ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 3, March 2016

A Study on a New Algorithm for Data Encryption against Brute Force Attacks

S J R Sai Nikhilesh[1], Sarath Kumar V[2]

Student, Dept. of ECE, Pondicherry Engineering College, Puducherry,India[1]

Student, Dept. of ECE, Pondicherry Engineering College, Puducherry,India[2]

ABSTRACT: Data encryption has been undergoing radical changes to adapt to the modern day attacks on the data. Even though new algorithms are being designed every day, the fact that every cipher text can be broken by sheer brute force attacks. With the advent of cutting edge processors, the number of keys tried per second has been on the rise exponentially. Ari Juels and Thomas Ristenpart proposed the idea of honey encryption in their paper [1] which provides a valid plaintext to the attacker on application of wrong keys. Based on this concept of thwarting brute force attacks, we propose an efficient method to secure data against brute force attacks by providing an efficient way for encoding the data such that a person trying to brute force the data is overwhelmed the sheer size of the key

KEYWORDS*:* Data Encryption, Brute Force, Cryptography, Honey Encryption.

I. INTRODUCTION

Cryptographic algorithms in the present day are vulnerable to simple brute force attacks on it. The attacker simply produces every possible key in the key space (assuming he knows the encryption algorithm used) and hence it is simply a matter of fast processing speed to guess the keys. All the attacker has to do is to use a generated key to decrypt the algorithm, compare the outputted text to a dictionary and find if the outputted text has a meaning to it, it can be assumed to be the text. As the number of processors increases, the speed at which keys are tried also increases and hence the security of any algorithm decreases.

Ari Juels and Thomas Ristenpart proposed a ground breaking method called honey encryption to provide security against the brute force attacks by providing a set of valid looking plain texts to the attacker so that he gets confused as to which plaintext is actually the message. This method though efficient has a problem in the fact that the plaintexts need to be stored hence increasing the overhead. This plaintext might be susceptible to attack by the user hence thus turning the whole encryption methodology obsolete. Instead in our algorithm re-encrypts the data so that the data remains unbreakable.

II. PROPOSED MODEL

We propose a new model for CD’s and hard drive’s which shall enable the data to remain secure and accessible only when accessed by the intended recipient. The data is encrypted using any of the standard encryption algorithm along with the key which should be stored in a secure model. The data can be opened by a specific key only which shall be user inputted. In case the user enters the wrong key ‘n’ number of times where, n is a number determined by the sender, the data will re-encrypt itself in the following way:-

1. The key generated can either be from a generator or a simple shifted version or a decimated version of the original key according to the algorithm used.
2. The key can then be used to re-encrypt the data present so that it may be stored from the brute force attackers.

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 3, March 2016

The additional overhead here will be the data/code required here for re-encrypting the data. The most beautiful feature of the proposed algorithm is that it supports any type of data encryption algorithm use by the sender to protect it against brute force attacks.

The algorithm provides flexibility in the fact that even some of the easily breakable algorithms can be used to re-encrypt the data. If AES algorithm is use for encrypting the data for the first time, the chances of breaking the encryption becomes very less and with each passing mistake the chance to find the correct key combination decreases even further.

IV. KEY GENERATION TECHNIQUES

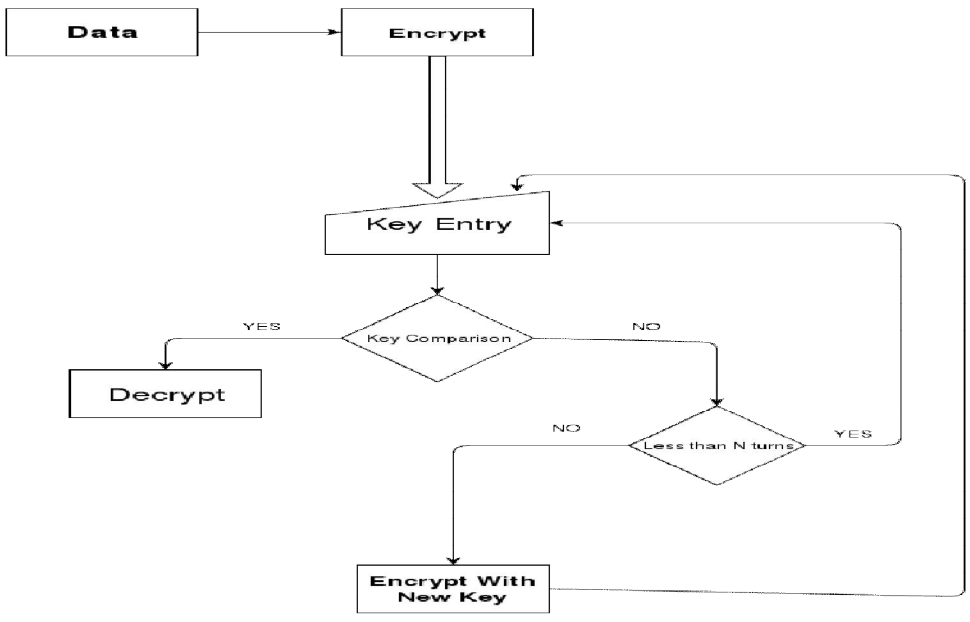
The keys which are used to re-encrypt the data can be generated in the following ways:-

