International Journal of Information Processing and Communication (IJIPC) Vol. 9 No. 1&2 [May, 2020], pp. 221-233

Special Issue on Computing and Communication Technologies
Online: ISSN 2645-2960; Print ISSN: 2141-3959

Comparative Analysis of Selected Encryption Algorithms to Secure SMS on Android Operating Systems

A. Musa and Moshood A. Ojulari

Centre for Telecommunication Research and Development, Kwara State University, Malete, Nigeria

Email: *twhid2001@yahoo.com; ojularim@gmail.com*

Abstract

This work investigates and analyzes the performance of popular cryptographic schemes on Android-based mobile devices over varying input data scenarios. The cryptographic operations of key generation, message encryption and decryption are experimented and the time it takes to execute these operations are used as performance metric of cryptographic algorithms. Short Message Service (SMS) is one of the oldest and most popular non-voice information transfer methods over mobile communication systems. However, exchange of information via SMS is not secure because it can be easily intercepted. To prevent unauthorized access to messages, cryptography is usually applied to render the transferred information into a format that will be indiscernible to others. To do this, cryptographic keys need to be generated, message needs to be encrypted before sending and the received encrypted message is then decrypted. Three symmetric ciphers and two asymmetric ciphers were selected and chosen using purposive sampling method as criteria. The performance of the selected cryptographic algorithms on cryptographic key generation, SMS encryption and decryption was tested in a virtual environment using some comprehensive sets of Android codes. These cryptographic actions are generally considered to be highly resource-demanding, particularly for resource-constrained devices like mobile phones. The results show Blowfish as the most efficient in terms of key generation and decryption time, while the RSA is the most efficient in encryption time.

Keywords: *Encryption, decryption, SMS, cryptographic algorithms, Android OS, key generation.*

1. Introduction

Networks like the internet and the telephone are indispensable in today's world as they connect people from far and near. However, their use may pose security threat because they allow considerable amount of anonymity and identity theft (Noroozi *et al.*, 2014). A very popular and widely deployed distributed network is the Global Service for Mobile Communications (GSM). GSM services include Short Message Service (SMS), voice call and data transmission (Azubuike and Obiefuna, 2014). SMS is a global and most widely used wireless service that enables the transmission of alphanumeric messages between mobile subscribers. Figure 1 is the block diagram of main network elements of the GSM architecture illustrating the flow of SMS from one user to another.

Special Issue on Computing and Communication Technologies
Online: ISSN 2645-2960; Print ISSN: 2141-3959

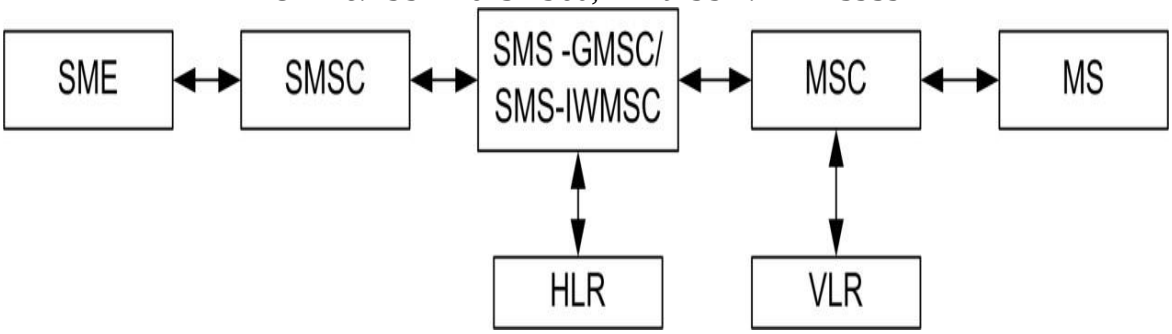


Figure 1. Basic SMS Network Architecture

Hundreds of thousands of SMS are sent daily across the globe because of the simplicity and low cost. These huge SMS exchange also includes the transfer of sensitive information like online purchases and banking transactions for which using SMS may not be ideal because unauthorized persons can read SMS when they have physical access to a user's device and the fact that network providers have access to copies of SMS transmitted. Therefore, if SMS must be used for secure communication purpose, cryptography features need to be added.

