The effectiveness of mixnets – an empirical study


Nazmul Islam


Kazi Md Rokibul Alam


Ashiqur Rahman

Nazmul Islam, Kazi Md Rokibul Alam and Ashiqur Rahman, Khulna University of Engineering and Technology, Bangladesh

A mixnet is a multi-stage system that accepts encrypted messages as its input and generates a new altered output while exploiting cryptographic operations and repeated permutations to ensure the untraceability between the input and the output messages. Based on the employed cryptographic operations, the main types of mixnets and their variants are: decryption, re-encryption, universal re-encryption and hybrid mixnets.

Our research evaluated mixnets based on several criteria such as: the number of messages traversing through the mixnet; the key length of the underlying cryptosystem; and the number of mix-routers involved in a mixnet. Finally, we compare mixnets on the basis of the computation time requirement for the above mentioned criteria while sending messages anonymously.

Introduction

Today, anonymous communication has become popular in many social applications where the privacy of individual opinion is crucial. In cryptographic social applications such as electronic voting, anonymous email, anonymous telecommunication, location privacy in wireless network, anonymous Internet communication and so on, where the disclosure of person's opinion on sensitive social issues is unsafe, an anonymous channel is used for privacy.[1,2,3,4,5] Thus an anonymous channel is essential for various protocols that include anonymous message passing between peers where the sent messages cannot be traced back to the sender.

Until now, the modern computer network is unable to provide this privacy because of its IP nature, as shown in Figure 1.

To ensure privacy in communications, an anonymous channel was introduced that is usually implemented through a mixnet.[6] In order to provide anonymity, each mix-router involved in a mixnet accepts a batch of encrypted messages as input then executes mixing (ie, cryptographic transformation and random permutation) to change the appearance of the input batch and generates an output batch. Here, the output cannot be traced back to the input, hence untraceability is achieved. In a mix-router, inputs may arrive at different times. After receiving a complete batch of input, the mix-router starts its operations and then forwards the batch to the next mix-router.[7] Figure 2 shows the behavior of a mixnet.

Some other practical implementation of anonymous channels, beside mixnets, include the Onion Routing Program, the Tor Project etc.[8,9] In the Onion Routing Program, the anonymous communication system has been implemented to create a secured system for Internet-based anonymous activities. The mechanism proposed in the Tor Project is for mass use where anyone can install a client and actively participate in the network. Previously, only decryption and re-encryption types of mixnets have been implemented employing 512- and 1024-bit encryption.[10] However, the empirical study presented in this article has performed simulation of RSA- and ElGamal-based decryption, re-encryption, universal re-encryption and hybrid types of mixnets employing both 1024- and 2048-bit encryption while varying the number of messages and the number of mix routers. Then experimental results have been shown

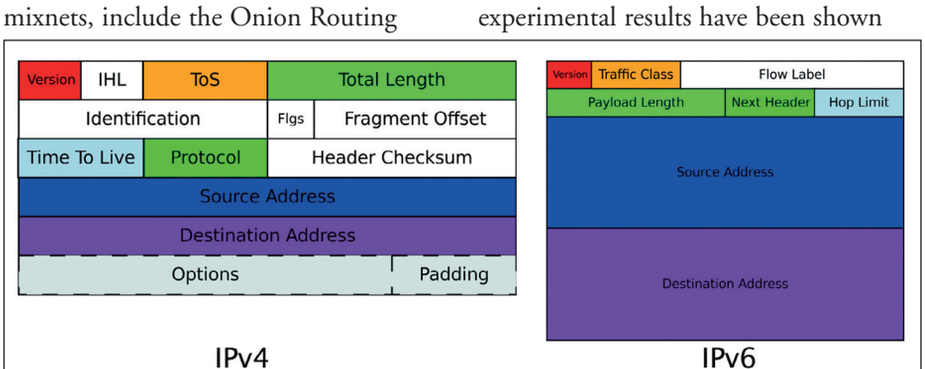

Figure 1: The headers of existing Internet Protocol (IP) packets contain both sender's and receiver's addresses. Consequently anyone can trace both the sender and the receiver. Moreover, the receiver can always identify the sender.

