International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 04 Issue: 09 | Sep -2017 www.irjet.net p-ISSN: 2395-0072

Securing Messages from Brute Force Attack by Combined Approach of Honey Encryption and Blowfish

Rasmita Sahu[1], Mohd. Shajid Ansari[2]

[1]MTech Scholar, Department of Computer Science & Engineering, RSR-RCET, Bhilai, Chhattisgarh, India
[2]Assistant Professor, HOD, Department of Computer Science & Engineering, RSR-RCET, Bhilai, Chhattisgarh, India

---

Abstract - *Nowadays, Messaging service is being used in many daily life applications like banking, business processes, mobile commerce, and so on. Mostly the current messaging based transactions use traditional encryption for security of the message. But the message may be hacked easily by the brute force attacker when he/she guess the key. Encryption also poses a threat as if key is hacked then the message could be intercepted during transmission. The proposed approach aims to develop a secure messaging transmission scheme that uses Honey Encryption technique for preventing the text based messages from brute face attacks in order to make messaging communication more secure and efficient. In this research paper we present a secure framework for messaging in android based device by using Honey Encryption with an efficient symmetric key encryption algorithm. We also perform a comparative analysis between the Honey Encoding with AES and Honey encoding with Blowfish algorithm. Experiments show that the Honey Encryption produce best result with Blowfish as compared to AES because Blowfish takes less processing time.*

Keywords-Messaging, Brute force Attack, Honey Encryption, AES, Blowfish

I. 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. People send confidential information through the SMS so security is needed.

For providing security mostly encryption technique is used by which the message is converted into an unreadable format known as cipher text. But still the encryption is weaker because brute force attacker [2] hack message easily if he find the key through some hacking methods. The security totally depends upon the key size and quality of key. If the key length is n bit then there is a possibility of generating 2^n possible keys [1]. So the attacker needs to try 2^{64} possible operations for cracking 64 bit key.

II. EXISTING SYSTEM

In the existing system the message security is provided by the conventional cryptographic algorithms like symmetric key and asymmetric key. The conventional Password Based Encryption algorithm [1] provides a single layer security that is the message is encrypted by shared secret key at sender side and produces a cipher text. The cipher text is in unreadable format and it travels through the network. So in this case there is a chance of happening Brute force attack with the cipher text, because the attacker tries to hack the message by trying all the possibility of keys to decrypt the message. If he gets the password then he can easily hack the message. In the reference paper [1] , the authors provides a best Encryption technique known as Honey Encryption which provides security against the Brute force attack. The research paper presents a messaging system for a desktop based application using Honey Encryption scheme. We know HE is two step approach, so at first step the authors of paper [1] implement DTE by the Statistical coding scheme and at step two they propose a symmetric key algorithm AES to encrypt the encoded message. So we need an improved framework for Honey encryption for messaging services, which is proposed in our research work.

III. PROBLEM IDENTIFICATION

The major Issues identified from the traditional approaches are:

1. The Present AES encryption technique employed in Honey encryption needs more Processing plus it demands more rounds of communicating.
2. Present approach is implemented on desktop computer. Today for messaging people prefer Android based devices so we need a messaging application with HE which can execute on Android based devices
3. DTE is the core of Honey Encryption. So the good quality of DTE design for messaging application is a very challenging task because for every plain message the DTE should generate a meaningful output message so that the attacker get confuse.

IV. PROPOSED ALGORITHMS

A. Honey Encryption

Honey encryption (HE) is a simple Method of encrypting messages using Non min-entropy keys for example passwords. He's intended to produce a cipher text which, when decrypted with any of a number of keys that were incorrect, yields plausible appearing but bogus plaintexts called honey messages. Honey Encryption [2] turns each wrong password guess made by a hacker to a perplexing dead-end. When an application or user sends and enters a password key to get an encrypted database or database, so long as the password is correct, the data is encrypted and accessible in its original, and readable, format. In case the password key is incorrect the data will last to be encrypted and invisibly. Hackers who steal databases of user logins and passwords simply need to guess just one correct password to be able to get access to the data [2]. How that they understand they have the right password is as soon as the database or file becomes readable. By way of example, if a hacker created 100 password attempts, they would get 100 plain text success. Even if one of the passwords were correct, the real data would be equal from the bogus data.

