JOURNAL OF SOFTWARE ENGINEERING & INTELLIGENT SYSTEMS
ISSN 2518-8739
30th April 2017, Volume 2, Issue 1, JSEIS, CADMEI Copyright © 2016-2017
www.jseis.org

Design of a secure android chatting application using end to end encryption

[1]Ammar H. Ali, [2]Ali Makki Sagheer
[1,2]College of Computer Sciences and Information Technology, University of Anbar, Iraq.
Email: [1]p.a.alfahad@gmail.com [2]ali_makki@computer-college.org

ABSTRACT

Smart phones have become an essential part in the life. The most prominent uses are in chatting and conversation applications. Large number of these applications claim that they are providing security, confidentiality and integrity of user's information. The security and privacy-preserving features of different mobile applications have gone under the spot-light. Yet there are very few mobile chat applications that give an End-to-End security and privacy-preserving service to their clients. In this paper, a secure chatting application with end-to-end encryption for smart phones, that used the android OS, have been proposed. This is achieved by the use of public key cryptography techniques. The proposed application used the ECDH algorithm to generate the key pair and exchanged 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 are used. The generated key (160 bit) is minimized to 128 bit length in order to be used by the AES algorithm. For the voice and image security processes the proposed application used the symmetric algorithm RC4. RC4 provides less security than AES but it performs faster and this is required for such types and sizes of data. The idea behind the RC4 attack is the biases in RC4 key streams. Thus, a new shift function is introduced that eliminated the biases as showed in the results, which guarantees the RC4 algorithm with the modification it can be considered secure.

Keywords: chat application; End-to-End; ECDH; AES; RC4;

1. 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, allow for instant communication to 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 light of the explosion of popular, chatting application. WhatsApp, Telegram, Viber and different applications all offer free text messaging. What's more, that is not withstanding 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.

There are a large number of mobile chat applications that claim to provide a secure service, their complete architecture is not publicly available. To our best knowledge there are not many publications that describe such systems. Sagheer et al., (2013) proposed a solution that provides confidentiality and integrity for SMS data by applying a crossbred cryptographic scheme which joins 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]. Chen et al. (2014) exhibited another idea about Mobile Text Chat utilizing a revolution session key based transposition cryptosystem plan [2]. 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. Akram et al., (2014) evaluated the security and privacy preserving features introduced in the current mobile chat services. They additionally put advances in 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 of the technical difficulty of satisfying the specified security and privacy requirements [3]. Chen et al., (2014) 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]. Dashtinejad (2015) investigate current security features of common messaging applications in the mobile market [5]. A list of requirements for acceptable security is generated and based on these requirements an architecture is developed. A demo is also implemented and evaluated.

2. ELLIPTIC CURVE CRYPTOGRAPHY

For the similar grade of security that public key cryptosystems, for instance RSA have, elliptic curve cryptography (ECC) offers the advantage of littler key sizes, henceforth little memory and processor requirements. The Diffie Hellman key exchange, ElGamal encryption and the Digital Signature Algorithm (DSA) would all be able to be executed in ECC. This makes ECC an extremely alluring algorithm for wireless devices, for example, handhelds and PDAs, which have restricted transmission capacity and processing power. Running on the same platform, ECC runs more TLS/SSL transactions per second than RSA [6]. ECC has been around since the mid-1980s. ECC gives the same level of security as RSA or discrete logarithm systems give with extensively shorter operands (roughly 160–256 bit versus 1024–3072 bit). ECC based on the generalized DLP [7].

3. ELLIPTIC CURVE DIFFIE–HELLMAN

In the elliptic curve Diffie-Hellman (ECDH) key exchange, the two communicating clients $client_A$ 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 [8].

3.1 Elliptic curve cryptography domain parameters

The public key cryptographic systems include arithmetic operations on Elliptic curve over finite fields which is dictated by elliptic curve domain parameters. The ECC domain parameters over F_q is characterized by the D = (q, a, b, G, n, h), where:

1. q: prime power, that is q = p or $q = 2^m$, where p is a prime.
2. a, b: field components, they determine the equation of the elliptic curve E over F_q, $y^2 = x^3 + ax + b$.
3. G: A base point symbolized by G= (x_g, y_g) on E (F_q).
4. n: Order of point G, that is n is the littlest positive integer such that nG = O.
5. h: cofactor, and is equal to the proportion $\#E(F_q)/n$, where $\#E(F_q)$ is the curve order.

It should be noted that the public key generated needs to be validated to ensure that it satisfies the arithmetic requirement of elliptic curve public key [9].