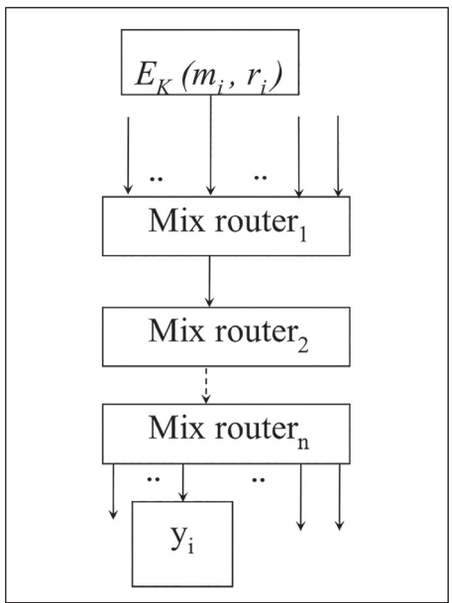

Figure 2: A mixnet's basic architecture with *n* mix-routers. Sender *i* encrypts m_i with random string r_i using public key *K*. For decryption, mixnet *K* is the public key of the mixnet and for the re-encryption mixnet it is the public key of the receiver.

for these implementations which provide some guidelines such as:

- Which mixnet implementation is faster?
- Does the performance of mixnet depends on the number of messages?
- Does the key length of the underlying cryptosystem affect the speed?
- Does the number of mix-routers involved in a mixnet influence the time of total transmission of the messages from the sender to the receiver?

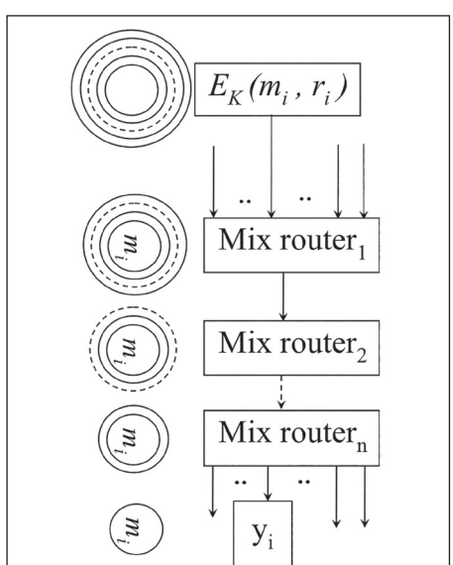

Figure 3: The process of sending messages through a decryption mixnet. *K* is the public key of the mixnet. On the way, each router pulls off a layer from all onions of the batch. Here a single onion is shown.

Mixnets basically fall into two types – decryption and re-encryption. In a decryption mixnet, each mix-router only decrypts and permutes the input batch whereas with each mix-router in a re-encryption mixnet, re-encrypts and permutes the input batches and finally they are decrypted. Hybrid and universal re-encryption mixnets are variations of decryption and re-encryption mixnets.

Decryption mixnets

RSA and ElGamal are two well-known cryptosystems that are used for decryption mixnets.

In an RSA-based decryption mixnet, senders encrypt their messages with the public keys of the mix-routers of a mixnet using the RSA cryptosystem. Then each mix-router decrypts its input batch with the corresponding private key that changes the appearance of the messages and randomly permutes them, ensuring untraceability between the input and output batches. In a decryption mixnet, a sender i concatenates its message with a random string r and encrypts it with the public keys of n mix-routers in the anonymous path:

$$forwarding\ onion_n = E_{K_x}(m) \,||\, r_n$$
$$forwarding\ onion_{n-1} = E_{K_n}(forwarding\ onion_n) \,||\, r_{n-1}$$
$$\vdots$$
$$forwarding\ onion_0 = E_{K_1}(forwarding\ onion_2) \,||\, r_1$$
$$E_K(m,r) = forwarding\ onion_0$$

Here $K=(K_1, K_2, \ldots, K_n)$ are the public keys of n mix-routers, and $r = (r_0, r_1, \ldots, r_n)$ are the random strings used to randomise the encryption at each mix-router. K_x is the public key of the receiver. So the sender broadcasts an n layer onion which is given as:

$$E_K(m, r) = E_{K_1}(E_{K_2}(\ldots(E_{K_n}(E_{K_x}(m) \,||\, r_n))\ldots) \,||\, r_2) \,||\, r_1$$

Each mix-router on the path peels off a layer from the onion – ie, decrypts using the private key $D_{Kj}=K_j^{-1}$ as:

$$D_{K_j}(E_{K_j}(forwarding\ onion_{j+1}))$$

Thus mix-router j decrypts other onions of its input batch, received from the mix-router j–1. After the completion of the decryption operation, mix-router j permutes all the onions of the batch using a random permutation $\pi_j : l \rightarrow l\ \pi_j : l \rightarrow l$ where l is the batch size. This concludes the mixing operation of mix-router j. The resulting quantities are the forwarding onions which have been reduced in size and constitute the mixed output batch of mix-router j. These onions are forwarded simultaneously to the next mix-router j+1. All the remaining mix-routers, on the path, repeat the same operations until, finally, the last mix-router n outputs the decrypted quantity which is sent to the receiver.

