Message Security on Chat App based on Massey Omura Algorithm

Taronisokhi Zebua[1], Rivalri Kristianto Hondro[2], Efeleroni Ndruru[3]
[1]AMIK STIEKOM Sumatera Utara
[2,3]STMIK Budi Darma Medan
[1]taronizeb@gmail.com, [2]rivalrihondro@gmail.com, [3]ronindruru@gmail.com

Abstract

Security of message on chat apps is very important to do so that messages that distributed always safety for others who do not have access permission. However, not all chat apps currently have tools that used to secure messages. This is still often overlooked, thus providing an easy space for the attackers to hack messages that are distributed. This research explains the usage of the massey-omura algorithm to secure text type message in chat apps when message distributed.

Keywords: *Cryptography, Massey Omura, Three-pass protocol, Message, Chat*

1. Introduction

Chat app is one of the most frequently used apps of today's users to distribute messages. Messages that are distributed can be either secret messages or not. Not all chat apps currently have message security tools, so attacks and misuse of messages by others who do not have access permissions occur frequently. Utilization of message security techniques in a chat app is essential to ensure access rights and message originality[1]. Cryptography technique is one of the common technique used in securing data or messages by encoding the original message into other characters that can hide the pattern and meaning of the original message [2][3].

Cryptography technique algorithm based on the key consists of two are symmetric and asymmetric cryptography. Massey-omura algorithm is one of the asymmetric cryptography algorithms, where the key used in the encryption process is different with the key used in the decryption process. Massey-omura algorithm works based on the concept of a three-pass protocol in which the key used in the encryption or decryption process is generated by the sender and receiver based on a mutually agreed value. Exponential modulo operations and prime numbers are used by both parties to generate the keys used [4][5]. The concept of three-pass protocols in this algorithm can maintain the security of keys used both in the process of encryption and decryption messages.

This research describes how to apply massey-omura algorithm to chat application. Any messages distributed via the chat app will be automatically encrypted by chat app by utilizing the key values that have set by the sender and receiver who are communicating. Implementation of this algorithm is expected to minimize the misuse of messages that have been distributed through chat applications by parties who are not given access.

2. Rudimentary

2.1. Cryptography

Cryptography techniques are one technique that can be used to secure information that is confidential. The cryptography technique secures an undisclosed message by encoding it into another form that can no longer be understood its originality [3][6]. The aims to be achieved in applying cryptography techniques are confidentiality (the message cannot be understood), integrity (the authenticity of the message), authentication (the originally of the sender or receiver identity) and non-

ISSN : 2580-7250

repudiation (the recipient or the sender cannot deny that he or she has sent or received the message) [7][8]. Based on the key used, the cryptography algorithm is divided into two, namely symmetric cryptography algorithm (using the same key in the encryption and decryption process) and asymmetric cryptography (using different keys both in the encryption process and on the decryption process)[9].

2.2. Asymmetric Key

An asymmetric key is one of the key types in the cryptography algorithm. Asymmetric key type cryptography algorithm works using different keys in both encryption and decryption processes [10]. This type of asymmetric key is commonly known as a public-key. An asymmetric key will generate one public key used for encrypting messages and a secret key which is a public key pair and can only be used to for messages decryption [4][8]. The process of generating the key on an asymmetric key algorithm involves each communicating party, so it can be ensured that the secret key to decryption more secure. Some known asymmetric key (public-key) algorithms such as RSA, El-Gamal, Massey-omura, Elliptic Curve, and others.

2.3. Three-pass Protocol

Think that eliminates the key exchange process between the recipient, and the sender of the message are the basic concepts of the three-pass protocol [11]. The key will be generated solely by the sender and receiver based on a prime number mutually agreed upon by both. Encryption process based on the three-pass concept will be done three times the process of message exchange starting from encryption process by the sender, encryption process by the receiver and re-encryption process by the sender. Both parties (sender and receiver) use their own key in the process of encryption and decryption [5].

