University of Ibadan
Journal of Science and Logics in ICT Research

Development of A Honeyed Advanced Encryption Standard Algorithm (HAESA)

Asoro, B. O. raymondblessing5@gmail.com

Emezue, H. C. handel.emezue@gmail.com,

Osunade, O. seyiosunade@gmail.com

Department of Computer Science,
University of Ibadan, Ibadan, Nigeria

Abstract

The rate at which Information is transmitted over the Internet (which is an open channel to all) keeps increasing in recent times. It has become very important for users to make use of encryption schemes that will protect their information from unauthorized parties. This research work presents an easy and safe encryption algorithm for data security since data encrypted with this system cannot be read, edited or copied through illegal access. It provides a middle ground for both speed and security of data regardless of the key strength used for encryption of data. With the use of this improved Honeyed Advanced Encryption Standard (HAESA), attackers will only access plausible looking messages called *honeyed messages*. These honeyed messages are meant to thwart the effort of any illegal access on this system. Honeyed messages are provided by the user of this system. The strength of HAESA lies in the fact that only the actual seeds are encrypted with AES after it must have been encoded with DTE to messages. On decryption, the message will only be displayed to the intended recipient with the right key. From the evaluation results, it was noted that this algorithm assured faster data encryption and decryption time. This research work made use of parameters like throughput and speed.

Keywords: *Cipher-text, Encode Distribution Transforming Encoder (DTE), Encrypt and Decrypt, AES, Honey Encryption.*

1. INTRODUCTION

Online transactions have become a norm for various sectors like health, e- commerce, recruitment processes etc. Users are required to fill forms that includes personal information such as telephone numbers, addresses, and credit card information in cases of business transactions [14]. There is need for users to know that their online information will be used as expected and for no other purposes. According to Abomhara, *et. al,* [1] to protect information, it is imperative to ensure privacy of users from unauthorized persons since the Internet can easily be accessed by anyone. While security is optional for some system users, it's a major priority for others and businesses of any size [2].

From research Zaidan, *et. al.* [22] defined encryption as one of the major methods of data security and privacy. It works by shuffling data so it becomes meaningless to unauthorized parties. Encryption can also be seen as the method of securing a message from every other party except the recipient of that message [20].

As stated by Kessler [8], Cryptography is the science of using ciphers and codes to write messages; to prevent unauthorized persons from accessing message contents through eavesdropping, sniffing, message interception and other methods. Also, cryptography is the use and learning of methods for safe data sharing by preventing attackers from gaining access [3]. The primary functions of cryptography are Privacy/Confidentiality,

Asoro, B. O., Emezue, H. C. and Osunade, O. (2020). Development of A Honeyed Advanced Encryption Standard Algorithm (HAESA), *University of Ibadan Journal of Science and Logics in ICT Research (UIJSLICTR),* Vol. 4 No. 1, pp. 1 – 9

Integrity, Non-repudiation and Authentication [5].

Cryptography can be categorized into two parts; the Symmetric (secret) and Asymmetric (public) key encryption [3]. This research will be focused on symmetric algorithm. The symmetric algorithm is known as a one key system because it uses same key for both encryption and decryption scheme. Example includes Data Encryption Standard (DES), Triple DES, International Data Encryption algorithm (IDEA), Twofish, Blowfish and Advanced Encryption Standard (AES). While the asymmetric algorithm is known as the two-key system. It needs one key for data encryption known as the public key and a separate one for decryption known as the private key. Example includes Diffe- Hellman, Elliptic Curve Cryptography (ECC) and RSA.

From research, it was noted that the Advanced Encryption Standard (AES) and Honey Encryption has limitations. AES takes considerable amount of time to encrypt and decrypt message as the message and key size increases. The weakness of the Honey Encryption scheme lies in the fact that if prior knowledge of the encrypted information is known, then HE security becomes invalid.

However, this research aims to curb these drawbacks with the development and application of a hybridized AES and Honey algorithm. This new algorithm makes use of a faster encryption time regardless of the key and message length.