Honey Encryption is broadly two step process, in first step DTE (Distribution Transforming Encoding) is happen with the input data then in second step SE (Symmetric Encryption) is happen to encrypt the encoded data. The following diagram shows the overall process of Honey Encryption Scheme.

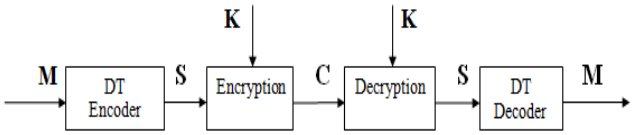

Figure-1: Honey Encryption Process

Where the following table present the symbos detail in the figure1

Table-1: HE Symbols

M	Message
S	Seed
C	Cipher text
K	Key
DTEncoder	Discrete Transforming Encoder
DTDecoder	Discrete Transforming Decoder

B. AES Algorithm

The Popular and widely adopted symmetric encryption algorithm likely to be encountered nowadays is the Advanced Encryption Standard (AES). It's found at least six times faster than algorithms like triple DES. AES comprises three block ciphers: AES-128, AES-192 and AES-256. Each cipher encrypts and decrypts data in blocks of 128 bits using cryptographic keys of 128-, 192- and 256-bits, respectively. First the plain text block is put into an array then the data is processed in no of rounds repeatedly to change its original meaning. The number of rounds is depends upon the key size. For the key length 128 bit it takes 10 rounds, for the key length 192 bit it takes 12 rounds and for key length 256 bit it takes 14 rounds.[10]

C. Blowfish Algorithm

Blowfish is a symmetric block cipher developed by Bruce Schneier in 1993.It is a fastest algorithm can be Utilized as a drop-in replacement for DES or IDEA. Blowfish takes 64 bit input block of message to encrypt and requires a variable-length key, from 32 bits to 448 bits, which makes it ideal for both domestic and exportable use [8]. Ever since that time it has been analyzed considerably, and it's slowly gaining acceptance as a strong encryption algorithm. Blowfish is unpatented and license-free, and is available free for all applications. Blowfish is a Feistel network algorithm employed (Feistel Network), which consists of 16 rounds [10]. In this paper we combine this algorithm with Honey Encoding for the best performance of message security.

V. PROPOSED METHODOLOGY

The methodology will consist of following phases:-

1. At sender side first Encode message with Honey Encoding and generate encoded text.
2. Encrypt the encoded output with Symmetric key algorithm and key to generate cipher text.
3. At receiver side decrypt the cipher text with decryption algorithm and with same key to produce the encoded text.
4. Then the encoded text is decoded with Honey Decoding to produce the original message.
5. Then calculate overall processing time

The above 5 steps are perform 2 times. First time with AES and second time with Blowfish and then compare the overall processing time in both the cases to find which encryption algorithm is best with Honey Encoding, whether it is AES or Blowfish.

5.1 Overall process of proposed algorithm

The following block diagram show the detailed steps of the proposed messaging system using Honey Encoding with AES.

