A Novel Approach for the Adaptation of Honey Encryption to Support Natural Language Message

Abiodun Esther Omolara, Aman Jantan, Oludare Isaac Abiodun and Howard Eldon Poston

Abstract— **Modern cryptosystems, even though they give confidentiality do not provide resilience against brute-force attacks. Honey encryption (HE) was proposed as a countermeasure to this problem of cryptography. HE is an encryption scheme that supplies valid-looking, but fake plaintext for every incorrect key used by an intruder to decrypt a message. All possible messages relative to the original message are mapped to a seed space such that any key supplied by the attacker when decrypting a message produces a relative, but fake message from the original message and this makes it difficult for him to determine if he has recovered the original message or not. However, a challenging problem with HE is adapting it to natural language message to produce convincing fake messages for documents such as emails. We propose a novel approach of generating decoy message using Stanford Dependency Parser and Wordnet from Princeton. The result shows that the proposed scheme effectively produces convincing decoy messages that fools the attacker. In addition, the structure, length and content of the original message are concealed. Finally, we verify the effectiveness of the proposed scheme by checking the entropy of decoy messages from the plaintext.**

***Index Terms*— Ciphertext, Brute-Force, Distribution Transforming Encoder (DTE), Honey Encryption (HE), Natural Language Processing (NLP), Plaintext**

I. Introduction

Cryptology continues to co-evolve with state-of-the-art communication and computing technologies [1], [2]. The reason, indisputably, is the evolution of computer and introduction of the internet which led to a rapid influence of technology and digital media on every aspect of human lives. It is a common knowledge that almost all application that fulfils some computing task, when deployed in the real world becomes susceptible to different types of attacks. Modern cryptographic encryption schemes use an n-bit key, where the security of the encryption increases with the size of the key. These schemes are considered secure because they are formulated on well-defined hard problems and they are concerned with the possibility or impossibility of securely realizing the message. The answer to whether a message can be acquired depends on the assumed power of the adversary. This security holds in a solid mathematical sense, implying that breaking a secure cryptographic scheme is either impossible or it would require solving some computational problem which currently has no solution. Notwithstanding, with enough computational power and time, these schemes are susceptible to brute-force attacks. In addition, an intruder can confirm a successful decryption of a ciphertext if his attack yields a valid-looking message, but more importantly, an invalid-looking output confirms an unsuccessful attempt. This is easily confirmed because all messages either in storage or transition have some structure – for instance, text, images, audio files, videos and this makes it easy for the intruder to determine if he has the correct structure or not during his attempt to acquire the message [3].

In information security, decoy or honey objects are powerful tools used to detect and distract an adversary from gaining access to a resource. Honeywords, honey accounts, honeytokens, honeypots or honey servers [4] - [6] are false resource that serves as baits to lure an adversary from gaining access to a system. Honey encryption proposed by [7] - [9] offers a counter-measure to this flaw of modern encryption scheme by supplying a valid looking but fake message when an intruder tries to decrypt a message using incorrect keys. A ciphertext that is honey-encrypted has the attribute that attempted decryptions with invalid keys yield valid-looking decoy messages. Thus, an intruder employing a brute-force attack gain no information from guessing and checking of keys.

The HE scheme was proposed within the context of password security where keys are of minimum entropy. Extending the scheme to work in other settings such as encoding reasonable sized human documents such as email poses a serious challenge. This is because it requires generating fake content and context-sensitive messages relative to the original messages while still hiding the content and structure of the original message. A number of researchers have made several attempts to solve this problem of adapting the scheme to human language message yet there has been no progress in such regards. A study by Bernadeau et al. [3] tried to extend the scheme to support encoding human message. Their technique produced convincing decoy messages for human message. However, partial content of the plaintext is revealed and their method fails to produce sane messages in some instance. These loopholes in their system may help the attacker acquire the plaintext. The aim of this paper is to give a detailed background of the honey encryption scheme,

Manuscript received December 27, 2017; revised January 16, 2018. This research was partially supported by the Fundamental Research Grant Scheme (FRGS) for ‘‘Content-Based Analysis Framework for Better Email Forensic and Cyber Investigation’’ [203/PKOMP/6711426], SFRG Lab, School of Computer Sciences, USM

