Journal of Computing and Information Technology - CIT 20, 2012, 3, 225–231
doi:10.2498/cit.1002101

Mixnets: Implementation and Performance Evaluation of Decryption and Re-encryption Types

Pance Ribarski and Ljupcho Antovski

Faculty of Computer Science and Engineering, University Ss. Cyril and Methodius, Skopje, Macedonia

The anonymous channels have been the essence of numerous protocols that include anonymous message passing between peers. The mixnet structure is one way to accomplish the anonymity . Since the publication of the Chaumian mixnet, there have been many practical implementations. There are two main approaches to implement mixnets: the decryption (Chaumian) and the re-encryption mixnets. In this paper we analyze four types of mixnets, of which one decryption and three re-encryption types. They were implemented in the Java programing language and evaluated on several criterias as: the number of messages, the total number of nodes, the number of threshold nodes, and the key length of underlying crypto system. In the results section we compare the results from the practical tests to answer the research question, which type of mixnets has better features.

Keywords: mixnets, decryption, re-encryption, e-voting, algorithms, performance, ElGamal, threshold, dealer

1. Introduction

The anonymous message passing through computer networks is essential to many security based communication 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 anonymity by adding sender and receiver IP addresses in every packet transmitted through the network. Knowing this, everyone could look at the packets and get the knowledge of who is sending the message and to whom it is intended. Further more, the receiver always knows who sent the message and can relate the message back to the sender.

The term "anonymous channels" was named by practicioners to underline that the sent message cannot be related back to the sender. Because of the mentioned vulnerability of computer networks another approach needs to be taken in order to achieve the anonymity of passing messages. One of the common implementation of anonymous channel are mixnets. Mixnet is a set of nodes that are included in the anonymous channel implementation Figure 1. When a sender wants to pass a message, he/she sends the message to one node of this set. All received messages in the node are permuted and passed to other node. Finally the message exits through the last node and gets to the receiver party. If at least one node is "honest" and functions correctly, meaning it does not expose the permutation, the anonymity of passed messages is guaranteed.

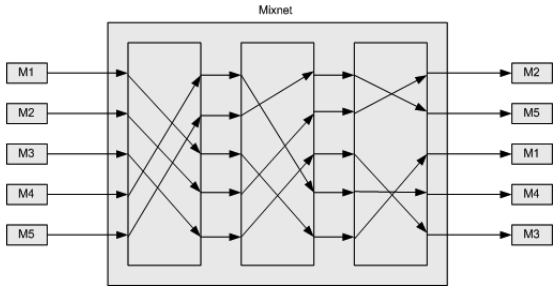

Figure 1. Mixnet with three nodes. There are five messages sent through the mixnet. Each node permutes the messages and forwards them to the next node. At the end the receiver cannot relate the messages to the corresponding sender.

In the following sections we will present the implementation of the two basic categories of

mixnets – decryption and re-encryption type. The results section presents the test results conducted for the four implementations. The tests were conducted in order to answer the research questions: Which mixnet implementation is faster, does the performance depend on the number of messages, does the key length of underlying crypto system affect speed, how does the number of nodes included in the mixnet change the performance time, and does the threshold value have impact on performance.

2. Related Projects

There are several research projects in the field of practical anonymous channels. One of the most recent active with promising results is the Onion Routing Program governed by US Navy [10]. This program analyses the anonymous communication systems and aims to create a practical system for internet-based anonymous activities. The third generation of the system is the Tor Project, available for practical public use [11]. The Tor Project is an implementation of anonymous channels for the massive use. Everyone can install a Tor client and actively participate in the Tor network. Every sent packet is going through a 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 numerous publications and proposed protocols for mixnets. Some of them are decryption, some are re-encryption. Several offer sort of verifiability for the mixnet's operations. The project Verificatum [12] intends to create practical and probably secure mixnet. This is an ongoing project with the aim to create a complete mixnet, including distributed key generation, proofs for verification and, most importantly, the mixnet to be practically feasible. There are other projects based on re-encryption mixnets like in [6].

Our research is mainly focused on the comparison of the decryption and the re-encryption mixnets. We have implemented already known mixnet algorithms and have tested them for performance evaluation on several parameters: number of messages, key length of underlying crypto system, number of nodes and number of threshold nodes if the mixnet supports threshold scenario.

3. Decryption Mixnets

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

The $E_{pk_j}(r,m)$ is the encryption protocol for the 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 the jth node with approrpiate private key sk_j of encrypted message m_{enc}.

