I. J. Computer Network and Information Security, 2017, 2, 29-35
Published Online February 2017 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijcnis.2017.02.04

Design of an Android Application for Secure Chatting

Ammar H. Ali
University of Anbar/ Department of Computer Science, Baghdad, 10013, Iraq.
E-mail: p.a.alfahad@gmail.com

Ali M. Sagheer
University of Anbar/ Information System Department, Baghdad, 10013, Iraq.
E-mail: ali_makki@computer-college.org

Abstract—Smart phones have become an essential part in the life of the individuals and their priorities at the present time. The most prominent uses are in chatting and conversation applications. Most of these applications do not provide the required protection and privacy of the data exchanged between users. Yet there are very few mobile chat applications that provides an End-to-End (E2E) security and privacy-preserving service to their clients. In this paper, a secure chatting application with end to end encryption for smart phones that use the android OS has been proposed. The proposed application uses the ECDH algorithm to generate the key pair and exchange to produce the shared key that will be used for the encryption of data by symmetric algorithms. The proposed Application allows the users to communicate via text messages, voice messages, as well as exchange photos. For the text message security the standard AES algorithm with a 128 bit key is used. The generated key (160 bit) minimized to 128 bit length in order to be used by the AES algorithm. For the voice and image security processes the proposed application uses the symmetric algorithm RC4 for this purpose. RC4 provides less security than AES, but it performs faster and this is required for such types and sizes of data.

Index Terms—Android, Chatting Application, ECDH (Elliptic Curve Diffie Hellman Key Exchange), AES (Advanced Encryption Standard), RC4 (Rivest Cipher 4).

I. Introduction

With the presentation of the internet, a lot of our correspondence has been done through the screen of our PC monitors or even through our telephones. One of the numerous additions to this was that of chatting application, allowing instant communication with anybody in the same city, in a different state, or anyplace on the planet.

The traditional SMS text message is quickly becoming ancient, particularly in the light of the explosion of popular chatting application. WhatsApp, Telegram, Viber and different applications all offer free text messaging. Moreover, that is not notwithstanding mentioning the options for voice and picture sharing to different clients. Chatting applications have turned into a lifestyle. They appear to be a more dependable type of communication than a phone call.

In this paper, a system is developed that provides end to end chatting application running on mobile devices that operate on android platform.

A. Problem Statement

Today's smartphones and tablets are more than communication devices. They are hip mounted PCs, with more memory and handling more power than portable PC of only a couple of years earlier. They are an incorporated portion of our lives. However, smartphones and tablets raise new security issues due to uncovering touchy information.

The chatting applications are the most widely used of the mobile application to communicate information and data over the internet. In the internet world, the governments and malicious people are always attempting to hack the servers and reveal information about the users that use this type of application.

There are different mobile chat applications accessible in the Apps markets. Large number of these applications claim that they are providing security, confidentiality and integrity of user's information. But the hacking information prove that many of developers do not consider security as the primary consideration of their applications.

B. Objectives

End to End Encryption: in order to achieve this property, ECDH key exchange is intended to provide the key pair (private and public) which will be exchanged between the two parties and generate the secure shared key that will be used as a key for the encryption algorithms.

Confidentiality, In order to achieve such a goal, encryption is the mechanism that provides confidentiality between two parties. A message is encrypted by the symmetric algorithm and this encrypted message can only be readable by the intended party.

Privacy, one of the most operators that is achieved in the proposed application. It will not grab metadata of the user which includes information about a user, such as location, name, and contact information.

Integrity, the proposed application insures that a message has not been edited or changed during transferring it between entities because any edition or even replace the message with a new one by an attacker, the message will not appear probably.

Efficient performance, this is achieved by encrypting the data by symmetric encryption algorithms (AES and RC4) and still provide end to end encryption from the using of ECDH key exchange to provide the key, this lead to maintain speed and the mobile processor capability.

