International Journal of Innovative Research in Science, Engineering and Technology (IJIRSET)

| e-ISSN: 2319-8753, p-ISSN: 2320-6710| www.ijirset.com | Impact Factor: 7.512|

||Volume 9, Issue 6, June 2020||

A Deception Model for Cloud Storage to Thwart Brute-Force Attack

Lijimol James[1], Dileesh E D[2]

M Tech Student, Department of Computer Science and Engg., Government Engg. College, Idukki, Kerala, India[1]

Assistant Professor, Department of Computer Science and Engg., Government Engg. College, Idukki, Kerala, India[2]

ABSTRACT: Cloud storage is a model of computer data storage in which the digital data is stored in logical pools that can be accessed from anywhere through the internet. Cloud security is the protection of data, applications, and infrastructures involved in cloud computing. Security in cloud storage deals with attributes such as confidentiality, integrity, and availability of data. The current strategy used to achieve confidentiality is to convert the plaintext to an unreadable form known as encryption. Most encryption techniques have one essential problem, they are vulnerable to brute-force attacks. The probability of the eavesdropper acquiring the key and recovering the data is high as he/she can distinguish a correct key from incorrect keys based on the output of the decryption. This is because data has some structure like texts, pictures, audios, and videos. Thus, an attempt at decrypting with a wrong key yields random gibberish that does not comply with the expected structure. Thus, causing the eavesdropper to learn the content of the data. In the proposed system, the objective is to reinforce the current encryption measures with a decoy-based deception model where the eavesdropper is discouraged from stealing encrypted data by confounding his resources and time. The decoy data are generated by using CNN and HMM. The conventional encryption scheme uses the AES-256 CBC mode of operation. The proposed model reinforces state-of-the-art encryption schemes and will serve as an effective component for discouraging eavesdropping and curtailing brute-force attacks on encrypted documents.

KEYWORDS: Deception, Encryption, Brute force attack, Honey Encryption, Decoy messages, DTE.

I. INTRODUCTION

Computer security is a battlefield between the attackers and the defenders. In recent years, information and communication security has evolved greater privacy awareness amongst both consumers and organizations. People and organizations communicate regularly via different social media platforms like Online Social Networks and store a large amount of data in cloud which can be accessed from anywhere and at any time.

Cloud storage is a model of computer data storage in which the digital data is stored in logical pools. Cloud computing gives high security, unrestricted data storage capability, and computing power. Using cloud computing, all kinds of information are within resources, services can be used at any time and anywhere. When storing large amounts of data onto the cloud, there are different issues generated. The main issues of cloud computing are data security, integrity, authentication, and confidentiality.

The very common method used by the defenders is encryption. Encryption is used to ensure that the data is secure, even if someone else gets the data. Although encryption has one essential problem, it can be cracked with a brute-force attack. Since computational power is ever increasing, the attackers have an easier time to break encrypted data and algorithms. So far, many different techniques have been made to prevent brute-force attacks. One of the techniques to prevent brute-force is increasing the length of the encryption key or computational complexity, so it takes a longer time to break it. However, these encryption schemes remain susceptible to brute-force attacks, causing attackers to acquire secret messages.

AES256/AES128 is the industry-standard encryption scheme currently used for securing most of our infrastructures. This end-to-end encryption scheme is invented to prevent third parties/eavesdroppers from gaining plaintext access to transmitted conversations or calls. However, it fails to give foolproof security as an eavesdropper can distinguish plausible chat messages from random gibberish based on the keys he/she supplies during decryption.

A new mechanism to prevent against brute-force attack was introduced and this mechanism is known as Deception Model. Our approach produces convincing decoy messages (which are coherentcontextually correct and domain-specific) to be served to an eavesdropper attempting to decrypt transmitted messages during communication. Any key supplied by the eavesdropper during a decryption process will yield a plausible message, thus, exhausting his time and resources, unlike conventional encryption scheme which yields random gibberish upon decryption with an incorrect key. The proposed deception model does not eliminate encryption but reinforces it with a degree of deception to exhaust the attacker's time and resources. The decoy data are generated by using CNN and HMM. The CNN model is

| e-ISSN: 2319-8753, p-ISSN: 2320-6710| www.ijirset.com | Impact Factor: 7.512|

||Volume 9, Issue 6, June 2020||

used to classify the domain (intent) to which the plaintext falls into, identify and extract important/special keywords from the plaintext. Based on the identified domain HMM generates a decoy file sharing similar domain as the plaintext but without any of the special keywords. The conventional encryption scheme uses the AES-256 CBC mode of operation.