Abiodun Esther Omolara (e-mail: styleest2011@gmail.com)
Aman Jantan (corresponding author’s email: aman@usm.my)
Oludare Isaac Abiodun (email: aioludare@gmail.com)
Howard Eldon Poston (email: howard.poston@gmail.com)
Security and Forensic Research Group, School of Computer Science, Universiti Sains Malaysia, Penang

cryptanalyze the state-of-the-art and propose a novel method of extending the scheme to support secure encoding of human message. The key contributions of this paper can be summarized as follows:

1. This paper describes a detailed attack analysis using both the conventional and honey encryption method and shows how the honey encryption scheme successfully deceives an adversary in a minimum-entropy key setting.
2. This paper gives an argument on why the present method of adapting the HE scheme fails to model human-generated message.
3. We propose an algorithm that produces decoys/fake messages for natural language messages. It produces reasonable length decoy messages capable of fooling the adversary.
4. The message structure in the proposed scheme is kept entirely secret and failed decryption produces radically different messages from the original messages.

The overall structure of this study takes the form of five chapters, including this introductory chapter. Chapter two begins by laying out the background studies and gives a detailed study of the scheme. The third chapter is concerned with the methodology used for this study. The fourth chapter presents the evaluation/experiment of the research. Finally, the fifth chapter gives a brief conclusion of the findings.

II. BACKGROUND STUDIES

A. *Terminologies*

This section outlines some terminologies used in the paper for basic understanding:

Attacker: Words like adversary, intruder or eavesdropper will be used interchangeably to mean an attacker.

Plaintext: It refers to a message in a readable format before it is encoded. We use any of the term original text/original messages, true-text to refer to the plaintext in this context.

Ciphertext: Ciphertext refers to the encoded plaintext; a cipher (an algorithm) is used to transform the plaintext to be in an unreadable format. Terms like, decoy, fake-text, false-text, honey messages is used interchangeably to mean the ciphertext in this context.

Message Space: The message space represented with M contains all possible messages M* of the plaintext. It is a set of all the fake messages modelled from the original message and tailored to appear like the real message.

B. *Conventional Encryption Scheme Setting*

In Conventional encryption, the sender and the receiver agree upon a secret key which is used in encrypting and decrypting by both parties. An adversary performing a brute-force attack gets gibberish (non-uniform distribution) or an error symbol as the expected output when he tries a wrong key. This output is a pointer that the key he is trying is an incorrect key. He continues his search till he gets the plaintext. During his attack, he quickly discards the output message when the distribution is non-uniform. This gives him more time to continue his search. The probability of acquiring the plaintext is high. Fig. 1 shows a transmission between two parties who share the same key during an encryption/decryption process and the output an attacker gets when he tries decrypting with an incorrect key. The sender uses a preshared key and an encryption algorithm to encrypt a message 'Hello Bob'. He sends the ciphertext to the receiver. The receiver uses the key to decrypt the message. An attacker using a different key to decrypt the message gets an invalid-looking output.

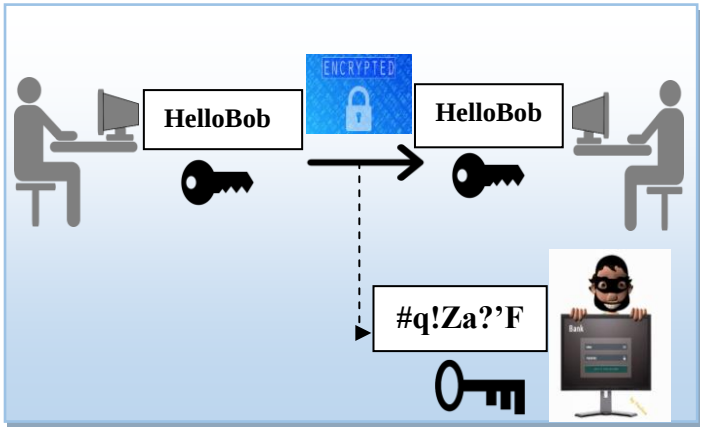

Fig. 1. Conventional Encryption Scheme

C. *Honey Encryption Setting*

Honey Encryption is an encryption scheme that serves up plausible-looking but fake message as a response to every invalid key supplied by an adversary during a brute-force attack. Thus, the adversary gains no information.