AES operation is a 4 x 4-dimensional matrix of bytes known as *state*. These states are filled in the matrix column by column. The number of columns denoted by (Nb) is the block size divided by 32. Depending on the nature of the operation used, at each stage of the permutation between plaintext and encrypted data, the block of data is changed from its existing state to the next new state. Before the bytes are processed, they are entered into the *State*. For a block size of 128 bit, the *State* will consist of 4 columns, for a block size of 192 bits, a corresponding 6 columns on the *State will exist* and for 256 bit of block size, there exist 8 columns. Table 1.1 depicts the mapping of a 4 x 4-dimensional matrix of bytes known as *state*.

Table 1.1; Depicting Cipher Input Bytes Mapped onto the State Matrix. Bashyam et al., [2].

A0,A0	A0,A1	A0,A2	A0,A3
A1,A0	A1,A1	A1,A2	A1,A3
A2,A0	A2,A1	A2,A2	A2,A3
A3,A0	A3,A1	A3,A2	A3,A3

Advanced Encryption Standard (AES) is a symmetric block cipher Encryption algorithm with variable key size ranging from 128,192, or 256 bits. The original 128-bit key called *Cipher Key* is divided into eight 16-bit and then used as the first eight key; this procedure is repeated until the last round of encryption has been completed for the chosen key size. Thus, output of one round becomes input for another round.

The data to be encrypted (plaintext) is added with the expanded key derived from the key expansion process. At each state of the round, a part of the *Expanded Key* will be used for that operation. At the second state which is the *initial round,* the AddRound key operation is performed. State three also known as the *Rounds* operation, involves sending the data and the new key from state two for the SubBytes, ShiftRows, MixColumns and AddRoundkey operation to be performed. For a 128 key bit having 10 rounds at this stage, 9 rounds will be performed. The output from state three is then transferred to state four for the 10th and *final round* to be performed (on the key and message).

The final round consists of the SubBytes, ShiftRows, and AddRoundkey. The *Round Key* is added to the current state matrix before starting with the round.

The process of converting the encrypted message into plain message is known as decryption. Each round of AES decryption which is the inverse of encryption process requires the cipher text to go through three of the four basics operations using inverse functions like; InvSubBytes, InvShiftRows, InvMixColumns.

Honey Encryption (HE) originated from a “honey system” which is a system that consist of baits and dummies that aims to deceive attackers. It works by producing honeyed but intelligible words (dummy words that have been mapped in a hash table that helps to detect unauthorized access) for every incorrect use of the key/password. HE consists of two schemes; the onetime pad (OTP) and Distribution Transforming Encoder (DTE) that helps to prevent brute force attack on a system. DTE is made up of a randomizer for encrypting and decrypting algorithms whose function is to encode messages to a particular space.

DTE model the message space M which entails all conceivable messages and maps these messages into seeds space S. Messages are first decrypted with the chosen symmetric algorithm key followed by the DTE decryption algorithm. If key is wisely chosen by the sender of the encrypted message, honey encryption still preserves the message integrity and security.

A fore hand knowledge of the message space M is needed in order for messages to be mapped by DTE to their seed ranges [19]. The size of the seed space has to be huge enough that messages with least chance in the message space is assigned at least one seed.

2. LITERATURE REVIEW

Kessler [8] was able to implement and carry out a comparison on performance based on CPU time, memory and battery power using parameters like speed, block size, and key size on four encryption algorithms; Advanced Encryption Standard (AES), DES, 3DES and E-DES. It was concluded that Educational-DES performs better than other DES, 3DES, and Advanced Encryption Standard (AES) algorithms. The limitation of this study is that E-DES uses 1024 bit which involves a lot of computational resources for key generation.

Gawai and Hakke [4] did a study on AES and Blowfish. Their evaluation was based on encryption speed, CPU utilization time and battery power consumption. The concluded that both algorithms were superior in terms of throughput and processing time.

More and Bansode [11] carried out an implementation of Advanced Encryption Standard (AES) with less time complexity for Various Input like text, video and images when related to existing Advanced Encryption Standard (AES). Mohammad and Al-Hazaimeh [10] Proposed a new approach for complex encrypting and decrypting of data using AES algorithm. They claimed that this new algorithm provides high security level with average encryption and decryption time.

Padmavathi and Kumari [16] surveyed AES, DES, RSA and LSB substitution technique in other to promote the performance of encryption methods and ensure security. They concluded that AES makes use of less encryption and decryption time compared to the other algorithms.

