Comparing Performance of Decryption and Re-Encryption Mixnets

Pance Ribarski and Ljupcho Antovski

Institute of Informatics, Faculty of Natural Sciences Skopje
{pance,anto}@ii.edu.mk
http://www.ii.edu.mk

Abstract. Anonymous channels have been the assumption of many protocols which include anonymous message passing between peers. One way of accomplishing this is by using a structure called Mixnet. Since the first Chaumian Mixnet there have been many implementations of mixnets. The two main types of developed mixnets are decryption (chaumian) and re-encryption mixnets. We analyze four types of mixnets, one decryption and three re-encryption mixnets. The mixnets are implemented in Java and tested by several criteria: message number, total nodes, threshold nodes and key length of underlying cryptosystem. The results are compared in order to answer the questions of type: which type of mixnet do we choose for specific type of job.

Keywords: mixnets, decrypting, reencrypting, evoting

1 Introduction

Anonymous message passing through computer networks is imperative in most secure protocols. Anonymous channels have been the assumption of many protocols which include anonymous message passing between peers. The IP nature of today's networks is breaking this anonymicity by adding sender and receiver ip addresses in every packet transmissed through the network. Knowing this, everybody can lookup the packets and get the knowledge of who is sending the message and to whom it is sent. Further more, the receiver always knows who sent the message and can relate the message back to the sender.

The term Anonymous channels was coined by scientists and practicioners to make clear that the sent message cannot be related back to the sender. Because of the mentioned "anomaly" in computer networks another approach needs to be taken in hand to achieve the anonimicity of passing messages. One of the common implementations of anonymous channels is Mixnets. Mixnets are a set of noders which are included in the anonymous channel implementation (see Fig. 1). When a sender wants to pass a message, he/she sends the message to one node of this set. Then the received messages are permuted and passed through the nodes. Finaly the message exits through the last node and gets to the receiver party. If at least one node behaves correctly and doesn't expose the

permutation, the anonimicity of passed messages is guaranteed.

In the following sections we will address the two basic categories of mixnets - decryption and reencryption. Then we will present results of tests made over four implementations. The tests are made to give answers to the questions like: which mixnet is faster, does the speed depends on the number of messages, does the keylength of underlying cryptosystem affects speed, how does the number of nodes included in the changes timings and so on.

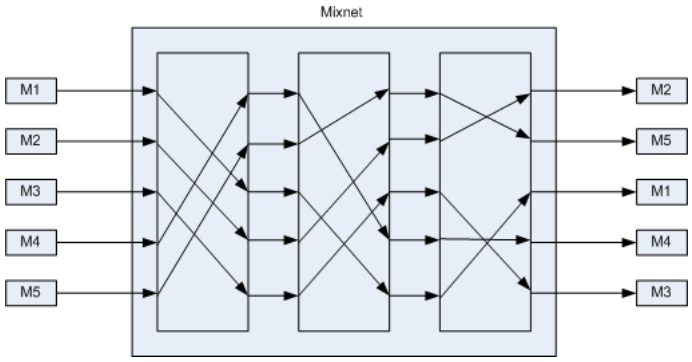

Fig. 1: Example of mixnet with three nodes. There are five messages sent through the mixnet. Each node permutes the messages and forwards to the next node. At the end the receiver cannot relate each message to the corresponding sender.

2 Similar work

There have been plenty of research in the field of practical anonymous channels. One of the most active and result giving projects is Onion Routing Program governed by US Navy [7]. This program is analyzing anonymous communication systems and is aiming to create a practical system for internet-based anonymous activities. Their third generation system is Tor, which is made for practical public user [8]. Tor Project is an implementation of anonymous channels for the masses. That means that everybody can install a Tor client and actively participate in the Tor network. Every packet sent is going through random route in the Tor network to the recipient. In the end, the recipient does not have information about the sender's ip address.

There are many publications and proposed protocols for mixnets. Some of them are decryption, some are reencryption. Some offer some sort of verifiability for the mixnet's operations, and some don't. There is a project Verificatum [9] meant to create practical and provably secure mixnet. This is ongoing project

that has a motivation to create complete mixnet, including distributed key generation, proofs for verification and most importantly practically feasible.

Our research will be mainly for comparison of decryption and reencryption mixnets. We will implement known mixnet algorithms and test them for reaction on several parameters: number of messages, keylength of underlying cryptosystem, number of nodes and number of threshold nodes if the mixnet supports threshold scenario.