In an ElGamal-based decryption mixnet, a sender i only needs to perform a single encryption for all the n stages as[11]:

$$E_K(m, r) = (g^r \,||\, mK^r)$$

Where g is a generator, r is a random string and K is the public key of the mixnet, computed from the public keys of the mix-router as:

$$K = \prod_{j=1}^{n} K_j = \prod_{j=1}^{n} g^{d_j} = g^{\sum_{j=1}^{n} d_j}$$

Where K_j and d_j, are the public and private key, respectively, of mix-router . Any stage j can randomly decrypt the sender i input, using its private key d_j and random string r_j as:

$$D_{K_j}(E_K(m,r)) = g^r g^{rj} \,||\, mK^r (g^r)^{-d_j} \left(\prod_{a=1,\, a \neq j}^{n} g^{d_a} \right)^{r_j}$$
$$= g^{r+r_j} \,||\, m \left(g^{\sum_{a=1,\, a \neq j}^{n} d_a r} g^{d_j r} \right) \left(g^{-d_j r} \right) \left(\prod_{a=1,\, a \neq j}^{n} g^{d_a r_j} \right)$$
$$= g^{r+r_j} \,||\, m \left(\prod_{a=1,\, a \neq j}^{n} g^{d_a r + r_j} \right)$$

In the first step of the above, the mix-router j uses the first component of its input in the sender's encryption to obtain $(g^r)^{-dj}$ and uses the product of public keys of the stages that are yet to decrypt. After decryption of more inputs to form a batch, the stage j broadcasts the mixed batch to the remaining n-1

stages. The process repeats, with another stage performing the decryption, until finally all n stages have decrypted using their private keys.

Hybrid mixnet

A hybrid mixnet is a more efficient variation of the decryption mixnet.[12] In a decryption mixnet, a sender performs expensive public key operations on the increasing size of the onions. A hybrid mixnet uses both public and symmetric key operations to achieve efficiency. Here, the expensive public key encryption is only required for symmetric keys that are non-increasing in size and relatively efficient symmetric key operations are used for increasing-in-size onions. In a hybrid mixnet, the forwarding onion received by mix-router j from mix-router $j-1$ is given by:

$$forwarding\ onion_j = E_{K_j}(K_j) \,||\, E_{K_j}[forwarding\ onion_{j+1}]$$

Similar to the decryption mixnet, mix-router j peels a layer with its private key K_j^{-1}. But here, on decryption mix-router j obtains a symmetric key h_j, which is then used to decrypt the forwarding onion to mix-router $j+1$. Hence, the complete sender i onion in the hybrid mixnet is given as:

$$E_K(m,r,h)=E_{K_1}(h_1) \,||\, E_{h_1}[E_{K_2}(h_2) \,||\, E_{h_2}[E_{K_3}(h_3) \,||\, \ldots \,||\, E_{h_{n-1}}[E_{K_n}(E_{K_x}(m) \,||\, r)]\ldots]$$

Where $K=(K_1, K_2, \ldots, K_n)$ are the public keys of the mix-routers, $h=(h_1, h_2, \ldots, h_n)$ are the symmetric keys chosen by the sender, and r is a random string.

Re-encryption mixnet

The basic idea of re-encryption mixnet is to utilise the re-encryption property of ElGamal encryption. Here, a sender i only needs to perform a single encryption with the public key of the mixnet and to change the appearance of the input a mix-router simply re-encrypts with a random string. Here, no predetermined order of mix-routers is required. A sender performs the following operation and broadcasts the message:

$$E_K(m, r)=(g^r \,||\, mK^r)$$

Where g is a generator, r is a random string and K is the public key of the receiver, computed as $K=g^d$ where d, is the private key of receiver. To change the appearance of the input, a mix-router simply re-encrypts them with a random string. The following cryptographic operation can be performed first, by any stage j, on the input of sender i.

$$E_K(m, r+r_j)=(g^r g^{r_j} \,||\, mK^r K^{r_j})=(g^{r+r_j} \,||\, mK^{r+r_j})$$