Jain [6] discussed ways to advance the abilities of various algorithms, using a symmetric hybrid cryptographic algorithm of Data Encryption Standard (DES) and International Data Encryption Standard (IDEA) which aim at producing better security for data. The drawback here is the time taken to encrypt data before transmission.

Rani and Kumar [18] carried out an analysis on different parameters of encryption algorithm and they decided that AES provides a very high security level due to the various length of keys while DES and RC4 provides lower security compared to that of AES basically due to their poor key scheduling

Palanisamy and Mary [17] carried out a hybrid cryptography implementation with RSA and Advanced Encryption Standard (AES) for data key security. The downside to this study is that encryption and decryption time will be increased due to the two pairs of keys generated by RSA.

Jules and Ristenpart [7] worked on honey encryption, an algorithm that can withstand brute force attack using honeyed words that gives an attacker the impression that the actual data has been accessed. This algorithm will prevent offline attack on system that uses low-entropy secrets like passwords and also offers protection for partial exposure of high min-entropy keys.

Zaidan *et. al.,* [22] analyzed ten data encryption algorithms using different parameters including speed. They considered algorithms like DES, 3DES, AES and RSA. From the comparison done, AES encryption time was faster. Gawai and Hakke [4] used the idea of honey encryption DTE which makes it possible for an attacker to get a set of valid looking messages making it look as if decryption was successful.

Noorunnisa and Afreen [13] studied Honey Encryption that provides security for weak passwords on sensitive data. In their study, it was observed that in situations of illegal access to data, the system yields a valid looking message to the attacker.

Singh and Supriya [20] proposed the use of Honey Encryption against brute force attack while data is transmitted. With this scheme, the attacker gets to see a dummy message without being any wiser. Nie, Song and Zhi [14] used the concept of honey encryption algorithm to propose a system for data security against brute force decryption by using continuous increase in key size for every attempted brute force attack, thus frustrating the attacker. The drawback of this proposed system is that there will be extra overhead for code generation for every wrong use of password that will lead to data re-encryption.

Mayur and Saraswat [9] presented an algorithm and the types of attacks it is susceptible to. They also did a review on Honey Encryption, exploiting the ways in which weak passwords can be chosen for a given password and how password based encryption schemes are susceptible to attacks. They discussed honey encryption on the one-time-pad and honey encryption with the Distribution Transforming Encoders (DTE). More and Konda [12] made use of Blowfish and Honey Encryption techniques to provide an enhanced algorithm with strong data security against hacking.

From the literature review above, the following observations were noted:

- Short key sizes are not advisable for use since it was observed that the strength of any security lies in its key length.
- Algorithm structures need not be simple if it is to be regarded as a strong one.
- Use of symmetric keys are preferred for encryption especially in cases where message size is large.
- The drawback of using algorithms with larger key sizes is that as the key size increases, the computational time complexity for encryption and decryption of data increases.

3 Methodology

This experimental research work was carried out using a combination of two encryption algorithms (Honey Algorithm and AES), the combined Algorithms were implemented using Java programming language. The encryption and decryption time were used as performance metrics. A new algorithm known as Honeyed Advanced Encryption Standard (HAESA) is proposed in this work shown in Figure 1.

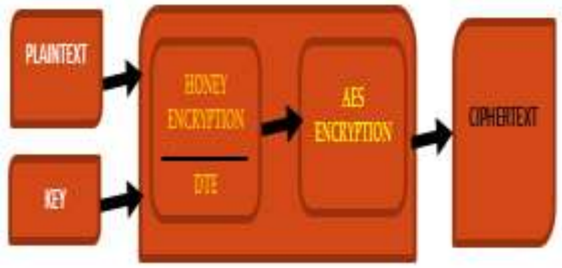


Figure 1. HAESA Encryption Model

Honeyed Advanced Encryption Standard (HAESA) is different from ordinary symmetric scheme due to how data decryption is handled. The messages encrypted with HAESA algorithm can only be accessed correctly if the recipient has the correct seed bit the message was encoded to. The recipient decrypts the seed to access the message with the key generated from Advanced Encryption Standard (AES). The use of a wrong key will generate, honeyed messages. At this stage, the hacker will be unaware of the added security to the data because the system will give the impression that the correct seed has been decrypted thus generating a “dummy message”.

