Advances in Mathematics: Scientific Journal **9** (2020), no.10, 8249–8258
ISSN: 1857-8365 (printed); 1857-8438 (electronic)
https://doi.org/10.37418/amsj.9.10.54 Spec. Iss. on AOAOCEP-2020

PROVIDING INFORMATION SECURITY USING HONEY ENCRYPTION

SANTHOSHINI SAHU[1]

ABSTRACT. Now a days internet usage has increased and because of this security concern also have increased. It has been a difficult task for providing security towards brute force attack. Brute-force attack is guessing passwords until they receive the original data. This paper proposes a new encryption method FDGA (Fake Data Generating Algorithm) that could frustrate hackers by giving them fake data while making it appear real. This algorithm makes use of Honey Encryption, which turns every incorrect password guessed by the hacker into a confusing dead-end which means they will not be able to recognize the original page. This algorithm takes the user id and password from the application and with the help of password key it tries to access the encrypted database or file. If the password is correct then the genuine access is given to the user but if the password is incorrect then the hacker will be given access to the fake page. Brute Force attack is highly possible as hackers who steal databases of user logins and passwords only have to guess a single correct password in order to get access to the data. They come to know that they have the correct password is when the database or file becomes readable. To speed up the process, hackers have access to sophisticated software that can send thousands of passwords each minute to applications in an attempt to decrypt the data and even by using higher speed, multi-core processors reduces the time it can take to break encryption. Even though after all these efforts, hackers fail to access the original data of the file. So, with the help of this algorithm we can achieve security for login systems against the brute-force attack.

2020 *Mathematics Subject Classification.* 68Q11, 94A05.

Key words and phrases. Honey Encryption, Fake Data, Brute Force Attack.

1. Introduction

For every application login is the basic security feature provided. Mainly login is done with the help of username or email or user ID along with the password. The password can be alphanumeric with special characters and with many rules. Even after providing so many rules the intruders easily detect the password with the help of trial and error methods or with the help of sophisticated algorithms that generate 1000 passwords within seconds. Using higher speed of multi-core processors will decrease the time it can take to break encryption. The guessing of passwords by the intruders is known as Brute Force Attack.

Password Based Encryption (PBE) methods are used to protect data against the brute force attack. These PBE methods provide correct data if the user name and password is correct and does not provide data if the credentials are wrong giving a clue to the hackers that the guessed password is wrong, this process continues until login access is given to the hacker and in this situation the main drawback is that we could not guess that this is an illegitimate authentication. So, to overcome this problem honey encryption is used [1].

Honey word is used in information security to specify false resource. Honeypot is a false server that usually contains fake data. Honeyword is false data that is stored in database. Based on honey encryption whenever encryption algorithm detects an intrusion it provides fake data which resembles the original data which confuses the intruder. Hackers who steal databases of user logins and passwords in order to get access to the data is to guess a single correct password. The way they know they have the correct password is highly difficult for the intruder.

Honey Encryption provides a different level of security to the encrypted data. When an intruder tries all possible combinations of credentials to crack password or guess the encryption key, at that point honey encryption plays role by providing fake data as a response to every guessed attempt. So, whenever a hacker makes an incorrect attempt, he receives spoofed data, which looks pretty similar to the actual data. Even if the attacker guesses the correct password eventually, the actual data will be lost in the crowd of spoofed data. Each decryption done by the intruder is going to look as if it is the real decrypted data. The intruder has no way to differentiate which is correct among the guessed [2].

This paper is organized with the five sections. We discuss literature review about the honey words generation algorithm and honey encryption process in section 2 and section 3 presents working model of the algorithm and the flow diagram of the model along with the encryption and decryption algorithms. In section 4, we focus on the results and discussions where we have given a set of output data generated by the algorithm to original and fake users with a comparative study of output of binary text and the text generated by the algorithm and then finally, in section 5 we conclude the paper.