This paper is orderly in addition the section I as follows; in section II we list some of related papers that is present number of secure application for smartphones. Section III center on Elliptic Curve Key Exchange. IV describe the AES algorithm that used for the text encryption and decryption. Section V describe the RC4 algorithm that used for the voice and image encryption and decryption. Section VI represents the proposed security model. The proposed model for end to end Android Chatting Application is briefed in section VII. VIII presents the implementation procedures and discusses the main results. The main conclusions is briefed in section IX.

II. Related Works

There are a large number of mobile chat applications that claim to provide a secure service, but their complete architecture is not publicly available. To our best knowledge there are not many publications that describe such systems.

In 2013 Dec, Ali Makki Sagheer et al, proposed a solution that provides confidentiality and integrity for SMS data by applying a crossbred cryptographic scheme which join the AES for encryption/decryption scheme and RC4 for key expansion and generation algorithms to fulfil more powerful security issues. The proposed model is implemented by Java programming language based on Net Beans platform. The proposed system was tested on various mobile devices such as the Nokia 5233 [1].

In 2014 May, H.C. Chen et al. [2] exhibited another idea about Mobile Text Chat utilizing a revolution session key based transposition cryptosystem plan. Their proposed scheme only deals with the secure text transposition for mobile chat system. It acclimatized the technologies of classical block cipher, substitution and transposition. In addition, the new session key can be produced by the matrix rotation technology. It could be easily applied to transmit via mobile devices using the quick encryption algorithm.

In 2014 July, R.N. Akram et al, evaluated the security and privacy preserving features introduced the current mobile chat services. They additionally put advances a fundamental system for an end to end security and protection mobile chat service and related necessities. They additionally put advances a fundamental system for an end to end security and protection mobile chat service and related necessities. Their proposal was implemented to produce proof-of-concept and valuation the technical difficulty of satisfying the specified security and privacy requirements [3].

In 2014 Nov, Hsing-Chung Chen et al, planned the essential system for secure end to end mobile chat plan and its related necessities. Their proposal is implemented to provide alternate authentication and prevent the password estimating attack and the undetectable on-line password estimating attack. In addition, the plan is a secret key based authentication and key agreement having simple recollected property [4].

In 2015 Jan, Pejman Dashtinejad [5], investigate current security features of common messaging applications in the mobile market. A list of requirements for acceptable security is generated and based on those requirements an architecture is developed. A demo is also implemented and evaluated.

III. ECDH Key Exchange

The security of the application depends largely on Elliptic Curve Cryptography, and using ECDH algorithm which is a variant of the Diffie-Hellman algorithm for elliptic curves. It is actually a key-agreement protocol, more than an encryption algorithm. This essentially implies ECDH characterizes how keys ought to be produced and exchanged between parties.

In the elliptic curve Diffie-Hellman (ECDH) key exchange, the two communicating client's client_A and client_B agree beforehand to use the same curve parameters and base point G. They each generate their private keys Pr_A and Pr_B, respectively, and the corresponding public keys $Pu_A = Pr_A * G$ and $Pu_B = Pr_B * G$. Both the client_A and client_B exchange their public keys, and each multiplies its private key with the other party's public key to derive a common shared secret $Pr_A * Pu_B = Pr_B * Pu_A = Pr_A * Pr_B * G$. An attacker cannot determine this shared secret key from the curve parameters [5].

IV. AES Algorithm

In January 1997, the United States National Institute of Standards and Technology (NIST) reported that it would hold an opposition to choose another block cipher to be known as the Advanced Encryption Standard, or AES to supplant DES [7]. The cipher takes a plaintext block size of 128 bits, or 16 bytes. The key length can be 16, 24, or 32 bytes (128, 192, or 256 bits). The algorithm is alluded to as AES-128, AES-192, or AES-256, contingent upon the key length [8].

The input to the encryption and decryption algorithms is a solitary 128 piece block. This block is delineated as a 4*4 square matrix of bytes. This block is replicated into the State array, which is adjusted at every phase of encryption or decryption. After the last stage, State is replicated to an output matrix. Likewise, the key is portrayed as a square matrix of bytes. This key is then ventured into an array of key schedule words. Each word is four bytes, and the total key timetable is 44 words for the 128 piece key. The cipher comprises of N rounds, where the quantity of rounds relies on upon the key length: 10 rounds for a 16-byte key, 12 rounds for a 24-byte key, and 14 rounds for a 32 byte key [9] [10].