3.2 Key generation

The clients public and private keys are associated with a particular set of elliptic key domain parameters (q, a, b, G, n, h). The clients generate their keys as following:

Step 1:

1. Client A chooses secrete random number a < n
2. Client B chooses secrete random number b < n

Step 2:

1. Client A computes PUA = a * G
2. Client B computes PUB = b * G

The two parties share their public keys and the common base point G

Step 3:

1. Client A compute S = a * PUB
2. Client B compute S = b * PUA

Step 4: Return (S)

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 [10]. 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 [11]. 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 expanded into an array of key schedule words. Every word is four bytes, and the aggregate key schedule is 44 words for the 128-bit key. The cipher comprises of N rounds, where the quantity of rounds relies on 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 [11, 12]. There are four fundamental strides, called layers that are utilized to form the rounds:

1. The Byte Sub Transformation (B S): 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 (S R): A straightforward permutation. This linear blending venture causes diffusion of the bits over multiple rounds.
3. The Mix Column Transformation (M C): A substitution that makes use of arithmetic over GF (28). This layer has a purpose similar to Shift Row.
4. Add Round Key (A R K): A simple bitwise XOR of the current block with a portion of the expanded key. The round key is XORed with the result of the above layer [11, 13].

5. RC4 ALGORITHM

RC4 is a stream cipher planned 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 [11]. 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 [14]. 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 [15]. The RC4 utilized heaps of standards and protocols for instance in the SSL/TLS standards that have been characterized for correspondence between Web programs and servers. It additionally utilized as a part of WEP protocol and WPA protocol [15]. Various papers have been distributed analyzing techniques for assaulting RC4. None of these approaches is practical against RC4 with a sensible key length, for example, 128 bits [11].

6. THE PROPOSED APPLICATION SECURITY MODEL

The security of the application depends largely on Elliptic Curve Cryptography. After the generation of the key pairs these keys 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 input key length differs from the generated key, the generated key submitted in key scheduling algorithm (KSA) in order to be in a suitable length form.

The application consists of a set of interface design, the user moves between them to perform the chat process with the rest of the users.

a) Registration screen

As shown in Figure 1 to hold new user Registry process. The registration process consists of insert a new user in the user class on the server, this is the easy side. This process involves recording the fixed information of the user which is the user name, e-mail in addition to the password. But the difficult side, the new user registration process is very complicated, as in the server a special class was created to contain changing user information, such as a user's status, whether online or offline also contains 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 is done and informs the user whether there are unread messages, and also used to indicate the status of other users.

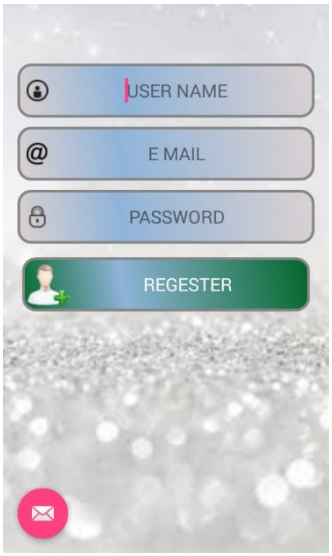

Figure. 1 The registration screen

b) The user list screen

During 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 and private key stored in the phone and the public key is submitted to the server. List of user's interface contains all the registered users of the application as shown in Figure 2. And provide the user with information about all other users such as, whether online or offline. Moreover, informs the user when there is unread messages sent to him by another user.

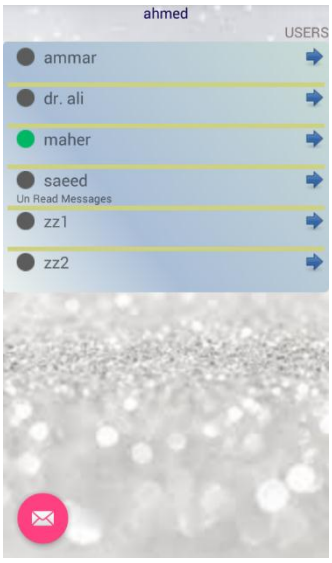

Figure. 2 The user list screen

c) 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 Figure 3. Which enables the user to write a text message, make voice record or open gallery to select image to be transmitted. The message that exchanged between the users are stored in the created prochat class. Each message stored in encrypted form with its own information. The information

is the sender, the recipient, the type, the date and other. This information is used in the queries by which the message is retrieved in the correct form and sequence.

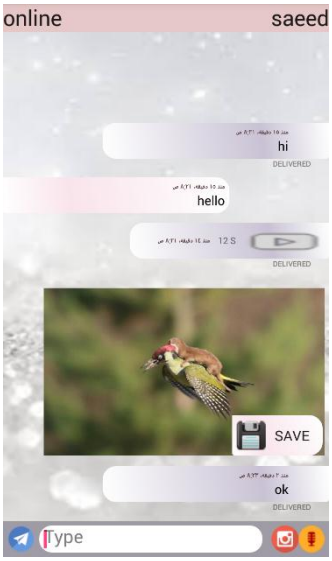

Figure. 3 The main chat screen

7. THE RESULTS

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
Samsung GT – S7272 Galaxy Ace 3	4.2.2	1 GB	1 GHz
Samsung GT - I9300I Galaxy S3 Neo	4.3	1.5 GB	1.2 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
Sony Xperia Z2	6.0.1	3 GB	2.3 GHz

The results of encrypting and decrypting pieces of text messages are shown in Table 2. The results are in terms of execution time in milliseconds.