The rest of this paper is organized as follows: the related works are presented in Section II. In section III, the proposed solution is presented. In Section III (B), the implementation of the proposed model within a cloud storage application is developed. Experimental results are presented in Section IV. Section V concludes the research.

II. RELATED WORKS

This section presents the related works regarding the encryption-based approach and deception-based approach.

A. ENCRYPTION BASED APPROACHES

A considerable amount of literature has been published on using encryption-based approaches for securing current communication network systems. In cryptography, encryption is the process of changing a message or information (plaintext) to a meaningless and unreadable form (ciphertext) so that only authorized parties can access it and those who are not authorized cannot. In an encryption scheme, the intended information or message, referred to as plaintext, is encrypted using an encryption algorithm to generate ciphertext that can be read-only if decrypted. The encryption and decryption keys are the same for symmetric encryption and the encryption and decryption keys are different for asymmetric encryption.

Yegireddi et al. [1] proposed a survey on a cryptographic algorithm, to make a comparative study for most important algorithms in terms of data security effectiveness, key size, complexity and time, etc. Table 1 gives the comparison between all algorithms with respect to the message he intends encoding and then uses any conventional encryption scheme as the second layer of encryption. From this analysis, it is understood that AES and ECC are best in security and encryption performance.

B. *DECEPTION BASED APPROACHES*

Deception-based strategies have been applied as countermeasures to delay, detect and confuse an adversary attempting to steal data on a network.

Bercovitch et al. [2] proposed Honey tokens that are artificial digital data items planted deliberately into a genuine system resource to detect unauthorized attempts to use information. "HoneyGen" is a novel method for generating honey tokens automatically by using data mining techniques. In this system, the process consists of three main phases that are rule mining, honey token generation, and likelihood rating. In rule mining, the various types of rules that characterize the real data are extracted from the production database; honey token generation in which based on the extracted rules an artificial relational database is generated; and the likelihood rating in which based on its similarity to the real data a score is calculated for each honeytoken. Based on the score honey tokens are generated. The limitation of this work is, it cannot support for long text documents.

Juels et al. [3] proposed honeywords (decoy passwords) to detect attacks against hashed password databases. The paper proposes an alternative approach that selects the honeywords from existing user passwords in the system to provide realistic honeywords. For each user account, the legitimate password is stored with several honeywords to sense impersonation. If honeywords are selected properly, a cyber-attacker who steals a file of hashed passwords cannot be sure if it is the real password or a honeyword for an account. Moreover, log in with a honeyword will trigger an alarm notifying the administrator about a password file breach. The generating methods are Chaffing-by tweaking, Chaffing-with-a-password-model, Chaffing with ”Tough Nuts” and Hybrid Method. The limitation of this work is that this method can only be used in passwords.

Hyun-Ju Jo et al. [4] proposed two countermeasures. The two techniques are honey encryption and the structural code scheme. Both methods look different; however, they have similar security goals and they both use feature distribution transforming encoders based on statistical schemes. The purpose of this scheme is to create false plaintext to protect the original plaintext. First, a system user trains text data to obtain the word frequency, given an assumption that plain text follows the Markov process. The next step is to generate a corpus, which is a database for an encoding and decoding process. The final step is to encode or decode user data with the corpus. The main limitation of this work is, repeated trials were required to build the syntactically meaningful plain text.

| e-ISSN: 2319-8753, p-ISSN: 2320-6710| www.ijirset.com | Impact Factor: 7.512|

||Volume 9, Issue 6, June 2020||

Algorithms	Key Size	Block Size	Round	Structure	Flexible	Features
DES	64 bits	64 bits	16	Fiestel	No	Not Strong Enough
3DES	112 or 168 bits	64 bits	48	Fiestel	Yes	Adequate Security and fast
RSA	1024 to 4096 bits	128 bits	1	Public Key Algorithm	No	Excellent Security and Low Speed
ECC	variable	variable	1	Public Key Algorithm	Yes	Excellent Security and fast Speed
RC6	128 to 256 bits	128 bits	20	Fiestel	Yes	Good Security
AES	124, 192, 256 bits	128 bits	10, 12,14	Substitution and Permutation	Yes	Security is excellent. It is best in security and Encryption performance

