IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received June 17, 2019, accepted June 27, 2019, date of publication July 11, 2019, date of current version August 9, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2928359

A Deception Model Robust to Eavesdropping Over Communication for Social Network Systems

ABIODUN ESTHER OMOLARA[1], AMAN JANTAN[1], OLUDARE ISAAC ABIODUN[1,2], KEMI VICTORIA DADA[3], HUMAIRA ARSHAD[1,4], AND ETUH EMMANUEL[5]

[1]Security and Forensic Research Group (SFRG) Laboratory, School of Computer Sciences, Universiti Sains Malaysia, Penang 11800, Malaysia
[2]Department of Computer Science, Bingham University, Karu, Nigeria
[3]Department of Statistics, Ahmadu Bello University, Zaria, Nigeria
[4]Department of Computer Science, The Islamia University of Bahawalpur, Bahawalpur, Pakistan
[5]Department of Mathematics, Arthur Jarvis University, Calabar, Nigeria

Corresponding authors: Abiodun Esther Omolara (styleest2011@gmail.com) and Aman Jantan (aman@usm.my)

This work was supported in part by the USM RUI, Penang, Malaysia, under Grant 1001/PKCOMP/8014017, and in part by the Center for Cyber Safety and Education, United States Internal Revenue Segregated Fund of (ISC)^2, Inc. Code. EIN: 45-2405127 through the (ISC)^2 Women Cybersecurity Scholarship Award, Clearwater, FL, USA.

ABSTRACT Communication security deals with attributes such as confidentiality, integrity, and availability. The current strategies used to achieve covertness of communication employs encryption. Encryption techniques minimize eavesdropping on the conversation between the conversing parties by transforming the message into an unreadable form. However, it does not prevent or discourage eavesdroppers from stealing and attempting to decrypt the encrypted messages using a brute-force attack or by randomly guessing the key. The probability of the eavesdropper acquiring the key and recovering the message is high as he/she can distinguish a correct key from incorrect keys based on the output of the decryption. This is because a message has some structure-texts, pictures, and videos. Thus, an attempt at decrypting with a wrong key yields random gibberish that does not comply with the expected structure. Furthermore, the consistent increase in computational power implies that stolen encrypted data may gradually debilitate to a brute-force attack. Thus, causing the eavesdropper to learn the content of the message. To this end, the objective of this research is to reinforce the current encryption measures with a decoy-based deception model where the eavesdropper is discouraged from stealing encrypted message by confounding his resources and time. Our proposed model leverages its foundation from decoys, deception, and artificial intelligence. An instant messaging application was developed and integrated with the proposed model as a proof of concept. Further details regarding the design, analysis, and implementation of the proposed model are substantiated. The result shows that the proposed model reinforces state-of-the-art encryption schemes and will serve as an effective component for discouraging eavesdropping and curtailing brute-force attack on encrypted messages.

INDEX TERMS Attacker, brute-force, chat, decoy, deception, eavesdropper, encryption and instant messaging.

I. INTRODUCTION

In recent years, information and communication security has evolved into a red-hot issue with notable data breach encompassing giant establishments, such as Facebook, Yahoo, Equifax and others, resulting in greater privacy awareness amongst both consumers and organizations [1]–[3]. The espousal of the internet has impacted every sphere of human lives particularly in the area of communication [4], [5]. People and organizations communicate regularly via different social media platforms and Online Social Networks (OSNs) such as Facebook, WhatsApp, Instagram, Snapchat, WeChat and others. The OSN service is mostly integrated with Instant Messaging (IM) applications connected to these services.

Communication via IM applications has become popular as both the young and old appreciate the convenience, speed, immediate receipt, acknowledgment and reply to messages transmitted. On the other hand, organizations enjoy the easy liaison, marketing and advertising opportunities these platforms offer. Ongoing analysis by the eMarketer predicted that

The associate editor coordinating the review of this manuscript and approving it for publication was Jafar A. Alzubi.

before the end of 2019, 65% of the world populace would utilize IM applications for their day-to-day transaction and correspondence [6].

Regardless of the increase in the use of IM applications, alongside its productivity and viability for easy interaction with others, the IM system suffers from a major drawback, intrusion of privacy by eavesdropping during communication [7], [8]. Consequently, IM systems providers have bolstered their IM applications with conventional encryption schemes [9]. Encryption techniques transform the transmitted messages by rendering the conversation to be unintelligible except to the conversing parties. However, conventional encryption schemes are susceptible to brute-force attacks. Messages are transmitted as bytes which are encoded in American Standard Code for Information Interchange (ASCII) form. Consequently, an adversary that intercepts an enciphered text and tries to decrypt the message using an incorrect key can filter out a part of the decrypted message by observing that some of the sequences are a combination of random symbols/characters which are non-uniformly distributed affirming the invalidity of the key.

Additionally, the security of current cryptosystem depends on the keyspace. Current practice involves designing the keyspace to be large such that it may take several billions of operations that exceeds the age of the universe for a classical computer to exhaust the keys. Thus, embarking on a brute-force attack in such circumstance is uneconomical and computationally infeasible. However, it is different in some cases. For instance, in the password-based encryption (PBE), the key space is intentionally designed to be small; thus, the necessary number of computations required for breaking the key and recovering the message is considerably less, coupled with users choosing simple, weak and poor passwords. If a plaintext M is a 15-digit American Express card with its number encoded via ASCII and a conventional encryption scheme is used for encrypting the card number, the likelihood that any Mi $\neq$ M is a valid ASCII encoding of a 15-digit string which is negligible, at t$(10/256)^{15} < 2^{-74}$. Thus, the adversary will discard incorrect messages and recover the plaintext, M with a high probability.

State-of-the-art encryption schemes, such as the Advanced Encryption Standard (AES-256) and its variants, AES-128 and AES-192 has inspired and continues to inspire the design of modern attacks, for instance, the bicliques attack, key-recovery attack, meet-in-the-middle (MitM), boomerang attacks and others. A MitM attack was proposed by [10], [11]. This kind of attack is a time trade-off attack where the attacker supposedly has access to the ciphertexts $\boldsymbol{C}$ and some side information (such as a set of plaintext $\boldsymbol{M}$) with the condition $E_{k1}(P) = D_{k2}(C) = r$, where the two keys are *k1* and *k2* and the internal variable is r. Thus, the attacker can easily compute $E_{k1}(P)$ for all key *k1* and $D_{k2}(C)$ for all key *k2*.The attacker is able to determine if the pair of keys are the correct candidate keys once matches are found. The MitM attack was improved at a low cost using bicliques, where bipartite graphs are exploited to carry out the cryptanalysis.

Some of these modern attacks are even faster than the brute-force attacks but are currently computationally infeasible. While the conventional encryption schemes are leveraged to secure our infrastructure today, they provide only temporary security. The prospective attacks on AES cipher using modern cryptanalytic techniques implies that the AES (and other conventional) ciphers may fail to provide the essential long-term threshold needed for foolproof security when computational power advances. The introduction of high-processing tools, high-performance parallel and distributed systems (such as GPU, FPGA-ASIC) allows the brute-force attack to be easier as malicious individuals become equipped with high computational power [12]–[16].

Popular IM system, such as WhatsApp, Treema enforces end-to-end encryption using conventional encryption schemes. For instance, WhatsApp has emerged as the most popular IM service on internet-enabled devices with over 1.5 billion monthly active users [17]. It allows users to exchange chat message (such as text, videos, voice and files) using AES256 in CBC mode for encryption and HMAC-SHA256 for authenticating the users [18]. 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 conversation or calls. However, it fails to give foolproof security as an eavesdropper can distinguish plausible chat message from random gibberish based on the keys he/she supplies during decryption as depicted in Fig. 1.

Recent reports have revealed a security flaw in IM applications such as WhatsApp, Threema, signal IM apps. This loopholes potentially allows third parties to eavesdrop on encrypted Group chats, thereby learning the message being communicated [19]–[21]. Researchers discovered a loophole that allows hackers to access group chats and share spurious news [22], implant bugs when users answer video call [23], impersonate users and send spoof messages [24]. These issues and more demonstrate the need for further research into the security of IM systems and communication in its entirety.

Current research effort continues to focus on increasing the keyspace or the computational complexity. However, these encryption schemes remain susceptible to brute-force attacks, causing attackers to acquire secret messages. To the best of our knowledge, the prior work that attempted to tackle the eavesdropping problem using a completely different method by Joo-Im and Yoon [7], proposed a deception-based approach using the statistical coding scheme. Their approach is built from the intuition of honey encryption, where the adversary is supplied with plausible-looking but fake message upon eavesdropping [7], [15], [16]. While their approach reinforced conventional encryption scheme and added more security to the current IM system, it is based on heuristics and cannot be implemented on real-world systems. The limitations of their approach are presented in related works in section II as one of the contributions of this study.

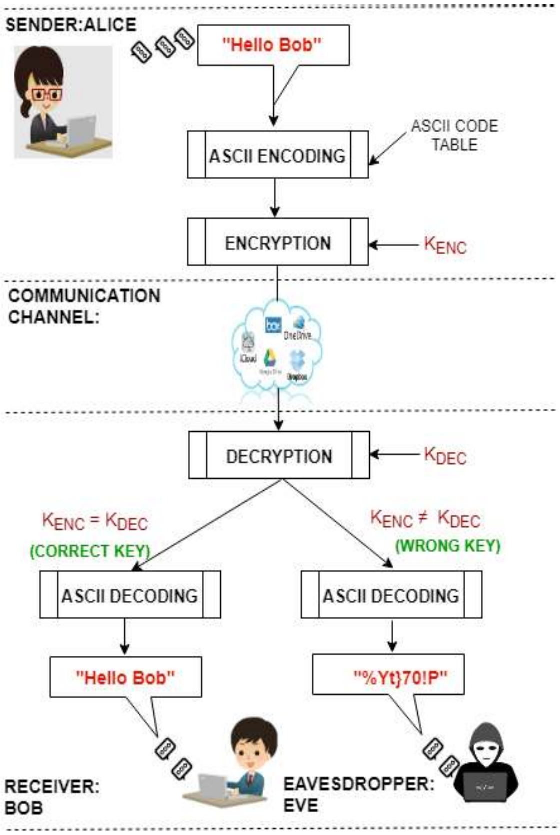


FIGURE 1. A conventional encryption scheme showing Alice sending a message to her friend, Bob. An eavesdropper, Eve steals the encrypted message and tries random keys to decrypt the message. Eve can determine if the key she tried is correct based on the output of her decryption. An incorrect/wrong key yields random gibberish while a correct key yields plausible message. On the other hand, Bob, the intended receiver decrypts the message with the correct key pre-shared with his friend, Alice to recover the message.

Following the approach proposed by [7], we developed a decoy-based deception model for preventing eavesdroppers from stealing encrypted messages and we apply it for the security of instant messaging on real-world deployment. Our approach produces convincing decoy messages (which are coherent, contextually correct and domain-specific) to be served to an eavesdropper attempting to decrypt transmitted message 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 attackers time and resources. Several security tests using automated tools and human subjects are carried out to ensure the proposed system is effective. The source code will be deposited in the Github repository for reproduction and testing purposes.