There are four fundamental strides, called layers that are utilized to form the rounds:

1. The Byte Sub Transformation (BS): Uses an S-box to play out a byte-by-byte substitution of the block. This non-linear layer is for resistance to differential and linear cryptanalysis assaults.
2. The Shift Row Transformation (SR): A straightforward permutation. This linear blending venture causes diffusion of the bits over multiple rounds.
3. The Mix Column Transformation (MC): A substitution that makes utilize of arithmetic over GF (28). This layer has a purpose similar to SR.
4. Add Round Key (ARK): A basic bitwise XOR of the present piece with a part of the extended key. The round key is XORed with the result of the above layer [10] [11].

V. RC4 Algorithm

RC4 is a stream cipher which was arranged in 1987 by Ron Rivest for RSA Security. It is a variable key size stream cipher with byte oriented operations. The algorithm depends on the utilization of an irregular permutation [9]. It has the ability of utilizing keys somewhere around 8 and 2048 bits. RC4 is utilized as a part of numerous business programming bundles, for example, Lotus Notes and Oracle Secure SQL. It is likewise part of the Cellular Specification [12]. It works in two stages, key setup and ciphering. Both stages must be performed for each new key. The key stream is totally autonomous of the plaintext utilized [13].

RC4 follows the design strategy utilized in stream ciphers. To extract the pseudorandom data bytes from a pseudorandom stage is the fundamental design principle of RC4 stream cipher. RC4 has two working modules: first there is a KSA with key K as input (with typical size of 40-256 bits), and second is PRGA which creates a pseudo-random output sequence [14].

RC4 has a variable length of the key that ranges between (0 - 255) bytes for initializing an array of 256 bytes in the initial state (State [0] to State [255]) [15].

At the point when the key size is little, it must be exceptionally effective and encryption time be very fast, numerous encryptions that are utilized in wireless devices depend on symmetric key encryption such as RC4 algorithm [16].

VI. The Application Security Model

The security of the application depends largely on Elliptic Curve Cryptography. After the generation of the key pairs these key will be used to generate the secure shared key, which is 160 bit key length. The data will be encrypted in asymmetric algorithms (AES 128 for text, RC4 for voice and image) by using the generated secure shared key. Hence, the encryption algorithms take key length which differs from the generated key, the generated key is submitted in key scheduling algorithm (KSA) in order to be in suitable length form.

The proposed chatting application employs a symmetric key encryption technique where the message is encrypted and decrypted with the generated secret key. The selected algorithm to be employed in this system for the text message is AES 128-bits with cipher block changing mode (CBC).

Before encrypting the message, the generated key (160 bit) is minimized to 128 bit length. Toward the beginning of the Cipher, the input is copied to the State array utilizing the traditions. After an initial Round Key expansion, the State array is changed by actualizing a round function 10, 12, or 14 times (contingent upon the key length 128, 192, 256 bit), the proposed application uses 10 rounds function with 128 bit key length. With the final round differing slightly from the first nine rounds, All ten rounds are identical with the exception of the final round, which does exclude the MixColumns() change. The last State is then replicated to the output. Also, at the decryption side, the generated key (160 bit) is minimized to 128 bit length. The decryption procedure is the inverse of the encryption process.

The procedure of decryption of an AES ciphertext is like the encryption procedure in the opposite order. Each round consists of the four processes (InvShiftRows, InvSubBytes, AddRoundKey and InvMixColumns) except the last round that not perform the InvMixColumns. Since sub-processes in each round are backward way, not at all like for a Feistel Cipher, the encryption and decryption algorithms should be independently executed, despite the fact that they are closely related.