Table 1: Comparison of Cryptographic Algorithm

Beunardeau et al. [5] proposed a new technique known as Probabilistic context-free grammars (PCFG). In this technique, the sentence is converted to the syntax tree or parse tree to generate a Skelton. The rules are rewritten and applied to the Skelton to generate decoy data. In this work semantics of the decoy file are not achieved.

Joo-Im Kim et al. [6] proposed a technique that efforts to strengthen the security of the Instant Messaging (IM) system. Conventional message encryption uses a secret or private key. However, the key is vulnerable to a brute force attack, causing to acquire the original message. A countermeasure was introduced that is plausible looking but fake plaintexts, other than the conventional message encryption which is vulnerable to brute force attack. The technique is called HoneyEncryption (HE). The plaintext is encoded using distribution-transforming encoders (DTE) and cumulative massive function (CMF) served as a code table, converting hexadecimal numbers to characters and vice versa to generate decoy data. In this work, the length of the message must be greater than 33 and it cannot generate decoy messages from the diverse domain.

Whitham et al. [7] proposed four designs for automating the construction of honey file content. Each of the designs employed the same three stages: template selection, content extraction, and document population. The new designs select a document from the target directory as a template and employ word transposition and substitution based on parts of speech tagging and n-grams collected from both the target directory and the surrounding file system. The new designs were able to successfully mimic the content from the target directory. The limitation is, it does not yield an accurate domain-specific document.

Edwin Mok et al. [8] proposed a technique for cloud data protection known as extended honey encryption (XHE). An eXtended Honey Encryption (XHE) scheme is done by adding an additional protection mechanism on the encrypted data. When the attacker attempts to access these encrypted data by entering the incorrect password, instead of rejecting the access, the HE algorithm generates an indistinguishable bogus data, in which the attacker could not determine whether the guessed password is working correctly or not. XHE algorithm is divided into 2-sub algorithms in order to protect the file's name and file's extension. The file name is given as the input to the DTE encode1 algorithm to obtain a set of prefix seed values from seed space. Then the file extension is given as the input to the DTE encode 2 algorithms to obtain a set of suffix seed values from seed space. IS_encode1 and IS_encode2 algorithm are used to generate a series of fake file names and file extensions. In this work, only the file name and extension are generated/replaced, content is not changed.

Prakruthi et al. [9] proposed a novel comprehensibility manipulation framework (CMF) to generate comprehend fake documents, which can be used for fooling attackers and increasing the cost of data ex-filtration by wasting their time and resources. CMF requires an original document as input and generates fake documents that are both believable and readable for the attacker, possess no important information, and are hard to comprehend. CMF for

| e-ISSN: 2319-8753, p-ISSN: 2320-6710| www.ijirset.com | Impact Factor: 7.512|

||Volume 9, Issue 6, June 2020||

generating fake documents consists of four modules: input preparation, pre-processing, modification, and post-processing. In this work, it does not yield an accurate domain-specific document, and semantic is not included.

Omolara et al. [10] proposed HE 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 a fake message from the original message and this makes it difficult for him to determine if he has recovered the original message or not. Decoy messages are generated by using Stanford Dependency Parser and Wordnet from Princeton. This technique cannot support for large documents.

Omolara et al. [11] proposed a decoy-based deception model for preventing eavesdroppers from stealing encrypted messages and applied it for the security of instant messaging on real-world deployment. Any key supplied by the eavesdropper during a decryption process will yield a plausible message, thus, exhausting his time and resources, unlike conventional encryption scheme which yields random gibberish upon decryption with an incorrect key. The proposed deception model uses BLSTM-RNN and HMM to generate decoy messages. The BLSTM-RNN model will be used to classify the domain (intent) to which a plaintext falls into, identify and extract important/special keywords from the plaintext. The HMM scans through the identified domain and then generate a decoy message sharing similar domain as the plaintext but without any of the special/keywords. The limitation of this work is that RNN is only suitable for temporal data and cannot support long messages.

