www.ijecs.in

International Journal Of Engineering And Computer Science ISSN:2319-7242
Volume 6 Issue 9 September 2017, Page No. 22480-22485
Index Copernicus value (2015): 58.10 DOI: 10.18535/ijecs/v6i9.15

A Secure Framework for Messaging on Android Devices with Honey Encryption

Rasmita Sahu[1], Shajid Ansari[2]

[1]M.Tech Scholar, Department of CSE, RSR-RCET, Bhilai,
Chhattisgarh, INDIA
[2]Assistant Professor, HOD, Department of CSE, RSR-RCET, Bhilai,
Chhattisgarh, INDIA

Abstract:
Nowadays, there are various encryption techniques used to secure the messaging services before send it through the unsecure network. But still the attackers hack the message by using various hacking methods. So for providing more security and to avoid the Brute force type attack, we provide a best encryption technique with encoding scheme. The proposed approach aim to develop a secure messaging framework for Android based devices using Honey Encryption with Symmetric key cryptography techniques, for preventing the text based messages from brute face attacks in order to make messaging communication more secure and efficient. In this work we combine the Honey encoding with both AES and Blowfish separately and then compare the performance of both the algorithm. Experiments show that Blowfish algorithm when combined with Honey Encryption scheme produces result in less time.

Keywords: Messaging Communication, Honey Encryption (HE), Blowfish, AES, Distribution-Transforming Encoder (DTE), Cumulative Massive Function (CMF).

1. Introduction

The popularity of messaging services is increasing day by day as it is being used in many data centric applications including railways enquiry, news alert, mobile banking, and health care applications. User sends confidential information using messaging services .The contents of messaging are stored in messaging centre and it is visible to the network provider Staff it can modify he contain of the message and therefore, Messaging is not an appropriate communication medium for secure communications. A hacker can easily hack the messaging centre and read the message contain [6]. Message security has at least four security constraints to meet, as listed below:

Confidentiality: It prevents the unauthorized user to assess the private information. The encryption techniques are used to provide confidentiality to the message. [3]

Integrity: It is preventing anybody other that authorized parties from modifying the computer system assets like writing, changing status and deleting and creating files.

There are two main ways to encrypt the information: Secret Key Encryption Systems and Public Key Encryption Systems.

Secret Key Encryption: A single key is used for encryption as well as decryption process. For this system, secret key must be present towards all the parties involved in the communication. If attacker gets the key, then he/she will be able to decrypt and read the messages. Examples are DES, AES and Blow Fish etc.

DES is a type of block cipher. DES maintains confidentiality of information to a large extent. It is generally useful for authentication purpose. Weakness of DES is that it has Short Length Key. Keys of DES can be easily broken so it is unsecure [9].

AES is highly secure. It is mostly used for encrypt the messages during message communication. But AES requires more processing as compared to others. Configuration of AES is complex.[7]

Blow Fish is also a type of block cipher. It is not patented by any authority or organization. That's why it is licensing free [7]. It is energy efficient and has fast execution time. In case of blow fish, it is not sufficient for large documents. It has long initialization time period so it is time consuming.

Public Key Encryption: Public Key Encryption System uses a pair of keys. Both keys can decrypt the message encrypted by other. One of the keys must be kept secret. Communication is encrypted using public key and it can be decrypted using the private key with the receiver. So the communication is secure. It is infeasible to derive private key from public key because of algorithmic properties.[7]

Attacks in Messaging Communication:

The message travelling in the network is prone to various types of attacks. Attacks are of following type:

1. **Brute Force Attack**
   Brute Force Attack is a password -guessing attack, which is a common threat faced by the web users. In this attack the attacker determine a password by trying every possible combination of letters, symbols, and digits until he discover the correct combination that works. Brute force attack is also known as dictionary attack because, the attacker guess the password by using dictionary words instead of a random number or string.[1] Because most people use dictionary words instead of random words as a password. So the attackers mostly use some tools or software's that use wordlists and smart rule sets to intelligently guess the password [4].