Threat Model: For an encryption C=enc(M, K) of message M. If K and M are drawn from a known distribution. The target of an adversary is to recover the message M. He tries to decode C using different keys. For every key he tries, he gets M1,..., Mn. For a minimum entropy distribution like passwords, M is guaranteed to appear on his list. This is possible because users choose simple passwords that can be easily guessed. Also, attackers are aware of how users choose their passwords (from previously released details of leaked passwords on the internet). Therefore, the security here depends on the probability of the adversary been able to pick the message M from all n possible messages should one of the keys he tried was correct. The Fig. 2 shows the output an attacker gets when he tries an invalid key using the honey encryption scheme.

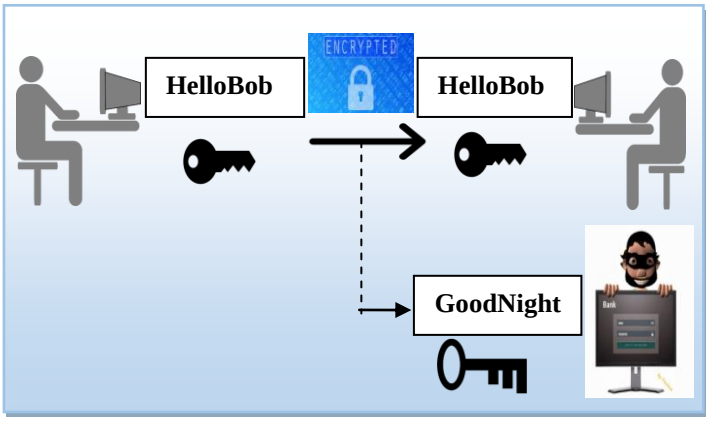

Fig. 2. Honey Encryption Scheme

ISBN: 978-988-14047-8-7
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

The sender encrypts the message 'Hello Bob' and sends to the receiver. The receiver decrypts the exact message using the same key as the encoder. An attacker using a different key gets a valid-looking output. In this case, he gets 'Good Night'. Therefore, the attacker is confused and cannot tell if he has the plaintext or not.

D. Distribution Transforming Encoder (DTE)

The HE scheme is designed with a cryptographic primitive called the Distribution Transforming Encoder (DTE). The DTE is a set of algorithm DTE= (encode, decode), where encode takes a *Message Space* M as an input and returns a value in the *Seed Space* S as output. Decode takes as input a value S and returns an output message M. Honey encryption involves a DTE-and-then-encrypt process. This means a sender applies the DTE to the original message he intends encoding and then uses any conventional encryption scheme as the second layer of encryption. The DTE represents the model of the message. A good DTE is designed to model the message distribution well such that if a seed is selected uniformly at random and applied to it, the message is recovered. The intuition here is to make the encoding process randomized to provide proper secrecy and make the decoding process deterministic.

The algorithm of the encode and decode process of HE is shown in Fig. 3.

HEnc(K,M)	HDec (K,C)
$S \leftarrow\$$ encode(M)	$(R,C_{2)} \leftarrow C$
$R \leftarrow\$ \{0,1\}^{rl}$	$S \leftarrow$ H(R‖K) $\oplus$ C_2
$C_2 \leftarrow$ H(R‖K) $\oplus$ S	$M \leftarrow$ Decode (S)
Return (R,C_2)	Return M

Fig. 3. The DTE-then-Encrypt Algorithm. H is the Hash Function, R is an integer representing random bits for the encrypt process, M is the Message or Plaintext, C is the Ciphertext.

E. State-of-the-Art

To date, various methods have been proposed to extend the HE scheme to support its adaptation to natural language message.

Jo et al. [11] proposed the statistical code system. In this proposal, the standard HE scheme is unified with a structural code system. This proposal generates plausible false text relative to the plaintext. However, the ambiguity between the plaintext and the false text is much. The attacker may use this vulnerability to recover the difference between the false text and plaintext and acquire the target message.

Chatterjee et al., [12] proposed a Natural Language Encoder called the NoCrack. The proposal is specifically for protecting Password Vaults/Manager. The intuition is to generate fake but realistic-looking vault to the attacker in the face of a brute-force attack. The attacker is not able to tell if it is the original or fake vault he has acquired. The system also forces the attacker to go online where his activities can be traced and prohibited. This proposal works reasonably well for password-related settings but cannot be extended to support large human-text.