III. PROPOSED DECEPTION MODEL (DM)

This section presents our proposed deception model (DM) designed to reinforce the current encryption scheme. We define the overall architecture as a DM-then Encrypt construction. The definition of DM-then-Encrypt is given as:
Definition 1: Let DM =(encode, decode) be a deception scheme whose outputs are in the space S = {0,1}.
Let SE = (encryption, decryption) be asymmetric encryption scheme with message space S and someciphertext space C.

Therefore, DM-then-Encrypt = [DM, SE]. Thus, DM is applied for the first layer of encoding, followed by the second layer of encryption using the symmetric encryption scheme, SE. The objective of the DM-then-Encrypt model is to ensure that a plausible-looking but fake plaintext is generated during decryption of the ciphertext under any given key. This implies the generation of decoy messages sharing similar distribution and indistinguishable characteristics as the plaintext.

Fig 2 shows an overall architecture of the system. First, the user can upload an encrypted file to the cloud. For the conventional encryption layer, we employed state of the art AES encryption scheme as it is the industry standard used for encryption at present. If a user downloads the file with the correct key, then the original file is been downloaded. If an attacker downloads with an incorrect key, then a plausible looking but fake decoy file is been generated and send it to the attacker.

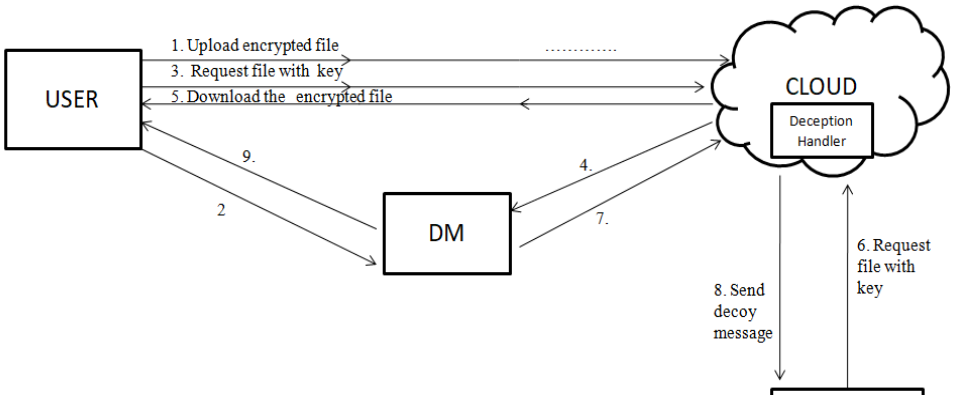

Fig 2: An architecture of the overall System

| e-ISSN: 2319-8753, p-ISSN: 2320-6710| www.ijirset.com | Impact Factor: 7.512|

||Volume 9, Issue 6, June 2020||

A. PRELIMINARY DESCRIPTION OF THE BUILDING BLOCKS OF THE PROPOSED DM

This research incorporates Convolutional Neural Network (CNN) and a Hidden Markov Model (HMM). The Convolutional Neural Network is a class of deep learning and Natural Language Processing. CNN are multilayer perceptrons. Multilayer perceptrons usually mean fully connected networks, that is, each neuron in one layer is connected to all neurons in the next layer. This makes CNN prone to over-fitting data. CNN takes advantage of the hierarchical pattern in data and assembles more complex patterns using smaller and simpler patterns. CNN uses little pre-processing compared to other text classification algorithm.

Hidden Markov Model is a statistical Markov model in which the system being modeled is assumed to be a Markov process with unobservable states. The HMM can be represented as the simplest dynamic Bayesian Network.
In this work, the CNN model is used to classify the domain (intent) to which the plaintext falls into, identify and extract important/special keywords from the plaintext. Based on the identified domain HMM generates a decoy file sharing similar domain as the plaintext but without any of the special keywords.