2. **Impersonation Attack**
   Impersonation attack is an active attack in which attacker acts as the character of authenticated user. Attacker can get the confidential information by various methods. Attackers rely on shoulder surfing to get the passwords by just reading a password that is typed or monitoring the activity using a camera. Dictionary attack also helps to get the password of the user. Some attackers torture or force the user to extract the password. Sometimes attacker attack on the computer and monitors the history and saved passwords to gain access to the user's account. [10]

3. **Man in middle attack**- In Man in middle attack, attacker intercepts a connection between client and server. It then retransmits those messages substituting his/her own public key. Attacker acts as a proxy who is being able to intercept, add, delete or modify the messages. In this attack, Users, who are communicating feels that they are talking to each other but in the middle eavesdropper is intercepting their messages and may be modifying those or inserting new messages [4].

4. **Side channel attack**-In Side channel attack the attacker analyses information obtained from the physical implementation of cryptosystem. For instance, timing data, power utilization, electromagnetic releases or even sound can give an additional wellspring of data, which can be misused to break the framework. Technical knowledge of the internal operation of system is necessary for side channel attack. Many Side channel attacks are view of factual techniques spearheaded by Paul Kocher [10].

2. Problem Identification

I have studied various research papers based on the messaging security and found various problems and security threats in messaging communication. We need to solve them and improve the existing systems performance.

The major problems identified in the traditional approaches are as follow:-

1. **A single level security**
   Mostly in existing system a single level security is provided to the messaging applications using only a single level encryption algorithm. So the attacker can easily hack the message by guessing the key. So more security is needed.
2. **Desktop Implementation**
   Existing Honey encoding approach is implemented on desktop based messaging system, so we need to implement the approach in android based devices for messaging security against Brute force Attack.
3. **Time Consuming**
   AES encryption technique used with Honey encoding needs more processing time to produce result due to its complex rounds. So it is not convenient to use AES for messaging on Android based mobile because we need fast response.

3. Background

3.1 Honey Encryption

Honey Encryption (HE) is an interesting and useful encryption technique which is developed by Ari Jules and Thomas Ristenpart of the University of Wisconsin.

They presented a paper on this technique at the 2014 Eurocrypt cryptography conference [2]. Honey encryption produce an cipher text which when decrypted by any wrong key by the brute force attacker, then it looks like meaning full fake plain text which is not the original message. So this encryption provides security against brute force attack. We can say that HE yield a cipher text, which, when decrypted with an wrong key as guessed by the attacker, presents a plausible-looking fake or incorrect plaintext password or encryption key. Honey Encryption is actually an encoding and decoding scheme which is used together with encryption/decryption scheme. HE is initially introduced in the cryptography conference, but it focuses on encoding and decoding scheme. It is used with the conventional encryption technology, and the main principle is to make it difficult to distinguish a true output message from other fake output messages [1].

Operation of Honey Encryption

In order to recover the drawback of the traditional password based encryption (PBE) with low-entropy passwords, Juels and Ristenpart introduced Honey Encryption (HE) [2]. The main idea is that encryption of plaintext M is randomized with a password kp, and decryption of cipher text results in plausible-looking plaintext M´ with wrong password kp’. A distribution-transforming encoder (DTE) is used for encoding and decoding of message as bit string, denoted DTE = (encode, decode). In brief, overall process is HE[DTE,Sym.E] = (HEnc,HDec) where Sym.E means conventional symmetric encryption. The cipher text is C = HEnc(kp;M) and decryption works M = HDec(kp;C) .

3.2 Distribution Transforming Encoder (DTE)

Suppose p be a probability distribution over the original message space MS, meaning that a user selects M belongs to MS for encryption with probability p (M). A DTE encoder encodes M as an n-bit seed S belongs to {0, 1} power n or we can say that S ϵ $\{0, 1\}^n$. That is the original message is encoded into seed space S. One message may be represented by many seeds belongs to S from which the encoder selects one such seed uniformly at random.[1] (Every seed, however, corresponds to a unique message.). At the receiver side, inverse process of encoding is happen for generating the plain text from the corresponding seed value, which is known as decoding. In other words, given S, we can decode through the inverse DTE decode(S) = M, which returns seed space *S’s* unique corresponding message. With a DTE that gives strong security (as we explain later), decode accurately generates p. In this case, selecting S uniformly at random from $\{0, 1\}^n$. and decoding to obtain M = decode(S) returns approximately the original M. In other words, the DTE is a good model of the message distribution [2].