Where r_j is a random string used by the mix-router j to re-encrypt the sender's input. After re-encrypting and permuting a batch of input, mix-router j broadcasts the mixed batch to the remaining mix-routers for further mixing. This mixing operation terminates at stage n. Each quantity of mixed output batch can be decrypted as:

$$D_K\left(E_K\left(m, r+\sum_{j=1}^{n} r_j\right)\right)=\frac{mK^{r+\sum_{j=1}^{n} r_j}}{\left(g^{r+\sum_{j=1}^{n} r_j}\right)^d}$$

Universal re-encryption mixnet

With a universal re-encryption mixnet, a sender i broadcasts two ElGamal encryptions, one containing the message and other containing the public key of the receiver used to encrypt the message as follows:

$$E_K(m,r)\,||\,E_K(1,r')=(g^r\,||\,mK^r) \,||\, (g^{r'}\,||\,K^{r'})$$

After encryption, the sender broadcasts the message. A mix-router j performs the following cryptographic operation:

$$\left(g^r g^{r' r_j}\,||\,mK^r K^{r' r_j}\right)||\left(g^{r' r_j'}\,||\,K^{r' r_j'}\right)$$

$$=\left(g^{r+r' r_j}\,||\,mK^{r+r' r_j}\right)||\left(g^{r' r_j'}\,||\,K^{r' r_j'}\right)$$

$$=E_K\left(m, r+r' r_j\right)||E_K\left(1, r' r_j'\right)$$

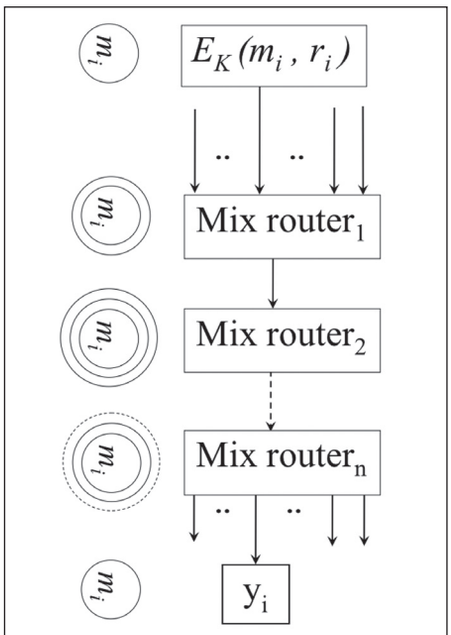

Figure 4: Process of sending a message through a re-encryption mixnet. *K* is the public key of the receiver. Each router re-encrypts the incoming batch. The re-encryption process of a single message of the batch is shown.

Where r', r_j are the random strings chosen by the mix-router j. All the other mixnets perform the same re-encryption operation with different random strings. As the sender includes the encrypted form of public key K, each stage can perform re-encryption operations without the knowledge of K. Only the message containing the portion of the mixnet output batch is broadcast by the mixnet as:

$$E_K(m,R_c)=(g^{R_c}\,||\,mK^{R_c})$$

Where R_c is a combination of random strings used by the mix-routers to re-encrypt sender i input.

Experimental analysis

This section evaluates the performance of mixnets based on varying the number of messages, the number of mix-routers involved in mixnets and the key length of the underlying cryptosystem.

To perform the experiments, the environment consisted of a 32-bit Windows machine (Windows 7), dual-core 2.5GHz CPU with 2048MB of RAM. The C# language and Microsoft Visual Studio 2010 with .Net Framework 4 was used

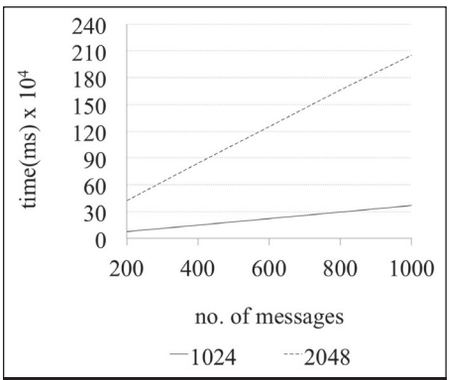

Figure 5: Traverse time requirement with variation of number of messages for RSA-based decryption mixnet.