2. LITERATURE SURVEY

Natural Language Encoder (NLE)[3] called NoCrack have used existing password models whose performance was calculated by finding the time needed to find out a particular vault and the amount of time which is needed to add a password to it. The main drawback of this paper is if there is large vault it is not supported. Beunardeau [4] proposed corpus quotation DTE which was built with help of Grammar model of language where users are required to find out the known public documents. But the disadvantage with this paper is quoting from a public document will not allow users to the main domain of the document and also fixed code book will not be able to provide all the essential combination of words. The authors [5] proposed that the statistical code scheme is a combination of structural code scheme which finds the syntax of natural language and honey encryption scheme, where the honey encryption finds the semantic feature of the natural language. The main disadvantage with this approach is it does not support any other data format and also generates fake messages only for short length messages and also the ambiguity between original message and the fake message is too much so the intruder can easily use this to find out the difference between original and fake data. The proposal from Golla [6] used Kullback-Leibler (KL) divergence and with help of this proposal the method of securing vaults had improved. But the main disadvantage of adopting this method is there will be intersection attack and there is no proper closure of providing maximum security. The authors [7] designed DTE that uses a common module for both encryption and decryption algorithms and an interface for the message space which contains cumulative probability function and the probability function which accept a plaintext message and gives CDF and

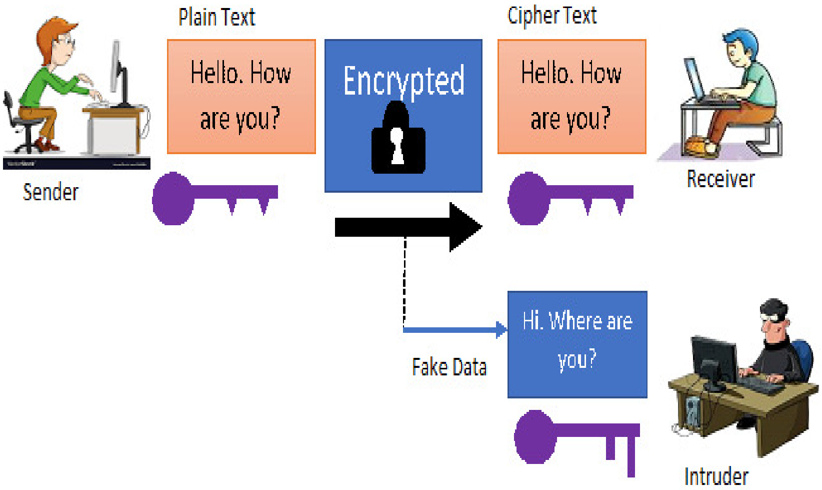


FIGURE 1. Model of Honey Encryption

PDF as output. The main disadvantage here is, it cannot be used for large message space because the overhead of processing the large message space will be too high, even if the application is not designed properly then it will not avoid brute force attack. The authors [8] have proposed two types of protocols that will help the authorized users to find the errors in the password. They have compared and analyzed the performance and security of the proposed scheme including the message recovery security. But still there is a flaw in the system by confusing legitimate users also. In this paper the proposed method helps to overcome the drawbacks of the referred papers and other related works with honey encryption and brute-force [9-10].

3. WORKING MODEL

In this paper a solution for brute force attack is provided because hacker is so sure in generating the multiple passwords and providing these at the same time and for which password the exact message is displayed that is the original password in his mind but in this paper, original message is provided to right user i.e. only if he have the correct password and fake data is given to the intruder which resembles the original data who tries to guess the password where the user ID is also hacked.

Sender: The person who send the message.
Receiver: The person who receives the original message.
Intruder: An unauthorized person who tries to access gain access of messages or system information.
Plaintext: It is the original message which is given by the sender and it will be in the readable format.
Ciphertext: It is the data sent which is in unreadable format with the help of encryption algorithm.