For the voice and image encryption processes, the proposed application uses the symmetric algorithm (RC4) for this purpose. In the RC4 encryption algorithm, the key stream is totally free of the plaintext utilized. For the voice/image encryption procedure, to generate the key stream, the cipher makes use of a secret internal state which consists of two parts:

1. A permutation of all 256 possible bytes (S).
2. Two index-pointers (i and j).

The permutation is initialized with the ECDH generated key (160 bit), using the key-scheduling algorithm (KSA). At that point the stream of bits is created by the PRGA. The algorithm utilizes a variable length key from 1 to 256 bytes to form a 256 byte state table. The state table is utilized for subsequent generation of pseudo-random bytes and afterward to create a pseudo-random stream which is XORed with the plain data bytes to give the cipher data bytes. Every element in the state table is swapped once in any event.

In the RC4 algorithm, key setup is the first and most troublesome period of this encryption algorithm. The encryption key is utilized to create an encrypting variable utilizing two arrays, state and key, and N-number of blending operations.

The PRGA modifies the state and outputs a byte of the key stream. In each iteration, the PRGA increments i, looks up the ith element of S, S[i], and adds that to j, exchanges the values of S[i] and S[j], and then uses the sum S[i] + S[j] (modulo 256) as an index to fetch a third element of S, which is XOR'ed with the next byte of the message to produce the next byte of either cipher data or plain data.

RC4 creates a pseudo-random stream of bits (a key-stream). Similarly, as with any stream cipher, these can be utilized for encryption by combining it with the plaintext utilizing bit-wise exclusive-or. Decryption is played out the same path (since exclusive-or is a symmetric operation).

VII. The Proposed Application Model

The system is android application that enables users to communicate with each other in a safe way and provides them with end to end security communication. This communication process is done through data encryption and submitted to the internet server in an encrypted format and then retrieved by certain queries and decrypted, then shown to the recipient user. The application consists of a set of interfaces design, the user moves between them to perform the chat process with the rest of the users.

Registration Screen: As show in the screen shot in Fig. 1, to hold new user Registry process. The registration process is consist of insert a new user in the user class on the server. And there are in the server special class was created to contain changing user information, such as a user's status, whether online or offline also the information that is constantly changing depending on the user status and activities. And this information be the basis of queries through which the exchange of declared keys done and inform the user whether there was unread messages, also used to indicate the status of other users.

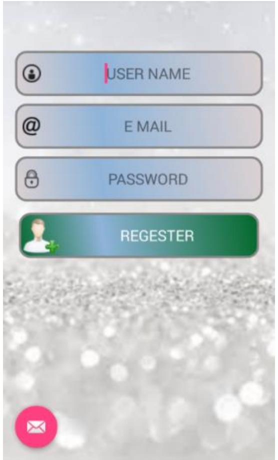

Fig.1. The Registration Screen

The User List Screen: When conducting the registration process the application saves user data in the phone to be used in the login process in the future. At this stage, the application generates a pair key. The private key stored in the phone and the public key is submitted to the server. List of users interface contain on the list of all the registered users of the application as show in the screen shot in Fig. 2, and provide the user with information about all other users.

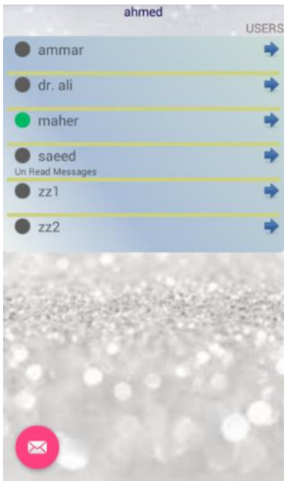

Fig.2. The user List Screen

The Main Chat Screen: The main chat interface consists of a small bar at the top shows the user name and the user status, list of the data exchanged, and the taskbar at the bottom as show in the screen shot in Fig. 3, which enables the user to write a text message, make voice record or open gallery to select image to be transmitted. Each message stored in encrypted form with its own information. This information are used in the queries by which the message retrieved in the correct form and sequence.