3.3 AES

AES is a symmetric cipher which is widely used for data encryption and decryption. It is six times faster than triple DES. AES is a block cipher and take 128 bit block of input data for encryption and decryption. In this algorithm mainly 3 different size keys used for encryption so it comprises of three block ciphers: AES-128, AES-192 and AES-256. Each cipher

encrypts and decrypts data in blocks of 128 bits using keys of 128-, 192- and 256-bits, respectively. AES is a secret key algorithm so it uses same key for encryption and decryption.[6]

Working:

First the plain text block is put into an array then the data is processed in no of rounds repeatedly to transform it. The number of rounds is decided by the key size. for key length 128 bit, 192 bit and 256 bit it takes 10 rounds, 12 rounds and 14 rounds respectively.

3.4 Blowfish Algorithm

Blowfish is a symmetric block encryption algorithm designed in 1993 by Bruce Schneier, a fast, alternative to existing encryption algorithms such as AES, DES and 3 DES etc. It takes a variable-length key, from 32 bits to 448 bits and a block size of 64 bits. Blowfish is unpatented and license-free.Blowfish takes less time to encrypt data so it is faster algorithm as compare to AES, IDEA, DES etc. It encrypts data at a rate of 26 clock cycles per byte. on large 32-bit microprocessors. It uses simple operations like addition, XOR, lookup table with 32-bit operands. It takes less memory that is, less than 5K of memory. It is a feistel network algorithm, which consists of 16 rounds with input is 64-bit elements, [5]

4. Proposed Methodology

The methodology will consist of following phases:-

Phase 1: Registration

1. Any user participating in message based communication first registers by providing credentials.
2. The IMEI no of the device being registered is stored on the server alongside other credentials.
3. The registered user is provided an appropriate key which is entered by user as an authentication procedure before communication

Phase 2: Secure Message communication using Honey Encryption.

1. The message is in form of plain text. This text is encoded and decoded as bit string using distribution-transforming encoder (DTE).
2. In the current case the entropy of message is high. Entropy is the measure of the uncertainty of the information in the information theory. Because the message is of variable length the entropy is high.
3. In the current process the messages are encoded using Statistical coding scheme (SCS).In the scheme a Cumulative Massive Function (CMF) is used as code table and the equation to build CMF of nth character of message is :

$$f_{cmf}(c_k) = \sum_{k=0}^{S} \frac{p(x_i=c_k \,|X_{i-1:i-n})}{\sum_{j=0}^{S} p(x_i=c_j|X_{i-1:i-n})} (1)$$

   In the above equation S is the possible number of character set in the code table and n is the order of Markov process. In the equation $x_i = c_j|X_{i-1:i-n}$ indicates that the i-th character of message is influenced by near n-1 characters. The probability of each character is calculated and adjusted according to the frequency of appearance.
4. A code table constructed according to SCS is shared between sender and receiver for encoding and decoding of messages.
5. The encoded /decoded message is again encrypted using Blowfish encryption algorithm. The public key used for encryption/decryption process is shared between sender and receiver.

Phase 3: Prevention of Brute force attack

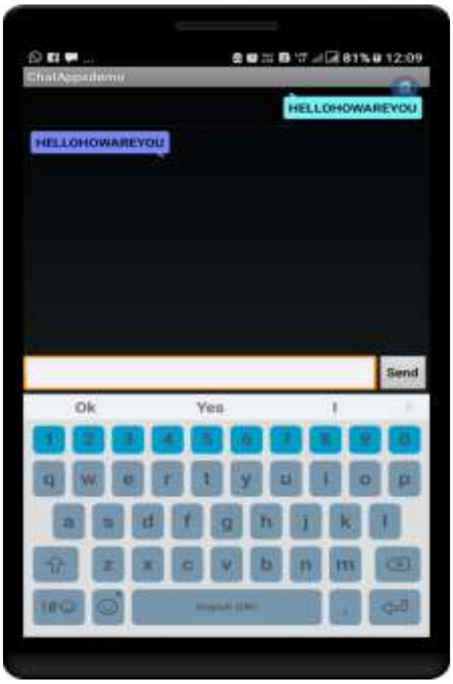
1. The N-Gram model is used to generate plausible looking text in case of a Brute Force attack.
2. If an unauthorized IMEI number device is detected by the server during communication process. For every unauthorized authentication attempt an N-Gram string is generated thus confusing the hacker.
3. The N-Gram is constructed using OpenNLP tools and a combination of antonyms and synonyms for the words in messages. Any punctuation word is replaced by an alternative from the word in a predefined dataset. Also a noun word if present is replaced by antonyms or synonyms.