B. IMPLEMENTATION

The basic DM system was developed using Python 3.0. The cloud used is Dropbox. The application uses a central server known as Deception Model (DM). The user first encrypts the file and uploads it to the cloud. The encryption algorithm used is AES-256 in CBC mode of operation and HMAC-SHA256 for authenticating the users. At the same time, the user encrypts the key, file name, and domain using DM's public key and sent it to the DM. So that the DM can decrypt it with their private key. If the user downloads the file with the same key, then the deception handler sends the key to the DM and DM verifies it whether it is the correct key or not. If the key is correct, then the user can download the file. If the key is incorrect, then the DM generates a plausible looking but fake decoy file and send it to the attacker. So, the attacker thinks that this was the original key and he/she does not try with any other key. By this brute-force attack can be prevented. DM also sends an alert to the user, if an attacker tries to access his/her file.

In designing CNN, first, the dataset is pre-processed and trained to build a model. Based on the model intent/domain classification is done. The datasets are generated from diverse domains based on emotions. In the CNN model, the first convolution is performed on the sentence matrix to generate a variable length feature map. An activation function ReLu is applied to each feature map. Then max pooling is done that is, the maximum value from each feature map is taken and concatenate together to form a single feature map. Then softmax function is used for multiclass classification. For example, if the plaintext is:

" I like this movie very much"

The domain/intent identified will be movie liking. So, the decoy message will be generated based on the same domain. But the special words from the plaintext are not revealed during decoy generation. HMM is used to generate decoy messages. The training dataset (classified by labels) is pre-processed to build the HMM to generate a meaningful output sentence related to the same domain. To generate a decoy file, first, the state space is defined and a transition matrix is created for the hidden states. Then the observation probability matrix is created. Then find the maximum probability of each path that gets to state. At the end of the sequence, create the most likely path by selecting the state that won each time.

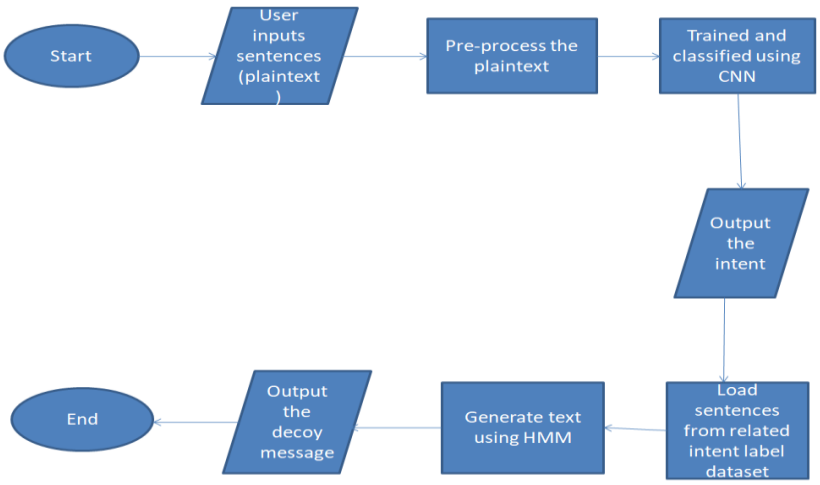

Fig 3: Flowchart showing how files are processed and how they generate decoy files

| e-ISSN: 2319-8753, p-ISSN: 2320-6710| www.ijirset.com | Impact Factor: 7.512|

||Volume 9, Issue 6, June 2020||

Fig 3 shows the flowchart showing how files are processed and how they generate decoy files using Hidden Markov Model (HMM). Fig 4(A) shows a text document that is uploaded to the cloud. Fig 4(B) shows a decoy file that is generated when an attacker tries to access the file with an incorrect key.