Honey encryption is explained in the figure-1. The sender send message Hello.How are you? to the receiver. Before sending the message in the network, the message is been encrypted with the help of encryption algorithm and that message is known as cipher text. The cipher text is decrypted at the receiver side. If the receiver is the original user then he gets the original message Hello. How are you?, but if there is an intruder in the network and tries to get the data then he will get the message from the honey pot which is maintained by the system as a separate server which is activated whenever there is an brute force attack on the system. Since the intruder gets the message Hi. Where are you?, he thinks that this is the original message.

If he gets doubt and tries with another password then also, he will get some fake data from the honey pot and each time the data differs but it is highly impossible for the intruder to guess which is the original message. Since the honey pot is maintained by the system and it contains some default messages which resembles the original message so it is difficult for the intruder to guess which is the right password and the right message.

Figure 2 explains the complete flow diagram of the working model. This algorithm can be used by websites to check whether the user who is trying to read the messages which are encrypted is genuine or not. If a person wants to send a message then he must be logged into the system. Before he tries to send message, he must have the user id of the user to whom he wants to send his message i.e. the receiver. When the message is given and he click send then the encryption algorithm runs and the data is encrypted by the system and sent to the receiver.

ENCRYPTION ALGORITHM
Input: Plain Text

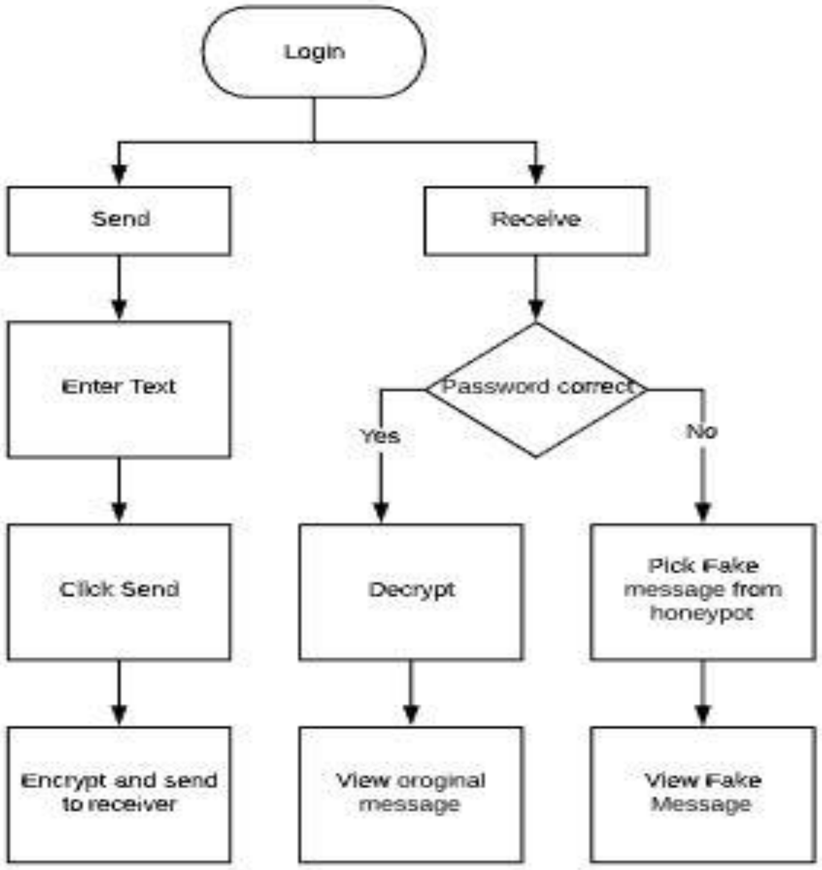


FIGURE 2. Flowchart of working model

Output: Cipher Text
Step 1: Consider Plain Text (M)
Step 2: Get Password from Database (P)
Step 3:n1= number of letters in text and n2= number of letters in password.
Step 4: $n3 = \frac{n1}{n2}$
Step 5: Merge text with password by placing password into text for every n3 (M1).
Step 6: Consider M1 as a character and convert it into ASCII code (M2).
Step 7: Consider M2 as a character and convert it into Binary (M3).
Step 8: Swap every 4 bits with next 4 bits which is the Cipher Text (C).