Cryptography is one of the most popular tools for ensuring data security on potentially insecure platforms like the GSM network and its services like the SMS. Cryptography is applied on the networks to enable secure communication requirement of confidentiality, integrity, authentication and non-repudiation (Schneier, 1996).

There are billions of mobile phones in use today. Most of them, however, are powered by a handful Operating Systems (OS); Android, IOS, Windows etc. The Android OS will be used in this research work as it is the most popular mobile OS with largest market share as shown in Table 1.

Table 1. Market Share of Popular Mobile OS (Vangie B., 2018).

Period	Android	iOS	Windows Phone	Others
2016Q1	83.4%	15.4%	0.8%	0.4%
2016Q2	87.6%	11.7%	0.4%	0.3%
2016Q3	86.8%	12.5%	0.3%	0.4%
2016Q4	81.4%	18.2%	0.2%	0.2%
2017Q1	85.0%	14.7%	0.1%	0.1%

The report of this work has been presented in a way that introduction is given in Section One, while Section Two reviews the theoretical and related works. In Section Three, methodology and implementation procedures are given, while the results and discussion are presented in Section Four. Conclusion is made in Section Five.

2. Theoretical and Related Works Review

Short Message Entity (SME) is a messaging software embedded in a mobile station (MS) that handles the sending and receiving of SMS. From the MS, an SMS is sent (and received) via a radio link communication through the Base Station System (BSS) whose basic functionality is transmission of voice and data traffic signals to and from the MS. This is the only part where encryption is optionally applied using very weak A5/1 and A5/2 cryptographic ciphers (Saxena and Payal, 2011; Toorani and Beheshti, 2008). The message is then routed to an appropriate Mobile Switching Centre (MSC). Thereafter, the SMS is further routed to the Short Message Center (SMC) or Short Message Service Center (SMSC) which is the entity responsible for the store-and-forward feature

Special Issue on Computing and Communication Technologies
Online: ISSN 2645-2960; Print ISSN: 2141-3959

of SMS messaging. The SMS is now forwarded to an appropriate MSC that serves the recipient MS using routing information from Home Location Register (HLR). Depending on whether the message is intra-network or inter-networks, it may involve the Gateway MSC (GMSC). The GMSC is a mobile network's point of contact with other networks. The visitor location register (VLR), like the HLR, is a database that contains subscribers' temporary information.

As highlighted in Section One, traditional SMS messaging is not secure and cryptographic layer needs to be added by using specific cryptographic algorithm for appropriate security. Figure 2 contains a block diagram of encryption and decryption.

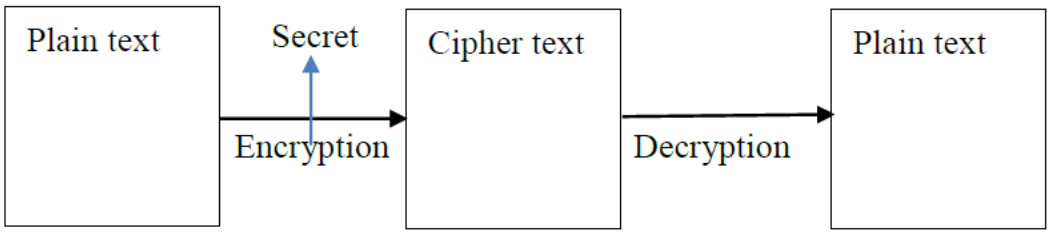


Figure 2. Block diagram of encryption and decryption.