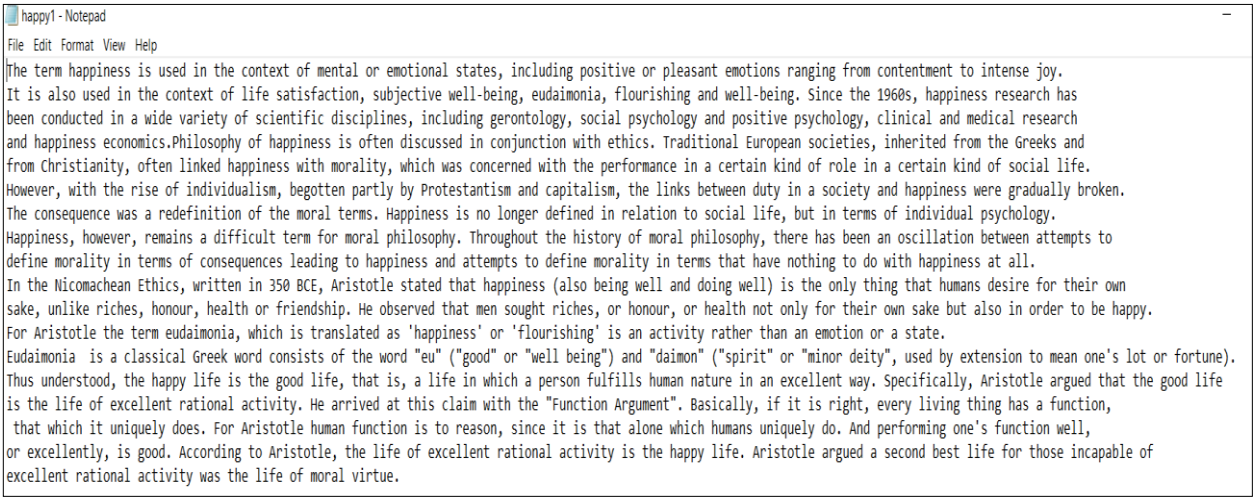

Fig 4(A): Simulation of original text document that is uploaded to the cloud

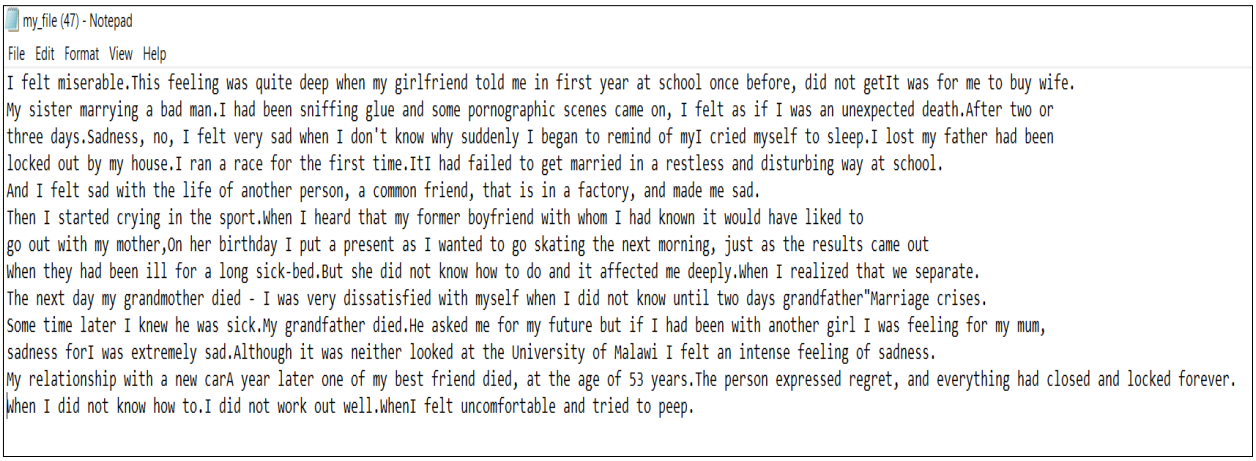

Fig 4(B): Simulation of decoy file that is generated

IV. RESULT ANALYSIS

We evaluate the performance of the proposed DM to determine its effectiveness. The evaluation is divided into two parts. The first part is to evaluate the proposed DM based on the accuracy using CNN and RNN. The second part is to determine the performance of the DM. The evaluations are based on accuracy and Hamming distance.

A. COMPARISON OF ACCURACY

Accuracy is the degree to which the result of a measurement, calculation, or specification conforms to the correct value or a standard. That is the quality or state of being correct or precise. The accuracy of the proposed DM using CNN is 89 percentage and the accuracy of the proposed DM using RNN is 85 percentage. CNN has higher accuracy compared to RNN. Therefore, CNN is better for classifying large text documents.

