Journal of Engineering and Applied Sciences 12 (1): 156-160, 2017
ISSN: 1816-949X

Design and Implementation of Secure Chatting Application with End to End Encryption

[1]A.H. Ali and [2]A. Makki Sagheer
[1]Department of Computer Science,
[2]Department of Information System,
College of Computer Science and Information Technology, University of Anbar, Anbar, Iraq

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 encryption service to their clients. Due to that, there is a need for secure chatting application with end to end encryption that provide complete security and privacy for the user. In this study, 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, AES for text encryption and RC4 for voice and image encryption.

Key words: Android, chatting application, ECDH, AES, RC4

INTRODUCTION

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 few publications that describe such systems (Akram and Ko, 2014; Chen and Epa, 2014; Chen *et al*., 2014; Dashtinejad, 2015; Sagheer *et al*., 2013).

Mobile phones are presently a portion of human lives and have developed from mere communication gadgets to getting to be cameras, coordinators, texting devices and address books. The rising of mobile applications are the principle strengths behind the expanding needs of mobile systems (Hasan *et al*., 2010).

Web based instant message applications permit clients to send/get messages over the web. It requires internet connection with exchange messages starting with one device then onto the next device. There are different applications like BBM (Black Berry Messenger), Ping Chat, Imo and so on (Mehrotra *et al*., 2014).

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 >17 billion SMS messages (Zhang *et al*., 2015).

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, IM 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 (Medani *et al*., 2011).

Cell phones vulnerable to dangers and the dangers connected with the technology. The dangers can be with wireless technology, software, hardware and inside the devices itself (Bartz, 2015). For any computing device that contains touchy data and gets to the internet, security is a noteworthy issue. In the 1980s, security issues were hardly noticed however, security is a noteworthy issue for clients today which incorporates cell phones (Dwivedi, 2010).

MATERIALS AND METHODS

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 users can communicate using text messages, voice messages and photos.

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,

Corresponding Author: A.H. Ali, Department of Computer Science, College of Computer Science and Information Technology, University of Anbar, Anbar, Iraq