3 Decryption Mixnets

The first notion of mixnet comes from David Chaum in 1981 [2]. These mixnets are in his name called Chaumian, or by the type of inner workings - decryption mixnets. In this type of mixnet the nodes have a pair of private and public key. There is some kind of PKI involved in the process of key distribution and usage.

The $E_{pk_j}(r, m)$ is the encryption protocol for jth node with public key pk_j on message m and random padding r. The $D_{sk_j}(m_{enc})$ is the decryption protocol for jth node with according private key sk_j od encrypted message m_{enc}.

Sender which wants to send a message m in a mixnet with five nodes would have to prepare a message:

$$m_{enc} = E_{pk1}(r_1, E_{pk2}(r_2, E_{pk3}(r_3, E_{pk4}(r_4, E_{pk5}(r_5, m)))))$$

This is calles onion encryption because we encrypt the message in layers, opposite of the direction of decryption. Message prepared like this have to pass the designated nodes in the correct order. The jth node will decode the received message $m_{enc_{j-1}}$ from $j-1$th node:

$$m_{enc_j} = D_{sk_j}(m_{enc_{j-1}})$$

The final (in our case the fifth) node will decrypt the received ciphertext and get the original message m:

$$m = D_{sk_5}(m_{enc_4})$$

In this algorithm there is given a possibility for individual verifiability. Let's make up a scenario where a sender sends a message to a public bulletin board. Now the sender can use the decryption mixnet to send their public key as a first message. After verification that the correct message is on the bulletin board after the mixing process the sender can send the intended message through the mixnet.

The biggest pitfall of Chaumian Mixnets is the robustness of the network. The prepared message have to be decrypted and passed on in the correct order. If only one node is inactive in the phase of mixing, the message will not be successfuly

received by the receiver. That is the reason that in practice the decruption mixnets are not widely accepted and user. The re-encryption mixnets are more suitable for internet-based networks where robustness is a wanted property.

The results of testing the decryption mixnet are given in figure 2. The figure shows that the decryption mixnet processes one thousand messages in 96 seconds using 512 bit keys, and 733 seconds using 1024 bit keys. The difference is huge because the underlying cryptosystem is RSA, where the encryption and decryption operations are expensive exponentiations. Therefore the bitlength of keys is dictating the speed of decryption mixnet.

4 Re-encryption Mixnets

The Re-encryption mixnets are network of nodes which also provide anonymization of passed messages. But instead of decrypting a previously onion-encrypted message, each node re-encrypts the message with fresh randomization. Then permutates the received messages and proceeds them to the next node. Therefore, as previously noted, if only one node keeps the permutation secret, the messages passed will be anonymous.

There are many reencryption mixnets which employ different cryptosystems. In this paper we will test three implementations of reencryption mixnets with the ElGamal [3] cryptosystem. The three implementation are given as subsections to this section.

4.1 Threshold Re-encryption Mixnet with Designated Dealer

This algorithm is described in [1]. It uses Shamir's threshold decryption [6] and Pedersen [5]. The underlying cryptosystem is ElGamal with threshold properties using Shamir secret sharing.

In the initialization phase a designated dealer creates ElGamal private key. Then this key is divided by using the polynomial $f(x) = \sum_{i=0}^{k} a_i x^i$. Here $a_0 = x$ represents the ElGamal private key that we want to share across the nodes, and the rest a_i are random numbers from the ElGamal group. The order of polynom k is threshold value - meaning that k nodes need to cooperate in the decryption process to recreate the private key. Using $f(x)$ we distrubute the secret share to n nodes by giving them $x_i = f(i), i = 1, ..n$.

The sender prepares their message as ElGamal ciphertext (u, v) using the published public key. This message is then passed to entering node of the mixnet which collects messages and permutates their order. Each j-th node then re-encrypts the message calculating:

$$z_j = u^{x_j}$$

When at least s, $s \geq k$ nodes have re-encrypted a message it is possible to decrypt using:

$$m = \frac{v}{\prod_{i=0}^{s} z_i^{l_i}}$$

$$l_i = \prod_{i'=0, i' \neq i}^{s} \frac{i'}{i' - i}$$

Testing the practical implementation of this algorithm gave interesting results. The figure 3 is showing the different timings in using 512 bit and 1024 bit keylengths with messages from one to one thousand. As in decryption mixnets, we also see big difference in timings comparing keylengths.