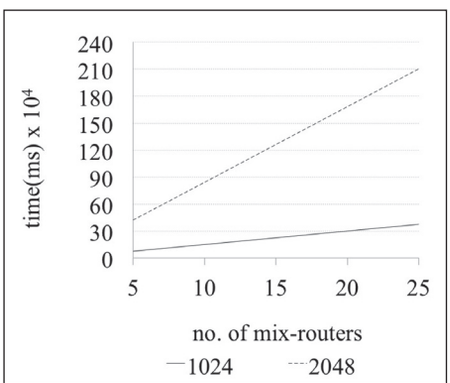

Figure 6: Traverse time requirement with variation of number of mix-routers for RSA-based decryption mixnet.

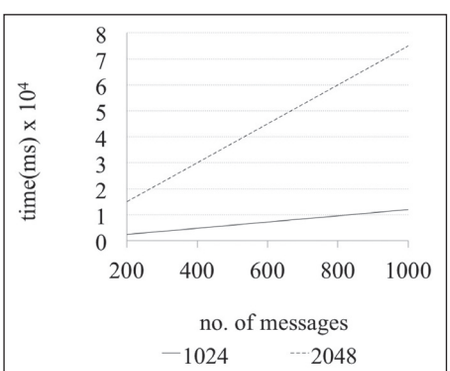

Figure 7: Traverse time requirement with variation of number of messages for ElGamal-based decryption mixnet.

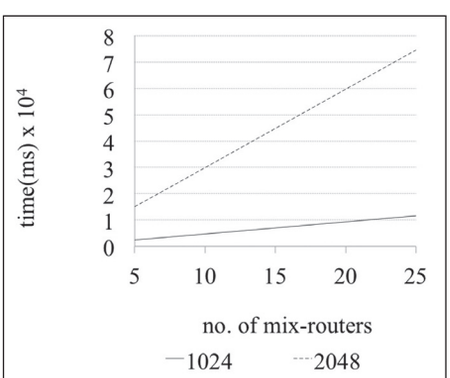

Figure 8: Traverse time requirement with variation of number of mix-routers for ElGamal-based decryption mixnet.

for coding. The ender-receiver address is not considered while simulation and router-to-router travelling time is assumed to be negligible – ie, all computation times do not include the communication time. Moreover the different operations of privacy-related applications that are not related to cryptography have not been considered. The mixnets are simulated using the key length of 1024 bits and 2048 bits and keeping the message length as 40 digits. To prevent trace-back from output to input, the Fisher Yates algorithm was used for shuffling.[13] For RSA encryption key generation, the Euclidian algorithm is an efficient method and is used to compute the greatest common divisor (GCD) of two integers. [14] The extended Euclidian algorithm is also used to find the modular multiplicative inverse of the encryption key – ie, to generate the decryption key.

Decryption mixnet

Both RSA and ElGamal cryptosystem-based decryption mixnets were simulated. In RSA based decryption mixnet, as the number of encryption is equal to the number of mix-routers of a mixnet, therefore it requires huge time for messages to reach from the sender to the receiver. Here Figs. 5 and 6 show that the time requirement is proportionally increasing when the number of messages as well as the key length increases (although thereby the security increases).

In Figure 5, for 200 messages of 40 digits, it requires 75075ms and 421006ms to traverse through five mix-routers employing 1024- and 2048-bit keys respectively. This time requirement increases if the number of massages increases. For example, when the number of messages is 800, the time requirement is 291057ms and 1681105ms for 1024- and 2048-bit keys respectively. Similarly the time requirement increases when the number of mix-routers increases although the number of messages is fixed is shown in Figure 6. Unlike the RSA-based type, an ElGamal-based decryption mixnet requires less time as there is one single encryption on the sender's side. The results

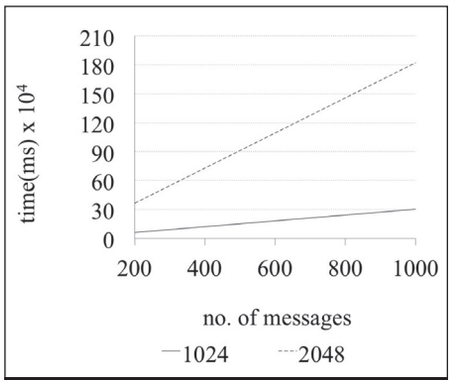

Figure 9: Traverse time requirement with variation of number of messages for hybrid mixnet.

