Cryptography using Random Rc4 Stream Cipher on SMS for Android-Based Smartphones

Rifki Rifki[1], Anindita Septiarini[2], Heliza Rahmania Hatta[3]
Department of Computer Science, Faculty of Computer Science and Information Technology,
Mulawarman University, Jl. Panajam Kampus Gn. Kelua, Samarinda, Indonesia.

***Abstract*—Messages sent using the default Short Message Service (SMS) application have to pass the SMS Center (SMSC) to record the communication between the sender and recipient. Therefore, the message security is not guaranteed because it may read by irresponsible people. This research proposes the RC4 stream cipher method for security in sending SMS. However, RC4 has any limitation in the Key Scheduling Algorithm (KSA) and Pseudo Random Generation Algorithm (PRGA) phases. Therefore, this research developed RC4 with a random initial state to increase the randomness level of the keystream. This SMS cryptography method applied the processes of encryption against the sent SMS followed by decryption against the received SMS. The performance of the proposed method is evaluated based on the time of encryption and decryption as well as the average correlation value. Based on the time, it shows that the length of the SMS characters sent affects the time of encryption and decryption. Meanwhile, the best correlation value achieved 0.00482.**

***Keywords*—Cryptography; SMS security; RC4 stream cipher; random initial state; correlation value**

I. Introduction

Cryptography is a science to protect data or information from irresponsible people by turning it into a form where the attacker cannot recognize the data or information while in the processes of storing and transmitting [1]. Moreover, it can be applied to communication services through wireless systems for the communication applications of cellular and wireless [2]. It consists of two-phase, namely encryption and decryption. The encryption is implemented to make data unreadable, invisible or incomprehensible during transmission or storage. While the opposite of it is decryption to reverse the encryption data become an original text [3]. Nowadays, there are various types of smartphones are widely used by the public, one of them is an Android-based smartphone. However, SMS service has not a security method on Android smartphones. SMS is a text messaging service that allows cellular customers to send the text to each other. Global System for Mobile Communication (GSM) uses as a tool for sending SMS messages. SMS message sent by the user, then it was stored by the SMSC to forward to the target mobile device. SMSC uses a store-and-forward technique to store messages to forward it to the target device. If the Home Location Register (HLR) of the target mobile device is active, then SMSC will transfer the SMS message to target mobile device. SMSC receives the verification message that confirms the delivery status of SMS message to the target mobile device [4]. The maximum length of an SMS without the image/graphic is 160 characters using 7 bits or 70 characters using 16 bits of character encoding [5].

Cryptographic methods are divided based on key-based and keyless [6]. Several conventional keyless cryptographic methods have implemented for improving data security such as Caesar ciphers [7], Vigenere ciphers [8], [9], Zigzag ciphers [10], and Playfair cipher [11]. Those methods are more complex and consume a significant amount of power when applied in the resource-constrained devices for the provision of secure communication [12]. Another method that has used is key-based with Symmetric Cryptography. The type of encryption that used is to provide end-to-end security to SMS messages. This method is appropriate for mobile devices because of limited resources, namely limited power/energy, insufficient memory, and less processing power [4]. The examples of symmetric key cipher block cryptography are AES, DES, and 3DES [3].

Several methods have been performed on SMS services such as AES [13,14], Blowfish [5,15], One-Time Pad Cipher [16], MNTRU [17], Certificate-Less Public Key Cryptography (CL-PKC) to Authentication over a GSM System [18]. Several previous works have developed RC4 for WEP [19], combined RC4 with a genetic algorithm [20] and compared RC4 with other methods. RC4 is one of the most popular stream ciphers in symmetric key cryptography since it uses in several security protocols. Moreover, it has the higher speed and the lower complexity than other stream ciphers. The data of statistics show that the RC4 algorithm is used to protect 50% of TLS traffic as the most widely used secure communication protocol on the internet nowadays [21]. RC4 has a secret internal state and works by generating the pseudorandom stream of bits [22]. The internal state of RC4 consists of an S-box array permutation of 256 bytes from the number 0...., N - 1 and two indices i, j $\in$ {0,. . . , N - 1}. The index i is determined and known to the public, while j and S-box permutations remain confidential [23,24]. The RC4 algorithm consists of the Key Scheduling Algorithm (KSA) used for initializing S-box using variable length key and Pseudo-Random Generation Algorithm (PRGA) to generate keystream bytes.

