iJETRM
International Journal of Engineering Technology Research & Management

HONEY ENCRYPTION ALGORITHM TO SAFEGUARD AGAINST BRUTE FORCE ATTACKS

Prithvik H C*[1]
Naman Jain[2]
Rakesh K R[3]
*[1]Prithvik H C (Department of Telecommunication Engineering, Bengaluru, India)
[2]Naman Jain (Department of Telecommunication Engineering, Bengaluru, India)
[3] Rakesh K R (Department of Telecommunication Engineering, Bengaluru, India)

ABSTRACT
Cryptography is an essential and integral part of the ever developing and advancing communication technologies. With these enhancements in the technology, there arises a need for extending security features for the same. There has been numerous algorithms, standards, and cryptographic systems developed to ensure the data security, but most of them are vulnerable to brute-force attack. To encounter this a crypto-system i.e., Honey Encryption (HE) was introduced. This technique ensures that any adversaries trying to guess the password or keys with partial or no knowledge by repetitive attempts is fed with incorrect or false information that might look the same as the original data. We were able to develop program to demonstrate the HE for a banking application and analyse its resilience towards brute-force attack and also see its applications in various fields.

Keywords
Brute-forceattack, Cryptography, Distribution Transforming Encoder (DTE), Honey Encryption (HE)

INTRODUCTION
With the ever-extending advancements in the communication technologies, there exists greater threat to the information being shared through the pubic network. This burden is laid on the cryptographic systems to ensure that the information is secure and not been tampered by adversaries. There has been a great amount of research and development in the field of Cryptography to develop and implement various algorithms and crypto-system to ensure the data security and integrity. In this view various types of Symmetric and Asymmetric algorithms were developed. Though the implementation of these complex and advanced encryption systems has helped in securing information and its integrity, they are still prone to Brute-Force attack and being hacked with the help of powerful system. To overcome these problems in 2014 Ari Juels and Thomas Ristenpart [6] came up with an ingenious solution to these type of attacks.

Since the Roman empire, people have been using cryptographic techniques to ensure the safety of their data. Many such techniques were developed over the course of time and these were mostly categorised as Symmetric encryption techniques, where the encryption and decryption takes place with the help of the same key used at sender and receiving ends. These were effective but were mostly prone brute-force attack and the crypto-system could be broken easily. Later, with the advent Asymmetric encryption systems, which used different keys for encryption/decryption and the complexity of these algorithms made the security for the information being shared through public channel exponentially secure. But as the development of powerful processing systems even these complex crypto-systems fell prey to the brute-force attack. Honey Encryption (HE) provides an efficient method to counter the brute-force attack and provides an unusual security for the information being shared.

HE uses a set of deceitful messages that seems to be correct but are decoy or false messages, when a wrong key is used for decryption, the system can yield a valid-looking plaintext message [4, 5]; therefore, the attacker cannot tell whether the guessed key is correct or not. The HE uses the Distribution Transforming Encoder that (DTE) [2, 3, 6] at its core whose operation depends on the probabilities of the messages in the message space. These messages are mapped into a message space and XORed with a key to get the ciphertext. For decryption,

iJETRM
International Journal of Engineering Technology Research & Management

the reverse operation achieves the messages back. The HE has various applications in the field of banking like PIN management, sensitive details management etc., Cloud data management, IOT, etc. [3]
In the next sections, the working of HE system and the implementation of HE encryption and its applications in banking especially in handling bank account numbers. Later, the paper concludes with a summary of HE and the ways that could extend the efficacy of the HE crypto-system.

METHODOLOGY

HE is application specific and for each application the HE system has to be altered accordingly and it provides protection to messages that have some common features (e.g., bank account or credit card numbers). A collection of messages is called a message space. We determine the message space before-hand i.e., before the encryption process and these messages are sorted tin a specific manner. The probability of occurrence of each message in the message space is calculated using probability distribution function (PDF) and also the cumulative probability (CDF). A seed space is created for Distribution Transforming Encoder (DTE) that maps each message in the message space to a seed in the seed space. The seed range is determined by the DTE on the PDF and CDF of each message and it is made sure that the PDF of each message is equal to the ratio of the corresponding seed range to the seed space.
The CDF is a random value which gives the beginning of the seed range of a particular range and the PDF gives the range of values that a particular message is mapped into the seed range. Now, if a particular message is being communicated, the corresponding seed value is take and it is XORed with key to get the ciphertext. The decryption process is just the inverse process of this. The ciphertext is XORed with the key and then the seed value is taken to map it back to the message. If the key that is being used for decryption is taken it results in a different message. With this core architecture of the HE system, a list of honey words [4, 5, 7] were created such that they act as trap passwords and when these passwords are used it sends out a warning message that there has been an unauthorized attempt to access the information. So when the adversary persistently tries to crack the information by repeatedly trying various different combination of the passwords, he/she fed with the wrong or deceitful information that might look correct or original but are in fact false information that look the original. In the next chapter we seen the implementation of the same.

IMPLEMENTATION and RESULTS

This section looks into the design and implementation of the HE in handling the bank account numbers when they are being communicated with clients. Here the program prompts the user to enter their message, and the password. We have shown three simulations where we see different scenarios played out. Firstly, when the end-user enters the correct password, secondly, when the end-user enters the password from the list of probable passwords or 'Honey Passwords' which the brute-force attacker might use and lastly when the end-user uses a wrong password.
The program initially prompts to enter a message and then the message. Our program is designed to generate ten probable password that an attacker might use. This number is exponentially higher un the real world applications with the use of powerful word processors to generate more and more probable passwords.