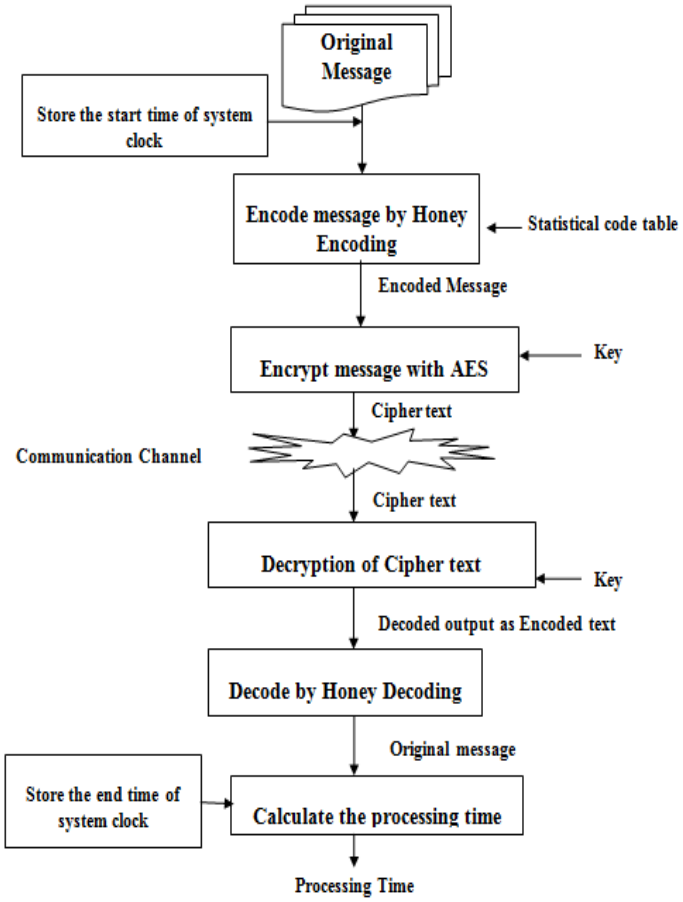

Figure -2: Proposed HE with AES

Similar to the above diagram, The following Block diagram also show the detailed steps of the proposed system using Honey Encoding with Blowfish algorithm. Here also the overall processing time for HE _ Encoding, Blowfish _ Encryption, HE _ Decoding and Blowfish _ Decryption is calculated to compare the time taken by AES with HE.

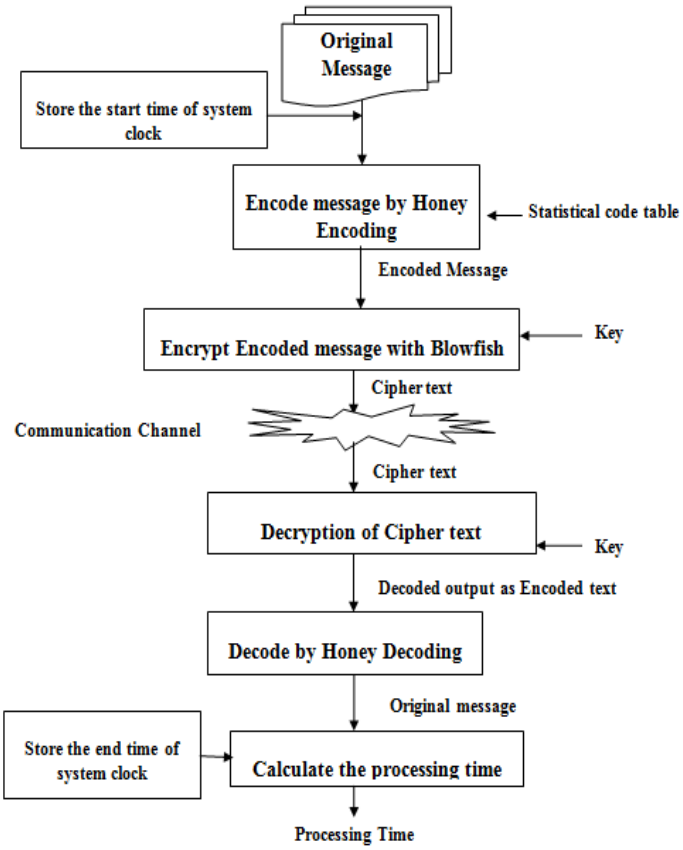

Figure-3: Proposed HE with Blowfish

In the above two process in figure 1 and figure 2 the processing time are calculated at each case. We compare the processing time consumed by AES with HE and Blowfish with HE. Here the Blowfish algorithm takes less time as compare to AES. We implement this in our experiment bellow.

VI. IMPLEMENTATION AND RESULT ANALYSIS

In this section we implement the proposed methodology in Android studio by developing a simulated messaging application and the language used is java.

Simulation Environment:

The simulation uses the supplied classes in java environment to mimic the operation of AES and Blowfish. The implementation uses managed wrappers for AES and Blowfish available in java .cypto and java. Security [Crypto Spec] that wraps unmanaged Implementations available in JCE (Java Cryptography Extension) & JCA (Java Cryptography Architecture).
The Cipher class provide the functionality of a cryptographic cipher used for encryption and decryption. It forms the core of the JCE framework.