In the previous researches, RC4 stream cipher compared to AES [25] shows that the performance of RC4 is better than AES which based on the throughput, CPU processing time, memory utilization, encryption time and decryption time. Subsequently, RC4 compared to Blowfish method [17] shows that RC4 has better encryption performance while Blowfish

has better decryption performance for small message texts. However, RC4 has better performance in power consumption for communication. Furthermore, the comparison of RC4 with RSA [23] shows that the algorithm of RC4 better than RSA based on the presented experimental and analytical results of both algorithms evaluated. RC4 has more excellence in execution speed and throughput compared to several other cryptographic methods such as VMPC, HC-128, HC-256, Salsa20 and Grain [24]. Nevertheless, RC4 has a limitation in the KSA and PRGA phases due to the initialization process which produces sequential numbers (0,1,2, ..., 255) that may provide the opportunities for hackers [26]. The development of RC4 with a random initial state will increase the randomness level of the keystream produced by RC4 [23,24].

The development of cryptography in SMS service security is an important and challenging issue. It caused the hackers may steal the contents of the original message of the SMS sent. This research proposed a cryptography method using the RC4 stream cipher on SMS for Android-based smartphones to overcome this issue. The contribution of this research is the use of the initial random state to increase the randomness level of the keystream. This method is expected to increase the level SMS service security.

II. Research Method

This research aims to implement cryptography on SMS for Android-based Smartphones using the RC4 stream cipher method with a random initial state to increase the randomness level of the keystream. The proposed method consists of two main stages, namely encryption, and decryption. The illustration of the cryptography system on sending the message via SMS is shown in Fig. 1. Based on Fig. 1, the initial process of this system is the sender and receiver as the user must apply the process of login. Afterward, in the encryption process, the sender should send the message (plaintext) and the key simultaneously. The message is sent as ciphertext as the implementation result of the RC4 method. Subsequently, the information of the sender's identity, key, and keystream are saved in the server. Meanwhile, in the decryption process, the receiver who has successfully login receives the ciphertext, key, and keystream from the server according to the message. Messages can be decrypted into plaintext based on the key similarity during message encryption.

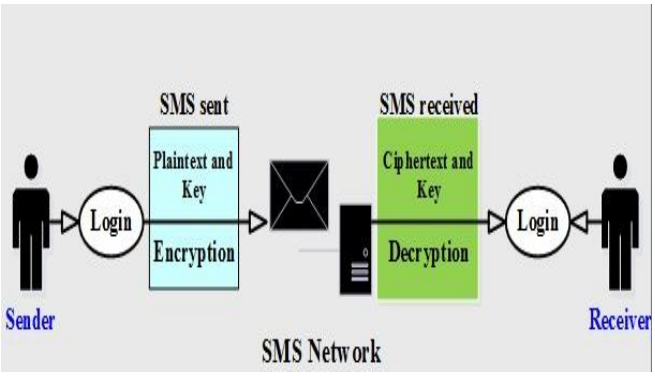

Fig. 1. The Illustration of the Cryptography System.