5. Proposed Messaging Application in Android

For implementing the proposed methodology, we use the tool Android Studio and the language Java to code the concept. First the input plain text is Encoded using Honey encoding technique to produce Encoded text. Then the encoded text is encrypted by using secret key algorithm then a cipher text is produced at the sender side. At the receiver side, the opposite process is happen, that is the received cipher text is first decrypted by using the shared secret key and the same symmetric cipher to produce the encoded text. Then the encoded text is decoded by using the Honey Decoding process to produce the original message. At the Encoding phase, Distribution Transforming Encoders (DTE) is used to produce plausible looking intermediate text. Here for DTE we use N-gram model concept. The N-Grams are generated using Opennlp-tools .The class used is NGramGenerator class that generates an nGram, with optional separator, and returns the grams as a list of n-Gram strings.

6. Experimental Analysis

Our experiment is a simulated approach of messaging system, where a single android device is treated as both sender and receiver. The following figure 1 shows the output screen of messaging.



HELLOHOWAREYOU

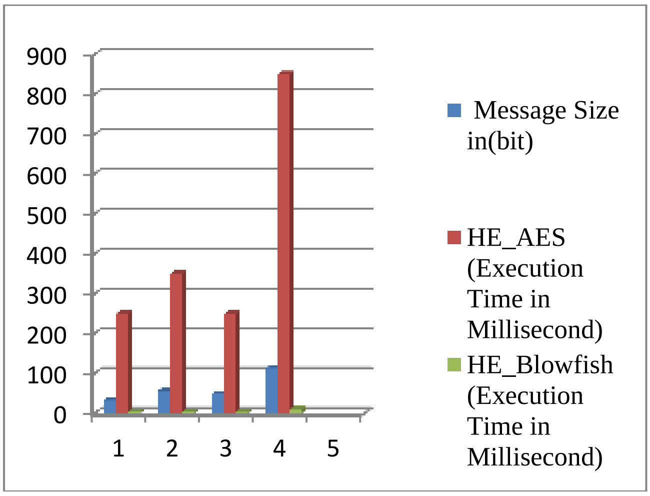
HELLOHOWAREYOU

Figure 1: Messaging Screen

Here when we send a message, the Honey encoding and encryption process are happen at sender side. We first apply AES algorithm with honey encoding and calculate the processing time and then apply Blowfish algorithm with the honey encoding and calculate the processing time for the same input message. Here we see that Blowfish with HE takes less time as compare to AES with HE. So we can take the second approach Blowfish with HE for secure messaging communication in android devices. So problems like Brute force attack and more processing time can be reduced by Honey encoding and Blowfish respectively.

Message Size in(bit)	HE_AES (Execution Time in Millisecond)	HE_Blowfish (Execution Time in Millisecond)
32	250	3
56	350	3
48	250	2
112	850	10

Table 1: Execution Time Example



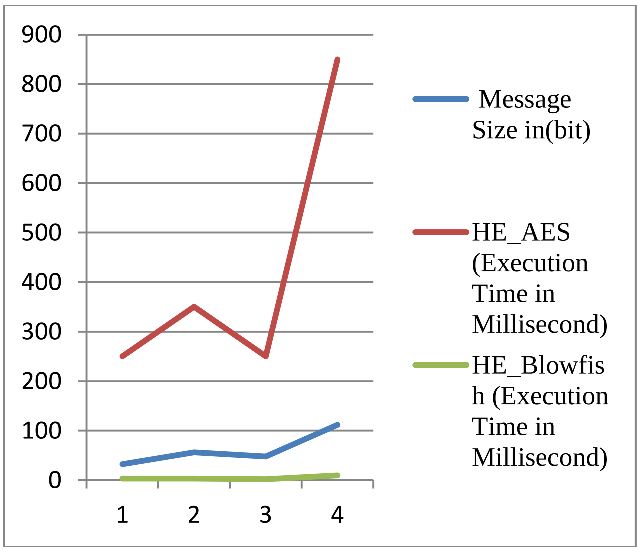


Figure 2 : Execution Time Chart with HE

The following output screen shows our implementation on Android Studio in a simulated environment. Here we develop a messaging application using our proposed methodology and follow the steps for encoding and encryption. When we execute the application on an Android device just like a smart phone, then the messaging screen will appear to the user. When the user type a message and send then

