Live chat alternative security protocol

R Rahman J P*, E Nugraha and A Febriany
Computer Science Study Program, Department of Computer Science Education, Faculty of Mathematics and Natural Sciences Education, Indonesia University of Education.

*Corresponding author’s e-mail: Rizky_rjp@upi.edu

Abstract. Indonesia is one of the largest e-commerce markets in Southeast Asia, as many as 5 million people do transactions in e-commerce, therefore more and more people use live chat service to communicate with customer service. In live chat, the customer service often asks customers’ data such as, full name, address, e-mail, transaction id, which aims to verify the purchase of the product. One of the risks that will happen is sniffing which will lead to the theft of confidential information that will cause huge losses to the customer. The anticipation that will be done is build an alternative security protocol for user interaction in live chat by using a cryptographic algorithm that is useful for protecting confidential messages. Live chat requires confidentiality and data integration with encryption and hash functions. The used algorithm are Rijndael 256 bits, RSA, and SHA256. To increase the complexity, the Rijndael algorithm will be modified in the S-box and ShiftRow sections based on the shannon principle rule, the results show that all pass the Randomness test, but the modification in Shiftrow indicates a better avalanche effect. Therefore the message will be difficult to be stolen or changed.

1. Introduction
Electronic Commerce (e-commerce) is the use of communication and computer networks to carry out business processes. According to IdeA Association e-commerce Indonesia, is one of the largest e-commerce markets in Southeast Asia, with a population of about 248 million people, 39 million people use the internet and 5 million of them make transactions in e-commerce.

Live Chat helps communication between sellers and buyers, which enabling the users to send real time messages via internet [1]. Live chat interests for use on e-commerce sites have grown significantly in recent years. Live chat has many advantages for e-commerce, such as, it provides space for customer to ask customer service in real time to improve customer satisfaction level, save money which previously used to contact customer service by number, time used more efficient, and its ability to deliver human work in e-commerce [2].

Some of the e-commerce who have applied Live Chat are Lazada.co.id, blibli.com, and Blanja.com. Usually in live chat the customer service or the seller will ask the customer’s name, email address, and transaction id in order to verify the product purchased by the customer. Those are customer data that must be protected, one of the risks that will occur is Sniffing which will lead to theft of confidential information and will result in great loss to the customer [3]. Another risk is that chats stored in databases are more vulnerable to be stolen. To provide guarantee that the relevant data will be safe, then the technique done by develops is to improve the security of live chat, such as firewall and encryption technology [4]. Access Chatting must exist when it is needed, but should be...
able to maintain the trust of consumers [5]. Some works can be reduced by interaction between live chat users, such as product selection, confusing checkout process, addressing customers who want to communicate directly with the customer service, and increasing purchases from e-commerce [6].

An anticipation which will be done to anticipate the risk of live chat is to create a live chat alternative security protocol using cryptographic algorithm. From some cryptographic purposes, what is needed by live chat is Confidentiality and Data Integrity, which is by using encryption and hash function. The cryptographic algorithm will be used is the Rijndael Algorithm [7], RSA Algorithm, and SHA256. Additionally, the selected Rijndael is the 256 bits, because the Rijndael 256 bits has an expansion of 14 rounds and has a higher complexity than the others. In a research conducted by Kalyani and Prof. Vaishali, they point out that [8], RSA algorithms cannot be used for a large data, and AES has security flaws in key transmission. So in the design of web service, the weaknesses owned by the algorithm AES and RSA should be considered. To overcome the weakness of both algorithm, this research will modify Rijndael algorithm. The standard Rijndael algorithm is good, but to increase the complexity required modification by following Shannon's principles of Confusion and Diffusion. For Confusion, the one that will be modified is the S-Box, while for Diffusion, the one that will be modified is the ShiftRow, according to research that already conducted in 2015 [9]. According to Claude Shannon [10], a good encryption algorithm should have two operating properties, namely Confusion and Diffusion.

To create an alternative security protocol on a web-based live chat application, a modified Rijndael algorithm will be implemented to encrypt and describe text message using RSA algorithms for symmetric key exchange for symmetry keys to be securely delivered. The encryption process is done before the message is sent, and the decryption is done after the message is received. The key to encrypt and decryption the message is encrypted again using the RSA algorithm to secure the key delivery. Meanwhile for the hash function, it is done before the message is sent, and after the message is received, then it is compared if the hash result is the same then there is no change of data, yet if it is not the same then the message has changed. Therefore, it can be difficult for third parties to steal and change the information contained in the text message.