Overall Performance with Honey Encryption:

The comparative analysis of working time among the algorithms with Honey Encryption is as given below.

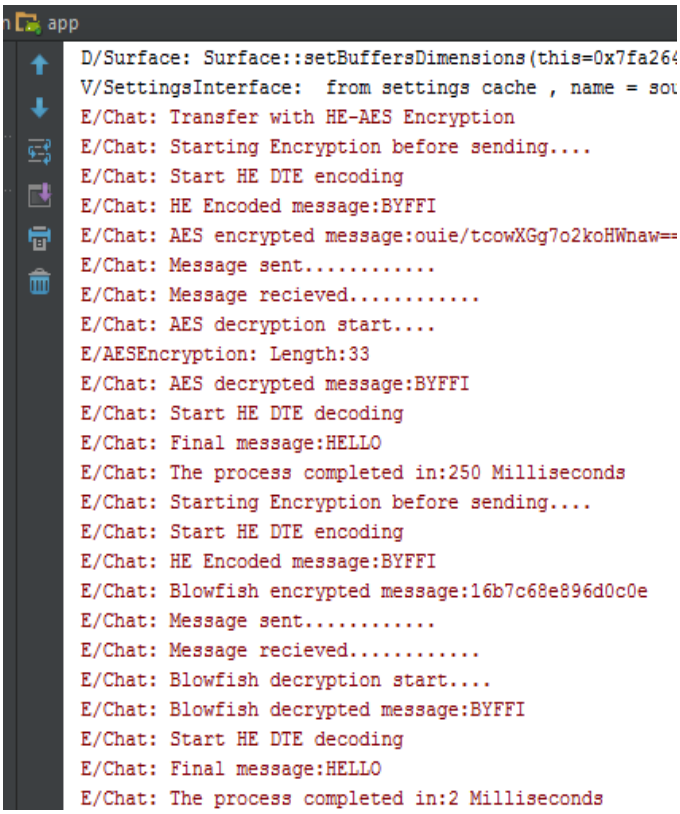

Figure-4: The Simulated Result in Android Environment

Comparative Analysis of AES and Blowfish with HE

Performance Results using ECB and CBC Mode:

To select any one encryption technique between AES and Blowfish, which is good for the Honey encoding we have to compare them according to their performance. The experiments were conducted using ECB and CBC mode with AES and Blowfish, the results are shown in figure 1 and Table 1 below.

Table-2: Performance of AES and Blowfish with modes

Techniques	AES ECB mode	Blowfish ECB mode	AES CBC mode	Blowfish CBC mode
Time in Milliseconds	120	10	110	10

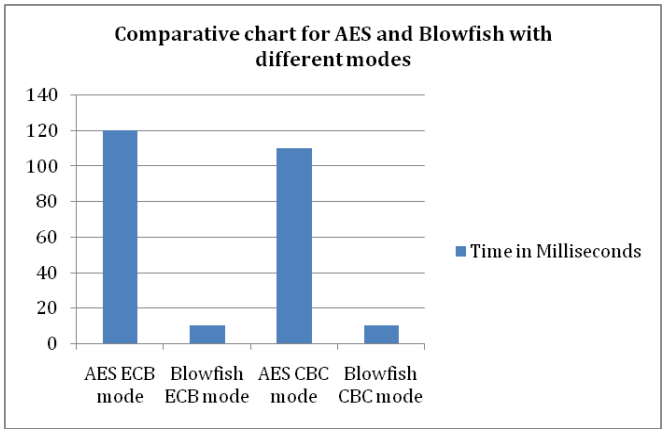

Figure-5: Performance graph of AES and Blowfish with ECB and CBC mode

The result shows the excellence of Blowfish algorithm as compare to AES in terms of the processing time. It shows also that AES consumes more processing time with its ECB and CBC mode than Blowfish. So it is a good decision to choose Blowfish with Honey Encoding to save time and provide security.