More also, this research will be the first work on the realization of natural language decoy within a real-world IM system. It will be beneficial to researchers, security experts, industry practitioners, academia and developers in the field of communication security.

The rest of this paper is organized as follows: the related works on current measures of securing IM system are presented in Section II. In section III, we present the adversarial model to describe further how the adversary can breach the current encryption scheme. In section IV, the proposed solution is presented. In Section V, the implementation of the proposed model within an instant messaging application is developed. Experimental results are presented in Section VI. Section VII concludes the research.

II. RELATED WORKS

In this era of advanced cybercrime, secure communication is increasingly sought for in infrastructures and services. Recent trends have shown that regardless of how much effort used in securing our digital technology, there is no assurance that some form of compromise may not occur either now or in future. A case that hit home is the PGP and S/MIME encryption, which are widely used for encryption and authentication in the e-mail system. The users, in this case, have a pair of keys used for encryption of signed messages and signatures which they continue to use for years. An attacker that steals encrypted data, perhaps, via device compromise and gains access to the decryption keys years after, may use the key pair to recover all past and even future emails sent with the key pair. Similar holds for instant messaging systems, for instance, Threema [25], some previous version of Viper [26] and others. Thus, the constant changes in security requirements demonstrate the need for innovative solutions that can withstand emerging problems.

A. ENCRYPTION-BASED APPROACHES

A considerable amount of literature has been published on using encryption-based approaches for securing current communication network systems and instant messaging systems. Luo *et al.* [27] proposed a certificateless signcryption scheme based on bilinear pairings for ensuring the confidentiality, integrity and authentication for IM systems. Wang *et al.* [28] developed an identity-based cryptosystem for IM applications which employs a group of private key generators. This approach curtails the key escrow problem by producing a master key using a secure distributed key generation protocol, however, at the expense of a high computational cost. A HyperElliptic Curve Cryptosystem (HECC) was designed by Wanda and Hantono as a secure and effective authentication approach for IM systems [29]. Their approach cannot be actualized in real life deployment due to the difficulty of transforming the point of the HECC into plaintext. Loukas *et al.* developed a public key infrastructure and AES for enhancing the security of social network systems. This approach prompts the user to set a pseudonym each time he/she logs into the server to start a session [30]. An authorization delegation scheme which depends on a proxy re-encryption and bilinear maps was presented by Feng *et al.* [31]. This scheme achieves fine-grained authorization delegation for sharing copyrighted or confidential multimedia resources. Chen *et al.* designed an encryption scheme based on a Pseudo One Time Pad and the Diffie-Hellman key exchange protocol for the secure transmission of messages for the IM system [32]. A PKI and AES based encryption scheme was proposed by Karabey and Akman for the secure sharing of messages for IM systems [33]. A comprehensive encryption scheme that caters for attacks such as replay attack, requiring low processor and ensures secure signature and message authentication is presented by [8]. The aforementioned schemes improve the privacy of IM and OSNs. They provide a form of security one way or the other; however, they cannot withstand a brute-force attack in the face of a computationally unbounded attack. This and other more reasons are explained further.

The client-server architecture of IM systems allows chat messages to pass through central servers. Consequently, encrypted messages may be exposed and stolen by attackers, creating an opportunity for a brute-force attack. The brute-force attack searches for one key, which returns plausible plaintext by computing all the keys in the key space. While state-of-the-art encryption schemes are designed to have a large keyspace, the continual increase in computational power shows that such an advantage may soon fail to protect current encryption schemes. Moore's law expresses that computing power doubles every 18 months while the costs remain constant. In cryptography, this implies that, if today, a computer worth $1,000 can break a cipher Z in a month, the cost for breaking Z in the next 18 months is $500 and in 3 years it becomes $250 [34], [35]. This exponential increase in computing power implies that less time and resources are needed to decrypt the encrypted message as time passes.

The high increase in data breach over time implies that the current measures for securing data may be insufficient and requires in-depth research [36]. Additionally, the advent of quantum computers will render current state-of-the-art encryption schemes to be insecure [37], [38]. This is because quantum computers can be used to solve the underlying mathematics that forms the foundational security of conventional ciphers. Complex problems and computations, such as the integer factorization and discrete logarithm problems, which are the core of asymmetric encryption schemes (such as Rivest Shamir Adleman (RSA), Elliptic-curve Cryptography (ECC), Diffie–Hellman (D-H), and Digital Signature Algorithm (DSA)) will be broken in few seconds using a quantum computer. Consequently, information protection products in enterprise security frameworks will immediately become susceptible to attacks with a quantum computer. Whilst the time in which large quantum computers will be built remains an open problem, the general conviction of the impossibility of constructing an industry-standard quantum computer has been crushed as the first quantum logic gate has already been built on silicon [39]. Hence, the need for reinforcing current encryption schemes.

Although, several quantum-based encryption schemes have been earmarked as potential encryption schemes for the quantum era. For instance, just as the security of RSA encryption is contingent on the difficulty of factoring large composite integers, quantum-based encryption schemes hinge on problems that are anticipated to remain hard to solve with quantum computers, such as, solving the shortest vector problem in a lattice, solving systems of multivariate quadratic equations over finite fields, solving problems in an error-correcting code and others [40]–[42]. However, they remain work in progress, as their security is yet to be established, proven and standardized [43]–[45].

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 [46]. In 2015, Kaghazgaran and Takabi [47] proposed honey permission, a deception model integrated into a role-based access control model to address insider threat using access control. Mor *et al.* [48] proposed HoneyFaces which allows the generation of realistic-looking but fake synthetic faces as a countermeasure for eavesdropping attacks on the biometric system. In [49], [50], HoneyFiles are proposed as countermeasures for protecting theft of confidential documents by serving the attacker with plausible-looking but fake files during their attack. The deception-based method has been applied in different fields and has been established as effective methods for curtailing attacks on several systems. However, it has not been applied for the security of the IM system except a preliminary work presented by [7].

The basis behind the use of fake/decoy messages is quite established as potential measures for identifying, detecting and preventing malicious threats. However, there is minimal evidence to show their application in curtailing data theft in real-world deployment [51]. This challenge is due to the current approach used in generating deceptive content. Current decoy generation methods employ random unrealistic bogus data, words that are frequently searched on the internet, random word extracted from web data, corpus-based generation, manually created templates for database fields and file attributes [51]–[54]. These approaches fail to convince malicious persons as they fail to generate words and sentences expected to reside within a document, they reveal the underlying file names and metadata, they leak information about the underlying message, thus leading the attacker to reconstruct and extract the underlying message from the decoy document.

Several studies have indicated that creating realistic, enticing, adaptive, with no distinguishable features (from the plaintext contents) is the essential pre-requisite for convincing an adversary to accept a decoy message as the underlying plaintext message during an attack. These features are possible with natural language and artificial intelligence as an adversary can only be convinced when the decoy message reflects the way human uses natural language [51]–[54].

The first and only proposal of a deception-based countermeasure on eavesdropping attack on instant messaging systems was carried out by Joo-Im and Yoon [7]. In their approach, they applied the concept of honey encryption using a statistical coding scheme and conventional AES. Their approach handles the brute-force attack of an unbounded adversary by serving up valid-looking, plausible plaintext in the event that an attacker tries to decrypt the transmitted messages using an incorrect key. They build a code table using the statistical coding scheme to generate fake chat messages. Chat messages are represented as n-gram models. The n-gram models were used to predict the probability of the next character in the sequence of the conversation been transmitted. A movie subtitle text-corpus was used for generating the decoy/fake conversations that would be decrypted and served to an attacker who tries to brute-force the conversation with an incorrect password. However, their approach is based on heuristics and the reasons are highlighted as follows:

- In the n-gram model employed by the researchers, the probabilities of each word are learned from a movie subtitle text-corpus. Each sentence allows several probable derivations. Different derivations in each instance will act as the decoy message. A chain of words is identified as the result of a production process that outputs one word at a time depending solely on the near past. However, the n-gram models alone are insufficient as they lack the long chain correlation feature of natural language. Presenting a sentence probability as a result of forward conditional probabilities does not fit effectively the long-range forward and backward interactions as used in natural language [55], [56]. Therefore, a problem of dimensionality arises when there is an increase in the number of parameters to better fit the data. Furthermore, the models are unable to grasp the concept of recursivity and syntactic properties of natural language and as such may fail grammatically except with some human intervention during the production process. Consequently, the approach fails to generate convincing correct decoy/fake chat messages in some instances.
- An attacker with the knowledge of the distribution of the corpus used can recover the message in a jiffy.
- The approach lacks a mechanism that checks and ensures that keywords/special words from the plaintext do not appear during a brute-force attack. The exclusion of this checking mechanism implies that a chosen ciphertext attack (CCA) may become possible, as an attacker may use the output results from prior decryptions to inform their choices of which ciphertexts have decrypted. Thus, their approach fails to prevent the eavesdropper from learning partial information of the chat message or from usefully mauling encrypted messages.
- The approach is restricted to a domain. In their research, the corpus used was for movie subtitles. The algorithm is restricted to the corpus used and cannot be used to generate decoy messages from diverse domain. This is one of the essential keys to successfully fooling an eavesdropper. For instance, an eavesdropper with side information expecting messages related to travel agency will not be fooled if the chat messages his incorrect key generated are from a flower-based domain. Thus, he will discard the message.
- The approach has a specific length of messages that must be greater than or equal to thirty-three (message length >=33) otherwise it fails grammatically, failing to convince the attacker.
- The approach fails to provide forward secrecy which is needed in the security of the IM system.

Other related studies by [57] and [58] also fail to convince an unbounded adversary as they share the same flaw as [7], thus failing to protect the underlying plaintext message. This research builds on the deception-based approach presented by Joo-Im and Yoon [7] with the additional component of addressing the limitation of their approach by employing artificial intelligence and testing it on live data.

III. ADVERSARIAL MODEL

IM systems broadcast all communication wirelessly over a large range, allowing cybercriminals with an antenna to sniff, eavesdrop or intercept messages. Thus, a vulnerability occurs where an adversary intercepts an encrypted message on the communication channel of an IM system and tries to decrypt the message by randomly trying all possible keys (brute-force). The adversary is able to acquire the original message (plaintext) by observing the difference of entropy between decrypted messages with wrong keys and the correct key. This process of recovering the message is easier as an incorrect key will yield non-uniform characters or an error symbol that forms no meaning. However, only one key which is the correct key will yield the original message. It is assumed that the adversary is computationally unbounded and can run through the Oracle in polynomial time. Furthermore, the brute-force process is executed faster as the attacker can access advanced hacking tools. Therefore, the goal of this study is to supply the adversary with plausible messages/conversations during his decryption using any of his candidate keys. For every random key he/she tries, the adversary will get a plausible message that shares the same distribution and domain as the plaintext.

IV. PROPOSED DECEPTION MODEL (DM)

This section presents our proposed deception model (DM) designed to reinforce current encryption scheme. The overall architecture is designed based on the intuition of honey encryption which applies a two-layer of security to the message [7], [15], [16] as shown in Fig.2. Following the [15], [16] approach, 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 $\mathrm{S} = \{0, 1\}^{\mathrm{s}}$. Let SE = (encryption, decryption) be a symmetric encryption scheme with message space S and some ciphertext space C.