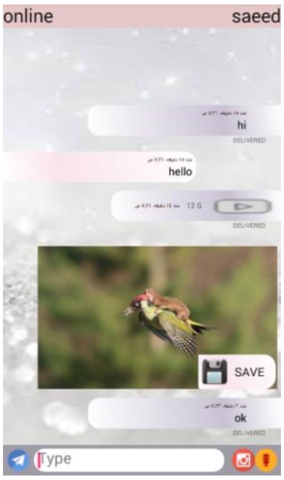

Fig.3. The Chat Screen

VIII. The Results and Discussion

The proposed system was installed and tested on multiple mobile phone devices that are based on android operating systems with various CPU capabilities and Random Access Memories (RAM), to ensure that it is able to work properly on all of them. Table 1 shows different types of phone devices used to apply and test the system on them and the specifications of these devices.

Table 1. Specifications of the Mobile Devices That the Application Has Been Experimenting with Them

Devise Name	*Android Version*	*RAM*	*CPU*
Samsung GT – S7272 Galaxy Ace 3	4.2.2	1 GB	1 GHz
Huawei ALE-L21 P8 Lite	5.0.1	2 GB	1.2 GHz
Samsung Galaxy S6 Duos	5.0	3 GB	2.1 GHz

The results of encrypting and decrypting pieces of text messages are presented in table 2. The results are in terms of execution time in millisecond. Fig.4. and Fig. 5. Show the encryption/decryption results plot of table 2. The algorithm used for encrypting text messages in the proposed application is the AES standard which is slower than other block cipher, but it provides a higher security. The results presented in table 2, shows acceptable execution speed suitable for the mobile phones processors which have constrained resources of power and cost, the real time computation requirements and other distinct characteristics such as limited programmability. The results are acceptable even for large blocks of data.

Table 2. Performance Time Metrics of the Text Message Encryption/Decryption

Size in Bytes	*Time (ms)*					
	Galaxy Ace 3		*Huawei P8 Lite*		*Galaxy S6 Duos*	
	Enc	*Dec*	*Enc*	*Dec*	*Enc*	*Dec*
32	60	63	19	22	7	10
128	63	68	20	23	8	11
512	68	72	21	24	14	12
2048	77	84	23	26	15	13
4096	90	92	24	27	20	13

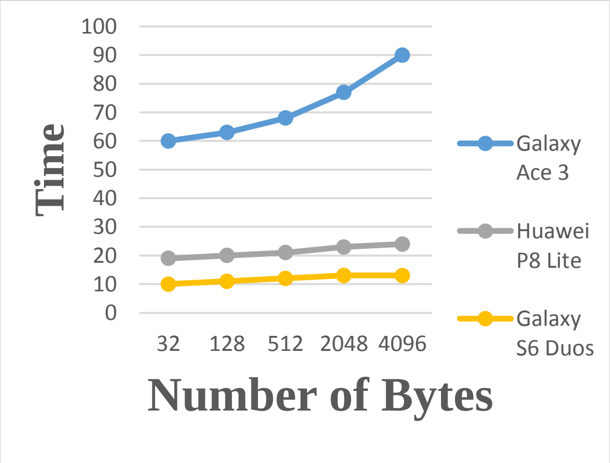

Fig.4. Encryption Time for the Text Message

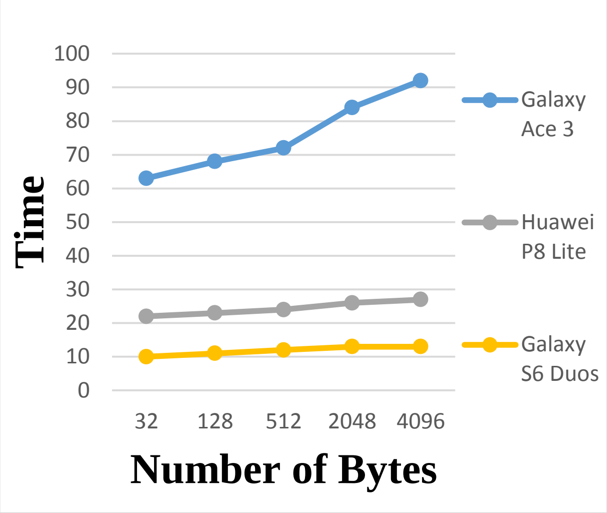

Fig.5. Decryption Time for the Text Message

Table 3 shows the duration and the size of the tested voice messages, hence the max length of the voice message allowed in the proposed application is 60 Sec, and therefore, it is the max length tested.