Huang et al., [13] expanded the standard Honey encryption scheme to support encoding of genomic data. This proposal suggested techniques for securing genetic materials and also protecting the genomes from mauling by an attacker with an unbounded time.

Golla et al., [14] analysed the proposal by Chatterjee et al. [12] and explained why it fails to model human language. This proposal uses statistical tools to build their adaptive NLE. Their method increased the message space, allowing more instances of online guessing of the original vault. However, this technique does not scale well as it cannot be extended to support the human-generated message.

Jaeger et al. [15], pointed out how message recovery setting in the standard HE scheme is lacking. They suggested ways of strengthening the scheme to conceal partial information of the target message while still providing security and protecting the acquired message from mauling.

Kim et al. [16], proposed the statistical code scheme HE. In this proposal, an adversary eavesdropping on their conversation at another end of the channel is supplied with valid-looking but fake chat message when he tries his incorrect keys.

Yoon et al.[17], proposed the visual HE which employs an adaptive DTE in a Bayesian framework. This proposal introduced a novel method of using the Bayesian framework to secure images and videos to produce fake but normal-looking videos to an adversary during transmission of images/videos.

Tyagi et al.[18] implemented the standard HE scheme on short messages and PINS. Chatterjee et al.[19], Choi et al. [20] proposed techniques of solving typo problems in the honey encryption scheme.

From our studies, all methods proposed by [11-18] worked relatively well for securing passwords but failed to accommodate its extension for the support of human-generated messages. To our knowledge, the only research that tried to address the open problem was by Bernadeau et al.[10]. We shall discuss the state-of-the-art approach proposed by Bernadeau et al.[3]. This proposal contended and proved why the methods of [11-18] failed to support the human-written message. According to them; the approaches used by [11-18] "*failed to model even simple sentences – let alone entire documents"*. For more details on their work, see [3]. The authors proposed the Corpus Quotation Distribution Transforming Encoder.

In the proposal described by [3], a list of grammatical roles is described where words are tagged using a clause level, phrase level, and word level labels. This approach uses an existing code-book with text and intervals to encode a combination of words. The words used to encode the message is realized inside an agreed public document. Therefore, the message takes the structure of the human language from the specified document. The grammar of a language is used in this approach to simplify the structure and a re-writing rule is applied to shuffle the message. Their use of grammatical role is an innovative approach as it produces realistic fake messages. However, a serious limitation of their scheme is that:

I. A user is restricted to quote only from a known public document. This presents a serious challenge as it is unlikely that a user will be able to encode a text

ISBN: 978-988-14047-8-7
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

from a computer-based domain using a code-book from a micro-biology domain.

II. Subjecting users to quote from a particular source may help to hide the empirical property of language but it does not show how human use their natural language in writing and sending messages.

III. A single vulnerability where the language distribution of the document is known will compromise the whole setting.

IV. Their approach produces weird sentences as honey messages when used in some context. For instance the sentence; "During his youth, Alex was tutored by a skilled architect until the age of 16". Architect was defined as a noun, so the decoy version might result in "Alex being tutored by a Baboon," which is obviously wrong.

V. The original idea of the HE scheme is to act as an additional layer to another known scheme. The basis of their scheme was that having the correct key will reproduce the original message. This fails to give security in some instances. For instance- if AES is used with their method (in this case, intervals represents the position in the codebook), the model impacts the text and fail to generate valid-looking fake text. For example, if we encode the interval position with 8-bit numbers we will allow only 256 symbols of the text to be used for encoding; in 16 bit that would be 65636 symbols and in average 10000. If the attacker tries to decode the message interval position with a large number of small text, he will be able to discard the key right away. When testing, some keys will be cancelled out. In this case, it will be possible to make an option to generate results produced by all keys combinations of a specific size.

III. Methodology

A. Natural Language Toolkit (NLTK)

As stated in the introduction, the primary objective of this research is to design an algorithm that generates fake but valid-looking message from the original message encoded so as to deceive the attacker. We are concerned with how human language is generated. Therefore, we opted for tools that work with human language.

The Natural Language Toolkit (NLTK) was used for parsing, tagging, classifying and organizing each message words by words, then processed as sentences to form the message [21], [22]. In this scheme, the syntactic structure of a sentence is described simply in terms of the words in a sentence and an associated set of directed binary grammatical relations that hold among the words.

