Design of Secure Chatting Application with End to End Encryption for Android Platform

Ammar Hammad Ali[1], Ali Makki Sagheer[2]

[1] College of Computer Science and Information Technology / University of Anbar
p.a.alfahad@gmail.com

[2] College of Computer Science and Information Technology / University of Anbar
ali.m.sagheer@gmail.com

Abstract: ***In this paper, a secure chatting application with end to end encryption for smart phones that used the android OS has been proposed. This is achieved by the use of public key cryptography techniques. The proposed application used the Elliptic Curve Diffie Hellman Key Exchange (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 and photos. For the text message security the standard AES algorithm with a 128 bit key are used. The generated key (160 bit) minimized to 128 bit length by selecting the first 128 bit of the generated key in order to be used by the AES algorithm. For the voice and image security processes the proposed application used the symmetric algorithm RC4 for this purpose.***

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

1. INTRODUCTION

The mobile instant message applications have overwhelmed the Short Message Service (SMS) operated by cellular network carriers, with 19 billion messages sent for every day contrasted and more than 17 billion SMS messages [1].

Instant message will assume an essential part later on business territories, which are prevalently known as m-commerce, mobile banking, administrative use, and everyday life correspondence. Moreover, instant message has turned into a famous wireless service all over the world as it encourages a client to be in contact with any mobile phone subscriber anyplace on the planet [2].

With the increasingly developing dependence on mobile chat system in one hand, and the developing number of vulnerabilities and assaults on the other hand, there is an undeniably interest for the security solutions. There are likewise some extra security issues in the wireless media that are not the situation in a wired framework. In this manner, extraordinary secure protocols are required for assortment mobile chat system platforms [3].

Customers utilize a mobile chat service to communicate with each other, a procedure that can incorporate relaying individual data. The security and protection of such communications ought to be considered important. In any case, late scenes of powerlessness in the significant chat services uncover that they won't be robustly actualizing security and protection highlights [4].

In the late years, Data Confidentiality, Authentication, Integrity, Non-repudiation, Access control, and Availability are the most imperative security services in the security criteria that ought to be considered in secure applications and frameworks. Notwithstanding, there is no arrangement for such security services in the mobile chat systems. Both mobile chat system customer and mobile chat system server are defenseless against both passive and active attacks. Passive dangers join arrival of message substance, and Traffic examination while active dangers consolidate adjustment of message substance, masquerade, replay, and denial of service (DoS). Truth be told, all the specified risks are appropriate to the mobile chatting communications [3].

The security and protection saving components of different versatile applications have gone under the spot-light. There are assorted security and protection highlights given by different mobile chat applications, yet there are not very many portable talk applications that give an End-to-End encryption administrations security to their customers [4].

2. RELATED WORKS

There are countless talk applications that claim to give a protected administration, however their total design is not freely accessible.

In 2013 Dec, Ali Makki Sagheer et al, proposed a solution that gives secrecy and uprightness to SMS data by applying a crossbred cryptographic plan which join the AES for encryption/unscrambling plan and RC4 for key extension and generation algorithms to satisfy all the more intense security issues. The proposed model is actualized by Java programming dialect in view of Net Beans platform. The proposed framework was tried on different cell phones, for example, the Nokia 5233.

Our work use Public Key Encryption algorithms that will save the time and cost spent to agree on a key between the users also the encryption time is minimized compered to this paper [5].

In 2014 May, H.C. Chen et al. [6] exhibited another idea about Mobile Text Chat utilizing a revolution session key based transposition cryptosystem plan. Their proposed conspire just manages the safe content transposition for mobile chat framework. It acclimatized the technologies of classical block cipher, substitution and transposition. Also, the new session key can be created by the network pivot innovation. 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 [4].

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 [3].

In 2015 Jan, Pejman Dashtinejad [7], 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.

3. ECDH KEY EXCHANGE

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. The clients generate their keys as following:

ECDH Key Exchange
Goal: generate secure shared key **Input:** EC parameter domain **Output:** secure shared key **S**
Step 1: 1.1. Client **A** chooses secrete random number a < n 1.2. Client **B** chooses secrete random number b < n **Step 2:** 2.1. Client **A** computes $PU_A = a * G$ 2.2. Client **B** computes $PU_B = b * G$ The two parties share their public keys and the common base point G **Step 3:** 3.1. Client **A** compute $S = a * PU_B$ 3.2. Client **B** compute $S = b * PU_A$ **Step 4: Return** (S)

An attacker cannot determine this shared secret key from the curve parameters [8].

4. 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 [9]. The cipher takes a plaintext square size of 128 bits, or 16 bytes. The key length can be 16, 24, or 32 bytes (128, 192, or 256 bits). The calculation is insinuated as AES-128, AES-192, or AES-256, dependent upon the key length [10].