Table 3. The Voice Message Duration and Size

NO	*Duration (Sec)*	*Size (KB)*
1	10	16
2	20	31
3	30	48
4	45	71
5	60	95

Table 4 shows the time of voice encryption and decryption processes in millisecond. Fig.6. and Fig. 7. Show the encryption/decryption results plot of table 4. The algorithm used for encrypting voice and image messages is the RC4 which is one of the fastest encryption techniques and it is suitable for the mobile device when encrypting vast amounts of data.

Table 4. Performance Time Metrics of the Voice Message Encryption/Decryption

No	*Time (ms)*					
	Galaxy Ace 3		*Huawei P8 Lite*		*Galaxy S6 Duos*	
	Enc	*Dec*	*Enc*	*Dec*	*Enc*	*Dec*
1	3	3	2	2	1	1
2	6	6	4	4	1	2
3	10	9	7	5	3	1
4	18	14	9	8	5	2
5	23	18	13	10	6	4

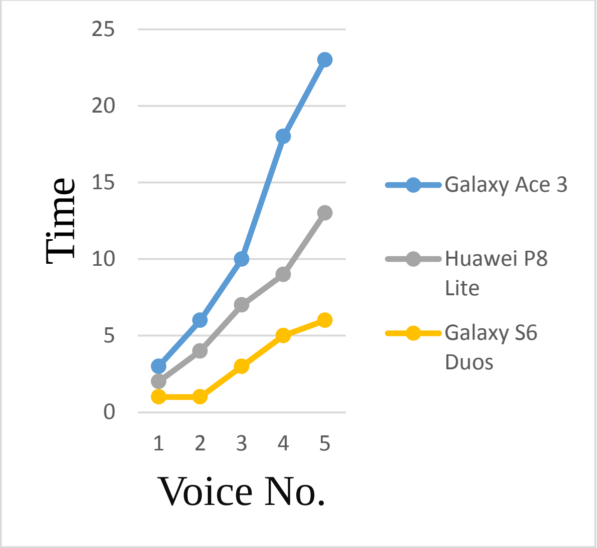

Fig.6. Encryption Time for the Voice Message

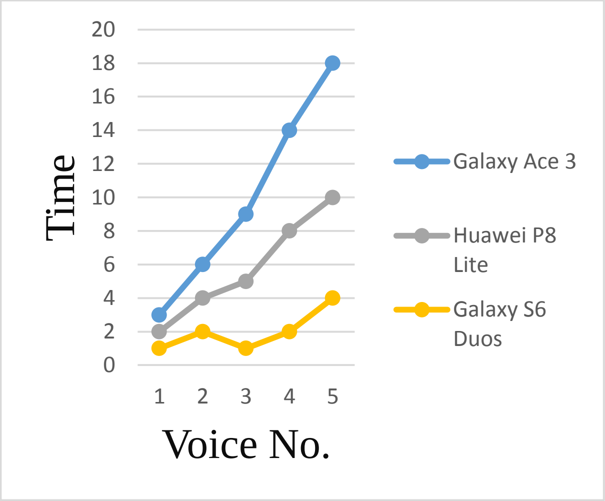

Fig.7. Decryption Time for the Voice Message

Table 5 shows the examined image size, NPCR and UACI. The NPCR and UACI are intended to test the quantity of changing pixels and the quantity of averaged changed intensity between encrypted pictures.

Table 5. The image message size, NPCR and UACI

NO	*Size (KB)*	*NPCR*	*UACI*
1	26	99.59	33.986
2	66	99.62	29.135
3	118	99.61	32.694
4	181	99.60	29.887
5	220	99.62	32.616

The proposed application allows transfer images that have size less than 250 KB. So, the tested images have the allowed size only. Table 6 shows the time of images encryption and decryption processes in millisecond. Fig.8. and Fig. 9. Show the encryption/decryption results plot of table 6.