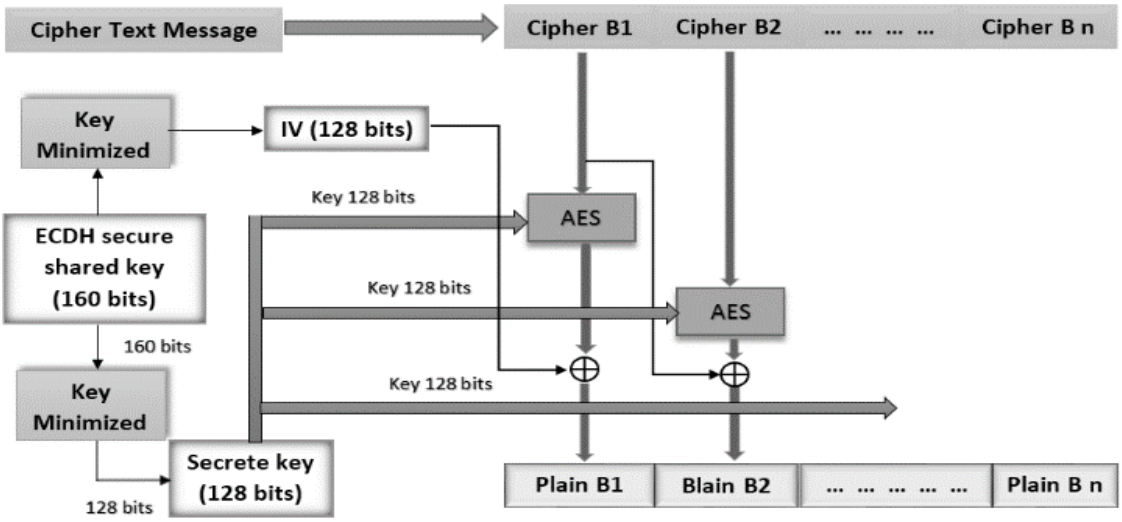

Fig. 1: Structure of the text encryption procedure

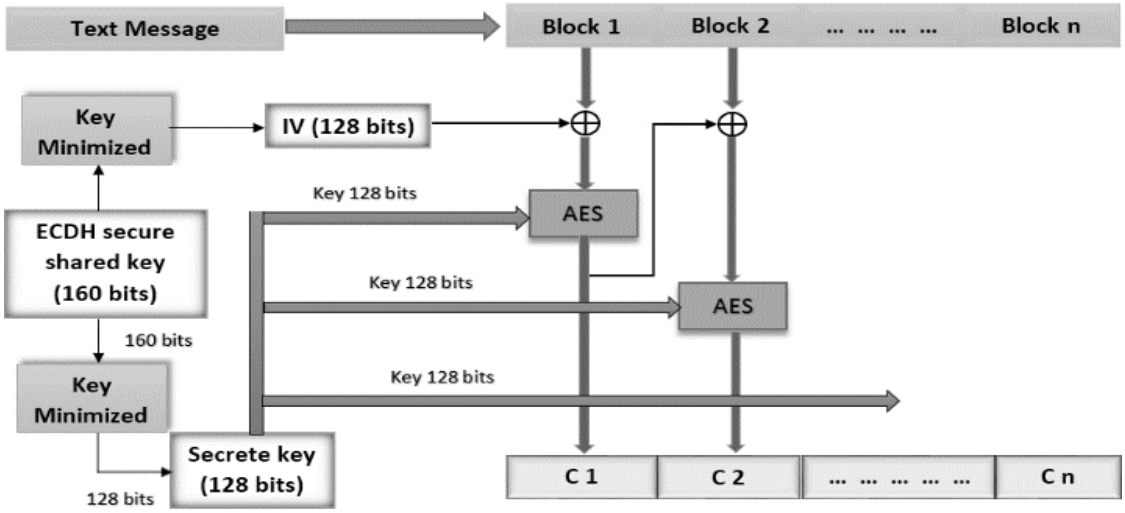

Fig. 2: Structure of the text decryption procedure

more than an encryption algorithm. This essentially implies 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. Figure 1 shows the implemented text encryption model. 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 (Fig. 2). Structure of the text decryption procedure bit key length. With the final

round differing slightly from the first Nr-1 rounds, All Nr 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. Figure 2 shows the implemented text encryption model.

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 security 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:

- A permutation of all 256 possible bytes (S)
- 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 1-256 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 (the key stream value K below) 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). The procedure in which the text, voice and image exchanged is illustrate in following algorithms.

Algorithm 1: Text message 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)

Algorithm 2: Voice message 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

Algorithm 3: Image message model
Step 1: The sender pick Image to by send (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

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.

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

Table 1: Specifications of the test devices

Devise name	Android version	RAM (GB)	CPU (GHz)
Galaxy S3 Neo	4.30	1.5	1.2
Huawei ALE-L21 P8 Lite	5.0.1	2.0	1.2
Sony Xperia Z2	6.0.1	3.0	2.3

Table 2: Performance time metrics of the text message encryption/decryption

	Time (ms)					
	Galaxy S3 Neo		Huawei P8 Lite		Sony Xperia Z2	
Size (bytes)	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: The voice message duration and size

Duration (sec)	Size (kb)
10	16
20	31
30	48
45	71
60	95

Table 4: Performance time metrics of the voice message encryption/decryption

	Time (ms)					
	Galaxy S3 Neo		Huawei P8 Lite		Sony Xperia Z2	
NBCR	Enc	Dec	Enc	Dec	Enc	Dec
99.59	3	2	2	2	3	1
99.57	7	4	4	4	5	2
99.56	11	7	7	5	6	3
99.64	15	11	9	8	10	5
99.60	29	20	13	10	16	6

Table 5: The image message size, NPCR and UACI

Size (kb)	NPCR	UACI
26	99.59	33.986
66	99.62	29.135
118	99.61	32.694
181	99.60	29.887
220	99.62	32.616