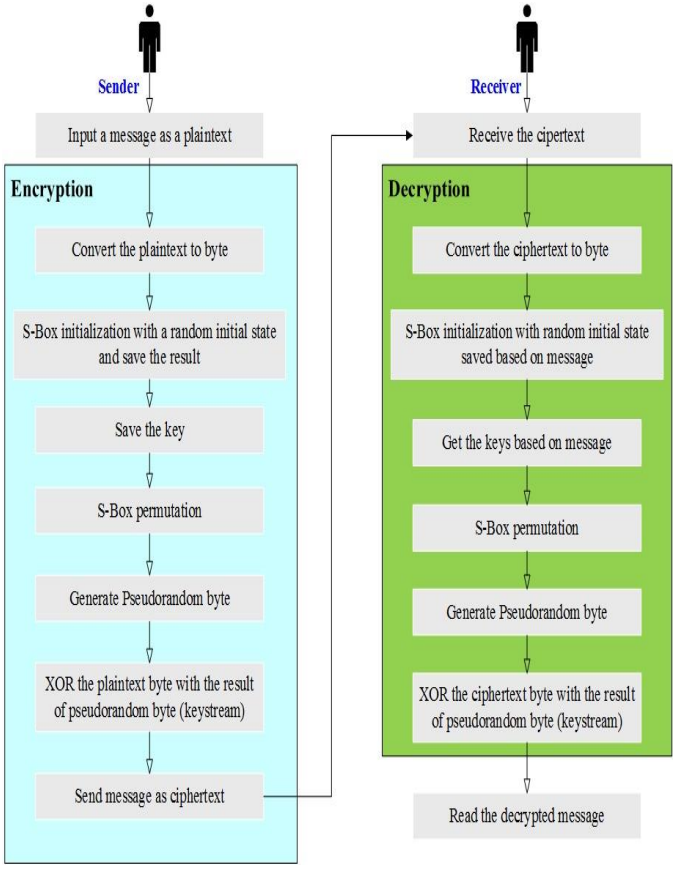

Fig. 2. The Stage Diagram of the Proposed RSA Method.

There are several similar processes in the stage of encryption and decryption, namely, (1) convert the plaintext/ciphertext to byte, (2) S-Box initialization with a random initial state, (3) S-Box permutation, and (4) generate pseudorandom byte to obtains the keystream. The keystream is used to implement XOR operation between the plaintext and ciphertext in a byte. The difference in both stages is in the process of saving the random initial state results and the key in the encryption process. In the decryption process, the results of the random initial state and keys select from the database based on the message to carry out the subsequent process. The detail process of the proposed method is depicted in Fig. 2.

A. Encryption

In this work, encryption is applied to encode the messages sent with the aim only the authorized people whose can access the messages. KSA phase of this work, RC4 allow producing the similar state even though two different keys used and a similar keystream output generated. This case is known as a key collision or related key pairs [27]. It is caused by the initialization process which produces the numbers of 0, 1, 2, .., 255, sequentially which opens opportunities for hackers. The proposed method of the RC4 stream cipher with a random initial state will increase the randomness level of the keystream. Development in KSA phase produces N values from 0 to N-1 without duplication by a pseudo-random number generator which distributes as an additional secret key. The steps of the encryption process in this work are as follows:

1) Get the ASCII values from the messages sent as the plaintext then they are converted to bytes.

2) In KSA phase, the initialization of the S-Box array with a random initial state followed by saving the key and the S-Box permutation. This step is implemented to produce random values between 0 and 255 without duplication.

3) Initialize the keys array then save it.

4) S-Box permutation is performed against the values in the S array by exchanging the contents of the S [i] with S [j].

The Pseudocode of this step is as follows:

```
INPUT: Plaintext L, Keys k, N
OUTPUT: State S
  For i ← 0 to N - 1 Do
  S[i] ← Random_i
  where S∩S=S∪S=S={0, 1, 2, 3, 4… N-1}
  j← 0
  For i ← 0 to N - 1 Do {
    j ← (j + S[i] + k [i mod keylength]) mod N
    Swap S[i] with S[j] }
  j← 0
  Return (S)
```

Input in KSA phase is Plaintext L, Keys k, and N where the message length of plaintext L is the initial key length in bytes, N is the size of the array S, and i and j are indexed pointers. The output of this phase is the array S.

5) In the PRGA phase, retrieving the values of S [i] and S [j] aims to sum up those values in the form of modulo 256. This phase obtains a keystream. The Pseudocode of this step is as follows:

```
INPUT: State S
OUTPUT: Key sequence Kseq
 j← 0
 i← 0
 For i ← 0 to N - 1 Do {
        i ← (i+1) mod 256
        j ← (j+S[i]) mod 256
       Swap S[i] with S[j]
Kseq ← S [(S[i] +S[j]) mod 256]
Return (Kseq)
```

Input in PRGA phase states S where N is the size of the array or state S, and i and j are indexed pointers. The output of this phase is array byte of Kseq using for XOR-ing with plaintext for obtaining ciphertext.