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 [10] [11].

There are four fundamental steps, 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] [12].

5. RC4 ALGORITHM

RC4 is a stream cipher which was organized in 1987 by Ron Rivest for RSA Security. It is a variable key size stream figure with byte situated operations. The algorithm depends on the utilization of an irregular permutation [10]. 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 [13]. 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 [14].

6. 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, which enable the user to perform the chat process with the rest of the users.

6.1 Registration Screen

As shown in the screen shot in Fig. 1 to performs new user Registry process. The registration process involve inserting a new user in the user class at 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.

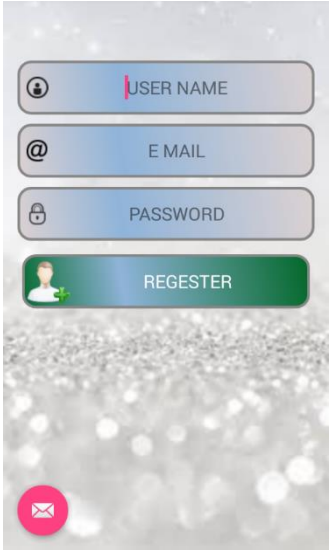

Figure 1: The registration screen

6.2 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 shows list of all the registered users as show in the screen shot in Fig. 2 and informs the user about the state of all other users.

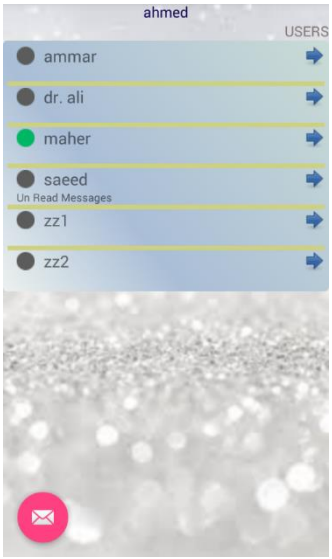

Figure 2: The user list screen

6.3 The Main Chat Screen

The main chat interface consists of a small bar at the top of the mobile screen that shows the user name and the user status, list of conversation, and taskbar at the bottom of the mobile screen as show in the screen shot in Fig. 3 which enables the user to type 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.

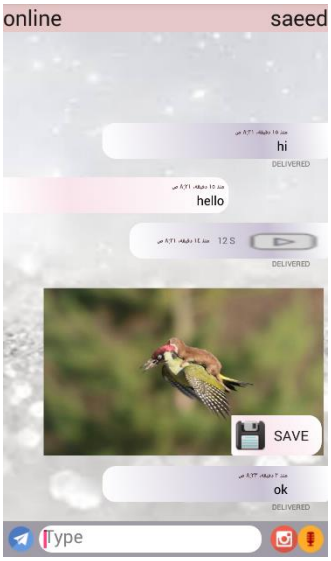

Figure 3: The chat screen

7. THE PROPOSED APPLICATION SECURITY MODEL

The security of the application depends largely on Elliptic Curve Cryptography, and using ECDH algorithm which is a variation of the Diffie-Hellman calculation for elliptic bends. It is really a key-understanding convention, more than an encryption algorithm. ECDH characterizes how keys ought to be produced and exchanged between parties.

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 by selecting the first 128 bit of the generated key. Toward the beginning of the Cipher, the input is copied to the State array utilizing the conventions. 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. 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 calculation utilizes a variable length key from 1 to 256 bytes to shape a 256 byte state table. The state table is utilizes for ensuing era of pseudo-irregular bytes and a short time later to make a pseudo-arbitrary stream which is XORed with the plain data bytes to give the cipher data bytes. Each component in the state table is swapped once in any occasion.

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 changes the state and yields a byte of the key stream. In every cycle, the PRGA increases i, gazes upward the ith component of S, S[i], and adds that to j, trades the estimations of S[i] and S[j], and afterward utilizes the aggregate S[i] + S[j] (modulo 256) as a file to get a third component of S, which is XOR'ed with the following byte of the message to deliver the following 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). The procedure in which the text, voice and image exchanged is illustrate in following algorithms.