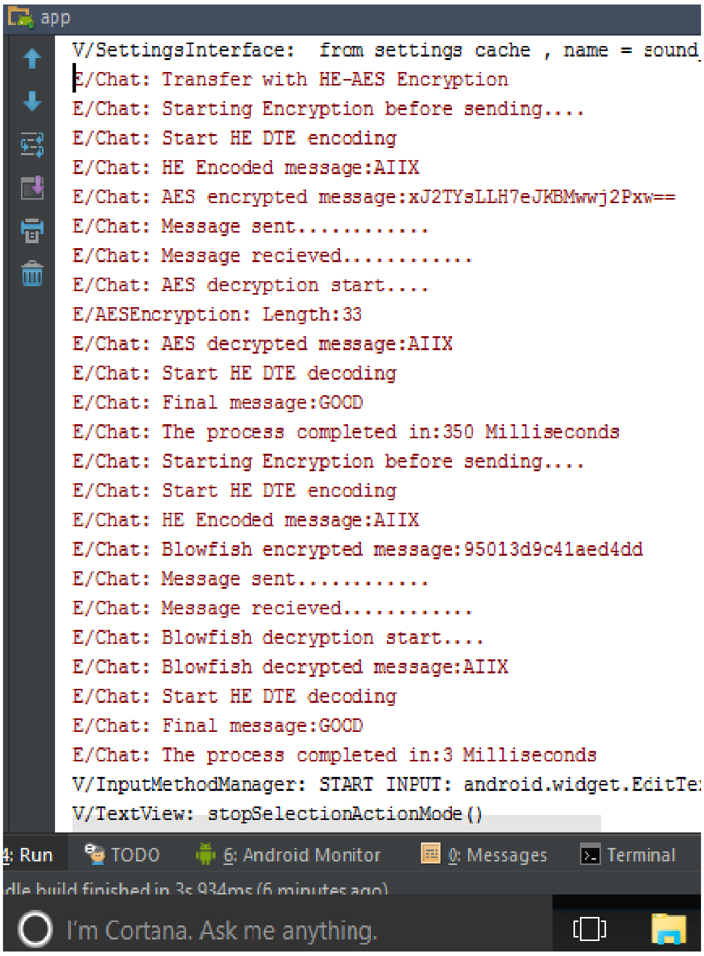
Step 1: The message is first encoded with the Honey encoding and then the encoded text.

Step 2: The encoded text is encrypted with AES algorithm and produces the cipher text and send to the receiver. At receiver side the cipher text is decrypted with the AES decryption algorithm and produces the encoded text. Then the encoded text is decoded using Honey decoding scheme and produce the original plain text. The overall processing time is calculated and display the AESBlowfish time in millisecond on the android monitor screen.

Step 3: Again the same message in step 1 is encoded with Honey encoding and produce the encoded text.

Step 4: The encoded text is encrypted with Blow fish algorithm and produces the cipher text, then send to the receiver. At receiver side the cipher text is decrypted with the Blow fish decryption algorithm and produces the encoded text. Then the encoded text is decoded using Honey decoding scheme and produce the original plain text. The overall processing time is calculated and display the HEBlowfish time in millisecond on the android monitor screen.