6) Performing XOR-ing keystream, plaintext bytes, obtain ciphertext.

7) Send the SMS message as ciphertext.

B. Decryption

The input of this stage is ciphertext. Decryption aims to reproduce the plaintext, which performed by decoding the ciphertext. The steps of the decryption process are as follows:

1) Get the ASCII values from the received message as the ciphertext then they are converted to bytes.

2) The initialization of the S-Box array in KSA has applied the similar step as in encryption based on the saved key.

3) Get the key based on the message.

4) The processes of S-Box permutation and generate Pseudorandom byte are performed similarly as in encryption.

5) Performing XOR-ing keystream, ciphertext bytes, to obtain the plaintext.

6) Receive the decrypted message as plaintext.

TABLE I. THE SPECIFICATION OF THE VARIOUS SMARTPHONES

Android Version	**Smartphone specification**	
	Processor	***RAM***
Version 4.4 (KitKat)	1.6 GHz	2 GB
Version 5.0 (Lollipop)	2.2 GHz	3 GB
Version 6.0 (Marshmallow)	1.8 Ghz	3 GB
Version 7.0 (Nougat)	2.4 Ghz	4 GB

III. RESULT AND ANALYSIS

This research used 225 message SMS with variations in character length as experimental data. Those data were sent using four smartphones with different specification. The specifications of the smartphone are summarized in Table 1.

The proposed method was evaluated based on the time of encryption and decryption as well as the correlation value between plaintext and ciphertext. The correlation value indicates the quality of encrypted data. This value lies between -1 and 1. The correlation value is defined as follows [28]:

$$|r| = \frac{n\sum(xy) - \sum x \sum y}{\sqrt{\left[n\sum(x^2) - (\sum x)^2\right]\left[n\sum(y^2) - (\sum y)^2\right]}} \quad (1)$$

Where *r* is the correlation value, *x* is the ASCII code value of plaintext and *y* is the ASCII code value of ciphertext. The correlation value should be close to 0 for a good method.

The proposed method aims to encode the SMS message sent as plaintext to ciphertext using the key with the RC4 with a random initial state. Several examples of the encryption result in the form of ciphertext obtaining based on the plaintext and key is shown in Table 2. The evaluation of this proposed method divides into two ways based on the time of encryption and decryption, and the value of the correlation between plaintext and ciphertext.

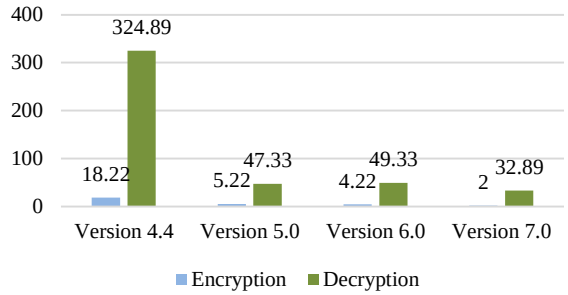

Fig. 3. The Evaluation Results based on the Time of Encryption and Decryption (in Millisecond) using Smartphones with the Different Android Version: Kitkat (Version. 4.4), Lollipop (Version. 5.0), Marshmallow (Version 6.0) and Nougat (Version. 7.0).

A. Evaluation based on the Time of Encryption and Decryption

This evaluation applied by comparing the time of encryption and decryption in four types of smartphones. Those smartphones built in the different Android version, namely version 4.4 (KitKat), version 5.0 (Lollipop), version 6.0 (Marshmallow) and version 7.0 (Nougat). The specification details of those smartphones are presented in Table 1. The performance of this proposed method based on the time evaluation against the SMS message and key with the various length of the character as shown in Table 2. Furthermore, the performance comparison of this proposed method based on the time of encryption and decryption in four types of smartphones is depicted in Fig. 3.