B. Wordnet

This research uses a large lexical database referred to as Wordnet. The database contains nouns, verbs, adjectives and adverbs and so on. They are all grouped into sets of cognitive synonyms referred to as synsets. All words in Wordnet are stored in nodes containing sets of synsets. All synsets have an id that can be used to uniquely identify them and it is possible to extract all synsets meeting certain requirements (same part of speech, etc.) from Wordnet. These two features are used to encode words within the proposed scheme [23] – [25].

C. Proposed Method

The proposed system is divided into the encoding and decoding process. The framework for the system is shown in Fig. 4. Messages are encoded and sent by the sender and at the other end, they are decrypted by the receiver.

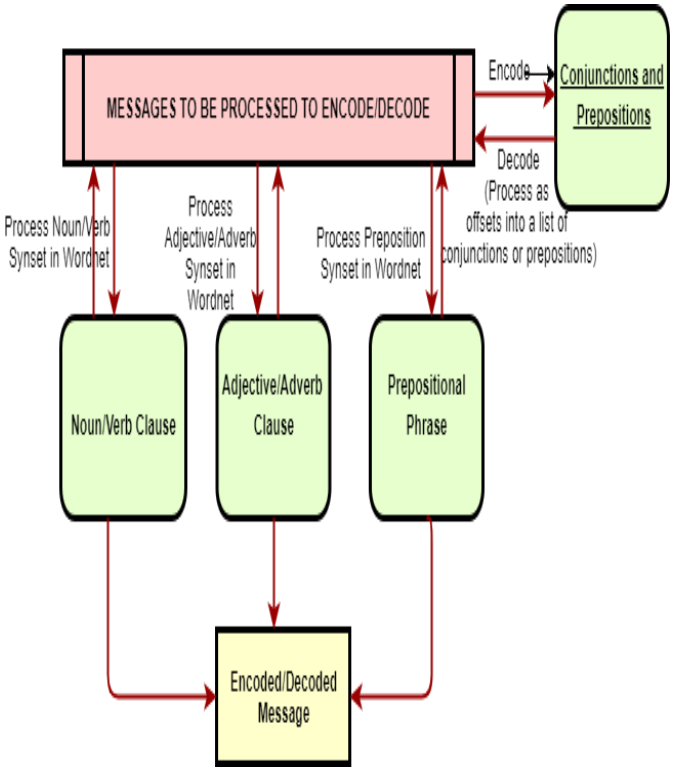

Fig. 4. A Framework of the Proposed Scheme

D. Pseudocode for Encoding Message

Using Standford Dependency Parser, each sentence in the message is parsed into a tree of Noun, Verb, Adjective, Adverb or Modifier Phrase for encoding. The steps for encoding a noun is shown below:

```
Step 1: Get synsets for noun
Step 2 : If personal noun
Step 3 : Drop synsets with non-personal meaning for noun
Step 4: else
   Drop synsets with personal meaning for noun
Step 5 : If synset list is non-empty,
   Randomly select synset from list
Step 6: else
   if noun is pronoun
Step 7: Set offset to length of list of nouns in wordnet plus one and encode in binary
Step 8 : Encode pronoun
Step 9     else
   Set offset to length of list of nouns in wordnet plus two and encode in binary
Step 10: Encode unknown word
   return binary string
Step 11: Get path from root of wordnet noun tree to selected synset
Step 12: If personal noun
```

ISBN: 978-988-14047-8-7
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

Drop nodes in path that are parents to "person noun" synset
Step 13: Return binary strings

E. Pseudocode for Decoding the encoded message described above

The steps for decoding a noun is shown below:

```
Step 1: Extract offset from binary string
Step 2: If pronoun
Step 3: return decoding of pronoun
Step 4: If unknown word
Step 5: return decoding of unknown word
Step 6: Get synset of noun at given offset in list of wordnet's
nouns and set as root
Step 7: While not at desired node (first character in binary
string is non-zero)
Step 8: Extract offset from binary string
Step 9: Set child at offset in list of root's children to new
root
Step 10: Extract lemma number from binary string
Step 11: Get specified lemma
return name of lemma (desired noun)
```