Table. 2 Performance time metrics of the text message

Size Byte	Time (ms)									
	Galaxy Ace 3		Galaxy S3 Neo		Huawei P8 Lite		Galaxy S6 Duos		Sony Xperia Z2	
	Enc	Dec	Enc	Dec	Enc	Dec	Enc	Dec	Enc	Dec
32	60	63	17	20	19	22	7	10	21	24
128	63	68	22	24	20	23	8	11	23	29
512	68	72	30	25	21	24	14	12	37	31
2048	77	84	34	27	23	26	15	13	39	33
4096	90	92	43	37	24	27	20	13	42	36

Table 3 shows the results that consist of the NBCR refer to Number of Bytes Change Ratio and the time of voice encryption and decryption processes in milliseconds.

Table. 3 Performance time metrics of the voice message

Voice Size In Kb	NBCR	Time (ms)									
		Galaxy Ace 3		Galaxy S3 Neo		Huawei P8 Lite		Galaxy S6 Duos		Sony Xperia Z2	
		Enc	Dec	Enc	Dec	Enc	Dec	Enc	Dec	Enc	Dec
16	99.59	3	3	3	2	2	2	1	1	3	1
31	99.57	6	6	7	4	4	4	1	2	5	2
48	99.56	10	9	11	7	7	5	3	1	6	3
71	99.64	18	14	15	11	9	8	5	2	10	5
95	99.60	23	18	29	20	13	10	6	4	16	6

Table 4 shows the time of images encryption and decryption processes in milliseconds.

Table 4. Performance time metrics of the image message

Image Size In Kb	Time (ms)									
	Galaxy Ace 3		Galaxy S3 Neo		Huawei P8 Lite		Galaxy S6 Duos		Sony Xperia Z2	
	Enc	Dec	Enc	Dec	Enc	Dec	Enc	Dec	Enc	Dec
26	58	57	89	74	53	47	69	28	124	51
66	125	118	163	182	107	103	112	54	132	102
118	185	184	296	291	168	164	130	95	155	161
181	266	269	420	399	248	242	167	101	171	124
220	287	284	463	424	261	257	201	138	213	149

8. CONCLUSIONS

In this paper, a secure chatting application was developed. The system was tested on different mobile devices. According to the obtained results the following are summarized as conclusions. End to End Encryption achieved by involving ECDH key exchange to provide the key pair (private and public) 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 application for a secure mobile chat application provides confidentiality, privacy and integrity. Users can be granted that nobody, even not the provider of the service, can read their messages. The exchanged data store only at the server, and nothing of them store 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 vast amounts of data.

ACKNOWLEDGEMENT

Authors give many thanks to the College of Computer Sciences and Information Technology, University of Anbar. Moreover, special thanks to the project supervisor Dr. Ali M. Sagheer for supporting us in this research.

REFERENCES

1. Sagheer, A.M., A.A. Abdulhameed, and M.A. AbdulJabbar. *SMS Security for Smartphone*. in *Developments in eSystems Engineering (DeSE), 2013 Sixth International Conference on*. 2013: IEEE.
2. Chen, H.-C. and A.L.V. Epa. *A rotation session key-based transposition cryptosystem scheme applied to mobile text chatting*. in *Advanced Information Networking and Applications (AINA), 2014 IEEE 28th International Conference on*. 2014: IEEE.
3. Akram, R.N. and R.K. Ko. *End-to-end secure and privacy preserving mobile chat application*. in *IFIP International Workshop on Information Security Theory and Practice*. 2014: Springer.
4. Chen, H.-C., J.-H. Wen, and C.-Y. Yang. *A secure end-to-end mobile chat scheme*. in *Broadband and Wireless Computing, Communication and Applications (BWCCA), 2014 Ninth International Conference on*. 2014: IEEE.

5. Dashtinejad, P., *Security System for Mobile Messaging Applications*. 2015.
6. Mogollon, M., *Cryptography and Security Services: Mechanisms and Applications: Mechanisms and Applications*. 2008: IGI Global.
7. Paar, C. and J. Pelzl, *Understanding cryptography: a textbook for students and practitioners*. 2009: Springer Science & Business Media.
8. Kumar, S., et al. *Embedded end-to-end wireless security with ECDH key exchange*. in *Circuits and Systems, 2003 IEEE 46th Midwest Symposium on*. 2003: IEEE.
9. Kumar, R. and A. Anil, *Implementation of elliptical curve cryptography.* IJCSI International Journal of Computer Science Issues, 2011. **8**(4).
10. Tayde, S. and S. Siledar, *File Encryption, Decryption Using AES Algorithm in Android Phone.* International Journel of Advanced Research in computer science and software engineering, 2015. **5**(5).
11. Stallings, W., *Cryptography and network security: principles and practices*. 2006: Pearson Education India.
12. Kizza, J.M., *Guide to computer network security*. 2009: Springer.
13. Trappe, W., et al., *Introduction to cryptography with coding theory.* The Mathematical Intelligencer, 2007. **29**(3): p. 66-69.
14. PATIL, B., *SMS SECURITY USING RC4 & AES.* Indian J. Sci. Res, 2015. **11**(1): p. 034-038.
15. Kurt, M. and N. Duru, *Email Encryption using RC4 Algorithm.* International Journal of Computer Applications, 2015. **130**(14): p. 25-29.

AUTHORS PROFILE

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.