Figure 4 shows the tests about choosing the k threshold value. The figure shows runs with 5, 10 and 15 as value for k, and the total number of nodes n is on the x-axis moving from 15 to 30. The results show that changing k doesn't significantly change timings. Therefore, choosing the threshold value in some mixnet will not affect the time in which the mixnet will operate. We see the same results in figure 5 in which we change the k value on x-axis for fix values of n. Figure 6 gives the dependancy of choosing keylengths 512 bit and 1024 bit for k value on x-axis. The conclusion is that even with larger keylengths the k value doesn't significantly change timings.

The timings for choosing different total number of nodes, in comparison with keylengths is given in figure 7. Here we see that timings drastically differ with larger keylength.

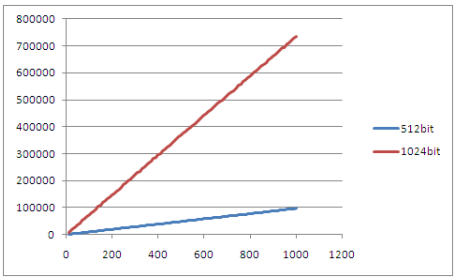

Fig. 2: Decryption mixnet showing time depending on number of messages using 512 bit and 1024 bit keylengths.

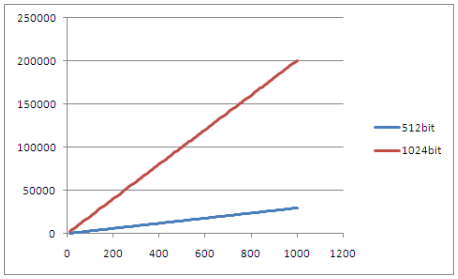

Fig. 3: Re-encryption mixnet with Dedicater Dealer showing time depending on number of messages using 512 bit and 1024 bit keylengths.

4.2 Threshold Re-encryption Mixnet with Designated Dealer, variant 2

This algorithm is also using designated dealer to divide the private key to the nodes of the mixnet. The initialization is the same as with the previous algorithm,

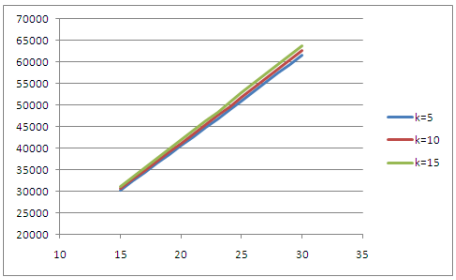

Fig. 4: Re-encryption mixnet with Dedicater Dealer showing time dependencies of fix k-threshold nodes.

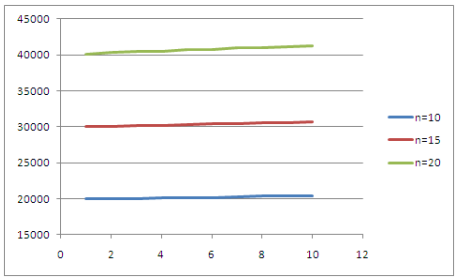

Fig. 5: Re-encryption mixnet with Dedicater Dealer showing time dependencies of total number of nodes.

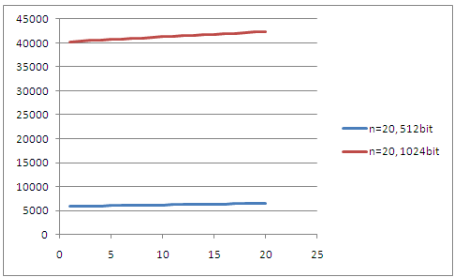

Fig. 6: Re-encryption mixnet with Dedicated Dealer showing time depending on k-threshold values against keylengths.

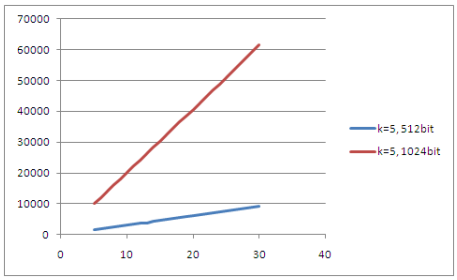

Fig. 7: Re-encryption mixnet with Dedicated Dealer showing time depending on k-threshold value of 5 with 512 bit and 1024 bit keylengths.

and the sender prepares their message as ElGamal ciphertext (u, v). Every j-th node re-encrypts the message with the following calculations:

$$u_j = u_{j-1} g^{r_j}$$

$$v_i = v_{j-1} h^{r_j}$$

At last, when at least s, $s \geq k$ nodes have re-encrypted a message it is possible to decrypt using:

$$m = v_s u_s^x$$

The results of testing this mixnet are given in figure 8. Here as in the previous mixnet we see big difference in timings between 512 bit and 1024 bit keylenghts. The other testing with k threshold value and n total number of nodes are coherent with the first variant of Re-encryption mixnet with Dedicated Dealer.