For this work, five honeyed messages were used. These messages were encoded with random string bit of 2^3 seed space. The actual message was assigned one bit while the remaining seven bits were used for encoding the honeyed message. For example, if a user has supplied five dummy messages, then the number of seed bit that can be allowed will be $2^{3.}$ Table 3.1 shows the basic distribution of space bit with the aid of the Look-Up table. If a user should supply just one honeyed message, then the message will be assigned one space which consist of 7bits. If the user supplies two honeyed messages, then the message will be assigned two spaces with 2bits and 5bits respectively, and so on till all messages have been assigned a bit space. These bits will then be encoded to random seed by the DTE.

HAESA Encryption Algorithm

Step 1; Pick a random seed bit between 000 and 111. Assign the actual and honeyed messages to seed bit with the aid of the Look-Up table.

Step 2; Encode both the actual and honeyed messages to their assigned seed bit using the Distribution Transforming Encoder (DTE). Both the DTE and Look-Up table are used to generate the encoding probability randomly. The number of seed space that can be assign to one message is decided by the Look-Up table. Table 3.1 illustrates the Look-Up table.

Step 3; Encrypt the seed encoded to the actual message with AES to get the cipher-text

Table 1. Look-Up Table for Five Honeyed Messages

No of Honeyed Messages	1	2	3	4	5
1	7				
2	2	5			
3	2	2	3		
4	2	2	2	1	
5	2	2	1	1	1

Decryption Algorithm

Step 1; Decrypt cipher-text to get the seed bit the actual message was encoded to using Advanced Encryption Standard (AES).

Step 2; Decode seed using Distribution Transforming Encoder (DTE).

Step 3; If seed corresponds with the actual message, display actual message, otherwise, display one of the honeyed messages.

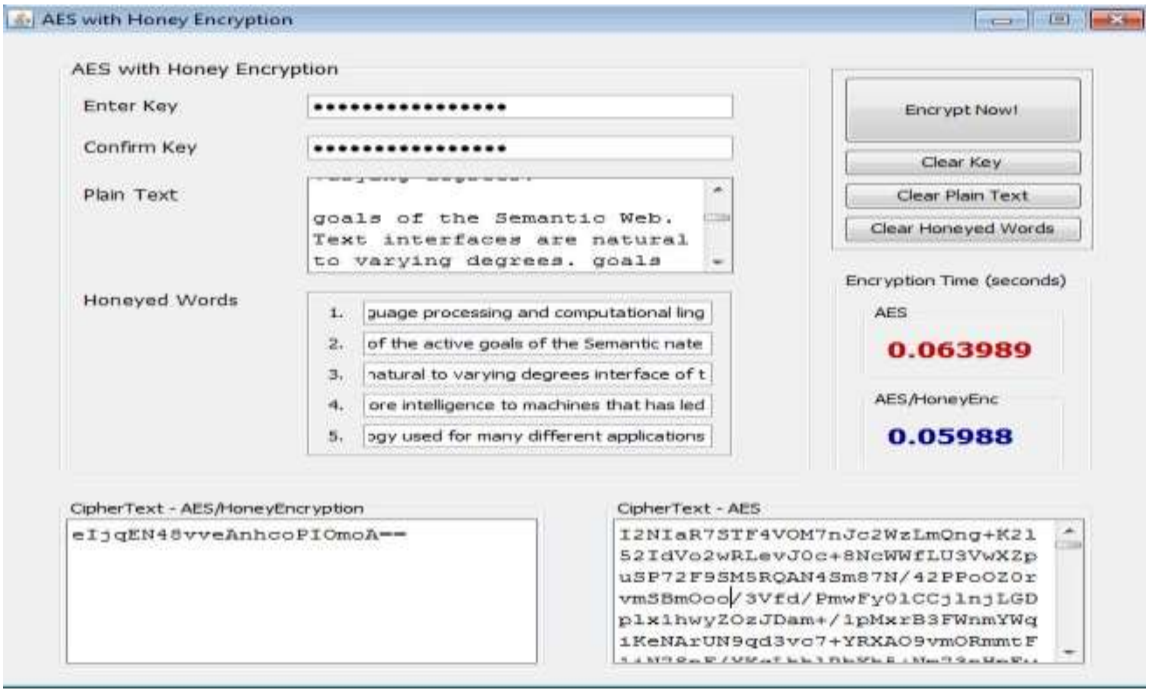
4. Results and Discussion

Table 2: Comparison of AES, Honey Encryption and HAESA with Respect to Different Parameters