There are several cryptographic algorithms even as they require varying degree of host's resources (i.e. power, RAM, etc.). The resource requirements for the algorithms on Android OS is not readily available in public domain. Cryptography comes in two flavors namely symmetric and asymmetric cryptography. While symmetric cryptography can provide confidentiality and integrity functionalities, asymmetric cryptography can provide all the four secure communication requirements (Lisonek and Drahansky, 2008). Although Kessler (2014) and Lisonek and Drahansky (2008) gave an overview of cryptography and SMS encryption for mobile communication, it is important to explain some of the symmetric and asymmetric ciphers and algorithms.

Advanced Encryption Standard (AES) Cryptography: The AES is a superior and one of the widely used cryptography also adopted by the U.S. Government. The algorithm described by AES is a symmetric key algorithm, i.e. the same key for both encryption and decryption of data. Using AES on Android platform, Muhammad and Adeel (2018) developed a secured SMS application. The application was very resilient to brute force attacks because of the strength and durability characteristics of the AES. However, key exchange mechanism was not implemented.

Blowfish Cryptography: Blowfish is a symmetric block that takes a variable length key from 32 bits to 448 bits making it ideal for both domestic and exorable use. The Blowfish has been as a fast, free alternative and replacement to some existing encryption algorithms. Twofish Cryptography: The Twofish is a well-known encryption algorithm commonly used in cryptography and steganography. Twofish algorithm is derived from Blowfish algorithm. It is a 128 bits block cipher that accepts a variable length key up to 256 bits.

Rivest-Shamir-Adleman (RSA): The RSA is one of the full-fledged and simplest asymmetric cipher (Schneier, 1996) used for both encryption and digital signing. RSA key size of 1024 bits is used in this work, as this is considered secured enough (Kessler, 2014). The RSA algorithm flow chart is a sequential adding process repeats until a target is met and satisfied. ElGamal Cryptography: The ElGamal Cryptography, described by Taher ElGamal, is an asymmetric key encryption algorithm for public–key cryptography which is based on the Diffie-Hellman key exchange.

While the work of Toorani and Beheshti (2008) was to secure SMS protocol for the M-Payment Systems, that of Saxena and Payal (2011) was for M-Commerce in GSM. Using dynamic digital signature for authentication

purpose, Noroozi et. (2014) also enhanced secured data hiding. The RSA encryption algorithm was used to encrypt messages for secure messaging on the SMS channel by Bodur and Kara (2015). This was validated on Android devices. The work mentioned different key sizes which could be used for key generation processes and changes occurring on encrypted messages. They also listed existing alternative solutions to be used for the encryption process in secure messaging. This include a method of message encryption based on the AES encryption algorithm providing a secured medium.

Mohammed et. al. (2017) also presented a parallel implementation of an AES cryptographic algorithm known as XTS-AES for storage encryption inside the Android-based architecture. The work achieved a performance speedup of 1.63 with over 80% efficiency in average. Besides, the authors clarified that mobile devices do have a long way for performance improvement especially in such a way that different data protection techniques can be easily integrated inside the mobile devices.

Suffices to add that Natanael et. al. (2018) also implemented an encryption algorithm for secure SMS. The scheme, however, did not include verification of the identity of communicating parties and their associated public keys. Finally, Dahlan et al. (2018) developed a multiple layer of security for information exchange. But the solution did not discuss how to handle key exchange problem inherent in symmetric-based cryptographic implementations. Bearing these in mind, two different cryptographic algorithms to give stronger security have been considered in this work. Besides, the key generation operations have also been considered and included.

