{j-u} = 0 \qquad \text{when } j \neq r \text{ and } r \in S$$

and

$$\prod_{u \in S-\{r\}} \frac{r-u}{r-u} = 1$$

.

Thus, $f(x) = g(x)$ over Z_Q. Evaluating at 0 yields

$$k = f(0) = g(0) = \sum_{j \in S} k_j \prod_{u \in S-\{j\}} \frac{-u}{j-u} \bmod Q$$

as claimed.