By applying the modified Rijndael algorithm, RSA and SHA256, it is expected that this can create a data security for text data in web-based live chat applications and it can work well and solve existing problems.

2. Methods

2.1. Rijndael Algorithm
In the Rijndael algorithm, the number of input blocks, output blocks, and states are 128 bits and 256 bits. With 128 bits of data, means Nb = 4 indicating the data length of each row is 4 bytes. With an input block or data block of 128 bits, the key used in the rijndael algorithm should not have the same magnitude as the input block.

Outline of the Rijndael algorithm that operates on a 128-bit block with a 128-bit key is as follows (outside the round key generation process). This is the encryption process of Rijndael's algorithm. AddRoundKey: perform XOR between initial states (plaintext) with cipher key, This stage is also called initial round. Round as many as N - 1 times, The processes performed on each round are: SubByte, ShiftRows, MixColumn, AddRoundKey. Final round: the process for the last round is SubByte, ShiftRows, AddRoundKey.

2.2. Scenario Rijndael 256 bits Algorithm Modification
To avoid cryptanalis breaking secret keys faster, it is necessary to increase the complexity and reduce the gap in the Rijndael algorithm, hence it is needed a modification which refers to Shannon's Confusion and Diffusion principles.
2.2.1. S-Box Key Dependent. In the SubByte section, modifications are made when generating S-Box which will be used in encryption process and Rijndael decryption algorithm. This modification is based on Shannon's principle of Confusion [11]. The S-box which will be raised will be different from the existing S-Box. S-Box Key Dependent or S-box which depends on the key is generated by performing XOR operations on each key element, as described in Figure 1.

![Figure 1. Example of XOR Operation on the key element](image1)

After finding XOR results for each key element, the next step is to repeat the XOR, XOR key with each element of the existing S-Box, it will generate S-Box_XOR_Key. In Figure 2 we will describe the XOR operation on all S-Box elements with XOR keys.

![Figure 2. Example of XOR operation of all S-Box elements with XOR key](image2)

The next step is to rotate the S-box as much as the result of the XOR key, this research was done in 2008 by Krishnamurthy and Ramaswamy [12] however, in that research the thing done was to rotate S-Box as the first element in the key. Here the researcher will rotate S-Box as much a XOR key result, in Figure 3 will explain the process of rotating Sbox as many as XOR keys.

![Figure 3. Example of the rotate of S-Box as much as XOR Key](image3)
The next step is to make the Inverse S-Box which depends on the key to be used in the decryption process, the way is to move the position of each S-Box\_XOR\_key element with the algorithm shown in Figure 4.

\begin{verbatim}
for i = 1 to 256
    inv_s_box(s_box(i) = i-1;
endfor
\end{verbatim}

**Figure 4.** Pseudocode inverse of the S-Box

### 2.2.2. ShiftRow Dependent Key

In the ShiftRow section, the modification is done based on Shannon's Diffusion principle. The modification is done when will do the shift of each state where the shift is affected by the key, this research has been done by Iman M. Ramadhan et al in 2015 [9].

The first step, each row in RoundKey the XOR operation will be done, then, after getting the XOR result of each row it will be done a ranking to determine the number of shifts that will be done by each row. Ranking is sorted from the largest to the smallest. For the first rank there is no shift like the usual ShiftRow on the first row, for the second row the shift to the left is done once, and so on until the fourth rank. For more details can be seen an example in Figure 5.

**Figure 5.** Examples of the ranking of each row

Once the rank of each line is obtained, then it can be determined the number of shifts from each line. Inverse ShiftRow is done in the same way but the direction of the shift to the right.

### 2.3. RSA Algorithm

Of the many public-key cryptography algorithms ever made, the most popular algorithm is the RSA algorithm. RSA algorithm security lies in the difficulty of factoring large numbers into prime factors. Factoring is done to obtain private key. During the factoring of large numbers into prime factors, it have not found the algorithm that effective, so during that time the RSA algorithm security is also guaranteed [13].