2.4. Massey-Omura Algorithm

Massey-omura is one algorithm that works with asymmetric key concepts and developed based on the concept of the three-pass protocol. The three-pass protocol works with the concept that each party (the recipient and sender of the message) uses their own key to perform the process of encrypting and decryption the messages. One of the advantages of the massey-omura algorithm is the difficulty of discrete logarithmic compute that similar to other public key algorithms such as RSA, and others [5][11]. Massey-omura algorithm causes a message encryption process is done three times through different protocol (three-pass protocol). It is intended that the sender and receiver can synchronize the keys they are using an encryption and decryption process.

2.5. Chat App

Today, chat apps are widely used as one of the easiest alternatives to communicating. Users can easily exchange any messages through the chat apps. One of the advantages of chat application is its ability to enable users to exchange messages either in the types of text, video, audio or image [12]. Chat apps generally work with peer-to-peer (client-server) and centralized systems on a server. The chat app gives users the freedom and convenience to identify friends who are online and can choose to whom he/her wants to communicate [13]. The chat app should be able to provide a secure chat situation for clients and servers, therefore chat apps desperately need a good security aspect.

3. Research Methods

3.1. Research Framework

This research was conducted with several steps which include analysis of message security problem on chat app, review of various literatures about data security, implementation of the massey-omura algorithm (key generation, encryption and decryption process) into chat app, and evaluation or testing.

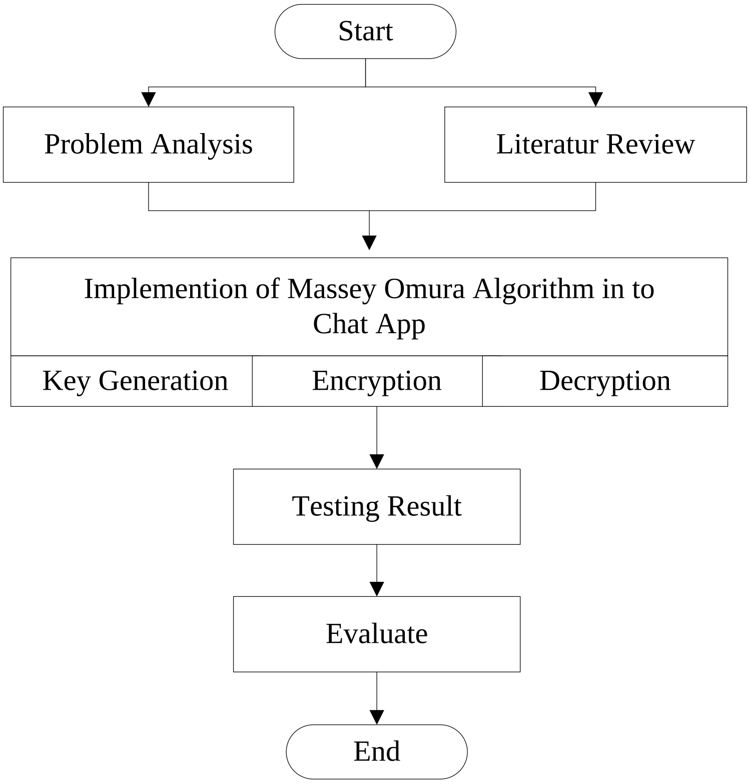


Figure 1. Research Framework

3.2 Encryption and Decryption Steps based on Massey Omura Algorithm

Generally, there are three processes in massey-omura algorithm that is the process of generating public keys and private keys, the process of encryption and decryption. Steps of massey-omura algorithm [4][5], are:

1. The receiver and sender have agreed a larger prime (p) greater value (eg > 256)
2. Sender (the first step)
   a. Determined a value used to perform the first encryption process. Assumed eA where $1 < eA < p - 1$ and eA is co-prime with $p - 1$
   b. Generate the decryption key with looking for the inverse of eA value based on an equation:
   $dA \times eA \pmod{p - 1} = 1$ (1)
   c. Do the first encrypt process to the messages to resulting C1 based on an equation:
   $C1 = M^{eA} \bmod p$ (2)
   d. The result of the cipher (C1) is sent to the recipient.
3. Receiver (the second step)
   a. Determined a value used to perform the second encryption process. Assumed eB where $1 < eB < p - 1$ and eA is co-prime with $p - 1$
   b. Generate the decryption key with looking for the inverse of dB value based on an equation:
   $dB \times eB \pmod{p - 1} = 1$ (3)
   c. Do the second encryption process to C1 for resulting C2 based on an equation:
   $C2 = C1^{eB} \bmod p$ (4)
   d. C2 be resent to the sender.
4. Sender (the third step)
   a. Accept the C2 from a recipient
   b. Encrypt C2 (the third encryption step) and set as C3, based on an equation:
   $C3 = C2^{dA} \bmod p$ (5)

c. Resend C3 (final cipher) to a recipient

5. Receiver (the fourth step)
   a. Accept C3 (as a message that has encrypted) from a sender
   b. Decryption C3 (message from sender) based on an equation:
   $$M = C3^{dB} \bmod p \qquad (6)$$
   c. The result of C3 decryption process is an original message from a sender.

4. Result and Discussion

4.1. Scheme of Massey Omura Algorithm on Chat App

Today, chat apps are still one of the most commonly used apps to communicate, but not all the chat apps have message security tools to protect the message against the action of others who do have not permission. Utilization of cryptography algorithms is one solution that can be used to minimize the misuse of messages distributed through the chat app. Utilization of cryptography algorithms is one solution that can be used to minimize the misuse of messages distributed through the chat app. The security of chat apps, of course, gives comfort to the users to distribute the message.

Procedure to implement massey omura algorithm in securing messages on chat application is done by adding tools of encryption and decryption process in chat apps. This additional tools will be enabled when the user performs the process of sending messages and receiving messages.