S/NO	Basis for Comparison	AES	Honey Encryption	HAESA
1	Algorithm Type	Symmetric	Symmetric	Symmetric
2	Encryption Time	High; It takes considerable amount of time when encrypting large data sizes	Good	Low; It takes a small amount of time regardless of the data size to be encrypted
3	Hardware and Software Implementation	Good and Effective	Good and Proficient	Faster and Effective
4	Power Consumption	High	Low	Low
5	Developed	2001	2014	2017
6	Security Level	Good Security Level	Adequate Security	Excellent Security
7	Use of Wrong Key	It produces meaningless data when a wrong key is used for decryption	It produces a cloned format of the actual data	It produces meaningful but wrong messages when a wrong key is applied

Various data sizes of 32KB, 64KB, 96KB, 128KB, 160KB, 192KB, 224KB, 256KB 288KB, and 320KB were tried out for encryption and decryption of the existing AES against the new encryption scheme. The evaluation of both systems was carried out with respect to computational time which consist of the time it takes each of these algorithms to generate an encrypted data.

Figures 2 and 3 shows the Java program interface. The output of AES and HAESA encryption are given as ciphertext and plaintext. The secret key is used to lock and unlock the message. The time for encryption and decryption is shown on the interface.



AES with Honey Encryption

Enter Key: ●●●●●●●●●●●●●●●●●

Confirm Key: ●●●●●●●●●●●●●●●●●

Plain Text:

```
goals of the Semantic Web.
Text interfaces are natural
to varying degrees. goals
```

Honeyed Words:

1. guage processing and computational ling
2. of the active goals of the Semantic nate
3. natural to varying degrees interface of t
4. ore intelligence to machines that has led
5. ogy used for many different applications

Encryption Time (seconds)

AES: 0.063989

AES/HoneyEnc: 0.05988

CipherText - AES/HoneyEncryption:

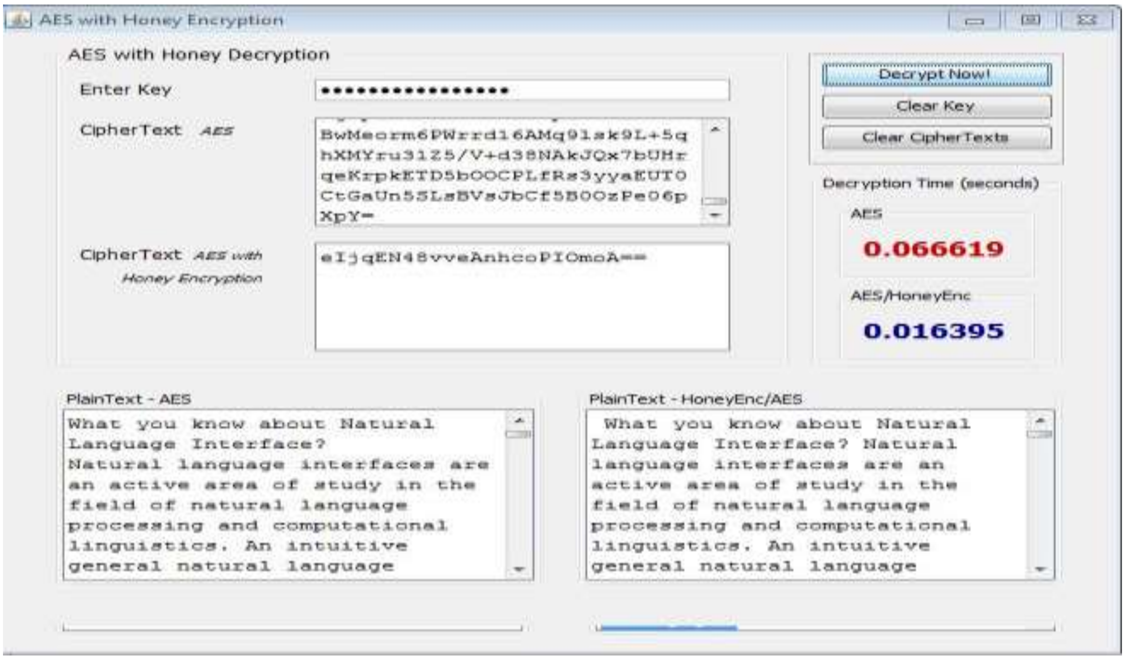
```
eIjqEN48vveAnhcoPIOmoA==
```

CipherText - AES:

```
I2NIaR7STF4VOM7nJc2WzLmQng+K21
52IdVo2wRLevJ0c+8NcWWfLU3VwXZp
uSP72F9SM5RQAN4Sm87N/42PPoOZ0r
vmSBmOoo/3Vfd/PmwFy01CCj1njLGD
p1x1hwyZOzJDam+/1pMxrB3FWnmYWq
1KeNArUN9qd3vo7+YRXAO9vmORmmtF
```

Figure 2: Encryption Interface

The five possible honey words for wrong decryption are provided in Figure 2. The key for decryption is also created at this interface.



AES with Honey Decryption

Enter Key: ●●●●●●●●●●●●●●●●●

CipherText *AES*:

```
BwMeorm6PWrrd16AMq91sk9L+5q
hXMYru3125/V+d38NAkJQx7bUHr
qeKrpkETD5bOOCPLfRs3yyaEUT0
CtGaUn55LsBVsJbCf5BOOzPe06p
XpY=
```

CipherText *AES with Honey Encryption*:

```
eIjqEN48vveAnhcoPIOmoA==
```

Decryption Time (seconds)

AES: 0.066619

AES/HoneyEnc: 0.016395

PlainText - AES:

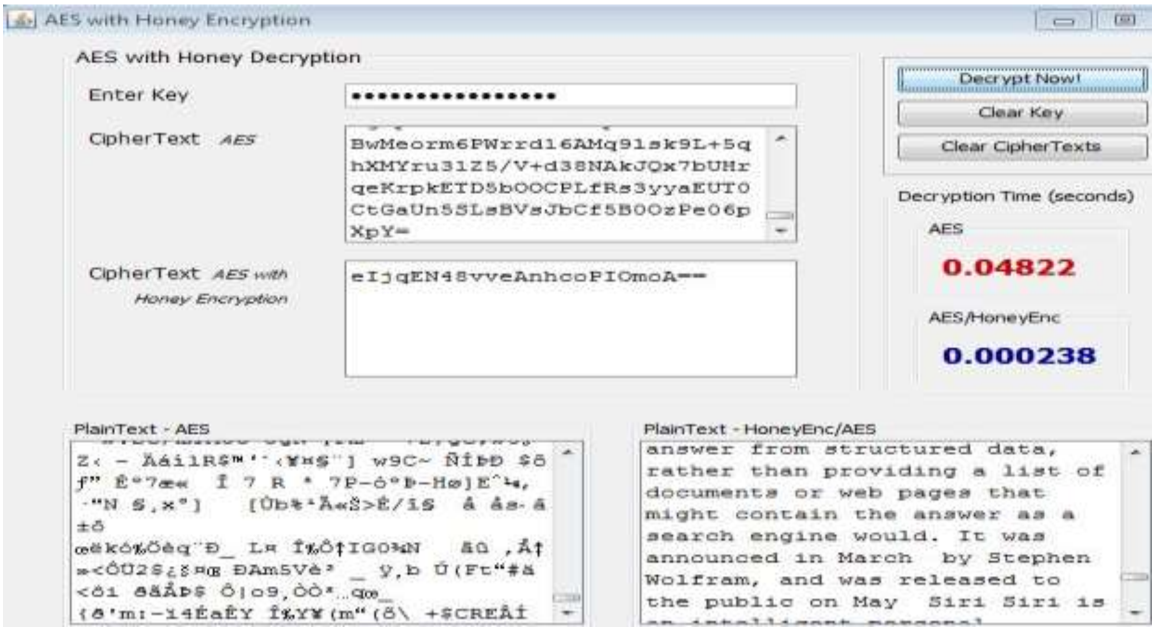
```
What you know about Natural
Language Interface?
Natural language interfaces are
an active area of study in the
field of natural language
processing and computational
linguistics. An intuitive
general natural language
```