If a user is registered into the system, he not only can send messages, but also, he can also view the messages. In this scenario if the user is the intruder then the system must know. Who is the real user and who is the intruder for this the decryption algorithm works better because if the user is legitimate user then he receives original message and if he is intruder he receives fake data which resembles the original data.

For every conversion of Binary we have to use the formula below

N = $b_i * q_i$
Where, N is the positive number
b is the digit
q is the base value
i is the integer

DECRYPTION ALGORITHM

```
Input: Cipher Text
Output: Plain Text
Step 1:Consider Cipher text from the sender( C )
Step 2:Take password from the receiver (P1)
Step 3: if P1 is original password
3.1: Swap every 4 bits with next 4 bits (C1)
3.2: Convert C1 into character (C2)
3.3: Consider C2 as ASCII code and convert it into character (C3)
3.4: Remove password from the text and give it as plain text (P)
Step 4:else
4.1: Pick a Random Text from the database.
```

A honey pot of messages is being maintained by the system, which is used when the system doubts of a brute force attack. By using the user id if the intruder tries to guess the password then each and every time, he gets new message for the same request. Even after trying hundred and thousands of time he will never know which is the original data as for every data he feels that it is the original data always. By using this we are confusing the intruder which is original and which is fake and he enters into a dead end of brute force attack.

4. Result and Discussions

In this paper the algorithm provides a secure mechanism toward the brute force attack i.e. the user once he login he can send or receive the data. If he wants to send the data, he needs to enter the text he wants to send i.e. the plain

TABLE 1. Generated messages

Original message	**Original Receiver**	**Intruder**
What are you doing?	What are you doing?	Come out and see
How are you?	How are you?	Who are you?
Where shall we meet?	Where shall we meet?	Meet you at 9pm
Im the original user	Im the original user	I want to go out
Send me the notes	Send me the notes	Take your notes
Give me your phone number	Give me your phone number	My phone is lost

text and after clicking send the text is given to the encryption algorithm. The encryption algorithm encrypts the data and sends to the receiver.