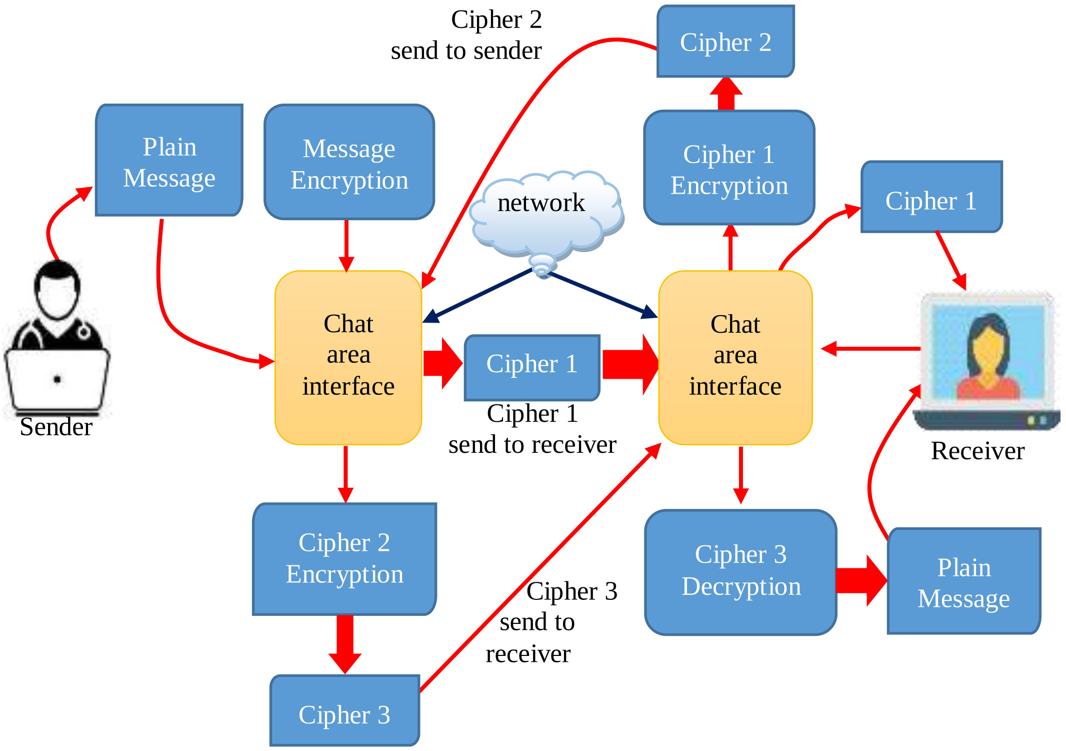


Figure 2. Scheme of Massey-Omura on Chat App

Based on figure 2 above, described that sender doing the encryption of the original message for generating the first cipher (C1) then sent to the receiver. C1 will be re-encrypt by the receiver and generate the second cipher of the message (C2). A receiver will resend the C2 to the sender. A sender will re-encrypt the C2 that has received, so that produced C3 (cipher of the original message) and the C3 will re-resend to the receiver as a final cipher of the message. In order for the cipher of the message be an original message, then receiver doing decryption process by the recipient.

4.2. Process of Generate Keys

Assume that the sender and receiver are online and use chat app. Both of them have approved a prime number greater than 256, for example, p = 263.

1. Sender chooses eA, where 1 < eA < p – 1 and eA co-prime with p – 1
   a. Assume eA = 79, because 79 is co-prime with 262. eA value is used in encryption process
   b. Sender computed the inverse of eA based on equation (1), and store in dA
   Assume dA = 199, if computed (199 x 79) mod 262 = 1 is qualified. dA value is use in encryption process
2. Receiver computed eB, where 1 < eB < p – 1 and eB co-prime with p – 1
   a. Assume eB = 125, because 125 is co-prime with 262. eB value is use in encryption process
   b. Receiver computed the inverse of eB based on equation (3), and store in dB
   Assumed dB = 109, if computed (109 x 125) mod 262 = 1 is qualified. dB value is use in decryption process

4.3. Process of Message Encryption

Assume a sender sent a message to the receiver using chat app and the original message is “HALLO ZEBUA”

1. Sender converting the character of message to an ASCII value, so :
   H = 72 A = 65 L = 76 L = 76 O = 79 SPACE = 32 Z = 90 E = 69
   B = 66 U = 85 A = 65
2. Encryption every character of the message based on equation (2).
   for H : 72^{79} mod 263 = 108
   for A : 65^{79} mod 263 = 158
   for L : 76^{79} mod 263 = 171
   for L : 76^{79} mod 263 = 171
   for O : 79^{79} mod 263 = 126
   for SPACE : 32^{79} mod 263 = 4
   for Z : 90^{79} mod 263 = 201
   for E : 69^{79} mod 263 = 187
   for B : 66^{79} mod 263 = 233
   for U : 85^{79} mod 263 = 116
   for A : 65^{79} mod 263 = 158
   so, is generated C1 = 108, 158, 171, 171, 126, 4, 201, 187, 233, 116, 158 or if converted to character will be resulting C1 = lž««~┘É»étž
3. Sender sent C1 to the receiver.
4. Receiver encrypt C1 based on an equation (4)
   for l : 108^{125} mod 263 = 153
   for ž : 158^{125} mod 263 = 155
   for « : 171^{125} mod 263 = 261
   for « : 171^{125} mod 263 = 261
   for ~ : 126^{125} mod 263 = 197
   for ┘ : 4^{125} mod 263 = 54
   for É : 201^{125} mod 263 = 87
   for » : 187^{125} mod 263 = 86
   for é : 233^{125} mod 263 = 17
   for t : 116^{125} mod 263 = 250
   for ž : 158^{125} mod 263 = 155
   so, is generate C2 = 153, 155, 261, 261, 197, 54, 87, 86, 17, 250, 155
5. Receiver sent C2 to the sender
6. Sender encrypt C2 based on an equation (5) and resulted in C3
   for 153^{199} mod 263 = 136
   for 155^{199} mod 263 = 191

for 261^{199} mod 263 = 177
for 261^{199} mod 263 = 177
for 197^{199} mod 263 = 189
for 54^{199} mod 263 = 62
for 87^{199} mod 263 = 180
for 86^{199} mod 263 = 144
for 17^{199} mod 263 = 151
for 250^{199} mod 263 = 215
for 155^{199} mod 263 = 191
So, is generated C3 = 136, 191, 177, 177, 189, 62, 180, 144, 151, 215, 191 or if converted to the character will be resulting C3 = ˆ¿±±½>´ —×¿