PlainText - HoneyEnc/AES:

```
 What you know about Natural
Language Interface? Natural
language interfaces are an
active area of study in the
field of natural language
processing and computational
linguistics. An intuitive
general natural language
```

Figure 3: Decryption with an Actual Key

The plaintext after AES and HAEAS are the same when the actual key is used for decryption as shown in Figure 3.



AES with Honey Decryption

Enter Key ●●●●●●●●●●●●●●●●●

CipherText *AES*

```
BwMeorm6PWrrd16AMq91sk9L+5q
hXMYru31Z5/V+d38NAkJQx7bUHr
qeKrpkETD5bOOCPLfRs3yyaEUT0
CtGaUn5SLsBVsJbCf5B0OzPe06p
XpY=
```

CipherText *AES with Honey Encryption*

```
eIjqEN48vveAnhcoPIOmoA==
```

Decryption Time (seconds)

AES 0.04822

AES/HoneyEnc 0.000238

PlainText - AES

```
Z‹ – ÃáiiR$™'¨‹¥¤$¨] w9C~ ÑÎÞÐ $õ
ƒ" Ê°7æ« Î 7 R * 7P–ô°Þ–Hø]E^¼,
·"N Ŝ,×°]   [Ûb*¹Ã«Š>Ê/ïS  Å Ås· Ã
±ô
œëkó‰Õèq¨Ð_ L¤ Î‰Ô†IGO¾N   Ã0 ,Å†
»<ÔÛ2S¿§¤Œ ÐAm5Vè³ _ ÿ,b Ű(Ft"#Ä
<ô1 ØãÅÞ$ Ô|o9,ÒÒ*…qœ_
{Ø'm:–ì4ÉaÊY Î‰Y¥(m"(ð\ +$CREÂÍ
```

PlainText - HoneyEnc/AES

```
answer from structured data,
rather than providing a list of
documents or web pages that
might contain the answer as a
search engine would. It was
announced in March  by Stephen
Wolfram, and was released to
the public on May  Siri Siri is
```

Figure 4: Decryption with a Wrong Key

Figure 4 shows when a wrong key has been applied for decryption process.

Table 3. Showing Encryption and Decryption Time of Data

Size Of Plaintext (KB) Encryption Time	Encryption Time		Decryption Time	
	AES	HAESA	AES	HAESA
32	0.024877	0.000135	0.022416	0.000134
64	0.071651	0.000094	0.135578	0.000228
96	0.156706	0.000083	0.290571	0.000199
128	0.228167	0.000089	0.416679	0.000159
160	0.300719	0.000064	0.540473	0.000229
192	0.364601	0.000093	0.681929	0.000279
224	0.525867	0.000068	1.001428	0.000174
256	0.565172	0.000301	0.093235	0.000456
288	0.593788	0.000087	1.106886	0.000172
320	1.014405	0.023611	1.297376	0.097829

5. Conclusion

The goal of this research work is to develop a better and faster symmetric algorithm for data security. This research work has been able to achieve it with the aid of the DTE and look-Up table that encode seed bit associated to each message which are in turn encrypted in place of the message; thus, making it faster than the conventional Advanced Encryption Standard (AES) by reducing encryption and decryption time. Also, this work assures complete data secrecy by giving an attacker plausible looking messages when a wrong key is being used for decryption. Financial transactions between businesses and high-profile clients will benefit from this work.

References

[1] Abomhara, M., Zakari. O., Othma. (2010). An Overview of Video Encryption Techniques. International Journal of Computer Theory and Engineering, Vol. 2, No. 1, 1793-820

[2] Bashyam, S.L.R., Shankar, K., Kadiyala, S., & Abuzneid. (2015). A Hybrid Cryptography Using Symmetric Key Encryption. Department of Electrical

Engineering and Computer Engineering, University of Bridgeport, CT.