4.3 Threshold Re-encryption Mixnet without Designated Dealer

The third re-encryption algorithm does not require Designated Dealer. It uses Gennaro's ElGamal secure distributed key generation scheme [4]. This scheme is giving opportunity for the network to create public ElGamal key, but the private part is not known until decryption time.

In the initialization phase every j-th node is creating polynom $f_j(x) = \sum_{i=0}^{k} a_{ji}x^i$. The value $z_i = f_i(0)$ is the secret sharing part in the key generation process. Then every i-th node calculates $s_{ij} = f_i(j)$ and sends them to every other j-th node. Let S be the subset of at least k nodes which correctly and successfully contributed in the key generation process. At the end of initialization phase the private key of every node is $x_i = \sum_{j=0}^{s} s_{ji}$ and the public key of every node is $y_i = g^{z_i}$. The public ElGamal key of the mixnet is:

$$Y = \prod_{i=0}^{s} y_i$$

The sender prepares their message as standar ElGamal ciphertext (u, v) using the public key Y of the mixnet. The prepared ciphertext is the sent to the mixnet.

Every node in the re-encryption process calculates $a_i = u^{x_i}$. After at least s, $s \geq k$ nodes calculated the re-encryptions, the ciphertext can be decrypted by calculating:

$$m = \frac{v}{\prod_{j=0}^{s} a_j^{l_j}}$$

$$l_i = \prod_{i'=0, i'!=i}^{s} \frac{i'}{i' - i}$$

The results of the mixnet with different number of messages passed to the mixnet with different keylenghts are given in figure 3. The conclusion of this test and the other test with k-threshold value and n total number of nodes are the same with the other re-encryption mixnets.

5 Conclusion

The comparison of all implemented mixnets is given in figure 10 and 11. Here we see comparison of timings with 512 bit and 1024 bit keylengths. We conclude that the decryption mixnet is very slow in both cases, and that the re-encryption algorithms are very close in timings and choosing between Dedicated Dealer and Non-Dedicated Dealer will not significantly change timings of running the mixnet.

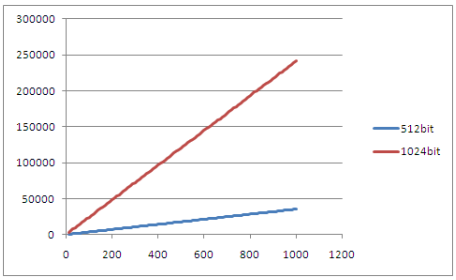

Fig. 8: Re-encryption mixnet with Dedicated Dealer, variant 2 showing time depending on number of messages using 512 bit and 1024 bit keylengths.

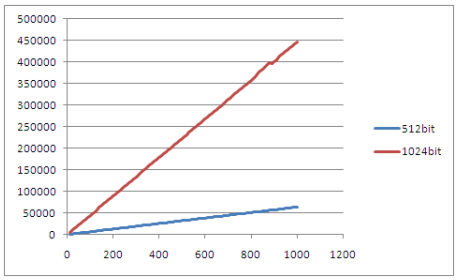

Fig. 9: Re-encryption mixnet without Dedicated Dealer showing time depending on number of messages using 512 bit and 1024 bit keylengths.