7. C3 is a cipher of the original message from a sender and sent to the receiver

4.4. Process of Decryption

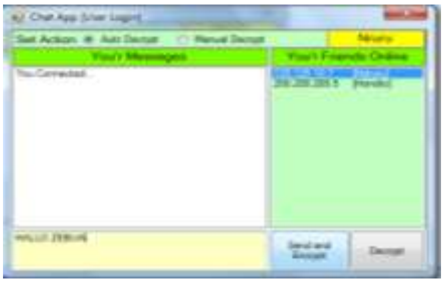
Decryption is done based on equation (6) and receiver generates an original message from the sender. Cipher of the message that has received by the receiver is ˆ¿±±½>´ —×¿ or in ASCII value is 136, 191, 177, 177, 189, 62, 180, 144, 151, 215, 191. So, this cipher will decrypt as follows :

for 136^{109} mod 263 = 72 in char is H
for 191^{109} mod 263 = 65 in char is A
for 177^{109} mod 263 = 76 in char is L
for 177^{109} mod 263 = 76 in char is L
for 182^{109} mod 263 = 79 in char is O
for 62^{109} mod 263 = 32 in char is SPACE
for 180^{109} mod 263 = 90 in char is Z
for 144^{109} mod 263 = 69 in char is E
for 151^{109} mod 263 = 66 in char is B
for 215^{109} mod 263 = 85 in char is U
for 191^{109} mod 263 = 65 in char is A
So, the receiver will get the original message is HALLO ZEBUA

4.5. Implementation

This research using one interface to send and receive the message as well as one interface to generate keys.

1. Chat area for the sender
   There is two option in the chat area is automatic or manual decrypt. When the user selects the automatically then the decryption process is done directly, but if manual option then the process of decryption is only done when the user clicking button of decrypt. Message encryption process is done automatically when user clicking button of send.



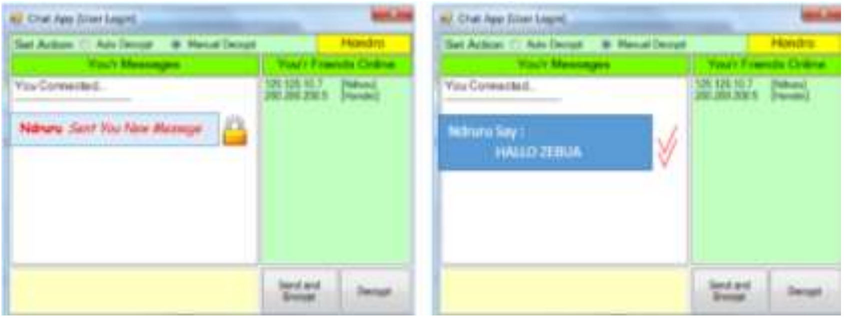
Set Action: ● Auto Decrypt ○ Manual Decrypt

You Connected...

HALLO ZEBUA

Figure 3. Chat Area for the Sender

2. Chat area for the receiver
   When the message is received by the recipient, the message is seen in the chat area still locked, so the recipient can decrypt it by clicking the button of decrypt (if manual decrypt).



You Connected...

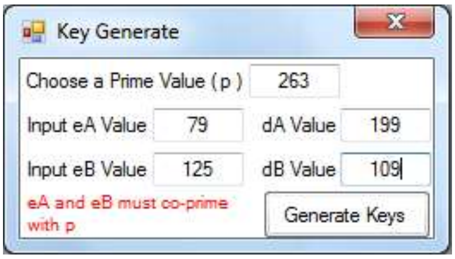
Ndruru Sent You New Message

Ndruru Say :
HALLO ZEBUA

(a) Receive Message (b) After Decryption

Figure 4. Chart Area for the Receiver

3. Interface for generating the key
   Before the sender and receiver distribute their message, they must approve a prime value (p), then input eA and eB values. eA and eB values are found by clicking the button of generate keys.