Alg. 1: Text message security model
Step 1: The sender type Text Message (TM)
Step 2: TM converted to Bytes Array (BA)
Step 3: Encrypt the BA (EBA): performed by AES with the generated ECDH secure key
Step 4: Convert the EBA to String (ES)
Step 5: Send the ES to the server
Step 6: The recipient receive the ES
Step 7: Convert the received ES to Bytes Array (EBA)
Step 8: Decrypt the EBA (BA)
Step 9: Convert the BA to string which is same the sender message (TM)

Alg. 2: Voice message security model

Step 1: The sender record Voice Message (VM)
Step 2: The VM converted to Bytes Array (BA)
Step 3: Encrypt the converted BA (EBA): performed by RC4 with the generated ECDH secure key
Step 4: Store the EBA to Audio File (AF)
Step 5: Send the AF to the server
Step 6: The recipient receive the AF
Step 7: Extract the EBA from the received AF
Step 8: Decrypt the extracted EBA (BA)
Step 9: Parse the BA to File Output Stream (FOS)
Step 10: Parse the FOS to the Media Player (MP)
Step 11: The recipient now able to play the VM

Alg. 3: Image message security model

Step 1: The sender picks an image to be sent (IM)
Step 2: The IM converted to Bitmap (B)
Step 3: Convert the B to Bytes Array (BA)
Step 4: Encrypt the converted BA (EBA): performed by RC4 with the generated ECDH secure key
Step 5: Store the EBA to Image File (IF)
Step 6: the IF send to the server
Step 7: The recipient receive the IF
Step 8: Extract the EBA from the received IF
Step 9: Decrypt the extracted EBA (BA)
Step 10: Convert the BA to bitmap to be shown to the recipient as IM

8. Results and Discussions

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 test devices

Devise Name	*Android Version*	*RAM*	*CPU*
Galaxy S3 Neo	4.3	1.5 GB	1.2 GHz
Huawei ALE-L21 P8 Lite	5.0.1	2 GB	1.2 GHz
Sony Xperia Z2	6.0.1	3 GB	2.3 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. 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. It is worth mentioning that time encryption / decryption in addition to CPU capabilities and Random Access Memories (RAM) affected by the available memory and the usage of the smartphone as appeared in (Sony Xperia Z2) results. Compared to the results obtained in [5], the result of this system was acceptable even for large blocks of data.

Table 2: Text message encryption/decryption time

Size in Bytes	*Time (ms)*					
	Galaxy S3 Neo		*Huawei P8 Lite*		*Sony Xperia Z2*	
	Enc	*Dec*	*Enc*	*Dec*	*Enc*	*Dec*
32	17	20	19	22	21	24
128	22	24	20	23	23	29
512	30	25	21	24	37	31
2048	34	27	23	26	39	33
4096	43	37	24	27	42	36

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. 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: voice message encryption/decryption time

No	*Time (ms)*					
	Galaxy S3 Neo		*Huawei P8 Lite*		*Sony Xperia Z2*	
	Enc	*Dec*	*Enc*	*Dec*	*Enc*	*Dec*
1	3	2	2	2	3	1
2	7	4	4	4	5	2
3	11	7	7	5	6	3
4	15	11	9	8	10	5
5	29	20	13	10	16	6

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.

Table 6: Image message encryption/decryption time

No	*Time (ms)*					
	Galaxy S3 Neo		*Huawei P8 Lite*		*Sony Xperia Z2*	
	Enc	*Dec*	*Enc*	*Dec*	*Enc*	*Dec*
1	89	74	53	47	124	51
2	163	182	107	103	132	102
3	296	291	168	164	155	161
4	420	399	248	242	171	124
5	463	424	261	257	213	149

9. CONCLUSION

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] Li Zhang, Chao Xu, Parth H. Pathak, and Prasant Mohapatra, "Characterizing Instant Messaging Apps on Smartphones", Passive and Active Measurement Lecture Notes in Computer Science, pp. 83-95, 2015.

[2] Medani1, A. Gani1, O. Zakaria, A. A. Zaidan, and B. B. Zaidan, "Review of mobile short message service security issues and techniques towards the solution", Scientific Research and Essays Vol. 6(6), pp. 1148-1165, March 2011.

[3] 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.

[4] 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.

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

[6] 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.

[7] Pejman Dashtinejad," Security System for Mobile Messaging Applications ", Thesis, KTH University, Jan 2015.

[8] 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.

[9] 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.
[10] William Stallings, "Cryptography and Network Security: Principles and Practice", Prentice Hall, Boston, 5th Ed, 2011.
[11] Joseph Migga Kizza, "A Guide to Computer Network Security", Springer, London, 2nd Ed, 2012.
[12] W. Trappe and L. Washington, "Introduction to Cryptography with Coding Theory", Pearson International, 2nd Ed, 2006.
[13] Bhimrao Patil, "SMS SECURITY USING RC4 & AES", Indian J.Sci.Res, Vol. 11(1), pp. 34-38, 2015.
[14] Meltem Kurt and Nevcihan Duru. "Email Encryption Using RC4 Algorithm", IJCA Vol. 130(14), pp. 25-29, 2015.