A sender that wants to send a message m in a mixnet with five nodes would have to prepare a message:

$$m_{enc} = E_{pk_1}(r_1, E_{pk_2}(r_2, E_{pk_3}(r_3, E_{pk_4}(r_4, E_{pk_5}(r_5, m))))) \quad (1)$$

This is called onion encryption because one encrypts the message in layers, opposite of the process of decryption. The message prepared with this protocol needs to pass the designated nodes in the correct order. The jth node will decode the received message $m_{enc_{j-1}}$ from the $(j-1)$th node:

$$m_{enc_j} = D_{sk_j}(m_{enc_{j-1}}) \quad (2)$$

The final (in this case the fifth) node will decrypt the received cipher text and get the original message m:

$$m = D_{sk_5}(m_{enc_4}) \quad (3)$$

In this algorithm there is a possibility for individual verification in a scenario where a sender sends a message to a public bulletin board. Now the sender can use the decryption mixnet to send the public key as a first message. After verification that a correct message is on the bulletin board following the mixing process, the sender can send the intended message through the mixnet.

The biggest pitfall of the Chaumian mixnets is the robustness of the network. The prepared message has 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 successfully received by the receiver. That is the reason why in practice the decryption mixnets are not widely accepted and used. The re-encryption mixnets are more suitable for internet-based networks where robustness is a wanted property.

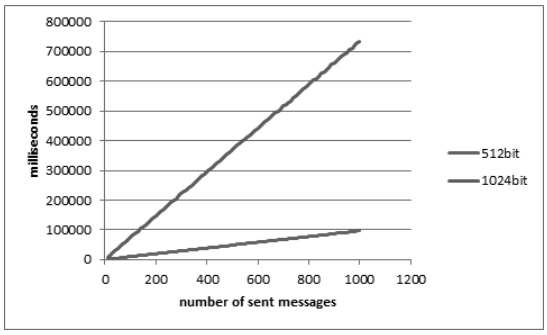

Figure 2. Decryption mixnet showing time depending on number of messages using 512 bit and 1024 bit key lengths.

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 in 733 seconds using 1024 bit keys. The difference is huge because the underlying crypto system is RSA, where the encryption and decryption operations are expensive exponentiations. Therefore the length of the keys is pacing the speed of decryption mixnet.

4. Re-encryption Mixnets

The re-encryption mixnets are networks 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 the node permutes the received messages and proceeds them to the next node. Therefore, as previously noted, if only one node is honest and keeps the permutation secret, the passed messages will be anonymous.

There are many re-encryption mixnets implementations that deploy different crypto systems. In this paper we present the results from the tests of three implementations of re-encryption mixnets with the ElGamal [3] crypto system. The three implementations are given as subsections to this section.

4.1. Threshold Re-encryption Mixnet with Designated Dealer

This algorithm is described in details in [1]. It uses Shamir's threshold decryption [9] and Pedersen's secret sharing protocol [7]. The underlying cryptosystem is ElGamal with threshold properties using Shamir secret sharing.

In the initialization phase a designated dealer creates an ElGamal private key. Then this key is divided among the nodes using the polynomial $f(x) = \sum_{i=0}^{k} a_i x^i$. Here a_0 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 polynomial k is the 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 $x_i = f(i), i = 1, \ldots, n$.

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

$$z_j = u^{x_j} \tag{4}$$

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

$$m = \frac{v}{\prod_{i=0}^{s} z_i^{l_i}} \tag{5}$$

$$l_i = \prod_{i'=1, i' \neq i}^{s} \frac{i'}{i' - i} \tag{6}$$

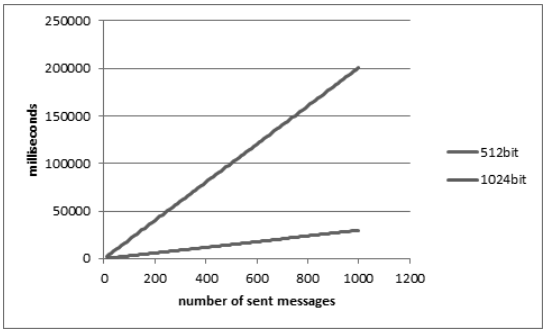

Figure 3. Re-encryption mixnet with dedicated dealer showing time depending on number of messages using 512 bit and 1024 bit key lengths.

The results of the implementation of this algorithm gave interesting results. Figure 3 presents the different timings when using 512 bit and 1024 bit key lengths with number of message from 1 to 1000. As in the case decryption mixnets, we also see big difference in timings comparing key lengths.