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 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 4 shows the results that consist of the NBCR which refers to Number of Bytes Change Ratio and 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 6: Performance time metrics of the image message encryption/decryption

Time (ms)					
Galaxy S3 Neo		Huawei P8 Lite		Sony Xperia Z2	
Enc	Dec	Enc	Dec	Enc	Dec
89	74	53	47	124	51
163	182	107	103	132	102
296	291	168	164	155	161
420	399	248	242	171	124
463	424	261	257	213	149

Table 5 shows the examined image size, NPCR and UACI that are used to appreciate the goodness of picture encryption. The NPCR and UACI are intended to test the quantity of changing pixels and the quantity of averaged changed intensity between ciphertext pictures.

The proposed application allows transfer images that have size <250 kb. So, the tested images have the allowed size only. Table 6 shows the time of images encryption and decryption processes in millisecond.

CONCLUSION

In this study, 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.

ACKNOWLEDGEMENTS

Researchers 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

Akram, R.N. and R.K. Ko, 2014. End-to-End Secure and Privacy Preserving Mobile Chat Application. In: Information Security Theory and Practice, Naccache, D. and D. Sauveron (Eds.). Springer, Berlin, Germany, ISBN:978-3-662-43825-1, pp: 124-139.

Bartz, R.J., 2015. Mobile Computing Deployment and Management. John Wiley & Sons, Hoboken, New Jersey,.

Chen, H.C. and A.L.V. Epa, 2014. A rotation session key-based transposition cryptosystem scheme applied to mobile text chatting. Proceeding of the 2014 IEEE 28th International Conference on Advanced Information Networking and Applications, May 13-16, 2014, IEEE, New York, USA., ISBN:978-1-4799-3630-4, pp: 497-503.

Chen, H.C., J.H. Wen and C.Y. Yang, 2014. A secure end-to-end mobile chat scheme. Proceeding of the 2014 Ninth International Conference on Broadband and Wireless Computing, Communication and Applications (BWCCA), November 8-10, 2014, IEEE, Taichung, Taiwan, ISBN:978-1-4799-4173-5, pp: 472-477.

Dashtinejad, P., 2015. Security system for mobile messaging applications. Master Thesis, Royal Institute of Technology, Stockholm, Sweden.

Dwivedi, H., 2010. Mobile Application Security. Tata McGraw-Hill Education, New York, USA., ISBN-13: 9780070701922, Pages: 444.

Hasan, M.H., N.S. Haron and N.S.S.M. Yazid, 2010. Development of multimedia messaging service (MMS)-based examination results system. Proceedings of the 9th WSEAS International Conference on Applications of Computer Engineering, March 23-25, 2010, World Scientific and Engineering Academy and Society, Stevens Point, Wisconsin, ISBN:978-960-474-166-3, pp: 157-163.

Medani, A., A. Gani, O. Zakaria, A.A. Zaidan and B.B. Zaidan, 2011. Review of mobile short message service security issues and techniques towards the solution. Sci. Res. Essays, 6: 1147-1165.

Mehrotra, P., T. Pradhan and P. Jain, 2014. Instant messaging service on android smartphones and personal computers. Int. J. Inf. Comput. Technol., 4: 265-272.

Sagheer, A.M., A.A. Abdulhameed and M.A. AbdulJabbar, 2013. SMS security for smartphone. Proceeding of the 2013 6th International Conference on Developments in E-Systems Engineering (DeSE), December 16-18, 2013, IEEE, Ramadi, Iraq, ISBN:978-1-4799-5264-9, pp: 281-285.

Zhang, L., C. Xu, P.H. Pathak and P. Mohapatra, 2015. Characterizing Instant Messaging Apps on Smartphones. In: Passive and Active Network Measurement, Jelena, M. and L. Yong (Eds.). Springer, Berlin, Germany, ISBN:978-3-319-15508-1, pp: 83-95.