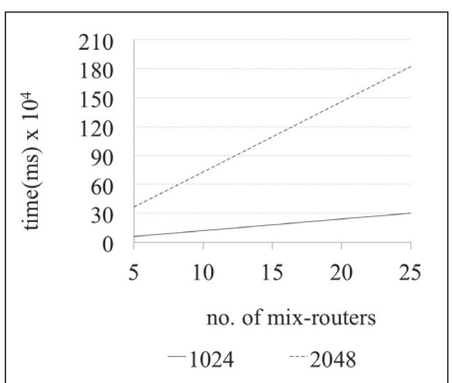

Figure 10: Traverse time requirement with variation of number of mix-routers for hybrid mixnet.

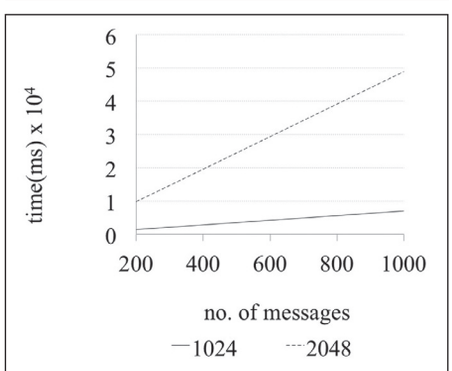

Figure 11: Traverse time requirement with variation of number of messages for re-encryption mixnet.

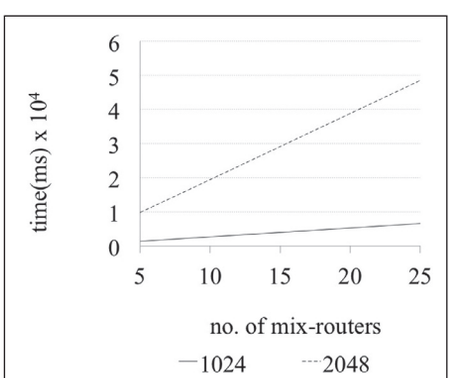

Figure 12: Traverse time requirement with variation of number of mix-routers for re-encryption mixnet.

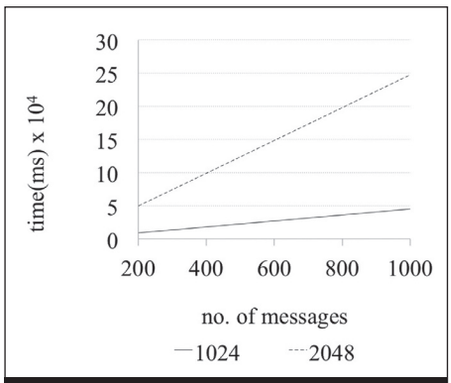

Figure 13: Traverse time requirement with variation of number of messages for universal re-encryption mixnet.

of our simulation are shown in Figures 7 and 8. The figures show that when the number of messages and mix-routers increase, the time requirement also increases respectively.

Hybrid mixnet

In this research, AES and RSA algorithms are used as symmetric and asymmetric encryption. For AES encryption, the block size and key size is kept at 128 bits and 256 bits respectively. Figures 9 and 10 show the simulation result for a hybrid mixnet. Figure 9 shows that 200 messages of 40 digits need approximately 60494ms and 364620ms respectively for 1024- and 2048-bit encryption with five intermediate mix-routers.

Similarly, Figure 10 shows the traverse time requirement with the variation of number of mix-routers for 200 messages. Here, messages are passed through 5, 10, 15 and 20 mix-routers and the time requirement gradually increases while the number of mix-routers increases.

Re-encryption mixnet

The simulation result of a re-encryption mixnet with the ElGamal cryptosystem is presented in Figures 11 and 12. According to this simulation, 200 messages require approximately 1398ms and 9786ms to traverse through a network of five mix-routers for 1024- and 2048-bit encryption respectively. For 400 messages, this time requirement increases to 2796ms and 19572ms. Similarly the time requirement increases when the number of messages increases. Figure 12 shows the increment of time requirement when the number of mix-routers increases. For a network of 10 mix-routers the time requirement is 2696ms and 19472ms for 1024- and 2048-bit encryptions keeping the number of messages fixed at 200.

Universal re-encryption mixnet

The universal re-encryption mixnet has the same mathematical calculation as the re-encryption mixnet but it requires some extra operations to ensure that the received ciphertexts are in a common group.[15] For this reason it has some extra overhead compared to the re-encryption mixnet. Figures 13 and 14 show the simulation results for the universal re-encryption mixnet.