Process 1:- Shifted version of the original key

Consider the key K. In general any secure Pseudo Random Key Generator (PRG) tries to produce a key which looks random in nature. The keys provided by the PRG have the property that the autocorrelation of the key is very less. This implies that even if a shifted version of the key is used for encrypting the data for second time or more, the key will not provide any knowledge about the original key. This method provides a very easy way to generate the key but has a limitation in the fact that the key needs to be accessed every time the data is to be re-encrypted, this may lead to a possibility of run-time attacks on the key.

Process 2:- Key generation using a new PRG

The new key for re-encryption can be generated by a new PRG which can be the part of the data as an overhead. This method of generation is useful because it means that the new key generated has no relation to the original key. This provides a complete lack of relation between the two keys and hence providing complete security against guess attacks. This method also ensures that the key generation is from a huge space hence enabling a larger guess space. The penalty here is paid in the form of a higher overhead which means that the data here is more since a code to create a secure PRG is to be written which will use more amount of data.



Proposed Model

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 3, March 2016

V. THEORETICAL ANALYSIS OF THE PERFORMANCE OF THE ALGORITHM:

Let us consider that the disk is encrypted with DES algorithm. The key used in DES algorithm is of size 56 bits. Suppose an attacker tries to brute force the data, thus the attacker has to guess from any of the 2^{56} to be the key.Now if we consider n=10, thus the probability that the correct key will be guessed within 10 chances is extremely low hence we can safely assume it as zero. Thus if the attacker enters the wrong key, we re-encrypt the data say using DES again. This re-encryption can be done by any key thus a new 56 bit key will be generated to re-encrypt the data. The effective key length will now become equal to 56(earlier) + 56(new) = 112 bits. This new 112 bits key provides a huge challenge for the attacker to brute force the key hence resulting in thwarting of the attack. This process can be looped again, that is, the data can be re-encrypted again to increase the key size to an extremely high value. This algorithm always provides security level greater than or equal to the security level provided by the standalone encryption (in most cases it provides better security than the stand alone algorithm).

This algorithm can find many applications in places where data sent is extremely sensitive. The algorithm provides high security against cipher text manipulation i.e. the even if by chance the attacker gets hold of the data he cannot add unwanted data to the cipher text to change the data.

VI. DISCUSSION

Even though the proposed algorithm provides high security, it is still has the problem of high overhead data. The extra encryption algorithm needs to be stored in a separate location and has to be called upon when need for re-encryption arises. The key also needs to be stored in a separate file which may also be susceptible to attacks (not encryption attacks but software attacks). This type of encryption algorithm is capable of being used only in CD’s and Flash/Hard drives which have a high amount of data. A more efficient way will be needed for enabling it on real time applications like security on web pages etc. The alternate method used in the network is by using honey tokens or captcha which are being used to stop direct computer attacks on the website hence needing human intervention.

VII. CONCLUSION

This algorithm is effective in terms of safety against any brute force attacks. Also, this algorithm makes sure that the inherent properties of the main encryption algorithm remain unchanged. The main task at hand in this algorithm is its correct implementation in the form of a software will result in a complete change in future of cryptography because the key size keeps increasing indefinitely and hence high amount of protection is provided.

REFERENCES

[1] Juels, A.; Ristenpart, T., ”Honey Encryption: Encryption beyond the Brute Force Barrier,” Security Privacy, IEEE , vol.12, no.4, pp.59,62, July-Aug. 2014
[2] Juels, Ari, and Thomas Ristenpart. ”Honey encryption: Security beyond the brute-force bound.” Advances in Cryptology–EUROCRYPT 2014. Springer Berlin Heidelberg, 2014. 293-310.