Table 6. Performance Time Metrics of the Image Message Encryption/Decryption

No	*Time (ms)*					
	Galaxy Ace 3		*Huawei P8 Lite*		*Galaxy S6 Duos*	
	Enc	*Dec*	*Enc*	*Dec*	*Enc*	*Dec*
1	58	57	53	47	69	28
2	125	118	107	103	112	54
3	185	184	168	164	130	95
4	266	269	248	242	167	101
5	287	284	261	257	201	138

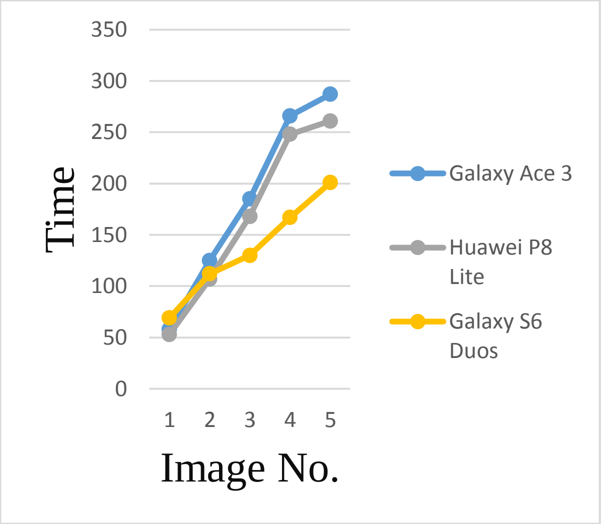

Fig.8. Encryption Time for the Image Message

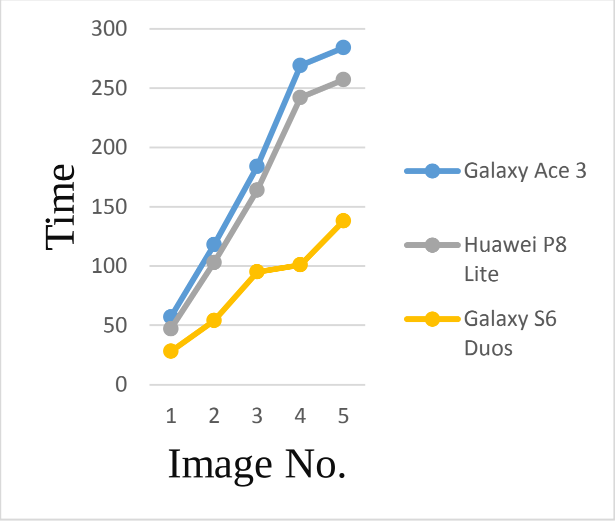

Fig.9. Decryption Time for the Image Message

IX. Conclusions

In this paper, a secure chatting application was developed. The proposed application was tried on various mobile devices. According to the obtained results the following are summarized as conclusions.

End to End Encryption is achieved by involving ECDH key exchange to provide the key pair, which will be exchanged between the two parties to generate the secure shared key that will be used as a key for the encryption algorithms. The proposed secure chatting application furnish confidentiality, privacy and integrity. Users can be granted that nobody, even the provider of the service, cannot read their messages. The exchanged data is store only at the server, and nothing of them is stored at the physical memory of the phone. The algorithm used for encrypting text messages is the AES standard which is slower than other block cipher but it provides higher security. The algorithm used for encrypting voice and image messages is the RC4 which is one of the fastest encryption techniques and it is suitable for the mobile device when encrypting immeasurable sums of data.

REFERENCES

[1] Ali Makki Sagheer, Ayoob Abdulmunem Abdulhameed and Mohammed Adeeb AbdulJabbar, “SMS Security for Smartphone”, Sixth International Conference on Developments in eSystems Engineering, 2013.