Choose a Prime Value (p) 263

Input eA Value 79 dA Value 199

Input eB Value 125 dB Value 109

eA and eB must co-prime with p

Figure 4. Area for Generate Keys

5. Conclusion

Based on the results of the analysis and examples of implementation, it is concluded that the chat app that has secured based on massey omura algorithm is very effective to secure the message because the message will be encrypted with layered encryption. However, based on the process time, that this algorithm takes a long time because the process must pass through three protocols. Key generation process that used in the encryption and decryption process is done based on the three-pass protocol concept by each party involved in the distribution of messages.

References

[1] K. Chouhan and S. Ravi, "Public Key Encryption Techniques Provide Extreme Secure Chat Environment," *Int. J. Sci. Eng. Res.*, vol. 4, no. 6, pp. 510–516, 2013.

[2] A. Singh, A. Nandal, and S. Malik, "Implementation of Caesar Cipher with Rail Fence for Enhancing Data Security," *Int. J. Adv. Res. Comput. Sci. Softw. Eng.*, vol. 2, no. 12, pp. 2277–128, 2012.

[3] M. Diana and T. Zebua, "Optimalisasi Beaufort Cipher Menggunakan Pembangkit Kunci RC4 Dalam Penyandian SMS," *J. Sains Komput. Inform.*, vol. 2, no. 1, pp. 12–22, 2018.

[4] R. Sivakumar and K. Thamodaran, "An Inventive Image Security System Based on Massey-Omura Encryption with Group," *Int. J. Adv. Res. Sci. Eng. Technol.*, vol. 3, no. 9, pp. 2656–2667, 2016.

[5] M. Reza, M. A. Budiman, and D. Arisandi, "Simulasi Pengamanan File Teks Menggunakan Algoritma Massey-Omura," *Dunia Teknol. Inf.*, vol. 1, no. 1, pp. 20–27, 2012.

[6] T. Zebua and E. Ndruru, "Pengamanan Citra Digital Berdasarkan Modifikasi Algoritma RC4," *J. Teknol. Infomasi dan Ilmu Komput.*, vol. 4, no. 4, pp. 275–282, 2017.

[7] R. K. Hondro, "Aplikasi Enkripsi dan Dekripsi SMS dengan Algoritma Zig Zag Cipher pada Mobile Phone Berbasis Android," *Pelita Inform. Budi Darma*, vol. 10, no. 3, pp. 122–127, 2015.

[8] R. MM, A. T, and R. A, "Development of Cryptography-Based Secure Messaging System," *J. Telecommun. Syst. Manag.*, vol. 5, no. 3, pp. 1–6, 2016.

[9] E. Setyaningsih, *Kriptografi dan Implementasinya Menggunakan Matlab*, 1st ed. Yogyakarta: CV. Andi Publisher, 2015.

[10] W. Stallings, *Cryptography and Network Security*, V. New York: Prentice Hall, 2011.

[11] A. A. Abdullah; R. Khalaf; M. Riza, *A Realizable Quantum ThreePass Protocol Authentication*. Mathematical Problems in Engineering, 2015.

[12] M. A. Mohamed, A. Muhammed, and M. Man, "A secure chat application based on pure peer-to-peer

architecture," *J. Comput. Sci.*, vol. 11, no. 5, pp. 723–729, 2015.

[13] R. N. Akram and R. K. L. Ko, "End-to-End Secure and Privacy Preserving Mobile Chat Application," in *Information Security Theory and Practice. Securing the Internet of Things*, 2014, pp. 124–139.

Authors

1st Author
Taronisokhi Zebua
Lecturer of AMIK Stiekom Sumatera Utara
taronizeb@gmail.com

2nd Author
Rivari Kristianto Hondro
Lecturer of STMIK Budidarma Medan
rivalryhondro@gmail.com

3rd Author
Eferoni Ndruru
Lecturer of STMIK Budidarma Medan
ronindruru@gmail.com