3. Methodology

Performance analysis to determine the best cryptographic algorithm among popular choices suitable for adding secure communication features to SMS communication on Android OS-powered mobile phones is carried out in this work. To achieve this, three symmetric ciphers and two asymmetric ciphers were selected and chosen using purposive sampling method and wide acceptance as criteria. Secondly, purposive sampling method was also used to choose mobile phone OS using popularity and market share of mobile phones OS (illustrated in Table 1) as criteria. Then a GUI-based mobile app (i.e. simulation in a virtual environment) was developed to test and measure the performance of the selected cryptographic algorithms on cryptographic key generation, SMS encryption and decryption using some comprehensive sets of Android codes. Results were obtained using appropriate mobile phone as the testing device. Lastly, the output performance was analyzed in terms of execution period for encryption and decryption and the storage size requirements for all the selected cryptographic algorithms.

4. Results and Discussion

Table 2 to Table 6 present the results obtained from the tests and analysis that were carried out on the five cryptographic algorithms. The tests were run fifty times each for each of the five algorithms. In Table 2 containing the Advanced Encrypted Standard (AES), the data size is 176 bytes, the average encryption time is 2.84ms, the average decryption time is 2.82 ms and the average key generation is 0.65 ms. In Table 3 containing the Blowfish algorithm, the data size is 168 bytes, the average encryption time is 2.12 ms, the average decryption time is 1.10 ms and the average key generation is 0.57 ms. In Table 4 containing the Twofish algorithm, the data size is 176 bytes, the average encryption time is 2.12 ms, the average decryption time is 2.24 ms and the average key generation is 0.61 ms.

Special Issue on Computing and Communication Technologies
Online: ISSN 2645-2960; Print ISSN: 2141-3959

Table 2. Advanced Encrypted Standard (AES)

ADVANCED ENCRYPTED STANDARD (AES)			
S/N	**Encryption time (ms)**	**Decryption time (ms)**	**Key generation time (ms)**
1	9	8	2
2	4	8	1
3	5	6	1
4	5	5	0
5	4	5	0
6	5	5	1
7	4	5	0
8	4	4	0
9	4	5	1
10	5	3	1
11	5	4	1
12	4	3	1
13	4	3	1
14	4	3	1
15	3	3	1
16	2	2	1
17	2	2	1
18	3	2	1
19	3	2	1
21	2	2	1
22	2	2	0
23	2	2	0
24	2	2	1
25	2	2	1
26	2	2	0
27	2	2	1
28	2	2	1
29	2	2	0
30	2	2	0
31	2	2	1
32	2	2	0
33	2	2	1
34	2	2	1
35	2	2	0
36	2	2	0
37	2	2	1
38	2	2	0
39	2	2	0
40	2	2	0
41	2	2	1
42	2	2	1
43	2	2	0

Special Issue on Computing and Communication Technologies
Online: ISSN 2645-2960; Print ISSN: 2141-3959

Table 2. Advanced Encrypted Standard (AES) (Cont..)

44	2	2	1
45	2	2	0
46	2	2	1
47	2	2	1
48	2	2	0
49	2	2	1
50	2	2	1
Total Average	**2.83673**	**2.81633**	**0.65306**

Table 3. Blowfish

Blowfish			
S/N	**Encryption time (ms)**	**Decryption time (ms)**	**Key generation time (ms)**
1	28	1	1
2	1	1	1
3	1	1	0
4	2	1	1
5	1	1	0
6	2	1	0
7	1	2	1
8	1	1	0
9	1	1	0
10	1	1	1
11	1	1	1
12	2	2	1
13	1	2	1
14	1	1	0
15	3	1	1
16	1	2	1
17	1	1	1
18	1	1	0
19	1	2	0
21	1	1	1
22	1	1	0
23	1	1	0
24	1	1	1
25	23	1	1
26	2	1	0
27	1	1	1
28	1	1	0
29	1	1	0
30	1	1	1
31	1	1	1
32	1	1	0
33	1	1	0

Special Issue on Computing and Communication Technologies
Online: ISSN 2645-2960; Print ISSN: 2141-3959

Table 3. Blowfish (Cont..)