We see that for each input message, here the Blow fish produce the best result with HE by taking less time as compare to AES with HE The above process is shown in the following output screen of android monitor.


app
4: Run
TODO
6: Android Monitor
0: Messages
Terminal
dle build finished in 3s 934ms (6 minutes ago)
I'm Cortana. Ask me anything.

```
V/SettingsInterface:  from settings cache , name = sound
E/Chat: Transfer with HE-AES Encryption
E/Chat: Starting Encryption before sending....
E/Chat: Start HE DTE encoding
E/Chat: HE Encoded message:AIIX
E/Chat: AES encrypted message:xJ2TYsLLH7eJKBMwwj2Pxw==
E/Chat: Message sent............
E/Chat: Message recieved............
E/Chat: AES decryption start....
E/AESEncryption: Length:33
E/Chat: AES decrypted message:AIIX
E/Chat: Start HE DTE decoding
E/Chat: Final message:GOOD
E/Chat: The process completed in:350 Milliseconds
E/Chat: Starting Encryption before sending....
E/Chat: Start HE DTE encoding
E/Chat: HE Encoded message:AIIX
E/Chat: Blowfish encrypted message:95013d9c41aed4dd
E/Chat: Message sent............
E/Chat: Message recieved............
E/Chat: Blowfish decryption start....
E/Chat: Blowfish decrypted message:AIIX
E/Chat: Start HE DTE decoding
E/Chat: Final message:GOOD
E/Chat: The process completed in:3 Milliseconds
V/InputMethodManager: START INPUT: android.widget.EditTe
V/TextView: stopSelectionActionMode()
```

Figure 3 : The Resulting Simulated Screen Showing Result

The experiment conducted shows the behavior patterns according to execution time. The above execution time reading clearly shows that the proposed approach has less execution time as compared to AES with Honey Encoding.

7 Conclusion

It is concluded that the proposed messaging approach with Honey Encoding with Blowfish produces best result by taking less processing time. The Brute force attack can be easily reduced with Honey Encryption. So the hackers can't attack the original message easily. This approach is good for Android based devices for do messaging securely. In future more secure frame can be manufactured by Incorporating Handshaking techniques for end to end empowerment between sender and receiver.

8 References

[1] Joo-Im Kim and Ji Won Yoon "Honey chatting: a novel instant messaging system Robust to eavesdropping over communication", Center for Information Security Technologies (CIST) Korea University, Seoul, Republic of Korea IEEE 2016 ,fjooimkim, jiwon_yoong@korea.ac.kr

[2] Joseph jaeger,Thomas Ristenpart & Qiang tang, "Honey encryption beyond message recovery security" , Eurocrypt, 2016.

[3] Ari Juels, Thomas Ristenpart "Honey Encryption: Security Beyond the Brute-Force Bound" EUROCRYPT 2014

[4] Navin Tyagi, Jessica Wang, Kevin Wen, Daniel Zuo "Honey Encryption Application", Computer and Network Security, Spring 2015

[5] G. Sowmya, D.Jamuna, M.Venkatakrishna Reddy, "Blocking of Brute Force Attack", International Journal of Engineering Research & Technology(IJERT), ISSN:2278-0181 Vol. 1 Issue 6, August-2012

[6] Nahri Syeda Noorunnisa1, Dr. Khan Rahat Afreen "Review on Honey Encryption Technique" International Journal of Science and Research (IJSR) ISSN : 2319-7064, 2015-16

[7] Ankita Verma1, Paramita Guha , Sunita Mishra, "Comparative Study of Different Cryptographic Algorith ms", International Journal of Emerging Trends & Technology in Computer Science (IJETTCS), Volume 5, Issue 2, March - April 2016, ISSN 2278-6856

[8] Neetesh Saxena ,Narendra S. Chaudhary "EasySMS: A Protocol For End-to-End Secure Trans-mission Of SMS" IEEE Transactions OnInformation Forensics And Security, Vol. 9, No. 7,July 2014

[9] P. Princy "A Comparison Of Symmetric Key Algorithms DES, AES, BLOWFISH, RC4, RC6: A Survey" International Journal of Computer Science & Engineering Technology ISSN : 2229-3345 Vol. 6 No. 05 May 2015

[10] Priyanka Chouhan, Rajendra Singh," Security Attacks on Cloud Computing With Possible Solution", International Journal of Advanced Research in Computer Science and Software Engineering, Volume 6, Issue 1, January 2016

[11] Varsha S. Bari1,Nileema R. Ghuge,Chaitali C. Wagh,Sayali R. Sonawane ,Mr.M.B. Gawali", SMS Encryption on Android Message Application", IJARIIE-ISSN (O)-2395-4396 Vol-2 Issue-2 2016