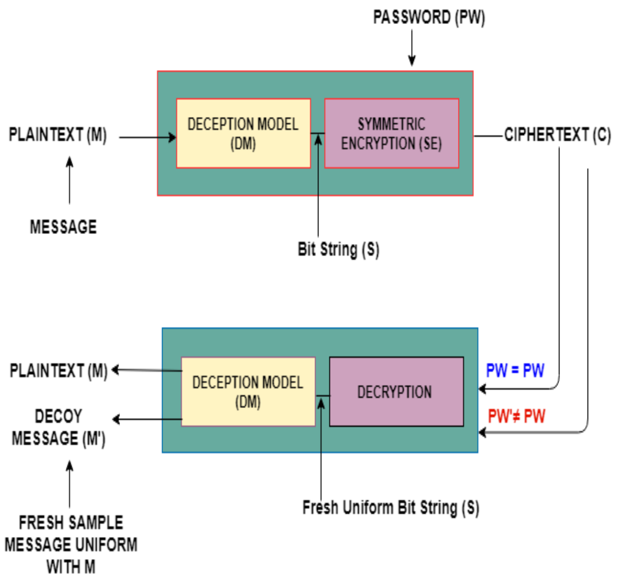


FIGURE 2. An architecture of the overall System. Messages are processed first with the deception model followed by a conventional encryption scheme. Decryption using correct keys yields the plaintext while decryption using incorrect keys yields plausible but fake message.

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.

In the first layer, the deception model will be developed, and the second layer offers a conventional encryption scheme. 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. The novelty of our research is in the deception model and as such, we concentrate on describing the process in the succeeding sub-section.

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

This research incorporates a bidirectional Long Short-Term Memory-Recurrent Neural Network (BLSTM-RNN) and a Hidden Markov Model (HMM) to address the limitations of [7]. The models and the justification for their application are substantiated further.

The Bi-directional Long Short-Term Memory (BLSTM) is a variant of the recurrent neural network (RNN) architecture employed in deep learning and Natural Language Processing (NLP). The key advantage of using the RNN over N-gram models is their greater representational intensity and their capability to perform intelligent smoothing by considering the semantic and syntactic features of natural language [59]–[61]. In an RNN, long-term dependencies are duly considered when modeling natural language. Besides, the transfer learning feature of RNN provides a better advantage over n-gram models. The bidirectional RNN is able to access the past and future context data, however, the range of the context may be limited due to the vanishing gradient problem consistent with the RNN model. Thus, this limitation is addressed by combining the bidirectional RNN with the LSTM, thus forming the BLSTM-RNN.

Messages transmitted during communication especially in IM system are processed in real time and such there is a need for accessing the future and past context in order to provide forward secrecy during transfer. The BLSTM-RNN model supplies the feature that explores the transmitted information from the past as well as the future. In this work, 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 essence of the aforementioned process is to minimize a CCA attack where an adversary with side information may be able to form the message with fragments recovered from the decoy message while decrypting. The LSTM stores the previous state in memory and memorizes the extracted keywords from the plaintext input. 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.

B. EXPERIMENTS FOR THE DEVELOPMENT OF THE DM MODEL

In the first phase of the development, an Intent Classification Model (ICM) is built. The intent represents the domain of the plaintext message. For instance, if the plaintext to be transmitted is;

“book a hotel for four people”

The domain/intent will be identified as hotel booking. Thus, the decoy message that will be generated will be from the same domain as the plaintext, in this case, it will be hotel booking. Also, this phase will identify special words/keywords from the plaintext and extract them, in this case, ‘book’, ’four’, ’people’ and the identified domain ‘hotel’. The special words are identified and extracted from the underlying plaintext so that they are not revealed during the generation of the decoy messages. This process is managed by the LSTM which acts as a form of storage in the cell memory. The bi-directional RNN accesses the information from before and after context, allowing the possibility of feedback for alternating layers. This feature protects the plaintext from an attacker with some side information while restricting the decoys that will be generated to the same domain as the plaintext to deceive him.

In designing the BLSTM-RNN, training datasets which are sentences with labels are preprocessed using the model to build an ‘intent classification model (ICM)’ as depicted in Fig.3. The datasets are extracted from diverse domains in public texts, dictionary, Wikipedia and others.

The steps to preprocess the data in Fig. 3 is shown (in Algorithm 1) as follows:

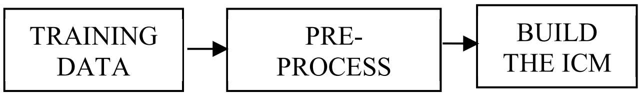


FIGURE 3. **The Bi-directional long short-term memory - recurrent neural network (BLSTM -RNN) model.**

Algorithm 1 Steps to Pre-Process the Data

```
1. Separate the sentence and the intent(label)
2. Remove the stop words from the sentences.
3. Get the maximum length of the sentence.
4. Tokenize the sentence to words.
5. Encode the words with numbers.
6. Get the vocabulary size from the tokenized words.
7. Pad the sequence to the maximum length.
8. Make the labels(intent) one hot encoding.
9. Split the data into train and test.
10. Pass the data to the HMM.
```

The input to the model should be of the same size. That is the reason we are padding the sentences in preprocessing step 7. This is especially important as the seed space of the encoder needs to be equal to the block size of the AES cipher. Furthermore, the RNN model needs vocabulary size and the maximum length of the sentence. The sentences are then passed line-by-line with the label (Intent). After which the data model then generates word embedding and then passes it to the bidirectional LSTM. The word embedding is depicted in Fig. 4.

Where LSTM (a,b,c…,m) are multiple layers of LSTM which outputs multiple intents from 1 to m. W_1, $W_2 \ldots$, W_n represents the word embedding. Intent_1, Intent_2,…, Intent_n represents the domain.

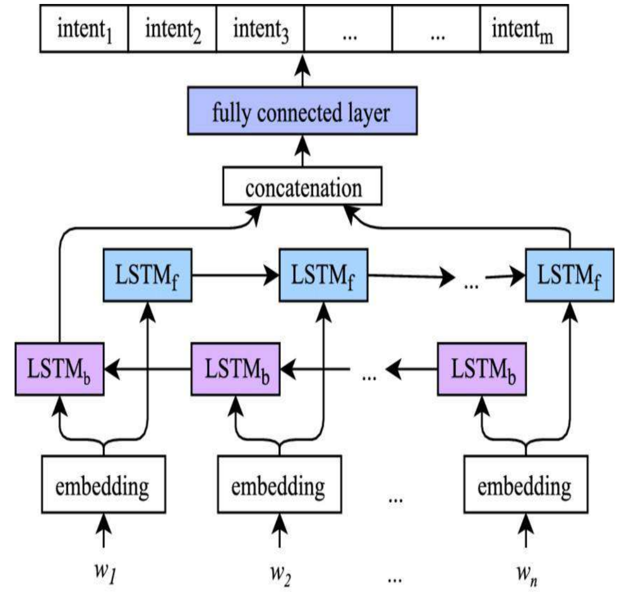


FIGURE 4. **The intent classification model (ICM). LSTM (a, b,…, m) are the multiple layers of LSTM, which outputs multiple intents from 1 to m.**

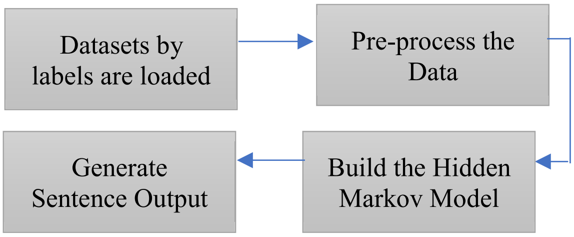


FIGURE 5. **(A) flowchart showing how the hidden markov model (HMM) processes the message.**

The Hidden Markov Model (HMM) was used as an enhancement to the N-gram model in this proposal. One of the problems of the N-gram model is that it has a number of parameters exploding with the size of the vocabulary, thus, the HMM is used to reduce this complexity by exploiting the Markov property. The training dataset (classified by labels) is pre-processed to build the Hidden Markov model to generate a meaningful output sentence related to the same labels. As shown in Fig. 5.

Thus, the flowchart of the proposed DM process showing how messages are incorporated and processed using the Intent Classification Model (ICM) and Hidden Markov Model (HMM) is depicted in Fig.6.

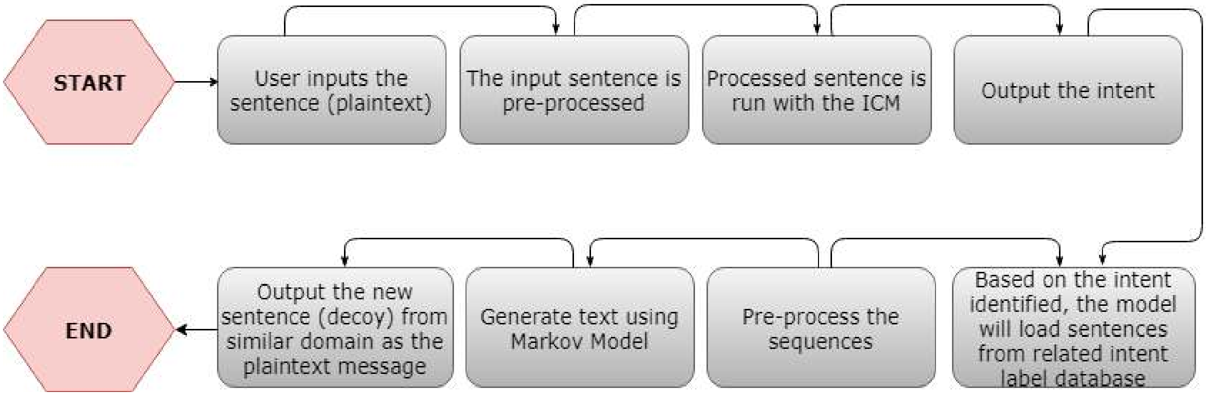

START
User inputs the sentence (plaintext)
The input sentence is pre-processed
Processed sentence is run with the ICM
Output the intent
Based on the intent identified, the model will load sentences from related intent label database
Pre-process the sequences
Generate text using Markov Model
Output the new sentence (decoy) from similar domain as the plaintext message
END

FIGURE 6. **Flowchart showing how messages are processed once they are transmitted and how they generate decoy messages.**

The pseudocode to show the generation of the decoy model is shown in Algorithm 2 as follows:

Algorithm 2 Function to generate BLSTM-RNN Model

```
def generate ():
        for i in range(number_of_sentences):
            sentence = []

            word0 = sample_word(initial_word)
            sentence.append(word0)

            word1 = sample_word(second_word[word0])
            sentence.append(word1)

            while True:
                word2 = sample_word(transitions[(word0, word1)])
                if word2 == 'END':
                    break
                sentence.append(word2)
                word0 = word1
                word1 = word2
            return ' '.join(sentence)
number_of_sentences = 1

        train_markov_model(cl)
        orginal=”
        count=0
        status=True
        while status == True:
            count=count+1

            eg=gots.lower()
            example_sent=eg.translate(str.maketrans(”,”, string.punctuation))

            stop_words = set(stopwords.words('english'))
            word_tokens = word_tokenize(example_sent)
            filtered_sentence = [w for w in word_tokens if not w in stop_words]
            if len(filtered_sentence)==0:
                original = ”
                status = False
                break

            eg1=generate().lower()
example_sent1=eg1.translate(str.maketrans(”,”, string.punctuation))

            stop_words1 = set(stopwords.words('english'))
            word_tokens1 = word_tokenize(example_sent1)
            filtered_sentence1 = [w for w in word_tokens1 if not w in stop_words]
```

V. IMPLEMENTATION OF THE PROPOSED MODEL INTO A BASIC INSTANT MESSAGING APPLICATION

A basic IM system was developed using Python 3.0 as a proof of concept (POC) for the proposed model. The application was installed on two phones. The application uses a central server for delivering messages between two conversing parties, Alice and Bob. An encrypted session is established after Alice and Bob enter their shared password. Their chat messages are processed using the proposed deception algorithm. The eavesdropper, Eve intercepts the encrypted conversation and tries random keys with the aim of learning the conversation shared between Alice and Bob as shown in Fig. 7. For this POC, we used labeled datasets from different domains.