Table-2: Overall Execution time

Techniques	HE with AES	HE with Blowfish
Time taken in Milliseconds for a message	250	2

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

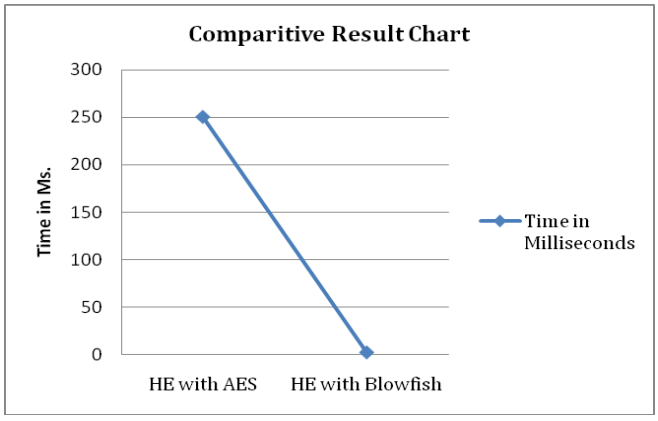

Figure-5: Overall Execution Time Chart

The above line chart show the time taken by Honey encoding with AES and Honey Encoding with Blowfish to encrypt and decrypt a single message. The HE with AES takes 250 milliseconds, while the HE with Blowfish takes only 2 milliseconds to perform Encoding, Encryption, Decoding and Decryption for a single message.

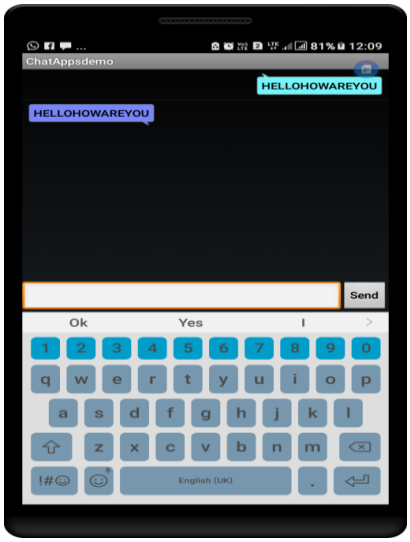

Figure-6: Messaging Screen

VII. CONCLUSION

The secure framework is successfully designed to provide security against Brute force attack with Honey encryption. For android based devices it is essentially needed that the messages must be send or receive quickly so we need an encryption algorithm which takes less time and secure to encrypt as compare to other, so Blowfish encryption algorithm is the good choice to integrate with Honey encoding. Because it takes less processing time as compare to AES. In future more secure framework can be developed by Incorporating Handshaking techniques for end to end empowerment between sender and receiver. Also we can apply the honey encryption with Blowfish in cloud computing security in future.

REFERENCES

[1] Joo-Im Kim and Ji Won Yoon, "Honey chatting: a novel instant messaging system Robust to eavesdropping over communication," ICASSP, IEEE 2016

[2] Ari Juels, Thomas Ristenpart, "Honey Encryption Encryption Beyond the Brute-Force Barrier," Copublished by the IEEE Computer and Reliability Societies , 2014

[3] N. Tyagi, J. Wang, K. Wen, and D. Zuo. "Honey encryption applications.," Network Security, 2015.

[4] Hoyul Choi, Hyunjae Nam, Junbeom Hur, "Password Typos Resilience in Honey Encryption," International Conference on Information Networking, IEEE 2017

[5] Amir Emad Ghassami, Daniel Cullina, and Negar Kiyavash, "Message Partitioning and Limited Auxiliary Randomness: Alternatives to Honey Encryption,"IEEE International Symposium on Information Theory, 2016

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

[7] 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

[8] Muhammad Kamran Asif, Yahya Subhi Al-Harthi, "Intrusion Detection System using Honey Token based Encrypted Pointers to Mitigate Cyber Threats for Critical Infrastructure Networks," International Conference on Systems, Man, and Cybernetics, IEEE 2014

[9] Sourabh Chandra, Smita Paira ," A comparative survey of symmetric and asymmetric key cryptography," International Conference on Electronics, Communication and Computational Engineering (ICECCE), IEEE 2014

[10] 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

[11] 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

[12] 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

[13] 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.