34	1	1	0
35	1	1	1
36	1	1	0
37	1	1	0
38	1	1	0
39	1	1	1
40	1	1	1
41	1	1	1
42	1	1	1
43	1	1	0
44	1	1	0
45	1	1	1
46	1	1	1
47	1	1	1
48	1	1	1
49	1	1	1
50	1	1	1
Total Average	**2.122449**	**1.102041**	**0.571429**

Table 4. Twofish

Twofish			
S/N	**Encryption time (ms)**	**Decryption time (ms)**	**Key generation time (ms)**
1	14	2	3
2	2	2	1
3	2	17	0
4	1	2	1
5	2	2	1
6	2	7	1
7	2	2	0
8	1	1	0
9	2	1	1
10	2	2	0
11	3	2	0
12	2	1	0
13	1	2	1
14	2	2	0
15	2	2	0
16	3	1	0
17	2	2	1
18	1	2	0
19	2	2	0
21	1	1	1
22	2	2	1
23	1	2	1
24	2	2	1
25	2	1	1

Table 4. Twofish (cont..)

26	2	1	0
27	2	2	1
28	2	3	0
29	2	2	0
30	1	2	0
31	1	1	1
32	2	2	1
33	2	2	1
34	3	1	0
35	2	2	1
36	2	2	1
37	1	2	1
38	2	2	1
39	2	2	0
40	2	2	0
41	2	2	1
42	2	2	0
43	2	2	1
44	2	2	1
45	2	2	0
46	2	2	0
47	2	2	1
48	2	2	0
49	2	2	1
50	2	2	2
Total Average	**2.122449**	**2.244898**	**0.612245**

In Table 5 containing the RSA where the data size is 128 bytes, the average encryption time is 1.67 ms, the average decryption time is 8.53 ms and the average key generation is 667 ms.

Table 5. RSA

RSA			
S/N	**Encryption time (ms)**	**Decryption time (ms)**	**Key generation time (ms)**
1	12	8	239
2	2	9	417
3	1	9	846
4	2	8	775
5	1	9	1482
6	1	9	1301
7	1	10	535
8	1	8	723
9	1	9	625
10	2	9	584
11	1	8	589
12	1	8	842

Special Issue on Computing and Communication Technologies
Online: ISSN 2645-2960; Print ISSN: 2141-3959

Table 5. RSA (cont..)

13	2	9	1865
14	1	9	110
15	2	9	139
16	2	10	66
17	2	8	806
18	1	8	362
19	2	9	1041
21	1	11	331
22	2	10	1326
23	2	12	438
24	2	7	614
25	1	8	835
26	1	10	657
27	1	7	1860
28	1	8	260
29	1	8	243
30	1	8	92
31	1	8	227
32	2	8	966
33	1	8	225
34	1	9	436
35	2	9	628
36	1	9	909
37	2	8	346
38	1	8	642
39	2	8	1043
40	1	8	730
41	2	8	405
42	1	7	1740
43	1	8	463
44	1	8	477
45	2	8	452
46	2	9	389
47	2	8	390
48	2	8	494
49	2	8	863
50	2	8	867
Total Average	**1.673469**	**8.530612**	**667.2449**

In Table 6 containing the ElGamal algorithm where the data size is 256 bytes, the average encryption time is 37.43 ms, the average decryption time is 13.92 ms and the average key generation is 26969 ms.

Special Issue on Computing and Communication Technologies
Online: ISSN 2645-2960; Print ISSN: 2141-3959