TABLE II. THE RESULT OF ENCRYPTION USING THE PROPOSED METHOD

Key	Plaintext	Ciphertext
first	first test	hZÚ9Áú·Ÿ
application user interface	Android device	Èå³åÖù'=£OÑ¤¬
SMS cryptographic system for safe data user	RC4 STREAM CIPHER	]oÂ âd‡ • ÏkÀ…L
Encryption	plaintext and ciphertext generated by system	ECMÐW)ãºuE÷Ff· ,pÎ XÂ˜Ìwt ±ˆŸÒt™>@ Ù]Å¸
medium SMS and medium key	RC4 has better performance and efficiency	fVDL]zwµãl¿(8ÍÏ¯…l^mÁAÄ 4 aþËpŠ?GT
Michael Jackson - earth song (cover by me)	what about sunrise what about rain what about all	-‰?;W¦•&)ððõlÚ‰qšçÊÏÔŽ ¹bX¢Ì h,]aŸ□ èá“›O¹D
Love	you say you love me, I say you crazy. We're nothing more than friends	âñf˜L!«:Ä□ /×kÃÂô&ÏÖ‰W.5Ìñï¿ ›ïc□ ›T“m1¼I"– 76móâ…´²OÎðÓøvCFs£ñu
Department of computer science	RC4 has become the most popular stream cipher in the history	Ü¡$Æ,ç=†□ Ô‡ úîIL□ˆ SKW2å\½”ðy\ • ÛOÖj× ®ÔÑK óß□ ©‘ä~ïyœ«—½ø
end to end encryption method for safe data	Key Scheduling Algorithm and Pseudo Random Generation Algorithm	E©úÖ=)Í§À0]àÛs¿‹ÞT6ÙVÿ¼gïl× 0\(å‰ƒƒ},J@4÷‘ÁU?w‡ž -!,§ •¿

Based on the experiment result as shown in Fig. 3, the time of encryption and decryption is influenced by the smartphone specification and the number of characters of the SMS message. Related to the smartphone specification, the time of encryption and decryption of Version 5.0 is faster than Version 6.0 due to the processor speed of Version 5.0 higher than Version 6.0 are 2.2 GHz and 1.8 GHz, respectively. In addition, the specification of RAM and the number of applications that are running on the smartphone may affect the time of encryption and decryption. Moreover, the increasing number of characters in an SMS message cause the time of the encryption and decryption to become longer.

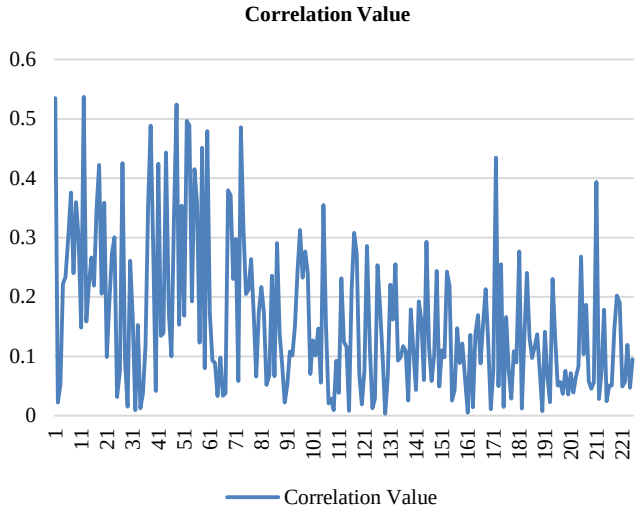

Fig. 4. The Evaluation Results of the Quality of Encrypted Data based on the Correlation Values of the Plaintext and Ciphertext.

B. Evaluation based on the Correlation Values

This research also evaluated the quality of encrypted data which obtains by the proposed method based on the correlation values. Low correlation value (close to 0) indicates that the encryption system is becoming more secure [26]. The correlation values are computed based on the ASCII values of plaintext and ciphertext using Eq. (1). All correlation values which generated from 225 examples of experimental data are summarized in Fig. 4. Based on the result is presented in Fig. 4, the correlation values of the best, the worst and the average achieve of 0.00337, 0.53716 and 0.10188, respectively. These results indicate that the increasing number of characters in the SMS message and the key, the correlation value is getting closer to 0. In this research, the correlation value is getting closer to 0 if the number of keys and plaintext character more than 30 and 25, respectively. Otherwise, the correlation value tends to closer to 1. The resulting correlation value only influenced by the number of characters from the plaintext and the key used but not affected by the smartphone specification used.