References

1. Aditya, R., Boyd, C., Dawson, E., Lee, B., Peng, K.: Batch verification for equality of discrete logarithms and threshold decryptions. In: Jakobsson, M., Yung, M., Zhou, J. (eds.) Applied Cryptography and Network Security Second International Conference on Applied Cryptography and Network Security (LNCS 3089). pp. 494–508. Springer-Verlag, Yellow Mountain, China (2004), http://eprints.qut.edu.au/23930/
2. Chaum, D., Acm, C.O.T., Rivest, R., Chaum, D.L.: Untraceable electronic mail, return addresses, and digital pseudonyms. Communications of the ACM 24, 84–88 (1981)
3. El Gamal, T.: A public key cryptosystem and a signature scheme based on discrete logarithms. In: Proceedings of CRYPTO 84 on Advances in cryptology. pp. 10–18. Springer-Verlag New York, Inc., New York, NY, USA (1985), http://portal.acm.org/citation.cfm?id=19478.19480
4. Gennaro, R., Jarecki, S., Krawczyk, H., Rabin, T.: Secure distributed key generation for discrete-log based cryptosystems. J. Cryptol. 20, 51–83 (January 2007), http://portal.acm.org/citation.cfm?id=1229121.1229123
5. Pedersen, T.P.: Non-interactive and information-theoretic secure verifiable secret sharing. In: Proceedings of the 11th Annual International Cryptology Conference on Advances in Cryptology. pp. 129–140. CRYPTO '91, Springer-Verlag, London, UK (1992), http://portal.acm.org/citation.cfm?id=646756.705507
6. Shamir, A.: How to share a secret. Commun. ACM 22, 612–613 (November 1979), http://doi.acm.org/10.1145/359168.359176
7. US-Navy: Onion routing program (Jun 2011), http://www.onion-router.net/
8. US-Navy: Tor project (Jun 2011), https://www.torproject.org/
9. Wikstrom, D.: Verificatum - provably secure mixnet (Jun 2011), http://www.verificatum.com/verificatum/index.html

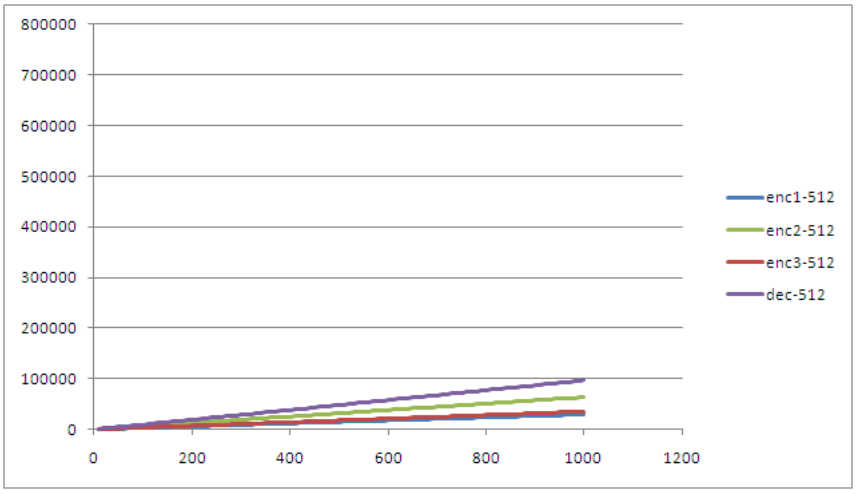

Fig. 10: Showing comparison of mixnets with 512 bit keylenghts.

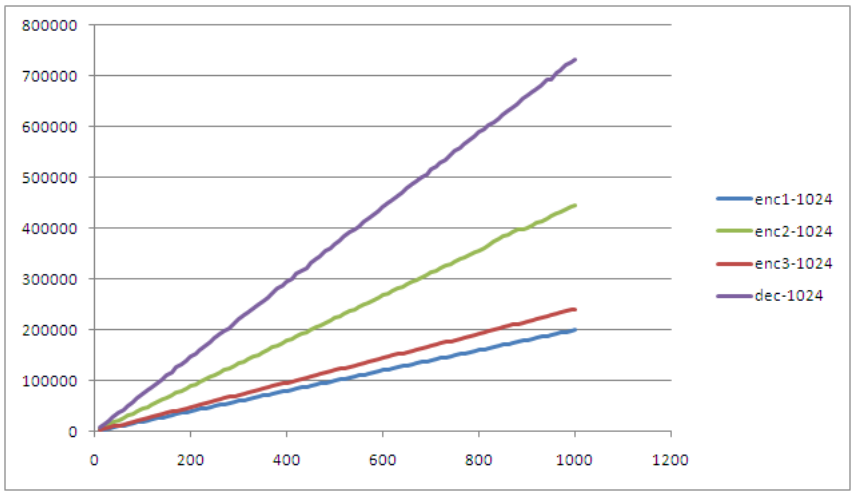

Fig. 11: Showing comparison of mixnets with 1024 bit keylenghts.