For a batch of 200 messages it requires 9028ms and 495045ms for 1024- and 2048-bit encryption respectively when the number of mix-routers is five. Again, for the batches consisting of 400, 600, 800, 1000 messages, the time requirement increases gradually. The mix-router and time requirement graph of a universal re-encryption mixnet is shown in Figure 14 – as before, keeping the batch of 200 messages fixed, and passing it through 5, 10, 15, 20 and 25 mix-routers.

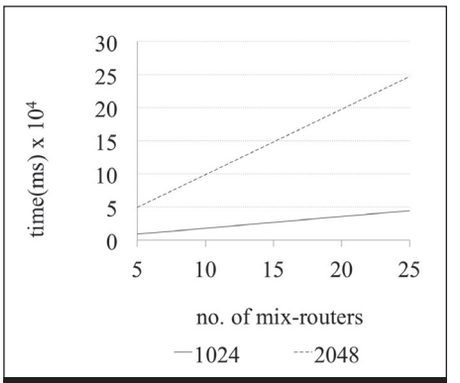

Figure 14: Traverse time requirement with variation of number of mix-routers for universal re-encryption mixnet.

Comparison

An RSA-based decryption mixnet requires a fixed sequence of mix-routers for mixing and if any mix-router is damaged, the receiver will not receive the message suc-

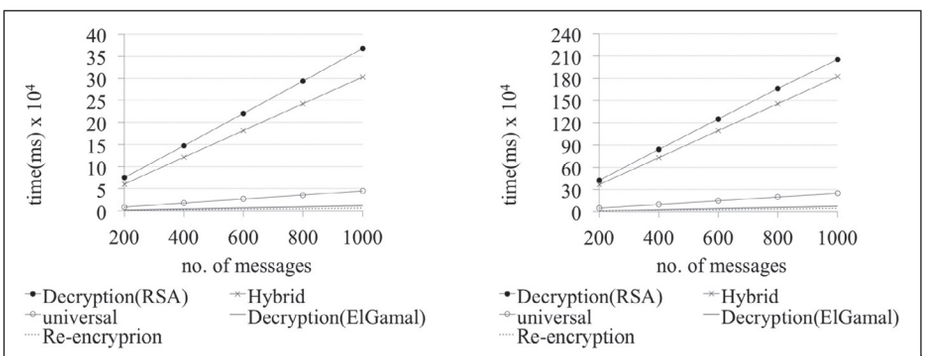

Figure 15: Comparison of number of messages – time requirement graphs of all mixnets for 1024-bit encryption (left) and 2048-bit encryption (right).

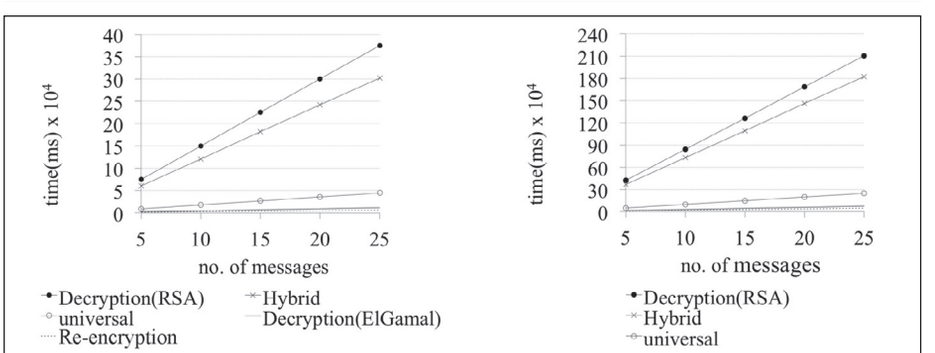

Figure 16: Comparison of number of mix-routers – time requirement graphs of all mixnets for 1024-bit encryption (left) and 2048-bit encryption (right).

cessfully. That's why this type of mixnet could not gain popularity. Moreover, it requires multiple encryptions by the sender. This problem is addressed by the ElGamal-based decryption mixnet where each sender performs a single encryption. But this type of decryption mixnet still requires all mix-routers to perform mixing.

"A mixnet is essential in applications where anonymity is a crucial issue. It can satisfy requirements for scalability, untraceability and security"

The ElGamal-based re-encryption mixnet requires single encryption by the sender but doesn't require all the mix-routers to perform mixing. An improved variation of re-encryption mixnet is universal re-encryption mixnet. This lets us construct a mixnet in which mix-routers hold no public or private key material and may therefore dispense with the cumbersome requirements of key generation, key distribution and private key management. One major limitation of the public key cryptosystem is that the plaintext size is restricted. Aimed at solving this problem, the hybrid mixnet uses both symmetric and asymmetric keys. The benefit is that this mixnet is able to accept input messages of arbitrary length. Figures 15 and 16 put together all the experimental data in a graph to compare all the mixnets described so far.