Other parts of speech such as adverbs, adjectives et cera are processed using the algorithm presented above.
Pronoun encoding/decoding: Have list of pronouns, encoded as offset in the list.
Unknown encoding/decoding: Each letter encoded as position in alphabet (0-25)

IV. IMPLEMENTATION AND EXPERIMENTAL RESULTS

Information theory defines entropy as a measure of the uncertainty of a dataset. The entropy of a dataset is proven to be high if the dataset is composed randomly [26].
The intuition here is for the transmitter to encode and send the original message to the receiver and if it is intercepted using incorrect keys then the original message must not be predictable from the decoy message his key generates. Furthermore, when the adversary guesses the right key, he should still not be able to figure out if it the original message he has acquired from all the other message he has recovered from his attack.

A. Experiment

To evaluate the performance of the proposed Honey encryption scheme, we considered the entropy of the decoy message. Subsequently, we check if the criterion of confidentiality and indistinguishability described in section (2E) above is fulfilled.
A hypothesis test was conducted to distinguish between decoy messages using the correct key K and incorrect keys K*. We used a typical email message as our plaintext to test our scheme. We encode the email message C= Enc(M, K) using the correct key K. Then, we try to decrypt the message using random but incorrect keys K* for 10,000 times. For all the 10,000 tries, we generate fake messages M*=Dec(C, K*) completely different from the original email message M encoded.
For the null hypothesis H_0, we state that; “There is no difference of entropy between the Plaintext M and the fake/decoy messages M*”. By this, we mean that the entropy of M is included within the scope of other messages M* (the message space). The alternative hypothesis H_1 is “There is a difference of entropy between the Plaintext M and the decoy message M*”, In this argument, it means that M and M* can be distinguished. Let test statistics M* be considered as the entropy distribution of decoy messages while the entropy of the original message M as observed value. The small P-value represents that the observed data M could not be included in the range of M*. Therefore, we reject the null hypothesis H_0 and accept H_1. This can be interpreted to mean that for all our incorrect key trials, we generate fake messages that are different from the original message.
The proposed method conceals the content and structure of the true message. Also, it secures the length of the original message. The messages scales, they produce real-looking decoy. Also, an attacker cannot distinguish the fake message from the original plaintext. The Fig. 5 shows the original plaintext that was transmitted. An attacker trying several keys is presented with the version of Fig. 6 and Fig. 7. There is no way the adversary can tell the original message from the fake message even if he acquires the original message.

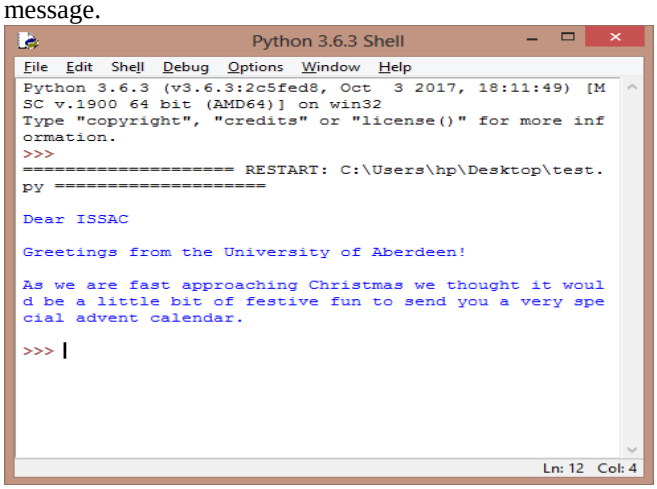

Fig. 5. Email-Message (Plaintext)