In Fig. 7(A), Alice was chatting with her friend Bob and he replied to her conversations. In the first and second line of chat, Alice sent ‘Hello’ and Bob replied with a, ‘Hi’. Eve tries to decode the messages using a random key and she gets ‘Good morning’ and ‘Good day’ as the conversation between Alice and Bob respectively. In the third and fourth line of chat, Alice said, ‘the weather is cold today’ and Bob replied with, ‘seriously, the weather is cold today?’. Eve’s incorrect key decoded the conversations to be ‘will there be snowfall this month’ and ‘what’s the forecast starting on July twenty’.

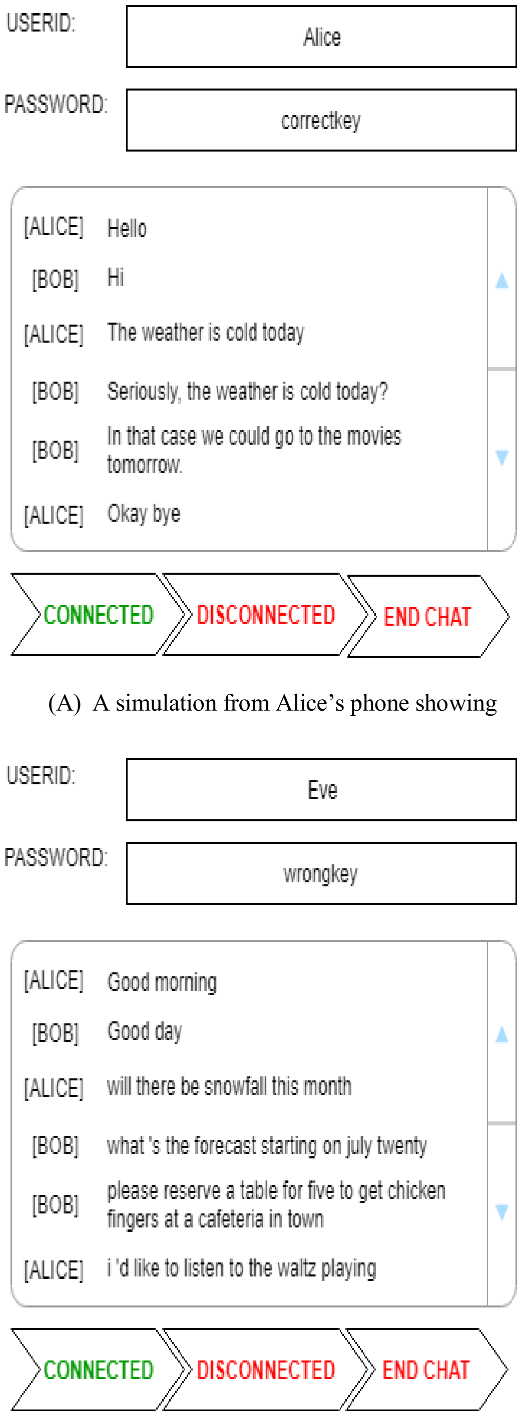
The novelty we will like to highlight here is that the proposed algorithm does not just generate random words. It checks the domain of the underlying plaintext (chat message) and generates fake/decoy chat message under similar domain without revealing any keywords/special word from the underlying chat message. This novelty is also observed in the fifth line of the chat when Bob suddenly said, ‘in that case, we could go to the movies tomorrow’ and Eve’s wrong key showed, ‘please reserve a table for five to get chicken fingers at a cafeteria in town’. The domain suddenly changed from a discussion of ***‘weather’*** to information that has to do with the ***‘restaurant’***.

Collectively, for every wrong password supplied by Eve, the algorithm searches for fake chat message that shares similar domain as the transmitted conversation and generates plausible-looking, semantically correct messages as decoy chat-message to confuse the adversary.

The proposed deception model is not only suitable for securing IM systems and OSNs, but it can also be used for securing human-generated messages and documents, as well as email messages.

VI. ANALYSIS AND RESULTS

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 for security and the second part was to determine the performance. The evaluations are based on the entropic property of the model, Hamming distance, Levenshtein distance, semantic coherence analysis, decoy turing test, comparison with state-of-the-art encryption schemes, functional comparison with the current deception-based model for IM system, generic evaluation of the model for an attacker with side information.



USERID: Alice

PASSWORD: correctkey

[ALICE] Hello

[BOB] Hi

[ALICE] The weather is cold today

[BOB] Seriously, the weather is cold today?

[BOB] In that case we could go to the movies tomorrow.

[ALICE] Okay bye

(A) A simulation from Alice's phone showing

USERID: Eve

PASSWORD: wrongkey

[ALICE] Good morning

[BOB] Good day

[ALICE] will there be snowfall this month

[BOB] what 's the forecast starting on july twenty

[BOB] please reserve a table for five to get chicken fingers at a cafeteria in town

[ALICE] i 'd like to listen to the waltz playing

(B) A simulation from the eavesdropper's (Eve) phone showing the conversation shared between Alice and Bob

FIGURE 7. (A) A simulation of conversations between Alice and her friend, Bob (B) The eavesdropper, Eve trying random keys to learn the conversation between Alice and Bob.

These metrics have been widely utilized to validate the effectiveness of the model in diverse branches of computer security. The experiments are further described in the following subsections.

A. COMPARISON OF ENTROPY

Entropy is used to measure the degree of unpredictability of a given system. In data security, this unpredictability must be provided in the decoy message to conceal the distribution/structure of the underlying plaintext. In information and communication security, a dataset composed randomly have a high entropic property. Introducing high entropy in the secret keys makes guessing/predicting the key to be difficult for the eavesdropper. Furthermore, it becomes difficult for the adversary to determine when his/her guess is correct. The entropy H(Q) of a discrete random variable Q is

$$Ha\,(Q) = -\sum q \in QP(q)log_{a}p(q) \qquad (1)$$

where, p(q) is the probability mass function which denotes $\Pr\{Q = q\}$, $q \in Q$ and a is the base of the logarithm, thus a = 2 in bits and a = 26 in lower-case letters [7], [63].

A significance test was carried out to evaluate the proposed model in terms of its entropic property. The test was carried out to check the degree of randomness and unpredictability of the plaintext message when a correct key and several incorrect keys are used to decrypt the plaintext. The test was carried out using MATLAB 2013(a). A plaintext was encoded using the proposed model. In the first instance, decryption was done using the correct key K, and it yields the plaintext message P. In the other part, decryption was done repeatedly using random keys, K* for 5, 000 times which yields P* for each decryption.

The first and second hypotheses are as follows,

H0 = There is no difference in the entropy between P and P*
H1 = There is a difference in the entropy between P and P*

To put it in a simpler context, H0 implies that the decrypted text P and P* cannot be distinguished while H1 implies that they can be distinguished. The entropic distribution of the wrong plaintext P* represents the test statistics while the entropy of the plaintext P represents the observed value. The significance level is at 0.05 with a confidence range of 95%. The result is depicted in Fig.8.

A small P-value compared to the 0.05 significance level indicates that the observed data P is not included in the scope of P* as there is a difference between P and P*. Thus, H0 is rejected. This implies that the adversary cannot predict or acquire the plaintext from the distribution of the ciphertext during the attack.

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 [7] and depicted in Fig. 9.

The average Hamming distance of [7] is 26.8 while the proposed DM is 29.1. The higher value implies that there

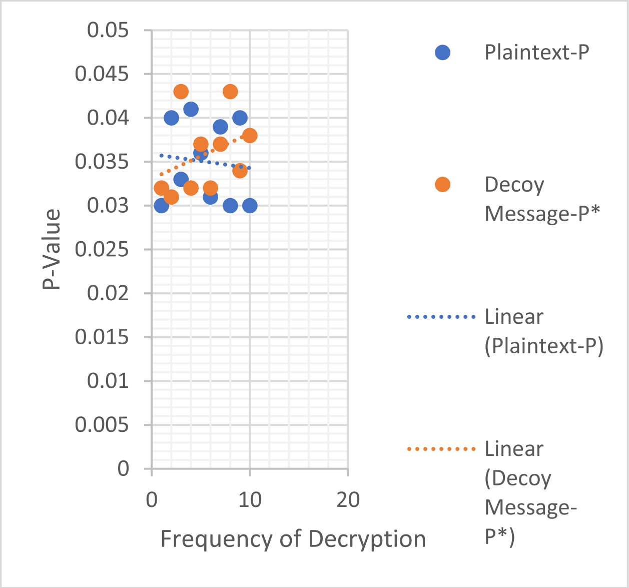


FIGURE 8. Evaluation for the entropic property of the plaintext, P and decoy message, P*.

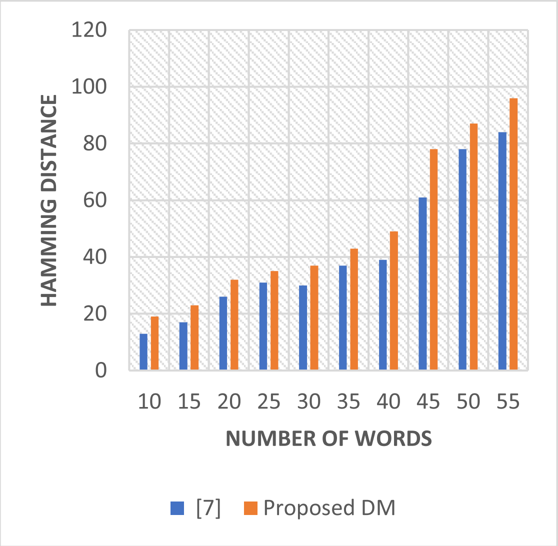


FIGURE 9. Hamming distance.

is a large difference between the plaintext message and the decoy message. This variation can also be observed from the graph when the number of words is compared to the Hamming distance, for instance, in the first instance, 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 in Fig 9. Thus, we conjecture that this may make our result to be inconclusive. We decided to carry out a Levenshtein distance analysis which is more appropriate for messages of different length as the plaintext message and decoy message generated in our proposed DM and also other related studies [7], [57] and [58] does not generate equal length during the brute-force decryption.