- End-User enters the correct password

In the first simulation (Fig. 1), we can see that the user gives his secret information and the password that is being associated with it. The end-user enters the correct password to retrieve the information and the right information is retrieved.

```
Please enter a secret message to store: The Bank Account number is 7429940982
Please enter a password: myUser
Your password is myUser, your seed value is 13, and your secret message is The Bank Account number is 7429940982
==================================
['myuser', 'MYUSER155', 'MYUSER', 'myUser12', 'myUser111', 'MYUSER205', 'myUser', 'myuser143', 'myuser183']
Enter a password to crack: myUser
The Bank Account number is 7429940982
Would you like to enter another inquiry (Y/N):
```

Fig. 1: End-User using correct password

- End-User enters the probable/honey password

iJETRM
International Journal of Engineering Technology Research & Management

In the second simulation (Fig. 2), we can see that the user enters all the required information and the end-user or the attacker tries to crack the message using Brute-force attack. With a ready set of possible or similar password list we check his entries and if the it matches then an alarm goes off in the background and attacker is provided with the incomplete or misinformation.

```
Please enter a secret message to store: The Bank Account number is 7429940982
Please enter a password: myUser
Your password is myUser, your seed value is 16, and your secret message is The Bank Account number is 7429940982
====================================
['myuser213', 'myUser', 'myUser141', 'MYUSER185', 'MYUSER235', 'myuser', 'myUser15', 'MYUSER', 'myuser173']
Enter a password to crack: myuser213
Intruder! SOUNDING ALARM!
The Bank account number is 982349823
Would you like to enter another inquiry (Y/N):
```

Fig. 2: End-User using honey password from the list

- End-User enters the Wrong password

  In the third simulation (Fig.3), we can see the end-user trying to decrypt the message by guessing the random passwords. If these attempted password are not in the probable list of passwords then the system doesn't allow the entry of the user to retrieve the information.

```
Please enter a secret message to store: The Bank Account number is 7429940982
Please enter a password: myUser
Your password is myUser, your seed value is 12, and your secret message is The Bank Account number is 7429940982
====================================
['myuser173', 'myUser11', 'myuser133', 'MYUSER', 'myUser', 'myUser101', 'MYUSER195', 'myuser', 'MYUSER145']
Enter a password to crack: User
Password not found or incorrect.
Would you like to enter another inquiry (Y/N):
```

Fig. 3: End-User using wrong password

CONCLUSION

The recent development of honey encryption offers many password-based security schemes resilience to brute force offline attacks by yielding plausible plaintexts under decryption by invalid keys. we addressed the key challenge of generating plausible honey messages for each of these spaces by researching the probabilistic distribution of the message spaces and constructing good DTEs for each. To look into the future the design of more powerful word processors that can generate possible honey passwords also contextually and semantically correct messages to be developed to ensure security. Currently, HE can be used efficiently on small messages due to message space limitations and with further improvements in cryptographic systems this could be implemented to large messages with improved message space. Finally, with altering the message space in the DTE along with recalculating the PDF and CDF of every message, the HE system can be used as firewall to specific applications.

REFERNCES

[1] Omolara; Abiodun Esther; Aman Jantan; and Oludare Isaac Abiodun. *A comprehensive review of honey encryption scheme*. Indonesian Journal of Electrical Engineering and Computer Science 13.2 (2019): pp. 649-656.

[2] Kapil; Gayatri. *A Novel Approach to Secure Big Data Using Attribute Based Honey Encryption.*Proc. of Department of Information Technology, School for Information Science and Technology, Babasaheb Bhimrao Ambedkar University, 2019.

[3] Uthayashangar, S., et al. *Efficient Group Data Sharing in Cloud Environment Using Honey Encryption*. 2019IEEE International Conference on System, Computation, Automation and Networking (ICSCAN). IEEE, 2019.

[4] Abdalla;NosibaAltoom Adam. *Preservation of Data Confidentiality Using Honey Encryption.*Proc.mof Sudan University of Science and Technology, 2019.

[5] Omolara, Abiodun Esther, et al. *A novel approach for the adaptation of honey encryption to support natural language message.*Proc. of the International MultiConference of Engineers and Computer Scientists. Vol. 1. 2018.

iJETRM
International Journal of Engineering Technology Research & Management

[6] Ari Juels; Thomas Ristenpart.*Honey Encryption: Beyond the Brute-Force Bound*. Advances in Cryptology-Euro-Crypt 2014, LNCS 8441, Springer, 2014, pp. 293–310.
[7] Joseph Jaeger, Thomas Ristenpart;Qiang Tang. *Honey Encryption Beyond Message Recovery Security*. Advances in Cryptology-Euro-Crypt 2016 ,pp. 758-788, 2016.
[8] Juels, A.; Rivest, R.L. *Honeywords: making password-cracking detectable.*in Proc. of the ACM SIGSAC Conference on Computer and Communications Security (CCS '13), pp. 145–160, ACM, November 2013.
[9] Chatterjee, R.;Athalye, A.; Akhawe, D.; Juels, A.; Ristenpart, T.. *Password typos and how to correct them securely*. Proc. of Security and Privacy (SP), 2016 IEEE Symposium, pp. 799–818, 2016.
[10]Choi, H.; Nam, H.;Hur, J.. *Password Typos Resilience in Honey Encryption*.IEEE Symposium. Proc. Of The 31st International Conference on Information Networking (ICOIN 2017), pp. 593-597, 2017.