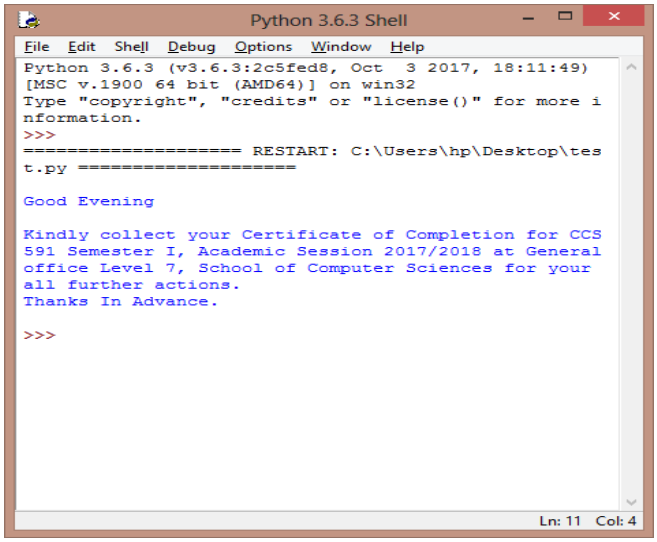

Fig. 6. Intercepted Message acquired by Attacker using incorrect keys

ISBN: 978-988-14047-8-7
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

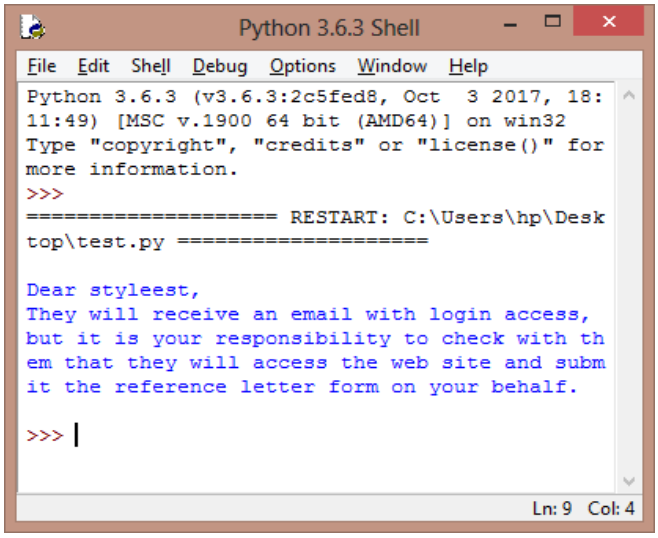

Fig. 7. Intercepted Message acquired by Attacker using incorrect keys

V. CONCLUSION

This paper proposed a cutting-edge solution for adapting the HE scheme to support encoding and decoding of human message such as emails, large written documents. The proposed scheme is implemented using Stanford's Dependency Parser from NLTK and Wordnet from Princeton. It was designed and tested using Python 3.6.3. We described an attack model where an email was intercepted and the attacker uses his incorrect key to try accessing the contents of the mail. It was observed that his incorrect-keys generate plausible-looking but fake messages. He has no way of determining if his attack recovered the real message or a fake message. If during his search he uses the correct key, he is still unable to tell the original message from the fake message he has acquired.

In conclusion, the structural skeleton, the contents and length of the original message are completely concealed in the face of an attack. To the best of our knowledge, this is the first attempt at successfully generating a reasonable-length human-language decoy message that fools humans and automated tools. This paper presents the initial results of our work. More comprehensive evaluation is being conducted to check for the speed, noise and enhance other features of the proposed scheme. More results will be reported in our future papers.

ACKNOWLEDGMENT

We gratefully acknowledge the constructive comments of the IMECS 2018 reviewers.

REFERENCES

[1] W. Diffie and M. Hellman, "New directions in cryptography." *IEEE Transactions on Information Theory*, vol.22, no. 6, pp. 644-654, November 1976.

[2] D. Chaum, R. L. Rivest and A.T. Sherman, "Advances in cryptology." *Proceedings of CRYPTO*. Vol. 82. 1983.

[3] M. Beunardeau, H. Ferradi, R. Géraud and D. Naccache, "Honey Encryption for Language." *In International Conference on Cryptology in Malaysia*, Springer, Cham, pp. 127-144, 2016.

[4] A. Juels, "A bodyguard of lies: the use of honey objects in information security," in *SACMAT*, pp.1-4, 2014.

[5] B. M. Bowen, S. Hershkop, A. D. Keromytis, and S. J. Stolfo, "Baiting Inside Attackers Using Decoy Documents," pp. 51–70, 2009.

[6] B. M. Bowen, V. P. Kemerlis, P. Prabhu, A. D. Keromytis, and S. J. Stolfo, "Automating the injection of believable decoys to detect snooping," in *WiSec. ACM*, pp. 81–86, 2010.

[7] A. Juels and R. L. Rivest, "Honeywords: making password cracking detectable," in Proceedings of the *ACM SIGSAC Conference on Computer and Communications Security*, pp. 145-160, ACM, Nov. 2013.