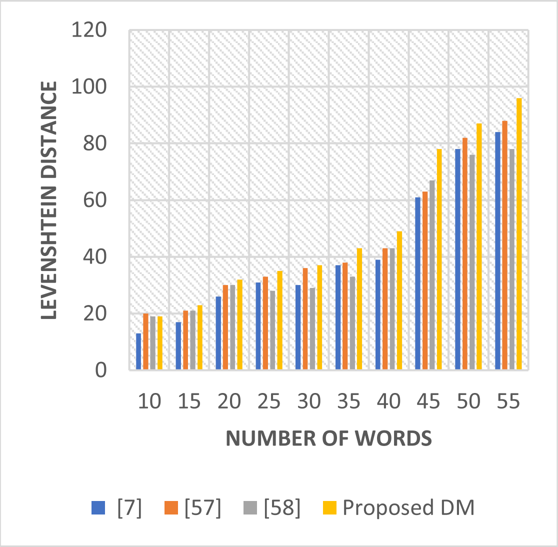


FIGURE 10. Levenshtein distance.

C. LEVENSHTEIN DISTANCE

It quantifies the minimum number of insertions, substitutions or deletions required to transform the plaintext into the ciphertext.

The Levenshtein distance between two strings p, q of length |p| and |q| is given by $\text{lev}_{\text{p,q}}$ (|p|, |q|) where

$$lev_{p,\,q}\,(a,\,b) = \begin{cases} min \begin{cases} \mathbf{max}(a,\,b) \\ lev_{p,\,q}\,(a-1,\,b)+1 \\ lev_{p,\,q}\,(a,\,b-1)+1 \\ lev_{p,\,q}\,(a-1,\,b-1)+1_{(p_q \neq Q_b)} \end{cases} \end{cases}$$
$$\mathbf{if\ min(a,b)=0} \qquad (2)$$

where $1(p_{q \neq Q_b})$ is the indication function equal to zero when $p_{q=Q_b}$ and equal to one, else, $lev_{p,\,q}\,(a,b)$ is the distance between the initial a character of p and the initial b characters of q.

The Levenshtein distance between the number of bit changes in the plaintext data and the decoy message of the proposed DM is compared with [7], [57] and [58] and depicted in Fig. 10.

The average Levenshtein distance of [7] is 41.6, [57] is 45.4, [58] is 42.4 while the proposed DM is 49.9. The higher value implies that there is a large difference between the plaintext message and the decoy message. A higher Levenstein value shows the large gap between the plaintext and the decoy message. While the Hamming distance and Levenshtein distance cater for the security at the word level, further analysis is required to check/cater for the meaning of the words and if they form context that can deceive the attacker while protecting the underlying plaintext message. Further analysis is carried out in the succeeding subsections.

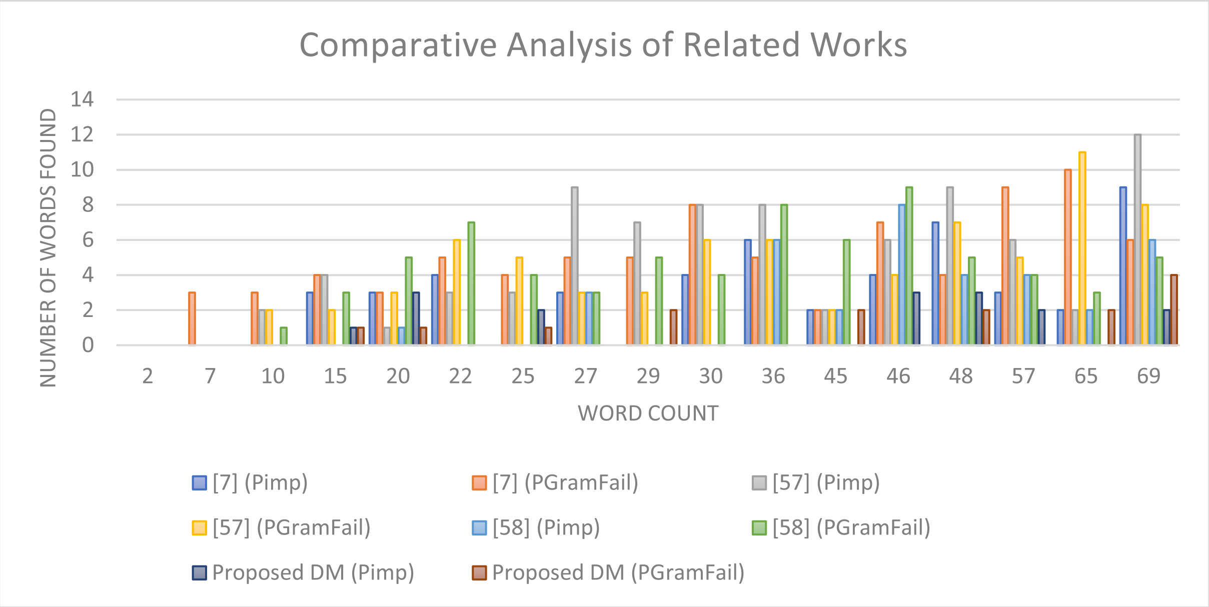


FIGURE 11. Semantic coherence analysis to check if special/keywords are protected during an attack.

D. SEMANTIC COHERENCE ANALYSIS

It was observed from the literature that some decoy-based production system fails to deceive the adversary into accepting the decoy message as the plaintext because the attacker can observe the decoy-message form no contextual meaning during decryption, such as in the work by [55]. Thus, the attacker discards such message knowing they are decoys. Semantic coherence analysis in this context was used to check if the message that would be generated during an attack has meaning and does not reveal the keywords/special words in the plaintext while successfully fooling the attacker.

This stage is necessary to verify if special keywords from the plaintext are hidden and not revealed in the decoy chat message during a brute-force attack. It is worth stating that some part of speech that connects other words such as prepositions, conjunctions may appear in both the plaintext (chat message) and ciphertext. This is because they help form contextual meaning to the message. Random keys were submitted to decrypt the Oracle, after which a comparison of the sanity of resulting messages was carried out. We define the terms used as follows:

Word Count: The number of words encoded

Pimp: This represents important keywords from the plaintext. These are words which must not appear in the decoy chat message during a brute-force attack otherwise an attacker with side information may exploit it to form the plaintext.

$P_{GramFail}$: This represents the number of words that fail grammatically in the context used when generating the decoys. Fig. 11 depicts the results from the evaluation of the semantic analysis.

The proposed DM shows a minimal word frequency compared to current studies, implying the difficulty of an adversary distinguishing the decoy message from the underlying message during an attack.

E. GENERIC EVALUATION OF THE MODEL FOR AN ATTACKER WITH SIDE INFORMATION

Aiming to evaluate the model in terms of generality such as its application to another message format (e-mails, human-generated messages), we performed a Decoy Turing Test (DTT). The DTT test is an imitation game presented by famous Cryptographer and Mathematician, Alan Turing [49], [64], [65]. It is used to show artificial intelligence to determine if human judges are able to distinguish between human conversational simulators and computers [66], [67]. The game is performed on a text-only communication route where the human judge engages in a conversation with the computer and human. The computer is said to have successfully passed the test in the event that the human judge is unable to distinguish between it and the human. Following the DTT approach, we enlist 50 participants who are students from the school of Computer Science, Universiti Sains Malaysia. The 50 students will act as the human judges and the justification of using this approach as the gold standard is based on the fact that attackers are humans and will judge the plausibility of the decoy message based on human perception of language. Therefore, the essence of this test is to check if the attacker can differentiate decoy messages from the real message (plaintext) based on their belief of the message been convincing.

The participants (attacker in this game) were each given a link to a brute-force interface where they were allowed to try several keys to decrypt an Oracle containing an encrypted message. They were allowed to try as many keys from the pool of keys and also as many times as possible after which

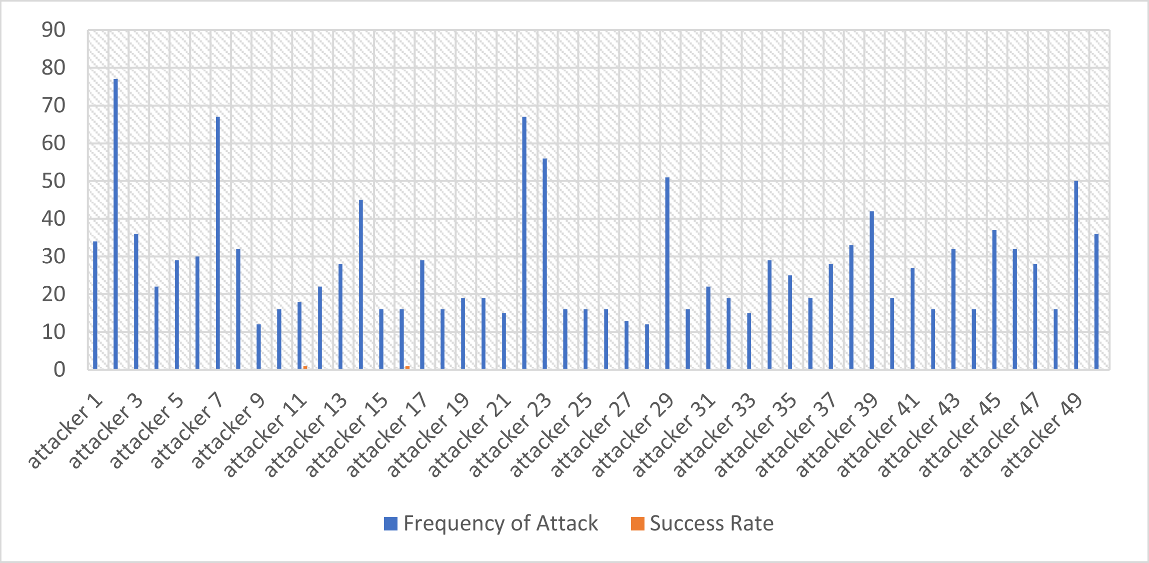


FIGURE 12. Evaluation on the believability of the decoy message based on a decoy turing test (DTT).

they are asked to submit a binary YES or NO answer for each decryption tried. A 'YES' means they think it is the plaintext and a 'NO' means it is not the plaintext but a decoy message. It is worth mentioning that the correct key was also supplied to the attackers without their knowledge in the pool of keys allowed. Fig. 12 depicts the result of the percentage of the believability of the decoy message.

The DTT test performed was to find out the percentage of the believability of the decoys. A 4% success rate in detecting the decoys from the plaintext implies that the model has a high probability of deceiving the attacker. We observed that the 4% was from attacker 11 (after 18 computations) and attacker 16 (after 16 computations). During our interaction with the two attackers, it was discovered that both attackers were able to distinguish the decoy message based on guesses as they both said they were unsure and just selected a 'YES' answer.

Finally, a test to check the domain security against an attacker with side information. Recall our discussion in Section II where we highlighted some of the flaws of current deception-based systems and the limitation of the work of [7]. In this case, we randomly selected some of the participants. We encrypted a message and informed the participant of the specific domain. Attackers are allowed to try several numbers of computations using the worst scenario of a small 4-bit key that can be exhausted in 2^4 operations until a large 256-bit key space. Fig. 13 shows the result of the attack.