Table 6. ElGamal

ElGamal			
S/N	**Encryption time (ms)**	**Decryption time (ms)**	**Key generation time (ms)**
1	27	14	15762
2	44	17	100512
3	33	19	90654
4	35	15	50564
5	45	15	45600
6	44	14	44000
7	46	15	35000
8	47	14	35456
9	48	14	34345
10	45	16	34000
11	34	15	28678
12	34	14	27456
13	37	14	27768
14	38	12	26789
15	47	12	25890
16	45	14	24123
17	46	13	20453
18	48	14	18435
19	46	15	18356
21	34	13	17890
22	35	13	17754
23	34	13	17543
24	34	15	16789
25	34	13	15765
26	35	14	15786
27	49	13	67900
28	29	14	45780
29	29	15	43987
30	28	12	14500
31	27	12	34598
32	27	13	26098
33	39	13	15765
34	38	13	12435
35	37	13	14675
36	39	13	14567
37	37	13	15980
38	36	14	23546
39	35	14	43670
40	32	15	12908
41	34	14	13567
42	35	14	10543
43	36	15	10567
44	38	15	11657
45	38	14	14890

International Journal of Information Processing and Communication (IJIPC) Vol. 9 No. 1&2 [May, 2020], pp. 221-233
Special Issue on Computing and Communication Technologies
Online: ISSN 2645-2960; Print ISSN: 2141-3959

Table 6. ElGamal (cont..)

46	44	14	13564
47	35	13	12987
48	34	13	13678
49	32	13	14789
50	31	13	13456
Total Average	**37.42857**	**13.91837**	**26968.88**

Figure 3 (encryption operations), Figure 4 (decryption operations) and Figure 5 (key generation) present the summarized results of all the tests carried out on all the five cryptographic algorithms, where A, B, C, D and E represent AES, Blowfish, Twofish, RSA and ElGamal respectively.

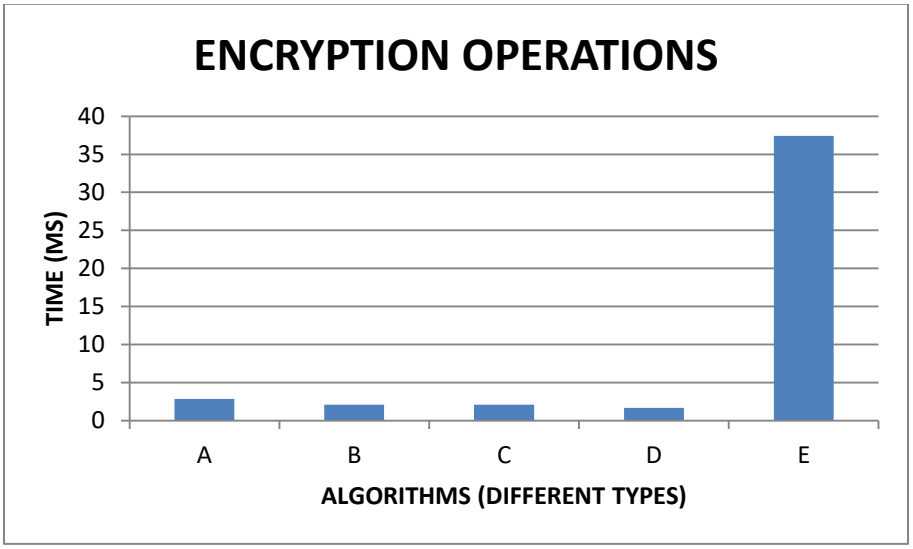
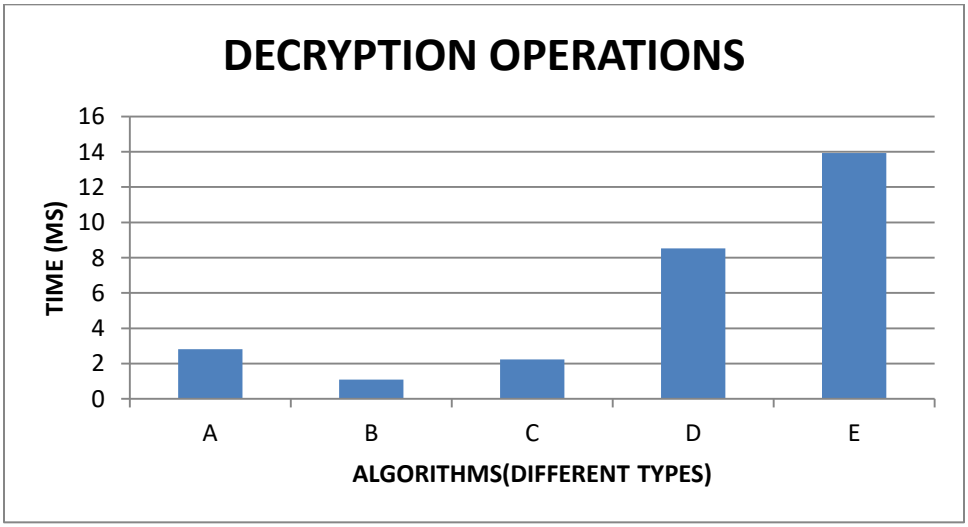