[8] A. Juels and T. Ristenpart, "Honey Encryption: encryption beyond the brute-force barrier," *Security Privacy, IEEE*, vol. 12, no.4, pp.59-62, July-Aug., 2014.

[9] A. Juels and T. Ristenpart, "Honey Encryption: security beyond the brute-force bound," *Advances in Cryptology-EUROCRYPT 2014*, vol. 8441 of Lecture Notes in Computer Science, pp. 293-310, Springer Berlin Heidelberg, Germany, 2014.

[10] W. Stallings and M. M. Manna. *Data and Computer Communications*. Vol. 6. Englewood Cliffs, NJ: Prentice-Hall, 1997.

[11] H. Jo and J. Won, "A new countermeasure against brute-force attacks that use high-performance computers for big data analysis," *Hindawi Publishing Corporation International Journal of Distributed Sensor Networks*, pp. 7, 2015. http://dx.doi.org/10.1155/2015/406915

[12] R. Chatterjee, J. Bonneau., A. Juels and T. Ristenpart, "Cracking Resistant Password Vaults using Natural Language Encoders," *Proceedings – IEEE Symposium on Security and Privacy*, no. 7163043, pp. 481-498, July 2015.

[13] Z. Huang, E. Ayday, J. Fellay, J. Hubaux and A. Juels, "Genoguard: Protecting genomic data against brute-force attacks," *IEEE Symposium on Security and Privacy*, pp. 447-462, 2015. DOI 10.1109/SP.2015.34

[14] M. Golla, B. Beuscher and M. Durmuth, "On the security of cracking-resistant password vaults," *Proceedings of the ACM Conference on Computer and Communications Security*, Vol. 24, no. 28, pp. 1230-1241, Oct. 2016.

[15] J. Jaeger, T. Ristenpart and Q. Tang, "Honey encryption beyond message recovery security," *International Association for Cryptologic Research*, Fischlin and J.-S. Coron (Eds.): EUROCRYPT 2016, Part I, LNCS 9665, pp. 758–788, 2016. DOI: 10.1007/978-3-662-49890-3 29

[16] J. Kim and J. Won, "Honey chatting: A novel instant messaging system robust to eavesdropping over communication," *IEEE In Acoustics, Speech and Signal Processing (ICASSP)*, pp. 2184-2188, 2016.

[17] J. W. Yoon, H. S. Kim, H. J. Jo, H. L. Lee, and K. S. Lee, "Visual honey encryption: Application to steganography," *in Proceedings of the 3rd ACM Workshop on Information Hiding and Multimedia Security*, New York, NY, USA, 2015, IH&MMSec '15, pp. 65–74, ACM.

[18] N. Tyagi, J. Wang, K. Wen and D.Zuo.: Honey Encryption Applications. 6.857 *Computer and Network Security, Massachusetts Institute of Technology*. (2015) http://www.mit.edu/~ntyagi/papers/honey-encryption-cc.pdf

[19] R.. Chatterjee, A. Athalye, D. Akhawe, A. Juels, and T. Ristenpart, "Password typos and how to correct them securely," *In Security and Privacy (SP), 2016 IEEE Symposium*, pp. 799–818, 2016.

[20] H. Choi, H. Nam and J. Hur, "Password Typos Resilience in Honey Encryption," *IEEE Symposium.The 31st International Conference on Information Networking* (ICOIN 2017), pp. 593-597, 2017.

[21] S. Bird, E. Loper and E. Klein, "Natural Language Processing with Python." O'Reilly Media Inc. 2009.

[22] W. B. Cavnar and J. M. Trenkle, "N-gram statistics for natural language understanding and text processing," in *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 2, pp. 164–172, 1979.

[23] G. A. Miller, "WordNet: A Lexical Database for English." *Communications of the ACM*, Vol. 38, No. 11, pp. 39-41, 1995.

[24] C. Fellbaum, "WordNet: An Electronic Lexical Database.", Cambridge, MA: MIT Press,1998.

[25] Princeton University "About WordNet." WordNet. Princeton University. 2010. <http://wordnet.princeton.edu>Entropy (Information Theory). Brilliant.org. Retrieved 03.03, December 27, 2017, from https://brilliant.org/wiki/entorpy-information-theory/