Furthermore, this research also calculated correlation values based on the plaintext and ciphertext which produced by other methods, namely Vigenere and Playfair. A summary of the performance comparisons between those methods and the proposed method is presented in Table 3. Table 3 shows that the proposed method produces the lowest correlation value of 0.10188. It indicates that the quality of the encryption data of the proposed method yields the best result than the other methods.

TABLE III. PERFORMANCE CORRELATION VALUE OF PREVIOUS METHODS AND PROPOSED METHOD

Method	Correlation value
Vigenere cipher	0.81229
Playfair cipher	0.21345
Proposed method	**0.10188**

IV. Conclusion

This paper developed the RC4 method with a random initial state. The random initial state is needed to increase the randomness of the keystream so that this method is safer than RC4 without a random initial state. The proposed method evaluated using 225 data. Based on the evaluation result, the time of encryption and decryption is influenced by the characters, number of the SMS message and the key as well as the smartphone specification. Meanwhile, the correlation value is only affected by the characters number of the SMS message and the key. The correlation value of the proposed method shows an improvement compared to the method of Vigenere and Playfair. For future works, other cryptographic methods are still possible to be developed to reduce correlation values.

Acknowledgment

The authors would like to thank KEMENRISTEK DIKTI for Bidikmisi Scholarship. We also would like to thank the Department of Computer Science, Faculty of Computer Science and Information Technology, Mulawarman University for the facilities during this research.

References

[1] A.A. Zaidan, A. Majeed, and B.B. Zaidan, “High Securing Cover-File of Hidden Data Using Statistical Technique and AES Encryption Algorithm,” Int J Comput Inf Eng, vol. 3, pp. 463–74, 2009.

[2] I. Memon, I. Hussain, R. Akhtar, and G. Chen, “Enhanced privacy and authentication: An efficient and secure anonymous communication for location based service using asymmetric cryptography scheme," Wireless Personal Communications, vol. 84, pp.1487–1508, 2015.

[3] H.O. Alanazi, B.B. Zaidan, A.A. Zaidan, H.A. Jalab, M. Shabbir, and Y. Al-Nabhani, “New Comparative Study Between DES, 3DES and AES within Nine Factors,” J Comput, vol. 2, pp. 152–7, March 2010.

[4] M.W. Khan, “SMS Security in Mobile Devices : A Survey,” Int J Adv Netw Appl, vol. 5, pp. 1873–82, 2013.

[5] H.M. El-Bakry, A.E. Taki_El_Deen, and A. H. El tengy, “Implementation of a Hybrid Encryption Scheme for SMS/Multimedia Messages on Android,” Int J Comput Appl, vol. 85, pp. 1–5, January 2014.

[6] M. F. Mushtaq, S. Jamel, A. H. Disina, Z. A. Pindar, N. S. A. Shakir, and M. M. Deris, "A Survey on the Cryptographic Encryption Algorithms," International Journal of Advanced Computer Science and Applications, vol. 8, pp. 333–344, 2017.

[7] A. Jain, and A. Patil, “Enhancing the Security of Caesar Cipher Substitution Method using a Randomized Approach for more Secure Communication,” Int J Comput Appl, vol. 129, pp. 975–8887, 2015.

[8] S.K. Mandal, and A.R. Deepti, “A Cryptosystem Based On Vigenere Cipher By Using Mulitlevel Encryption Scheme,” Int J Comput Sci Inf Technol, vol. 7, pp. 2096–2099, 2016.

[9] H. Hamdani, H. Ismanto, A.Q. Munir, B. Rahmani, A. Syafrianto, D. Suprihanto, and A. Septiarini, “The Proposed Development of Prototype with Secret Messages Model in Whatsapp Chat,” Int. J. Electr. Comput. Eng., vol. 8, pp. 3841-3849, 2018.