Given that the attackers are aware of the domain from which the decoy message is expected, they were still unable to distinguish the decoy message from the underlying plaintext. Collectively, the low percentage of success rate (at distinguishing decoy message from the real message) using several combinations of keys shows the effectiveness and plausibility of the proposed system.

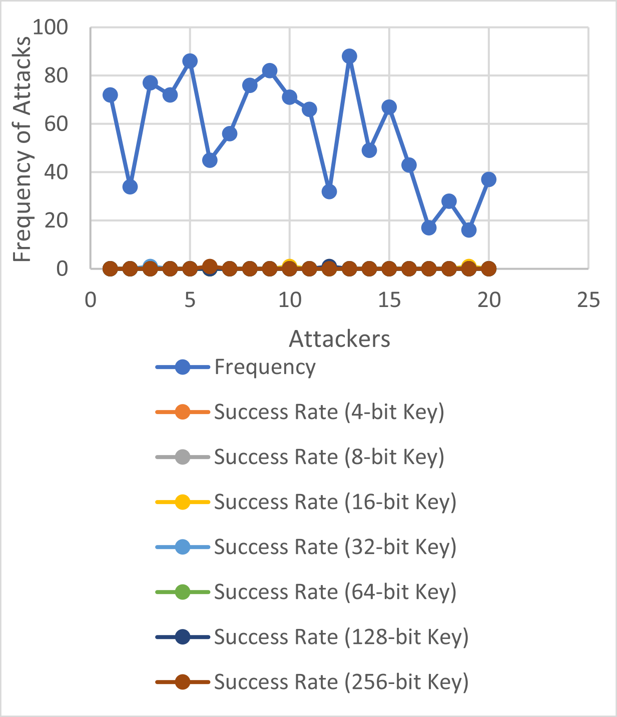


FIGURE 13. Security evaluation of the proposed model against an attacker with side information.

F. FUNCTIONAL COMPARISON WITH CURRENT DECEPTION-BASED MODEL FOR IM SYSTEM

A functional comparison is carried out between the proposed deception model and the work proposed by [7], [57], [58] and the proposed DM. The data presented in Table 1 shows the additional features of the proposed model which makes its deployment to real-world to be possible.

TABLE 1. Functional comparison of the proposed DM with current work by [7] and other related studies by [57] and [58].

	[7]	[57]	[58]	Proposed DM
Convincing decoy message in the context of attacker's believability.	**Domain-based Decoys**			
	X	X	X	√
	Semantics Coherence			
	X	X	X	√
	Syntactic Cohesion			
	√	√	√	√
Message Recovery (MR) Security	**MR for an attacker with side information**			
	X	X	X	√
	MR for an attacker with side no information			
	X	X	√	√
Decoys restricted to Corpus	√	√	√	X
Length (L) of message	L ≥ 33	N/A	N/A	L=∞

G. PERFORMANCE COMPARISON WITH CURRENT DECEPTION-BASED MODEL

The performance of the proposed model was carried out and compared with related studies in the literature. The time taken to encrypt and decrypt the same length of message using the proposed algorithm by [7], [57], [58] and the proposed DM is shown in Fig 14 and Fig. 15 respectively.

The average encryption time for [7] is 24.4, [57] is 24, [58] is 23.8 and the proposed DM is 23.9, implying there is not much difference in the time of encryption.

The average decryption time for [7] is 26.4, [57] is 24.8, [58] is 25.5 and the proposed DM is 27.2, implying our approach may require more time during the decryption process of generating the decoys. While this overhead occurs as a result of fetching the data from multiple sources, several approaches of optimizing the speed of decryption exist and this may be part of the future work.

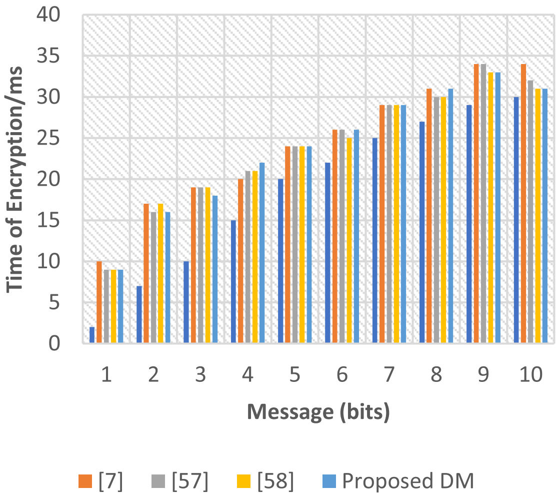


FIGURE 14. Encryption time.

H. DISCUSSION ON COMPARISON WITH STATE-OF-THE-ART ENCRYPTION SCHEMES

Conventional cryptographic schemes are designed using asymmetric or symmetric encryption scheme. While these measures may provide security for our infrastructures now, they cannot stand the test of time as they are susceptible to brute-force attack and may yield in the hands of a persistent adversary with high computational power.

The proposed deception model reinforces current encryption scheme and cannot be directly compared with current

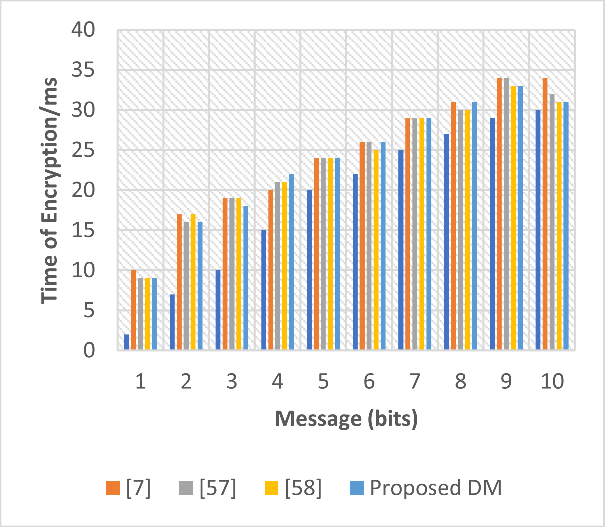


FIGURE 15. **Decryption time.**

cryptographic measures as their functional components are different. In current encryption schemes, more attention of researchers has focused on strengthening the key by increasing the length and complexity. However, as discussed in Section II, an adversary can distinguish the key based on the distribution/structure of the expected output.

A simple description is presented to further demonstrate how a conventional encryption scheme fails to withstand a brute-force attack:

If an adversary intercepts a ciphertext C = 50 45 54 45 52 20 50 41 55 4c Assuming he is aware of the encryption scheme used, for example, if the AES cipher is used. His next action will be to find out the language the message was encrypted in. If he finds out that it is an English word, then, he will try to brute-force the key using AES decryption to find out if the letters belong to the English alphabets.

$$\boldsymbol{C \leftarrow enc(P, K)}$$

Let's assume he has three 7-digits keys to pick from:

$$\begin{aligned} \mathrm{M}_{2789613} &= \text{ 41 2B 51 54 5B 4F 42} \\ \mathrm{M}_{1822761} &= \text{77 65 61 74 68 65 72} \\ \mathrm{M}_{9136434} &= \text{ 26 45 4C 2F 46 51 21} \end{aligned}$$

The effective way to distinguish between the keys is to check what each key corresponds to in the American Standard Code for Information Interchange (ASCII) character dictionary. By the time he checks, he will find out that:

$$\begin{aligned} \mathrm{M}_{2789613} &= \text{ 41 2B 51 54 5B 4F 42 = A+ QT[OB} \\ \mathrm{M}_{1822761} &= \text{77 65 61 74 68 65 72=WEATHER} \\ \mathrm{M}_{9136434} &= \text{ 26 45 4C 2F 46 51 21=\&EL/FQ!} \end{aligned}$$

He will discard $\mathrm{M}_{2789613}$ and $\mathrm{M}_{9136434}$, as they are not uniformly distributed, they do not spell out any meaningful word, and he can easily pick out the correct key as $\mathrm{M}_{1822761}$ which spells out 'WEATHER' and recover the message by applying the decryption function, $P \leftarrow dec(C, K)$ on the key and ciphertext

$$\begin{aligned} \text{C} &= \text{43 4F 4D 45 20 48 4F 4D 45 20 4E 4F 57} \\ &= \text{COME HOME NOW} \end{aligned}$$

In summary, after exhausting all 7-digit keys. The adversary finds only one message completely made of letters and which forms a plausible meaning. Hence k = 1822761 and the plaintext message decrypts to '**COME HOME NOW**'. The adversary may use these tactics to recover all the encrypted messages. However, in our deception approach, for every key the adversary tries, he has plausible messages. For example, trying any incorrect key yields plausible messages such as '**THEY ARE HERE', 'MEET HER LATER,' 'CHECK THE HALL'**. In essence, an attacker trying to steal the encrypted data by trying random keys (password guessing or brute forcing) in current cryptographic approach gets gibberish which is an indicator that he has not yet gotten the data. But in the proposed approach, trying random keys yields meaningful and plausible data to fool the adversary into thinking he has the data.

The intuition behind the proposed DM is to reinforce the current encryption scheme, it does not discard the current encryption schemes. Comparing state-of-the-art encryption schemes to the proposed DM may be synonymous with comparing apples and orange. This is because the proposed DM has some functional components which state-of-the-art encryption lacks. For instance, comparing the standard AES encryption scheme to the proposed DM produces random gibberish during a brute-force attack on the AES. However, integrating the proposed DM with AES as applied in the research produces plausible, contextually correct but fake message. In this way, the attacker will discard the random gibberish when he is decrypting using AES but will get confused when he is decrypting a data secured with the proposed DM.

I. HIGHLIGHT OF THE IMPROVEMENT OF THE PROPOSED DM OVER [7]

This section presents highlights of the innovative aspects of the proposed system over the method adopted by [7] as shown in Table 2.

VII. CONCLUSION AND FUTURE WORK

Recent data breach reports and incidence have heightened the need for addressing communication problems concerning security and privacy [1]–[3], [19]–[24], [68]. In this paper, a deception model is presented to reinforce the conventional encryption scheme for secure communication of online social networks. The implications of the results have three-fold merits over the existing works in the literature. Firstly, it contributes generally to the area of cryptography. Conventional encryption schemes present 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.

TABLE 2. Highlight of the innovations of the proposed DM over current research by [7].