Figure 4 shows the tests for the choice of the k threshold value. The figure shows tests with 5, 10 and 15 as values for k, and the total number of nodes n is on the x-axis in the range from 15 to 30. The results show that changing the threshold value k doesn't significantly change the performance. Therefore, the choice of the threshold value in some mixnets will not affect the performance of the mixnet.

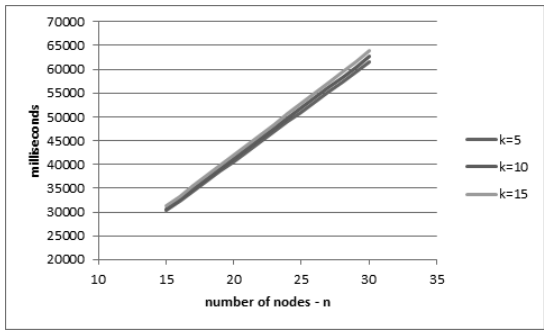

Figure 4. Re-encryption mixnet with dedicated dealer showing time dependencies of fixed k-threshold nodes.

We observe the same results in Figure 5 in which we change the threshold value k value on x-axis for fixed values of total number of nodes n. Figure 6 presents the dependency of the key lengths 512 bit and 1024 bit for threshold value k value on x-axis. The conclusion is that even with larger key lengths, the threshold value k value does not significantly change the performance of the algorithm.

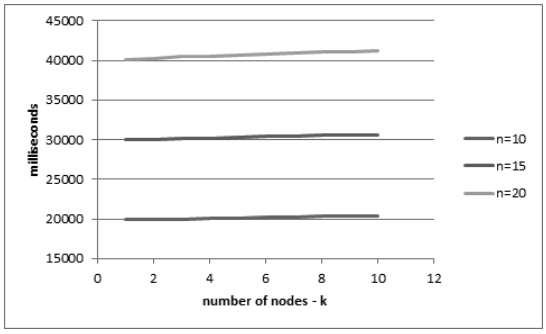

Figure 5. Re-encryption mixnet with dedicated dealer showing time dependencies of total number of nodes.

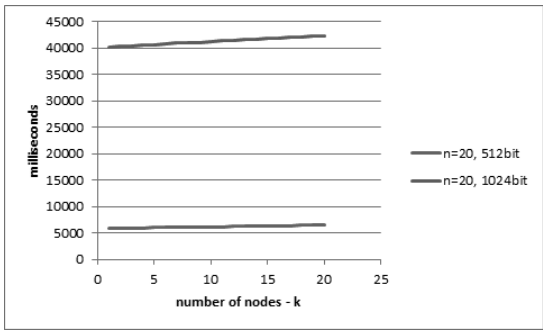

Figure 6. Re-encryption mixnet with dedicated dealer showing time depending on the k-threshold values against key lengths.

The timings for selecting different total number of nodes n, in comparison with key lengths is given in Figure 7. Here we see that timings drastically differ with larger key length.

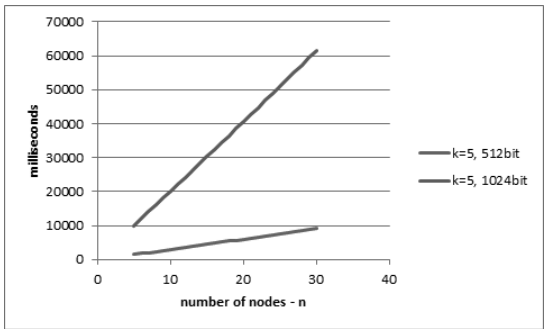

Figure 7. Re-encryption mixnet with Dedicated Dealer showing time depending on k-threshold value of 5 with 512 bit and 1024 bit key lengths.

4.2. Modified Threshold Re-encryption Mixnet with Designated Dealer

The modified 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 basic algorithm, and the sender prepares the message as ElGamal cipher text (u, v). Every j-th node re-encrypts the message with the following algorithm:

$$u_j = u_{j-1} g^{r_j} \tag{7}$$

$$v_j = v_{j-1} h^{r_j} \tag{8}$$

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

$$m = \frac{v_s}{u_s^x} \tag{9}$$

The test results of this mixnet are presented in Figure 8. We see that, as in the previous cases, there is a big difference in performance between 512 bit and 1024 bit key lengths. The test results with k threshold value and n total number of nodes are similar to the first version of Re-encryption mixnet with dedicated dealer.

4.3. Threshold Re-encryption Mixnet without Designated Dealer

The third re-encryption algorithm implemented does not require designated dealer. It uses Gennaro's ElGamal secure distributed key generation scheme [5]. This scheme offers the opportunity for the network to create the public ElGamal key, where the private part is not known until decryption time.