[2] H.C. Chen and A.L.V. Epa, “A Rotation Session Key-Based Transposition Cryptosystem Scheme Applied to Mobile Text Chatting”, Proceedings of The 28th IEEE International Conference on Advanced Information Networking and Applications (AINA2014), pp. 497 - 503, Victoria, Canada, May 2014.

[3] Raja Naeem Akram, and Ryan K. L. Ko. “End-to-End Secure and Privacy Preserving Mobile Chat Application”, Information Security Theory and Practice. Securing the Internet of Things Lecture Notes in Computer Science, pp.124-139, 2014.

[4] Hsing-Chung Chen, Jyh-Horng Wen and Cheng-Ying Yang, “A Secure End-to-End Mobile Chat Scheme”, Ninth International Conference on Broadband and Wireless Computing, Communication and Applications, 2014.

[5] Pejman Dashtinejad,” Security System for Mobile Messaging Applications “, Thesis, KTH University, Jan 2015.

[6] S. Kumar, M. Girimondo, A. Weimerskirch, C. Paar, A. Patel, and A. S. Wander, “Embedded End-to-End Wireless Security with ECDH Key Exchange”, 2003 46th Midwest Symposium on Circuits and Systems.

[7] Randhir Kumar and Akash Anil, “Implementation of Elliptical Curve Cryptography”, IJCSI, 8(2), pp. 544-549, 2011.

[8] Suchita Tayde and Seema Siledar. “File Encryption, Decryption Using AES Algorithm in Android Phone”, International Journal of Advanced Research in Computer Science and Software Engineering, Vol. 5(5), pp. 550-554, 2015.

[9] William Stallings, “Cryptography and Network Security: Principles and Practice”, Prentice Hall, Boston, 5th Ed, 2011.

[10] Joseph Migga Kizza, “A Guide to Computer Network Security”, Springer, London, 2nd Ed, 2012.

[11] W. Trappe and L. Washington, “Introduction to Cryptography with Coding Theory”, Pearson International, 2nd Ed, 2006.

[12] Bhimrao Patil, “SMS SECURITY USING RC4 & AES”, Indian J.Sci.Res, Vol. 11(1), pp. 34-38, 2015.

[13] Meltem Kurt and Nevcihan Duru. “Email Encryption Using RC4 Algorithm”, IJCA Vol. 130(14), pp. 25-29, 2015.

[14] Poonam Jindal, Brahmjit Singh,"A Survey on RC4 Stream Cipher", IJCNIS, vol.7, no.7, pp.37-45, 2015.DOI: 10.5815/ijcnis.2015.07.05.

[15] M. M. Hammood, K. Yoshigoe, and A. M. Sagheer, “RC4-2S: RC4 Stream Cipher with Two State Tables”. Springer, Lecture Notes in Electrical Engineering, 2013, 1, pp 13-20.

[16] P. Prasithsangaree, and P. Krishnamurthy, “Analysis of energy consumption of RC4 and AES algorithms in wireless LANs”, In Global Telecommunications Conference, 2003. GLOBECOM'03. IEEE, 2003, 3, pp 1445-1449.

Authors’ Profiles

Ammar H. Ali has received his B.Sc. in Computer Science (2013) from the University of Anbar, Iraq. He is a master student (2015, till now) in the Computer Science Department, College of Computer Sciences and Information Technology at Al-Anbar University. He is interested in the following fields; Mobile Computing, Information Security, Coding Systems.

Ali M. Sagheer is a Professor in the Computer College at Al-Anbar University. He received his B.Sc. in Information System (2001), M.Sc. in Data Security (2004), and his Ph.D. in Computer Science (2007) from the University of Technology, Baghdad, Iraq. He is interested in the following fields; Cryptology, Information Security, Number Theory, Multimedia Compression, Image Processing, Coding Systems, and Artificial Intelligence. He has published many papers in different scientific journals.

How to cite this paper: Ammar H. Ali, Ali M. Sagheer,"Design of an Android Application for Secure Chatting", International Journal of Computer Network and Information Security(IJCNIS), Vol.9, No.2, pp.29-35, 2017.DOI: 10.5815/ijcnis.2017.02.04