Figure 3. Encryption operations



Special Issue on Computing and Communication Technologies
Online: ISSN 2645-2960; Print ISSN: 2141-3959
Figure 4. Decryption operations

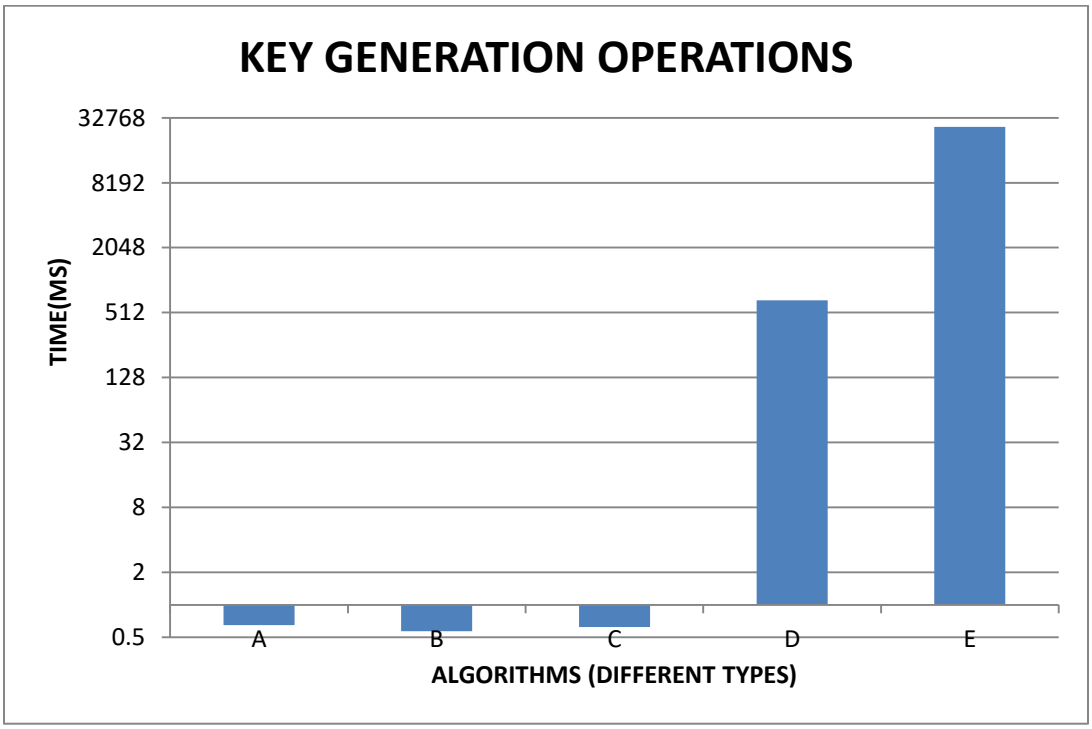


Figure 5. Key generation operations

5. Conclusion

Performance analysis of SMS encryption on Android operating systems using cryptographic algorithms has been carried out in this work. The focus is on the time each of the cryptographic algorithms will take to perform encryption, decryption and key generation operations as well as the resources they will utilize given the limited resources available to Android mobile phones. Five cryptographic algorithms have been selected and analyzed using metrics that include the time taken to encrypt messages, the time it takes to decrypt text messages and the size of the cipher text. This allows for determination of which of the algorithms is more efficient in terms of encryption and decryption periods and resources such as storage sizes. It also allows for determination of ideal cryptographic algorithms for secure communication purposes on Android-based mobile devices.