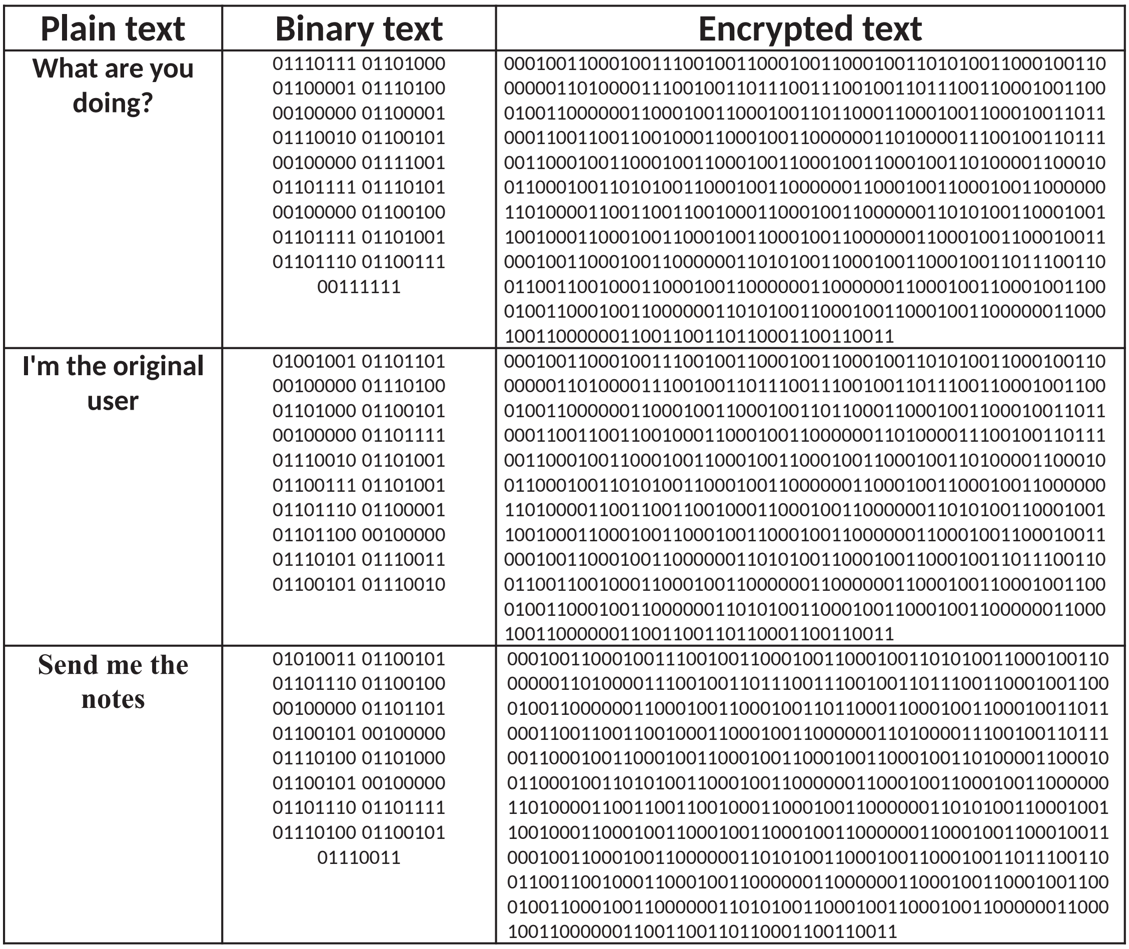
At the receiver side, the intruder part comes into play. The original user knowns his password so he has no difficulty in accessing his data but when it comes to the role of intruder, he tries to guess the password each and time until he gets the original access of data but according to our algorithm the original data is only given to the user with original password and the user who does not know the password is given the fake data which resembles the original data so finding out which is correct data and which is fake data is very difficult for the intruder.

Based on the results, we can easily say that we are successful in bluffing the intruder with the fake messages.

Based on the table 1 we can see what will be the data for the intruder and the legitimate user and figure 3 shows the comparison of conversion of plain text into normal binary text and the cipher text generated from the algorithm.

5. CONCLUSION

Using the algorithm FDGA (Fake Data Generating Algorithm) security for brute force against the login systems have been provided. Yet there is a limitation in this system i.e. they have limited field view. Honey pots need to be very large and, in some servers, this is not feasible. But this technology can be combined to network and host-based intrusion protection. A key challenge for using this technique is generation of honey messages through good DTEs for all type of problems naturally.



Plain text	Binary text	Encrypted text

What are you doing?	01110111 01101000
	01100001 01110100
	00100000 01100001
	01110010 01100101
	00100000 01111001
	01101111 01110101
	00100000 01100100
	01101111 01101001
	01101110 01100111
	00111111
		000100110001001110010011000100110001001101010011000100110
		000001101000011100100110111001110010011011100110001001100
		010011000000110001001100010011011000110001001100010011011
		000110011001100100011000100110000001101000011100100110111
		001100010011000100110001001100010011000100110100001100010
		011000100110101001100010011000000110001001100010011000000
		110100001100110011001000110001001100000011010100110001001
		100100011000100110001001100010011000000110001001100010011
		000100110001001100000011010100110001001100010011011100110
		011001100100011000100110000001100000011000100110001001100
		010011000100110000001101010011000100110001001100000011000
		1001100000011001100110110001100110011