[10] Mu. Annalakshmi, and A. Padmapriya, “Zigzag Ciphers: A Novel Transposition Method,” IJCA Proceedings on International Conference on Computing and information Technology, IC2IT(2):8-12, December 2013.

[11] S. Bhattacharyya, N. Chand, and S. Chakraborty, ”A Modified Encryption Technique using Playfair Cipher 10 by 9 Matrix with Six Iteration Steps,” Int J Adv Res Comput Eng Technol, vol. 3, pp. 307–12, 2014.

[12] K. Sharma, M Ghose, and D. Kumar, “A comparative study of various security approaches used in wireless sensor networks,” Int J Adv Sci Technol, vol. 17, pp. 31–44, 2010.

[13] B. Patil, “SMS Security Using RC4 & AES,” Indian J Sci Res, vol. 11, pp. 34–8, 2015.

[14] R. Rayarikar, S. Upadhyay, and P. Pimpale, “SMS Encryption using AES Algorithm on Android,” Int J Comput Appl, vol. 50, pp. 12–17, 2012.

[15] A. Kaur, and R. Dhadwal, “Performance Comparison of Symmetric Algorithms for SMS Communication,” Int J Adv Res Comput Commun Eng, vol. 4, pp. 62–64, 2015.

[16] M. Iqbal, M. A. S. Pane, and A. P. U Siahaan, “SMS Encryption Using One-Time Pad Cipher,” IOSR J Comput Eng, vol. 18, pp. 54–58, 2016.

[17] S. Jha, and U. Dutta, “Review on SMS Encryption using MNTRU Algorithms on Android,” Int J Comput Sci Inf Technol, vol. 6, pp. 3855–3858, 2015.

[18] I. Memon, M. R. Mohammed, R. Akhtar, H. Memon, M. H. Memon, and R. A. Shaikh, “Design and implementation to authentication over a GSM system using certificate-less public key cryptography (CL-PKC)," Wireless personal communications, vol 79, pp. 661–686, 2014.

[19] L. Stošić, and M. Bogdanović, "RC4 stream cipher and possible attacks on WEP," International Journal of Advanced Computer Science and Applications, vol. 3, pp. 110–114, 2012.

[20] B. Ferriman, and C. Obimbo, "Solving for the RC4 stream cipher state register using a genetic algorithm," International Journal of Advanced Computer Science and Applications, vol. 5, pp. 216–223, 2014.

[21] E. Taqieddin, O. Abu-Rjei, K. Mhaidat, and R. Bani-Hani, “Efficient FPGA Implementation of the RC4 Stream Cipher using Block RAM and Pipelining,” Procedia Comput Sci, vol. 63, pp. 8–15, 2015.

[22] J. Chen, and A. Miyaji, “Novel strategies for searching RC4 key collisions,” Comput Math with Appl, vol. 66, pp. 81–90, 2013.

[23] A. A. Okedola, and Y. N. Asafe, “RSA and RC4 Cryptosystem Performance Evaluation Using Image and Text File,” Int J Sci Eng Res, vol. 6, pp. 289–294, 2015.

[24] S.O. Sharif, and S.P. Mansoor, “Performance Analysis Of Stream And Block Cipher Algorithms,” Int. Conf. Adv. Comput. Theory Eng., vol. 1, pp. 522–525, 2010.

[25] N. Singhal, and J. P. S. Raina, “Comparative Analysis of AES and RC4 Algorithms for Better Utilization,” Int J Comput Trends Technol, vol. 2, pp. 177–181, 2011.

[26] M. M. Hammood, K. Yoshigoe, and A. M. Sagheer, “RC4 Stream Cipher with a Random Initial State,” Lect. Notes Electr. Eng., vol. 253 LNEE, pp. 407–415, 2013.

[27] P. Jindal, and B. Singh, “RC4 encryption - A literature survey,” Procedia Comput Sci, vol. 46, pp. 697–705, 2015.

[28] E. Setyaningsih, C. Iswahyudi, and N. Widyastuti, “Image Encryption on Mobile Phone using Super Encryption Algorithm,” TELKOMNIKA, vol. 10, pp. 837–845, 2012.