Of the five cryptographic algorithms analyzed namely AES, Blowfish, Twofish, RSA and ElGamal, it can be concluded that the Blowfish is the most efficient in terms of key generation and decryption time, while the RSA is the most efficient when it comes to encryption time. The least efficient algorithm in terms of key generation, encryption and decryption period was found to be ElGamal.

Special Issue on Computing and Communication Technologies
Online: ISSN 2645-2960; Print ISSN: 2141-3959

Reference

Azubuike, C. and Obiefuna, O. (2014). Wireless Communication: The Impact of GSM on the Economic Lives of the Nigerian Rural Users. Journal of Educational and Social Research, 4 (79), pp. 109 - 119.

Dahlan, A., Robbi, R., Andysah, P. U., Ananda, F. U., Zahratul, F., Malahayati, M. and Harun, H. (2018). Super-Encryption Cryptography with IDEA and WAKE Algorithm. 1st International Conference on Green and Sustainable Computing (ICoGeS). IOP Conf. Series: Journal of Physics: Conf. Series 1019, pp. 1-5. doi :10.1088/1742-6596/1019/1/012039.

Hüseyin, B. and Resul, K. (2015). Secure SMS Encryption Using RSA Encryption Algorithm on Android Message Application, 3RD International Symposium on Innovative Technologies in Engineering and Science, At Valencia, pp. 1 -10, 2015.

Kessler, G. C. (2014). An Overview of Cryptography. Available at: "http://www.garykessler.net/library/crypto.html." Retrieved 02-12-2014

Lisonek, D. and Drahansky, M. (2008). SMS Encryption for Mobile Communication. IEEE International Conference Security Technology (SECTECH), pp. 198-201.

Mohammed, A. A., Khairulmizam, S., Abdul, R. R., and Shaiful, J. H. (2017). Efficient Android-based storage encryption using multi-core CPUs, 2017 John Wiley & Sons, Ltd. Available at: https://doi.org/10.1002/sec.1727

Muhammad, N. R. and Adeel, I. (2018). Development of a Secure SMS Application using Advanced Encryption Standard (AES) on Android Platform. International Journal of Mathematical Sciences and Computing. DOI: 10.5815/ijmsc.

Natanael, D., Faisal, and Suryani, D. (2018). Text Encryption in Android Chat Applications using Elliptical Curve Cryptography (ECC). *Procedia Computer Science, 135*, pp. 283-291.

Noroozi, E., Salwani, M. D. and Sabouhi, A. (2014). Enhancing Secured Data Hiding Using Dynamic Digital Signature for Authentication Purpose. Jurnal Teknologi (Sciences & Engineering) 68(2): 39–44.

Schneier, B. (1996). Applied cryptography. 2nd ed., John Wiley and Sons, Inc., New York.

Saxena, N. and Payal, A. (2011). Enhancing Security System of Short Message Service for M-Commerce in GSM. International Journal of Computer Science & Engineering Technology (IJCSET) 2(4), pp. 402-410.

Toorani, M. and Beheshti, A. A. (2008). A Secure SMS Messaging Protocol for the M-Payment Systems. Proceedings of the 13th IEEE Symposium on Computers and Communications (ISCC), pp.700-705.

Vangie, B. (2018). Mobile Operating Systems (Mobile OS) Explained. Available at: https://www.webopedia.com/DidYouKnow/Hardware_Software/mobile-operating-systems-mobile-os-explained.html