[3] Goyal, S.: A Survey on the Applications of Cryptography. (2012). International Journal of Science and Technology olume 1 No. 3.

[4] Gawai, S., & Hakke. (2016) S Honey Encryption: Encryption beyond the Brute-Force Barrier. International Journal of Advance Research in Computer Science and Management Studies. Volume 4, Issue 3.

[5] Haldankar, C., & Kuwelkar, S. (2014). Implementation of AES and Blowfish. IJRET: International Journal of Research in Engineering and Technology. Volume: 03 Special Issue: 03.

[6] Jain, M. (2014). "Implementation of Hybrid Cryptography Algorithm,". International Journal of Core Engineering & Management (IJCEM). Volume 1, Issue 3.

[7] Jules, A., & Ristenpart, T. (2014)" Honey Encryption: Encryption beyond the Brute Force Barrier," Security Privacy, IEEE, vol.12, no.4, pp.59,62.

[8] Kessler, G.C. (2016). An Overview of Cryptography. http://www.garykessler.net/library/crypto.html

[9] Mayur, T., & Saraswat, L. (2016). A Review on Common Encryption Techniques to Brute Force Shielded Technique: Honey Encryption. IJSRD – International Journal for Scientific Research & Development. Vol. 3, Issue 12.

[10] Mohammad, O. & Al-Hazaimeh, A. (2013). A New Approach for Complex Encrypting and Decrypting Data. International Journal of Computer Networks & Communications (IJCNC) Vol.5, No.2. DOI: 10.5121/ijcnc.2013.5208

[11] More, S., & Bansode, R. (2015). Implementation of AES with Time Complexity Measurement for Various Input. Global Journal of Computer Science and Technology: E Network, Web & Security. Volume 15, Issue 4

[12] More, R., S, Konda, S, S. (2016). Resilient Security Against Hackers Using Enhanced Encryption Techniques: Blowfish and Honey Encryption. International Journal on Recent and Innovation Trends in Computing and Communication ISSN:2321-8169 Volume: 4 Issue: 6 98 – 102

[13] Noorunnisa, N.S & Afreen, K. R. (2016). Review on Honey Encryption Technique. International Journal of Science and Research (IJSR). Volume 5, Issue 2.

[14] Nie, T., Song, C., & Zhi X. (2010)."Performance Evaluation of DES and Blowfish Algorithms". IEEE International Conference on Biomedical Engineering and Computer Science (ICBECS), pp. 1-4

[15] Nikhilesh, S. J. R. S., & Kumar, S. V. (2016). A Study on a New Algorithm for Data Encryption against Brute Force Attacks. International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering. Vol. 5, Issue 3.

[16] Padmavathi, B., & Kumari, R. S. (2013). A Survey on Performance Analysis of DES, AES and RSA Algorithm along with LSB Substitution Technique. International Journal of Science and Research (IJSR). Volume 2 Issue 4

[17] Palanisamy, V. & Mary, J. (2011). Hybrid Cryptography by the Implementation of RSA and AES. International Journal of Current Research Vol. 33, Issue, 4, Pp.241-244.

[18] Rani, M., Kumar, S. (2015). Analysis on Different Parameters of Encryption Algorithms for Information Security. International Journal of Advanced Research in Computer Science and Software Engineering. Volume 5, Issue 8,

[19] Rohini., & Smita. (2016). Resilient Security Against Hackers Using Enhanced Encryption Techniques: Blowfish and Honey Encryption. International Journal on Recent and Innovation Trends in Computing and Communication. Volume: 4 Issue: 6 98 –102

[20] Singh, G., & Supriya. (2013). A Study of Encryption Algorithms (RSA, DES, 3DES and AES) for Information Security. International Journal of Computer Applications. Volume 67– No.19

[21] Vinayak P P., & Nahala, M. A. (2013). Avoiding Brute Force Attack in MANET using Honey Encryption. International Journal of Science and Research (IJSR). Index Copernicus Value

[22] Zaidan, B.B., Zaidan, A.A., Jalab.A.H., & Al-Frajat, A.K. (2010). On the Differences between Hiding Information and Cryptography Techniques: An Overview. Journal of Applied Sciences, 10: 1650-1655