### 2.4. Hash Function

Hash function is a one-way function in which if we enter data, it will generate a "checksum" or "fingerprint" of the data. A message passed to the hash function will produce an output called Message Authenticated Code (MAC). Another requirement of a good hash function is that changing one character (in a text file) or one bit in another data must produce a much different output, not just one bit. Some common hash functions are MD5, SHA (Secure Hash Algorithm) [13].

SHA-2 is not so widely used even for security, SHA-2 is one of the safest among other algorithms. This is due to the protracted process that causes the length of time in doing the hash. SHA-2 itself is divided into 4 types, depending on the output bit of its hash result, SHA-224, SHA-256, SHA-384, SHA-512.
2.5. Logistic Map
The logistic map is the simplest chaotic system in the form of an iterative equation [14]. This algorithm can be formulated as follows:

\[ x_{i+1} = r x_i (1 - x_i) \]  

The value of \( x_i \) is between \( 0 \leq x_i \leq 1 \), \( i = 0, 1, 2, \ldots \) and \( 0 \leq r \leq 4 \). The initial value (seed) of the iteration equation is \( x_0 \).

Equation (1) is deterministic because if the same value of \( x_0 \) is added then the same chaotic value (\( x_i \)) sequence is generated. Therefore, a random number generator with a chaos system is called a pseudo-random generator. The most important characteristic of the Chaos algorithm is its sensitivity to the small change in the initial value. This means that if the change in key values used, then the results which is obtained will not be the same.

Conversion of chaos value to integer is done by using the proposed cutting function in [13]. The way is to multiply the value of chaos (\( x \)) by 10 repeatedly until it reaches the desired number of sizes, then cut the multiplication to take its integer part only.

Mathematically the conversion function is as follows:

\[ T(x, \text{size}) = \| x \times 10^{\text{count}} \|, \quad x \neq 0 \]  

In this case the count starts from 1 and increases 1 to \( x \times 10^{\text{count}} > 10^{\text{size} - 1} \). The result is then taken only the integer part. For example, suppose \( x = 0.004276501 \) and size = 4, then starting from count = 1 to count = 6 is obtained

\[ 0.004276501 \times 10^6 = 4276.501 > 10^3 \]

From the result above is taken its integer part, the result is: \( \| 4276.501 \| = 4276 \)

2.6. Live Chat Encryption and Decryption Process
The security protocol which will be built on the live chat app will employ the modified Rijndael 256 bits algorithm, RSA algorithm, and SHA256.

2.6.1. Encryption Process. Encryption process is done on the customer side and customer service. Encryption on Live chat there are various stages that will be shown in figure 6.

\[ \text{Figure 6. The Chat Encryption Process} \]

In the Encryption Process, or on the sender side, the process starts from generating a Rijndael key of 64 bits using chaos theory, which is Logistic Map with the term of session Key. After that encrypt the plaintext or message using the previously generated key that will become the cipher text (CT). Then take the public key receiver (PBK) to encrypt the Rijndael (K) key, because the Rijndael cryptographic algorithm is symmetry then the key must be protected by Asymmetry cryptography algorithm. After encrypting the Rijndael (K) key, a hash function has to plaintext using SHA256 (H_{256}), the hash function is used to preserve the authenticity of the plaintext. Results from plaintext (CT) encryption, key encryption (ENCK), and a hash of plaintext (H_{256}) will be sent to the recipient.

2.6.2. Decryption Process. Decryption process is done on the customer side and customer service. Decryption on Live chat there are various stages that will be shown in figure 9.
In the decryption process, the messages sent from the sender will be parsed before the process. It is described into 3 parts: encryption plaintext / message (CT), encryption key (ENCK), and hash of plaintext / message (H_{256}). After that the rijndael key decryption uses the recipient's private key (PRK), after obtaining the rijndael (K) key and then plaintext / message (PT_{DEC}), after getting the plaintext / message re-do the hash function (H_{256}) in order to preserve the authenticity of the message, if the hash resulting from the plaintext / message is the same as the hash sent, then it means that there is no changes in the message, and the message can be displayed, if not the same then the message will not be displayed.