In the initialization phase every j-th node creates a polynomial $f_j(x) = \sum_{i=0}^{k} a_{ji}x^i$. The value $z_j = f_j(0)$ is the secret shared 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 that 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 $y_i = g^{zi}$. The public ElGamal key of the mixnet is:

$$Y = \prod_{i=0}^{s} y_i \tag{10}$$

The sender prepares the message as standard ElGamal cipher text (u, v) using the public key Y of the mixnet. The prepared cipher text is then 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-encryption, the cipher text can be decrypted by calculating:

$$m = \frac{v}{\prod_{j=0}^{s} a_j^{l_j}} \tag{11}$$

$$l_i = \prod_{i'=1, i' != i}^{s} \frac{i'}{i' - i} \tag{12}$$

The test results of the mixnet with different number of messages passed to the mixnet with different key lengths are presented in Figure 3. The conclusion is that in all performance tests with k-threshold value and n total number of nodes, the performance is the same in all re-encryption mixnets implementations.

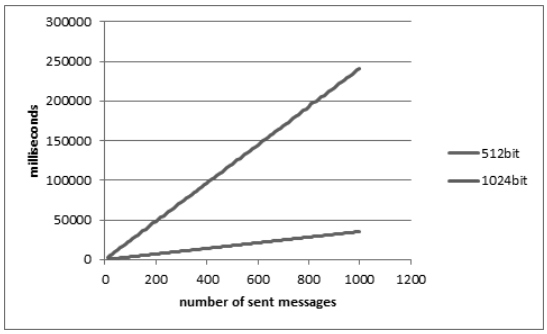

Figure 8. Modified re-encryption mixnet with dedicated dealer, presenting the dependence of time on the number of messages using 512 bit and 1024 bit key lengths.

5. Conclusion and Future Work

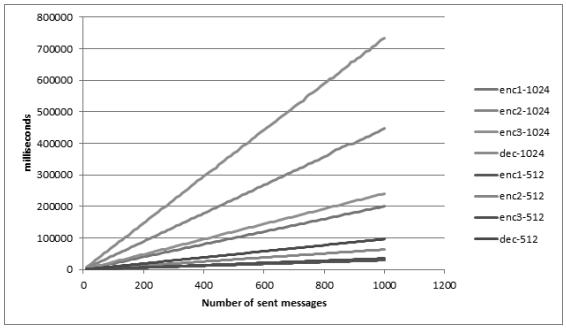

Figure 9. Showing comparison of mixnets with 512 bit and 1024 bit key lenghts.

The comparison of all the implemented algorithms for mixnets is given in Figure 9. The results obviously lead to conclusion that the decryption mixnets are very slow in both cases. The re-encryption algorithms are faster and in different implementations the performances are very very close, no regard if it is a dedicated

dealer or non-dedicated dealer. The modified algorithm will not significantly change performance of the running mixnet. The non-dedicated dealer algorithm is slightly slower than the other re-encryption algorithms, but the lack of dedicated dealer is a very wanted performance trade-off property in many scenarios. These scenarios cover key generation in environment where one does not trust any of the other involved parties concerning the private key.

This paper presents the practical implementation results of the proposed mixnet algorithms. Selecting a right type of mixnet will depend on several factors; in this paper we underline the performance when selecting appropriate values for the parameters such as: crypto system key length, total number of nodes and threshold value, and helps with the strategic choice between popular mixnet algorithms.

There are several open question regarding mixnets implementations. In the future work we will focus on the features of the mixnets with elliptic curves, on the possibility of the elliptic curves to improve the performance of mixnets [4]. We will investigate the approaches to clue the open doors for the various attacks on mixnets [8]. The results of the research will determine the choice of mixnet type that will be used.

References

[1] R. ADITYA, C. BOYD, E. DAWSON, B. LEE, K. PENG, Batch verification for equality of discrete logarithms and threshold decryptions. In *Applied Cryptography and Network Security Second International Conference on Applied Cryptography and Network Security (LNCS3089)* (M. JAKOBSSON, M. YUNG, J. ZHOU, EDITORS), (2004) pp. 494–508, Yellow Mountain, China. Springer-Verlag.
http://eprints.qut.edu.au/23930/

[2] D. L. CHAUM, Untraceable electronic mail, return addresses, and digital pseudonyms. *Commun. ACM*, 24(2) (February 1981), 84–90. ISSN0001-0782. doi: 10.1145/358549.358563.
http://doi.acm.org/10.1145/358549.358563