LIMITATION OF METHOD BY [7]	APPROACH USED TO ADDRESS THE LIMITATION
❖ N-grams model was used to derive each instance which will act as the decoy message. **Limitations:** ✚ Unable to grasp the concept of recursivity of natural language. ✚ Unable to capture syntactic and semantic correlation of natural language. ✚ Presenting a sentence probability as a result of forward conditional probabilities does not fit effectively the long-range forward and backward interactions as used in natural language. Thus, they fail grammatically in producing convincing decoys.	❖ The RNN model was used in place of the N-grams as it allows encoding dependencies between inputs. ✚ It is able to grasp recursivity of natural language as it can learn long-distance semantic information of the plaintext. ✚ The RNN model was supplemented with a bi-directional LSTM to allow the hidden-to-hidden connection layer to flow in both directions to exploit the message from the past and the future. Unlike an N-gram model that picks a word and process it alone to predict, other words, the BLSTM-RNN picks a word, explores other words behind it and after it before generating the decoy message, thus allowing the possibility of feedback for every layer and allowing a long-range forward and backward interactions as used during human conversation. Thus, the proposed model generates semantically, syntactically and contextually correct decoy messages which reflect how human use their natural language during a conversation.
❖ A movie subtitle text corpus is used for fetching the message. Supplying an incorrect key during decryption implies that the decoy message will be influenced by the literary style of the movie (chosen) text corpus. **Limitations:** ✚ An attacker with the knowledge of the distribution of the corpus used can recover the message in a jiffy. ✚ Contemporary cryptography work on Kerchoff's [49] principle of no obscurity in the algorithm but only the key, implies that the source of decoy message must be public knowledge. Thus, the corpus used must be exposed to the public.	❖ Large context-relevant information is needed as the source of data to fetch information that will be used as an Oracle to generate the decoy message. This study leverages its labelled training data from several sources such as Wikipedia, dictionary, newspapers, text corpus, public texts and others. ✚ The intuition behind the use of fetching the data from several sources is to make it very difficult for an attacker to be able to learn enough about the distribution of all the sources of data. ✚ This approach also targets the whole breadth of human natural language making it impossible to distinguish the decoy message that will be produced from the underlying plaintext/chat conversation.
❖ No provision for ensuring that keywords/special words contained in the plaintext do not appear during a brute force attack. **Limitations:** ✚ The exclusion of a keyword checking mechanism implies that a chosen ciphertext attack (CCA) may become possible, as an attacker may use the output results from prior decryptions to inform their choices of which ciphertexts have decrypted. ✚ The approach fails to prevent the adversary from usefully mauling encrypted messages to learn partial information of the underlying plaintext/chat message.	❖ This approach ensures that keywords/special words that are in the plaintext do not appear during the brute force attack. ✚ In the development of the BLSTM-RNN, an ICM model was built which identifies keywords/special words and extract it during the processing stage. The special words are stored in the LSTM which acts as a cell memory such that during a brute-force attack, the special word (s) are not revealed as part of the decoys. ✚ This approach specifically handles an adversary that has side information or one that may use part of the decrypted message to maul encrypted messages to form and recover the plaintext.
✚ The approach is restricted to a domain. In their research, the corpus used was for movie subtitles. The algorithm is restricted to the corpus used and cannot be used to generate decoy messages from diverse domain. This is one of the essential keys to successfully fooling an eavesdropper. For instance, an eavesdropper with side information expecting messages related to travel agency will not be fooled if the chat messages his incorrect key generated are from a flower-based domain. Thus, he will discard the message.	❖ HMM allows generation of a language in a way that elements from a family of strings can be listed to generate members of the same family. Thus, the proposed work leverages the HMM model to generate messages that share the same domain as the plaintext to further fool the adversary. ✚ This approach also overcomes an adversary with side information, for instance, an adversary that is expecting messages from biology and expects keywords from the same domain will be fooled as the decoy message will be from biology, However, the special words from the underlying plaintext will have been hidden and so it will be difficult to determine if his attack was generating the real message or a decoy message.
❖ The approach has a specific length of messages that must be greater than or equal to thirty-three (message length >= 33) otherwise it fails grammatically, failing to convince the attacker.	❖ The proposed approach handles messages line by line and is not dependent on a particular length and thus, does not fail grammatically in any instance.

The proposed model addresses this limitation of conventional encryption scheme by yielding domain-specific, coherent but fake message to an attacker who tries to brute-force/decrypt a ciphertext using incorrect keys. Secondly, this work contributed to the only work that proposed a decoy-based deception model in the security of instant messaging (IM) system upon eavesdropping. Thirdly, it contributes an approach of generating convincing decoy message tailored to the domain of the underlying message/plaintext while hiding special/keywords from the plaintext which an attacker may exploit. The proposed model does not eliminate encryption but complements the current encryption scheme to confound the time and resources of the attacker. A proof of concept was designed and evaluated using an instant messaging system to show how the system can be realized within a real-world scenario. The experimental results demonstrate that the proposed model is sufficient for deception when used on a real-world deployment. Consequently, the proposed model can be incorporated into current IM systems such as Whatsapp, Threema and others to address the ravaging eavesdropping attack of messages between conversing parties. It can also serve for encrypting e-mail messages, human-generated messages and documents. Furthermore, the quantum era is fast approaching when current conventional encryption will be unable to withstand the quantum computers. While the proposed scheme was implemented for the current encryption system, it can be adapted to a quantum-based encryption scheme in the quantum era.

This research has laid down the footprint of the plausibility of leveraging convincing natural language decoys in curtailing eavesdropping in the context of English texts. Future work may consider incorporating interactive media such as images, videos and audios. Generating natural language decoys from an image, video is an emerging aspect at the convergence of computer vision and NLP. It establishes the technical foundation and security of innovative applications such as photo, audio and video sharing in OSNs, visual intelligence in chatting robots, aiding the visually impaired persons to perceive visual content.

Another possible area of future research would be human cognitive deception systems. This construction will require capturing the process of human perception, reasoning, emotional state. State-of-the-art Neural Language Models (NLM) may be suitable models to leverage for this purpose. NLM exploits stratified models of nonlinear processing, innovated by biological neural systems to learn ingrained characterizations from language data, in ways intended to imitate/mimic human cognitive abilities. Interlacing such deception model with modern cryptographic schemes such as game-hopping proofs, cryptographic hardness assumptions and reductions may potentially produce a strong cipher that models the human mind. The implication is a very secure deceptive cryptographic scheme that can control the adversary. After all, attackers are humans and their deductions are based on their cognitive state.

REFERENCES

[1] G. Ioannou, P. Louvieris, and N. Clewley, "A Markov multi-phase transferable belief model for cyber situational awareness," *IEEE Access*, vol. 7, pp. 39305–39320, 2019. Available: 10.1109/access.2019.2897923.

[2] World Economic Forum. (2016). *The Global Risks Report*. [Online]. Available: http://www3.weforum.org/docs/GRR/WEF_GRR16.pdf

[3] T. Moore, "On the harms arising from the Equifax data breach of 2017," *Int. J. Crit. Infrastruct. Protection*, vol. 19, pp. 47–48, Dec. 2017.

[4] Y. Kassa, R. Cuevas, and A. Cuevas, "A large-scale analysis of Facebook's user-base and user engagement growth," *IEEE Access*, vol. 6, pp. 78881–78891, 2018.

[5] M. Wollschlaeger, T. Sauter, and J. Jasperneite, "The future of industrial communication: Automation networks in the era of the Internet of Things and industry 4.0," *IEEE Ind. Electron. Mag.*, vol. 11, no. 1, pp. 17–27, Mar. 2017. doi: 10.1109/mie.2017.2649104.

[6] Chatbot News Daily. (2017). *The Rise of Messaging Platforms*. Accessed: Jan. 12, 2019. [Online]. Available: https://chatbotnewsdaily.com/the-rise-of-messenger-platforms-and-its-legal-implications-62fe73355122

[7] K. Joo-Im and J. Yoon, "Honey chatting: 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.

[8] Z. Wang, Z. Ma, S. Luo, and H. Gao, "Enhanced instant message security and privacy protection scheme for mobile social network systems," *IEEE Access*, vol. 6, pp. 13706–13715, 2018. doi: 10.1109/access.2018.2813432.

[9] S. Muftic, N. B. Abdullah, and I. Kounelis, "Business information exchange system with security, privacy, and anonymity," *J. Elect. Comput. Eng.*, vol. 2016, Feb. 2016, Art. no. 7093642.

[10] A. Bogdanov, D. Khovratovich, and C. Rechberger, "Biclique cryptanalysis of the full AES," in *Proc. Int. Conf. Theory Appl. Cryptol. Inf. Secur.* Berlin, Germany, 2011, pp. 344–371.

[11] L. Grassi, "Mixture differential cryptanalysis: A new approach to distinguishers and attacks on round-reduced AES," IACR Trans. Symmetric Cryptol., Tech. Rep., 2018, pp. 133–160.

[12] A. Omolara and A. Jantan, "Modified honey encryption scheme for encoding natural language message," *Int. J. Elect. Comput. Eng.*, vol. 9, no. 3, p. 1871, 2019. doi: 10.11591/ijece.v9i3.pp1871-1878.

[13] M. Coutinho, M. Torquato, and M. Fernandes, "Deep neural network hardware implementation based on stacked sparse autoencoder," *IEEE Access*, vol. 7, pp. 40674–40694, 2019. doi: 10.1109/access.2019.2907261.

[14] D. B. Cousins, K. Rohloff, and D. Sumorok, "Designing an FPGA-accelerated homomorphic encryption co-processor," *IEEE Trans. Emerg. Topics Comput.*, vol. 5, no. 2, pp. 193–206, Oct. 2017. doi: 10.1109/tetc.2016.2619669.

[15] A. Juels and T. Ristenpart, "Honey encryption: Encryption beyond the brute-force barrier," *IEEE Secur. Privacy*, vol. 12, no. 4, pp. 59–62, 2014. doi: 10.1109/msp.2014.67.

[16] A. Juels and T. Ristenpart, "Honey encryption: Security beyond the brute-force bound," in *Proc. Annu. Int. Conf. Theory Appl. Cryptograph. Techn.*, 2014, pp. 293–310.

[17] Statista. (2019). *Most Popular Messaging Apps 2019*. Accessed: Jan. 15, 2019. [Online]. Available: https://www.statista.com/statistics/258749/most-popular-global-mobile-messenger-apps/

[18] *WhatsApp Security*. Accessed: Jan. 15, 2019. [Online]. Available: https://www.whatsapp.com/security/

[19] P. Paganini. (2018). *A Security Issue in WhatsApp Potentially Allows Attackers to Eavesdrop on Encrypted Group Chats*. Accessed: Jan. 15, 2019. [Online]. Available: http://securityaffairs.co/wordpress/67627/hacking/whatapp-group-chat-hack.html

[20] A. Greenberg. (2018). *WhatsApp Flaws Could Allow Snoops to Slide into Group Chats*. Accessed: Jan. 15, 2019. [Online]. Available: https://www.wired.com/story/whatsapp-security-flaws-encryption-group-chats/

[21] H. Pettit. (2019). *WhatsApp Security Flaw Lets ANYONE SPY on Private Chats*. Accessed: Jan. 15, 2019. [Online]. Available: http://www.dailymail.co.uk/sciencetech/article-5257713/WhatsApp-security-flaw-lets-spy-private-chats.html

[22] S. Shah. (2019). *WhatsApp Hack Warning—Newly Discovered Exploit Lets Hackers Access Your Group Chat to Spread Fake Messages*. Accessed: Mar. 15, 2019. [Online]. Available: https://www.thesun.co.uk/tech/6967953/whatsapp-hack-fake-group-chat-messages/

[23] (2019). *Gizmodo*. Accessed: Feb. 15, 2019. [Online]. Available: https://gizmodo.com/nasty-whatsapp-bug-left-users-vulnerable-to-attack-by-s-1829658664

[24] NewsComAu. (2019). *Terrifying WhatsApp Flaw Allows Hackers to Impersonate You*. Accessed: Apr. 15, 2019. [Online]. Available: https://www.news.com.au/technology/online/hacking/the-latest-whatsapp-hack-sees-users-impersonated-by-attackers/news-story/096085d222657dc25d919e39d3d9385e

[25] (2019). *Threema*. Accessed: Feb. 17, 2019. [Online]. Available: https://threema.ch/press-files/2_documentation/cryptography_whitepaper.pdf

[26] N. Kobeissi. *Remarks on Viber Message Encryption*. Accessed: Feb. 17, 2019. [Online]. Available: https://nadim.computer/2016/05/04/viber-encryption.html

[27] M. Luo, L. Cai, Y. Ji, and P. Huang, "A secure instant messaging scheme based on certificateless signcryption," *J. Comput. Inf. Syst.*, vol. 10, no. 14, pp. 6067–6074, 2014.