Conclusions

This research presents experimental analysis of mixnets and their variants. While the applications of mixnets are growing gradually, we have evaluated the performance of their types. A mixnet is essential in applications where anonymity is a crucial issue. It can satisfy requirements for scalability, untraceability and security between the senders and the receivers at the same time. Although anonymity can be provided by some other techniques or network architectures, for anonymous communication in terms of untraceability from the receiver to the sender, the mixnet is still the most attractive solution.

From this simulation the following conclusions can be drawn. An RSA-based decryption mixnet is very slow because of multiple encryptions by the sender and requires more time than other mixnets. With the hybrid mixnet, the expensive public key encryption is only required for the symmetric keys which are non-increasing in size and symmetric key operations are used for the increasing size onion. The operations of symmetric key are faster than asymmetric key operations. For this reason it is more time efficient than decryption mixnet. ElGamal-based decryption and re-encryption mixnets require much less time than RSA-based decryption and hybrid mixnets because a sender only needs to perform a single encryption for all mix-routers. But a universal re-encryption mixnet requires more time than a re-encryption mixnet as it requires two ElGamal encryptions.

About the authors

Dr Kazi Md Rokibul Alam is a professor at the computer science and engineering department of Khulna University of Engineering & Technology. He did his PhD at the University of Fukui, Japan and an MS at Bangladesh University of Engineering & Technology, Bangladesh. He is a member of IEEE. He contributed as a corresponding author for this paper. Nazmul Islam and Ashiqur Rahman hold BSc degrees in computer science and engineering from Khulna University of Engineering & Technology.

References

1. AKM Rokibul, S Tamura, S Taniguchi and T Yanase. 'An anonymous voting scheme based on confirmation numbers'. IEEJ Transactions EIS, 130 (2010) 2065-2073.
2. G Danezis, R Dingledine and N Mathewson. 'Mixminion: Design of a type III anonymous remailer protocol'. Proc. 2003 IEEE Symp. Security Privacy, (2003) 2-15.
3. A Jerichow, J Muller, A Pfitzmann, B Pfitzmann and M. Waidner. 'Real-time mixes: A bandwidth-efficient anonymity protocol'. IEEE J. Select. Areas Commun, 16 (1998) 495-509.
4. L Huang, K Matsuura, H Yamane and K Sezaki. 'Enhancing wireless location privacy using silent period'. Proc. IEEE Wireless Communications Networking Conf, (2005).
5. MG Reed, PF Syverson and DM Goldschlag. 'Anonymous connections and onion routing'. IEEE J. Special Areas Commun, 16 (1998) 482-494.
6. D Chaum. 'Untraceable electronic mail, return addresses, and digital pseudonyms'. Commun. ACM, 24 (1981) 84-88.
7. K Sampigethaya, R Poovendran. 'A Survey on Mix Networks and Their Secure Applications'. Proc. IEEE, 94 (2006) 2142-2181.
8. Onion routing. Accessed Jun 2013. www.onion-router.net/.
9. Tor project. Accessed Jun 2013. https://www.torproject.org/.
10. P Ribarski and L Antovski. 'Mixnets: Implementation and Performance Evaluation of Decryption and Re-encryption Types'. Journal of Computing and Information Technology, 20 (2012) 225-231.
11. C Park, K Itoh and K Kurosawa. 'Efficient anonymous channel and all/nothing election scheme'. Advances in Cryptology-Eurocrypt. New York: Springer-Verlag, (1994) 248-259.
12. B Moller. 'Provably secure public-key encryption for length-preserving Chaumian mixes'. Proc. CT-RSA. New York: Springer-Verlag, (2003) 244-262.
13. RA Fisher, F Yates. 'Statistical tables for biological, agricultural and medical research'. third ed, Oliver & Boyd, London, 1979.
14. AJ Menezes, PC van Oorschot and SA Vanstone. 'Hand-book of Applied Cryptography'. CRC Press, Boca Raton, FL, 1996.
15. P Golle, M Jakobsson, A Juels and P Syverson. 'Universal re-encryption for mixnets'. Proc. RSA Conf Cryptographers' Track, (2004) 163-178.