[3] T. EL GAMAL, A public key cryptosystem and a signature scheme based on discrete logarithms. In *Proceedings of CRYPTO 84 on Advances in cryptology*, (1985) pp. 10–18, NewYork, NY, USA. Springer-Verlag, NewYork, Inc. ISBN 0-387-15658-5.
http://portal.acm.org/citation.cfm?id=19478.19480

[4] J. FURUKAWA, K. SAKO, An efficient publicly verifiable mix-net for long inputs. In *Financial Cryptography and Data Security* (GIOVANNI DI CRESCENZO, AVI RUBIN, EDITORS), volume 4107 of Lecture Notes in Computer Science, (2006) pp. 111–125. Springer Berlin/Heidelberg. ISBN 978-3-540-46255-2.

[5] R. GENNARO, S. JARECKI, H. KRAWCZYK, T. RABIN, Secure distributed key generation for discrete-log based cryptosystems. *J. Cryptol.*, 20 (January 2007), 51–83. ISSN 0933-2790. doi: 10.1007/s00145-006-0347-3.
http://portal.acm.org/citation.cfm?id=1229121.1229123

[6] A. HUSZTI AND A. PETHŐ, A secure electronic exam system. *Publicationes Mathematicae*, 77(3-4) (2010), 299–312. ISSN 0033-3883.

[7] T. P. PEDERSEN, Non-interactive and information-theoretic secure verifiable secret sharing. In *Proceedings of the 11th Annual International Cryptology Conference on Advances in Cryptology, CRYPTO '91*, (1992), pp. 129–140, London, UK. Springer-Verlag. ISBN 3-540-55188-3.
http://portal.acm.org/citation.cfm?id=646756.705507

[8] A. SERJANTOV, R. DINGLEDINE, P. SYVERSON, From a trickle to a flood: Active attacks on several mix types. In *Proceedings of Information Hiding Workshop (IH 2002)* (F. PETITCOLAS, EDITOR, (2002) pp. 36–52. Springer-Verlag, LNCS 2578.

[9] A. SHAMIR, How to share a secret. *Commun. ACM*, 22 (November 1979), 612–613. ISSN 0001-0782.
http://doi.acm.org/10.1145/359168.359176

[10] US-NAVY, Onion routing program, June 2011.
http://www.onion-router.net/

[11] US-NAVY, Tor project, June 2011.
https://www.torproject.org/

[12] D. WIKSTROM, Verificatum – provably secure mixnet, June 2011.
http://www.verificatum.com/verificatum/index.html

Received: June, 2012
Accepted: August, 2012

Contact addresses:
Pance Ribarski
Faculty of Computer Science and Engineering
University Ss. Cyril and Methodius
st. Rugjer Boshkovikj 16
Skopje
Macedonia
e-mail: pance.ribarski@finki.ukim.mk

Ljupcho Antovski
Faculty of Computer Science and Engineering
University Ss. Cyril and Methodius
st. Rugjer Boshkovikj 16
Skopje
Macedonia
e-mail: ljupcho.antovski@finki.ukim.mk

PANCE RIBARSKI has been employed since 2009 as a teaching assistant at the Faculty of Computer Science and Engineering in Skopje. He lectures: data structures, network operating systems, security and cryptography, wireless and mobile technologies, software project management, software design and architecture, operating systems and software requirements. The research field of Pance Ribarski is the application of security and cryptography in software engineering, with special interest in electronic voting. He also has interest in mobile technologies and ubiquitous technologies in everyday life. Pance Ribarski is co-author of several papers in the field of cryptography, e-security and electronic voting systems. He completed the post-graduate courses and defended his master thesis in the field of electronic voting. Currently, he is enrolled as a PhD student. He is a member of Computer Society of Macedonia and is Secretary of the Association for Information and Communication Technologies ICT-ACT.

LJUPCHO ANTOVSKI is an associate professor of software engineering at the Faculty of Computer Science and Engineering, St. Cyril and Methodius University, Skopje, Macedonia. He lectures courses in requirements engineering, software project management, e-business, and mobile applications. He is the president of the IT technical committee at the Macedonian Institute for Standardization. Ljupcho is cofounder and board member of iVote, leading integrator of elections management information systems in South-Eastern Europe. He has vast experience in implementing election information systems worldwide. His academic research is focused on the requirements and usability engineering when implementing election information systems and the broad applicability of mobile voting. He has published widely in the fields of m-government and m-voting. He has authored books and numerous papers in leading journals and conference proceedings.