I'm the original user	01001001 01101101
	00100000 01110100
	01101000 01100101
	00100000 01101111
	01110010 01101001
	01100111 01101001
	01101110 01100001
	01101100 00100000
	01110101 01110011
	01100101 01110010
		000100110001001110010011000100110001001101010011000100110
		000001101000011100100110111001110010011011100110001001100
		010011000000110001001100010011011000110001001100010011011
		000110011001100100011000100110000001101000011100100110111
		001100010011000100110001001100010011000100110100001100010
		011000100110101001100010011000000110001001100010011000000
		110100001100110011001000110001001100000011010100110001001
		100100011000100110001001100010011000000110001001100010011
		000100110001001100000011010100110001001100010011011100110
		011001100100011000100110000001100000011000100110001001100
		010011000100110000001101010011000100110001001100000011000
		1001100000011001100110110001100110011

Send me the notes	01010011 01100101
	01101110 01100100
	00100000 01101101
	01100101 00100000
	01110100 01101000
	01100101 00100000
	01101110 01101111
	01110100 01100101
	01110011
		000100110001001110010011000100110001001101010011000100110
		000001101000011100100110111001110010011011100110001001100
		010011000000110001001100010011011000110001001100010011011
		000110011001100100011000100110000001101000011100100110111
		001100010011000100110001001100010011000100110100001100010
		011000100110101001100010011000000110001001100010011000000
		110100001100110011001000110001001100000011010100110001001
		100100011000100110001001100010011000000110001001100010011
		000100110001001100000011010100110001001100010011011100110
		011001100100011000100110000001100000011000100110001001100
		010011000100110000001101010011000100110001001100000011000
		1001100000011001100110110001100110011

FIGURE 3. Comparison of binary text and cipher text

REFERENCES

[1] C. HOYUL, J. JEONG, S.S. WOO, K. KANG, J. HUR: *Password typographical error resilience in honey encryption*, Computers and Security, **87** (2019), 101411.

[2] J. ARI, T. RISTENPART: *Honey encryption: Encryption beyond the brute-force barrier*, IEEE Security and Privacy, **12**(4) (2014), 59-62.

[3] R. CHATTERJEE, J. BONNEAU, A. JUELS, T. RISTENPART: *Cracking-resistant password vaults using natural language encoders*, In: Security and Privacy, 2015 IEEE Symposium, **17** (2015), 481–498.

[4] M. BEUNARDEAU, H. FERRADI, R. GÉRAUD, D. NACCACHE: *Cracking-resistant password vaults using natural language encoders*. In Security and Privacy, Honey Encryption for Language. In International Conference on Cryptology in Malaysia (2016) Dec 1, 127-144.

[5] H.J. JO, J.W. YOON: *A new countermeasure against brute-force attacks that use high performance computers for big data analysis*, International Journal of Distributed Sensor Networks, **11**(6) (2015), 406915.

[6] M. GOLLA, B. BEUSCHER, M. DÜRMUTH: *On the security of cracking-resistant password vaults*, In Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, 2016, 1230-1241.

[7] W. YIN, J. INDULSKA, H. ZHOU: *Protecting Private Data by Honey encryption*, Hindawi Security and Communication Networks Volume 2017, Article ID 6760532 on 2017 nov 21.

[8] H. CHOI, H. NAM, J. HUR: *Password Typos Resilience in Honey Encryption*, International Conference on Information Networking (ICOIN), 2017, 593-598.

[9] K.S.M. MOE, T. WIN: *Honey Encryption Algorithm for Increasing Message Space against Brute Force Attack*, 15th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON), Chiang Rai, Thailand, 2018, 86-89,

[10] R.V. YOHANANDHAN, R.M. ELAVARASAN, M. PREMKUMAR AND L. MIHET-POPA: *Cyber-Physical Power System (CPPS): A Review on Modeling, Simulation, and Analysis With Cyber Security Applications*, IEEE Access., **8** (2020), 151019–151064.

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
GMR INSTITUTE OF TECHNOLOGY
RAJAM, ANDHRA PRADESH, INDIA
Email address: santhoshini.s@gmrit.edu.in