3. Results and Discussion
The following is a discussion of the results of tests that have been performed on the modified Rijndael 256 bits algorithm. The test is to calculate the value of avalanche effect and randomness test.

3.1. Avalanche Effect
An avalanche effect is said to be good if the resulting change in bits ranges from 45-60% (50% is a very good result [15]). This is because the change means making the difference is difficult enough for the cryptanalyst to do the attack.

In their research, Eman Mohammed, Ahmed, and Talaat [16] obtained an avalanche effect of AES-128 by converting 1 bit in secret key by 41% to 61%. They concluded that it is very difficult to predict plaintext, and also reflect that this algorithm is resistant to cryptanalysis.

In table 1, there is an avalanche effect testing of algorithm Rijndael 256 bits with various keys. The value results can be seen around 50.7813%.

| No | Rijndael Algorithm 256 bits | Keys 00 | Keys 01 | Keys 10 | Keys 11 | Keys Alphabet | Keys Alphanumeric | Keys Logistic Map |
|----|-----------------------------|--------|--------|--------|--------|--------------|-------------------|-------------------|
| 1  | Standard                    | 41.4063| 46.0938| 51.5625| 52.3438| 44.5313      | 45.3125           | 46.0938           |
| 2  | S-Box Depend Key            | 41.4063| 46.0938| 51.5625| 52.3438| 44.5313      | 53.125            | 53.9063           |
| 3  | ShiftRow Depend Key         | 48.4375| 52.3438| 56.2500| 46.0938| 42.9688      | 47.6563           | 57.8125           |
| 4  | ShiftRow Depend Key         | 48.4375| 52.3438| 56.2500| 46.0938| 47.6563      | 60.1563           | 50.0000           |
3.2. Randomness Test

Randomness test is a series test to see randomness. The purpose of the series test to determine whether there is a particular pattern in the data or whether the data is a random sample. For the randomness test test, it is needed five basic tests for randomness, or 5 statistical tests which are usually used to determine whether a binary row has the characteristics as shown by a random row [17], the five basic tests for the deflection are the frequency test, serial test, poker test, run test. The randomness test was tested with the same plaintext and encrypted with different keys then analyzed its randomness using Crytpool 1.4.3 application. In Table 2. It is the result of randomness test with all the keys of 'universitas pendidikan indonesia', and Rijndael 256bits algorithm with the key of all 'universitas pendidikan indonesia' succeeded in passing the randomness test.

| Table 2. The Randomness Test with All Keys 'universitas pendidikan indonesia' |
|---------------------------------------------------------------|
| Randomness Test | Rijndael Algorithm 256 bits |
| Type | Max. Value | Standard | S-Box Depend Key | ShiftRow Depend Key | S-Box & ShiftRow Depend Key |
| Frequency | 3.841 | 3.841 | 0.281 | Pass | 0.125 | Pass | 0.00 | Pass | 2.00 |
| Poker | 14.07 | 14.07 | 3.714 | Pass | 5.61 | Pass | 4.094 | Pass | 8.285 |
| Long Run | 34 | 34 | 6 | Pass | 9 | Pass | 6 | Pass | 8 |
| Run | 9.488 | 9.488 | 3.760 | Pass | 4.514 | Pass | 0.908 | Pass | 3.350 |
| Serial | 5.991 | 5.991 | 0.309 | Pass | 5.222 | Pass | 0.118 | Pass | 3.00 |

In Table 3. It is the result of randomness test test with key obtained from logistic map '73285441765877398470565182757283', and Rijndael 256bits algorithm with key obtained from logistic map '73285441765877398470565182757283' has passed in randomness test.

| Table 3. The Randomness Test with logistic map '73285441765877398470565182757283' |
|---------------------------------------------------------------|
| Randomness Test | Rijndael Algorithm 256 bits |
| Type | Max. Value | Standard | S-Box Depend Key | ShiftRow Depend Key | S-Box & ShiftRow Depend Key |
| Frequency | 3.841 | 0.781 | Pass | 0.125 | Pass | 0.781 | Pass | 2.0 | Pass |
| Poker | 14.07 | 6.0 | Pass | 3.333 | Pass | 3.714 | Pass | 8.285 | Pass |
| Long Run | 34 | 6 | Pass | 8 | Pass | 5 | Pass | 8 | Pass |
| Run | 9.488 | 3.956 | Pass | 4.201 | Pass | 2.139 | Pass | 3.350 | Pass |
| Serial | 5.991 | 1.321 | Pass | 0.118 | Pass | 0.909 | Pass | 3.0 | Pass |