[28] C.-J. Wang, W.-L. Lin, and H.-T. Lin, "Design of an instant messaging system using identity based cryptosystems," in *Proc. 4th Int. Conf. Emerg. Intell. Data Web Technol.*, Xi'an, China, Sep. 2013, pp. 277–281.

[29] P. P. Wanda and B. S. Hantono, "Efficient message security based hyper elliptic curve cryptosystem (HECC) for mobile instant messenger," in *Proc. 1st Int. Conf. Inf. Technol., Comput. Electr. Eng.*, Semarang, Indonesia, 2014, pp. 245–249.

[30] A. Loukas, D. Damopoulos, S. Menesidou, M. Skarkala, and G. Kambourakis, "MILC: A secure and privacy-preserving mobile instant locator with chatting," *Inf. Syst. Frontiers*, vol. 14, no. 3, pp. 481–497, 2012.

[31] W. Feng, Z. Zhang, J. Wang, and L. Han, "A novel authorization delegation scheme for multimedia social networks by using proxy re-encryption," *Multimedia Tools Appl.*, vol. 75, no. 21, pp. 13995–14014, 2016.

[32] H. C. Chen, H. Wijayanto, C. H. Chang, F. Y. Leu, and K. Yim, "Secure mobile instant messaging key exchanging protocol with one-time-pad substitution transposition cryptosystem," in *Proc. Comput. Commun. Workshops (INFOCOM WKSHPS)*, San Francisco, CA, USA, Apr. 2016, pp. 980–984.

[33] I. Karabey and G. Akman, "A cryptographic approach for secure client-server chat application using public key infrastructure (PKI)," in *Proc. 11th Int. Technol. Secured Trans. (ICITST)*, Barcelona, Spain, Dec. 2016, pp. 442–446.

[34] J. Silverman, J. Hoffstein, and J. Pipher, *An Introduction to Mathematical Cryptography*. New York, NY, USA: Springer-Verlag, 2008.

[35] C. Paar, B. Preneel, and J. Pelzl, *Understanding Cryptography*. New York, NY, USA: Springer, 2009.

[36] J. De Groot. (2019). The History of Data Breaches. Digital Guardian. Accessed: Feb. 17, 2019. [Online]. Available: https://digitalguardian.com/blog/history-data-breaches

[37] P. W. Shor, "Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer," *SIAM Rev.*, vol. 41, no. 2, pp. 303–332, 1999.

[38] T. Monz, D. Nigg, E. A. Martinez, M. F. Brandl, P. Schindler, R. Rines, S. X. Wang, I. L. Chuang, and R. Blatt, "Realization of a scalable Shor algorithm," *Sci.*, vol. 351, no. 6277, pp. 1068–1070, 2016. doi: 10.1126/science.aad9480.

[39] M. Veldhorst, C. H. Yang, J. C. C. Hwang, W. Huang, J. P. Dehollain, J. T. Muhonen, S. Simmons, A. Laucht, F. E. Hudson, K. M. Itoh, and A. Morello, "A two-qubit logic gate in silicon," *Nature*, vol. 526, pp. 410–414, Oct. 2015. doi: 10.1038/nature15263.

[40] O. Reparaz, S. Roy, R. de Clercq, F. Vercauteren, and I. Verbauwhede, "Masking ring-LWE," *J. Cryptograph. Eng.*, vol. 6, no. 2, pp. 139–153, 2016. doi: 10.1007/s13389-016-0126-5.

[41] S. Streit and F. De Santis, "Post-quantum key exchange on ARMv8-A: A new hope for NEON made simple," *IEEE Trans. Comput.*, vol. 67, no. 11, pp. 1651–1662, Nov. 2018. doi: 10.1109/tc.2017.2773524.

[42] N. Sendrier, "Code-based cryptography: State of the art and perspectives," *IEEE Secur. Privacy*, vol. 15, no. 4, pp. 44–50, Aug. 2017. doi: 10.1109/msp.2017.3151345.

[43] D. Bernstein and T. Lange, "Post-quantum cryptography," *Nature*, vol. 549, no. 7671, pp. 188–194, 2017. doi: 10.1038/nature23461.

[44] N. Mouna, "Secure hardware implementation of post quantum cryptosystems," Ph.D. dissertation, Concordia Univ., Montreal, QC, Canada, 2017.

[45] L. Chen, "Cryptography standards in quantum time: New wine in old wineskin?" *IEEE Secur. Privacy*, vol. 15, no. 4, pp. 51–57, 2017. doi: 10.1109/msp.2017.3151339.

[46] K. E. Heckman, F. J. Stech, B. S. Schmoker, and R. K. Thomas, "Denial and deception in cyber defense," *Computer*, vol. 48, no. 4, pp. 36–44, 2015. doi: 10.1109/mc.2015.104.

[47] P. Kaghazgaran and H. Takabi, "Toward an insider threat detection framework using honey permissions," *J. Internet Services Inf. Secur.*, vol. 5, no. 3, pp. 19–36, 2019. Accessed: Apr. 16, 2019.

[48] O. Mor, O. Dunkelman, S. Gibson, and M. Osadchy, "HoneyFaces: Increasing the security and privacy of authentication using synthetic facial images," 2016, *arXiv:1611.03811*. Accessed: Apr. 16, 2019. [Online]. Available: https://arxiv.org/abs/1611.03811

[49] B. M. Brian, S. Hershkop, A. D. Keromytis, and S. J. Stolfo, "Baiting inside attackers using decoy documents," in *Proc. Int. Conf. Secur. Privacy Commun. Syst.* Berlin, Germany: Springer, 2009, pp. 51–70.

[50] 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 *Proc. 3rd ACM Conf. Wireless Netw. Secur.*, 2010, pp. 81–86.

[51] B. Whitham, "Design requirements for generating deceptive content to protect document repositories," in *Proc. Austral. Inf. Warfare Secur. Conf.*, 2014, pp. 20–30.

[52] N. C. Rowe and J. Rrushi, *Introduction to Cyberdeception*. New York, NY, USA: Springer, 2016.

[53] 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.

[54] X. Han, N. Kheir, and D. Balzarotti, "Deception techniques in computer security," *ACM Comput. Surv.*, vol. 51, no. 4, pp. 1–36, 2018. doi: 10.1145/3214305.

[55] T. Mainguy, "Markov substitute processes: A statistical model for linguistics," Ph.D. dissertation, Univ. Pierre Marie Curie, Paris, France, 2014.

[56] (2019). *Ludwig*. Accessed: Dec. 4, 2018. [Online]. Available: https://github.com/oswaldoludwig/Seq2seq-Chatbot-for-Keras

[57] 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.

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

[59] M. Tan, W. Zhou, L. Zheng, and S. Wang, "A scalable distributed syntactic, semantic, and lexical language model," *Comput. Linguistics*, vol. 38, no. 3, pp. 631–671, 2012. doi: 10.1162/coli_a_00107.

[60] O. Ludwig, "End-to-end adversarial learning for generative conversational agents," Jan. 2018, *arXiv:1711.10122*. [Online]. Available: https://arxiv.org/abs/1711.10122

[61] J. Turian, L. Ratinov, and Y. Bengio, "Word representations: A simple and general method for semi-supervised learning," in *Proc. 48th Annu. Meeting Assoc. Comput. Linguistics*, Uppsala, Sweden, 2010, pp. 384–394.

[62] O. Başkaya and D. Jurgens, "Semi-supervised learning with induced word senses for state of the art word sense disambiguation," *J. Artif. Intell. Res.*, vol. 55, pp. 1025–1058, Apr. 2016. doi: 10.1613/jair.4917.

[63] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. Hoboken, NJ, USA: Wiley, 2012.

[64] A. Turing, "Computing machinery and intelligence," Oxford Univ. Press Mind Association, vol. 59, no. 236, pp. 433–460, 1950. Accessed: May 26, 2019. [Online]. Available: https://phil415.pbworks.com/f/TuringComputing.pdf

[65] M. Ferguson, "*The man who knew too much: Alan turing and the invention of the computer*, by David Leavitt and Alan Turing, the Enigma, by Alan Hodges," *J. Homosexuality*, vol. 56, no. 8, pp. 1145–1153, 2009. doi: 10.1080/00918360903275559.

[66] 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 *Proc. 3rd ACM Conf. Wireless Netw. Secur.*, Mar. 2010, pp. 81–86.

[67] C. E. Shannon, "Communication theory of secrecy systems," *Bell Syst. Tech. J.*, vol. 28, no. 4, pp. 656–715, 1949.

[68] O. A. Alzubi, T. M. Chen, J. A. Alzubi, H. Rashaideh, and N. Al-Najdawi, "Secure channel coding schemes based on algebraic-geometric codes over hermitian curves," *J. UCS*, vol. 22, no. 4, pp. 552–566, 2016.

ABIODUN ESTHER OMOLARA is currently pursuing the Ph.D. degree with the School of Computer Sciences, Universiti Sains Malaysia. Her research interests include computer and network security, cyber-security, cryptography, artificial intelligence, natural language processing, and forensics.

AMAN JANTAN is currently an Associate Professor with the School of Computer Sciences, Universiti Sains Malaysia. He has published more than 100 articles in reputable journals and has won national and international recognition in some of his work. His research interests include artificial intelligence, natural language processing, cybersecurity, cryptography, forensic, computer, and network security.

OLUDARE ISAAC ABIODUN received the Ph.D. degree in nuclear and radiation physics from the Nigerian Defence Academy, Kaduna, and the Ph.D. degree in computer science from the Universiti Sains Malaysia, Penang, Malaysia. His research interests include artificial intelligence, robotics, cybersecurity, digital forensics, nuclear security, terrorism, and national security.

KEMI VICTORIA DADA received the M.Sc. degree in statistics from Ahmadu Bello University, Zaria, Nigeria. Her research interests include economic and forensic analysis, security information analysis, and cybercrime solution.

HUMAIRA ARSHAD is currently pursuing the Ph.D. degree in digital forensics with the School of Computer Sciences, Universiti Sains Malaysia. She is currently an Assistant Professor with the Department of Computer Sciences and Information Technology, The Islamia University of Bahawalpur, Pakistan. Her research interests include digital and social media forensics, information security, online social networks, cybersecurity, intrusion detection, reverse engineering, and semantic web.

ETUH EMMANUEL is currently pursuing the Ph.D. degree in computer science with the University of Nigeria, Nsukka. He is currently a Lecturer with Arthur Jarvis University, Nigeria, and has held several appointments including the Pioneer Coordinator of the Department of Mathematics and Computer Science. His research interests include software engineering, security web technology, and intelligent systems.