B. HAMMING DISTANCE

The Hamming distance is a metric used for estimating the edit distance between two strings of characters to determine the number of positions where the strings have different characters. It quantifies the minimum number of substitutions/errors required to transform the plaintext into the ciphertext. The Hamming distance between the number

of bit changes in the plaintext data and the decoy message of the proposed DM is compared with the N-gram model [6]. The average Hamming distance of [6] is 26.8 while the proposed DM is 32. The higher value implies that there is a large difference between the plaintext message and the decoy message. For example, if the number of words is ten (10) and the Hamming distance from the proposed DM is 9, this means out of 10 words in the plaintext data, 9 words are different. The Hamming distance is suitable for comparing strings of equal length and we had to pad and truncate the length of some of the generated decoys to get the Hamming distance. Table 2 represents the comparison of the proposed DM with N-gram and RNN.

		N- gram model	Proposed DM using RNN and HMM	Proposed DM using CNN and HMM
Convincing decoy message in the context of attacker's believability	Domain based	X	√	√
	Semantic based	X	√	√
	Syntax based	√	√	√
Length of text file		L>32	Short messages	Large text documents
Accuracy		-	83%	89%

Table 2: Comparison of the proposed DM with [6] and [11]

V. Conclusion And Future Work

The paper presents a deception model to reinforce the conventional encryption scheme for secure file storage on the cloud. Conventional encryption schemes have a vulnerability where an eavesdropper can determine if his candidate key is correct or not based on the structure or distribution of the output message. The proposed model addresses this limitation of a conventional encryption scheme by yielding a domain-specific, coherent but fake message to an attacker who tries to brute-force/decrypt a ciphertext using incorrect keys. The proposed model does not eliminate encryption but complements the current encryption scheme to confound the time and resources of the attacker. In the future, it can be extended to other media such as images, audios, and videos.

References

[1] Ramesh Yegireddi, R Kiran Kumar "A survey on Conventional Encryption Algorithms of Cryptography" IEEE International Conference, 2016.

[2] Bercovitch M, Renford M, Hasson L, Shabtai A, Rokach L, and Elovici Y (2011). "HoneyGen: An automated honeytokens generator", in Intelligence and Security Informatics (ISI), 2011 IEEE International Conference on, IEEE, pp. 131–136.

[3] Imran Erguler "Achieving Flatness: Selecting the Honeywords from Existing User Passwords" IEEE Transactions on Dependable and Secure Computing, 2015

[4] H. J. Jo and J. W. Yoon, ''A new countermeasure against brute-force attacks that use high-performance computers for big data analysis," Int. J. Distrib. Sensor Netw., vol. 11, no. 6, 2015, Art. no. 406915. doi: 10.1155/2015/406915.

[5] M. Beunardeau, H.Ferradi, R.Géraud, and D.Naccache, ''Honeyencryption for language-robbing Shannon to pay turing?'' in Proc. Int. Conf. Cryptol. Malaysia, Springer, 2016, pp. 127–144.

[6] K.Joo-Imand J.Yoon, ''Honeychatting: "A novel instant messaging system robust to eavesdropping over communication,'' in Proc. IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP), Mar. 2016, pp. 2184–2188. Accessed: Jan. 12, 2019.

[7] BenWhitham.2017. "Automating the Generation of Enticing Text Content for High-Interaction Honeyfiles". In Proceedings of the 50th Hawaii International Conference on System Sciences.

[8] P. Karuna, H. Purohit, R. Ganesan, and S. Jajodia, ''Generating hard to comprehend fake documents for defensive cyber deception,'' IEEE Intell. Syst.,vol.33,no.5,pp.16–25,Oct.2018 doi:10.1109/mis.2018.2877277.

[9] Mok, E., Samsudin, A., & Tan, S.-F. (2017). "Implementing the honey encryption for securing public cloud data storage". In In Proceedings of First International Conference on Computer Science and Engineering.

[10] Omolara,A.E.,Jantan,A.,Abiodun,O.I.,&Poston,H.E.(2018) "A novel approach for the adaptation of honey encryption to support natural language message" In Proceedings of the International MultiConference of Engineers and Computer Scientists (Vol. 1).

[11] Abiodun Esther Omolara, Aman Jantani, Oludare Isaac Abiodun, Kemi Victoria Dada, Humaira Arshad "A Deception Model Robust to Eavesdropping Over Communication for Social Network Systems" IEEE Access 2019, Digital Object Identifier 10.1109/ACCESS.2019.2928359.