In table 4. can be seen the test results Randomness test of ciphertext algorithm Rijndael 256 bits implementation results in live chat application pass from randomness test test.

| Table 4. Randomness Test implementation results in the Live Chat app |
|---------------------------------------------------------------|
| Randomness Test | Type | Max. Value | Standard |
| Frequency | 3.841 | 0.015 | Pass |
| Poker | 14.07 | 2.811 | Pass |
| Long Run | 34 | 11 | Pass |
| Run | 9.488 | 2.098 | Pass |
| Serial | 5.991 | 0.733 | Pass |

4. Conclusion

The first paragraph after a heading is not indented (Bodytext style). The modified 256 bits Rijndael algorithm in S-box and ShiftRow passes through some test options given that all results are below maximal test value, this proves the binary sequence of ciphertext results is completely random. In the
avalanche effect test results show that modification ShiftRow section gives a better results 50.2232%, while the results of avalanche effect testing on the modification of the S-box section gives 48.9955% results. So, modification in the ShiftRow section gives a better avalanche effect. From the research that has been done then it is summarized that the algorithm is very good and resistant to cryptanalyst.

Live Chat security protocol which is built with Algorithm Rijndael 256 bits that have been modified and combined with RSA Algorithm and SHA256 hash function into Live Chat application successfully implemented. Security protocol in live chat is already fulfilling some aspects of cryptography that is confidentiality and data integrity because it can protect messages between the customer and customer service in the form of text data.

5. References
[1] Ahmed E 2013 Applying The Technology Acceptance And Service Quality Models To Live Customer Support Chat For E-Commerce Websites The Journal of Applied Business Research 589-596
[2] Ahmed E Benefits 2011 Analyis of Live Customer Support Chat in E-Commerce Websites: Dimension of a new Success Model for Live Customer Support Chat in International Conference on Machine Learning and Applications
[3] Kartika I S 2015 Metode Keamanan E-Commerce Jurnal Informasi & Pengembangan Iptek 11 2 99-108
[4] Ahmed E, et al 2015 Factors Influencing Live Customer Support Chat Service: An Empirical Investigation in Kuwait Journal of Theoretical and Applied Electronic Commerce Research 10 3 63-76
[5] Víctor A, et al 2013 Trustiness in customer/user chat services: the importance of design http://www.upf.edu
[6] Dorine C A 2010 Online Customer Service Chat: Usanility and Sociability Issues (Georgetown University)
[7] Joan D and Vincent R 2002 The Design of Rijndael, AES-The Advanced Encryption Algorithm
[8] K G Kadam and P V Khairnar 2015 Hybrid RSA-AES Encryption for Web Service International Journal of Technical Research and Applications 51-56
[9] Eddy P N, Rizky R J P and Iman M R 2016 SMS authentication code generated by Advance Encryption Standard (AES) 256 bits modification algorithm and One time Password (OTP) to activate new applicant account Science in Information Technology (ICSITech) 2nd International Conference on IEEE 175-180
[10] Claude E S 1949 Communication Theory of Secrecy System Bell System Technical Journal 28 4 656-715
[11] William S 2003 Cryptography and network security: principles and practice (Pearson Education India)
[12] G N Krishnamurthy and V Ramaswamy 2008 Making AES stronger: AES with key dependent S-box IJCSNS International Journal of Computer Science and Network Security 8 9 388-398
[13] Rinaldi M 2004 Sistem Kriptografi Kunci-Publik (Bandung: Departemen Teknik Informatika Institut Teknologi Bandung)
[14] R D Stinson 2002 Cryptography Theory and Practice 2nd Edition Boca Raton (London: CRC Press Inc)
[15] Rohan A 2011 Advance Encryption Standard (AES) Modes of Operation (Maryland: University of Maryland)
[16] Eman M M, et al 2013 Dynamic AES-128 with key-dependent S-box
[17] A Menezes and P Van 1996 Oorschot, and S Vanstone-Handbook of Applied Cryptography