Natural Language Processing by Enhanced Honey Encryption Technique

Mrinal Paliwal

Abstract— ***Traditional encryption systems and techniques have always been vulnerable to brute force cyber-attacks. This is due to bytes encoding of characters utf8 also known as ASCII characters. Therefore, an opponent who intercepts a cipher text and attempts to decrypt the signal by applying brute force with a faulty pass key can detect some of the decrypted signals by employing a mixture of symbols that are not uniformly dispersed and contain no meaningful significance. Honey encoding technique is suggested to curb this classical authentication weakness by developing cipher-texts that provide correct and evenly dispersed but untrue plaintexts after decryption with a false key. This technique is only suitable for passkeys and PINs. Its adjustment in order to promote the encoding of the texts of natural languages such as electronic mails, records generated by man, still remained an open-end drawback. Prevailing proposed schemes to expand the encryption of natural language messages schedule exposes fragments of the plaintext embedded with coded data, thus they are more prone to cipher text attacks. In this paper, amending honey encoded system is proposed to promote natural language message encryption. The main aim was to create a framework that would encrypt a signal fully in binary form. As an end result, most binary strings semantically generate the right texts to trick an opponent who tries to decipher an error key in the cipher text. The security of the suggested system is assessed..***

Rundown Encryption, Cipher, Natural language, Encoding, Honey encryption.

I. INTRODUCTION

In information-conscious culture, encryption act as a building block to a variety of procedures that include electronic-commerce, online banking, business, social networks, expert system and so on. Digital development has influenced almost all individuals to handle their funds, obtain products and access to healthcare. Gradually individuals have started relying on the use of technology to get immediate access to data, company, family and friends. This rise in global dependency on contemporary infrastructure has given rise to continual security threats. The constant attacks on computer networks, card fraud, malware attacks, and cellphone hacking are proof of the continual security risk. There is a need for enhanced infrastructure, with strong security, for strengthening the computer depended on society for current and future generation and making use of the chances provided by this technology. The research of science and practice of concealing data is known as cryptography, which prevents a distrustful group from learning the content present in the data block. The modern standard of cryptographic structures are more suspected to be attacked by brute force such as an attempt to break a user's password is an attack by brute force and error approach is employed by an adversary in decryption. Traditional encryption structures facilitate the continuation of the use and achievement of an opponent in these kinds of attacks. All standard crypt systems are represented in letters of utf8 or ASCII characters. An opponent who interrupts a cipher text and attempts to decrypt the signal via a false pass key could, therefore, filter out certain decrypted signal and combinations of the same by noting that those sequences are a combination of characters from various languages and symbols that are non-homogeneously dispersed in separate contexts. It is clear that an opponent can judge whether his attack succeeds which is based on the construction/delivery of the text the malicious user recovers during the attacks. A non-homogenous distribution, which makes a meaning-less allocation, means a plaintext and core that is inaccurate. In addition, the emergence of handling tools such as FPGA and GPU also allows the brute-force attack very effective. An intruder who interrupts the encoding signal has, therefore, an elevated opportunity of retrieving the pass key and then using the distinctive method to encode the signal which is the decryption performance depending on the signal distribution/structure.

II. LITERATURE SURVEY

The Security experts and the cryptographic society have suggested various measures to reduce the attacks on standard encryption systems by reducing the duration and cost of the main computing systems. However, the fundamental issue that uses the distinguisher method to communicate decryption outcomes is not addressed by all these processes. An intruder can still decrypt an oracle repeatedly by attempting arbitrary codes until a feasible signal is found. Honey Encryption incorporates some of the other schemes ' known characteristics. In particular, decoy schemes [1] are used as defense mechanisms. Perhaps the most popular security decoy concept is honeypot, with some exciting, though false data that is attractive for the intruder. The honey denotes the attractiveness of the false information and the capability to get invaders to aim the honeypots. Honey encryption technique includes a false email, which it generates when required, by Honey words [2]. A wide-ranging evaluation of honey word [3] security was carried out in the year 2018. A method was proposed [4] to avoid off-line attack by saving fake passcodes with real user pin code. It was specifically aimed at memory archives because false emails generated in the scheme of six

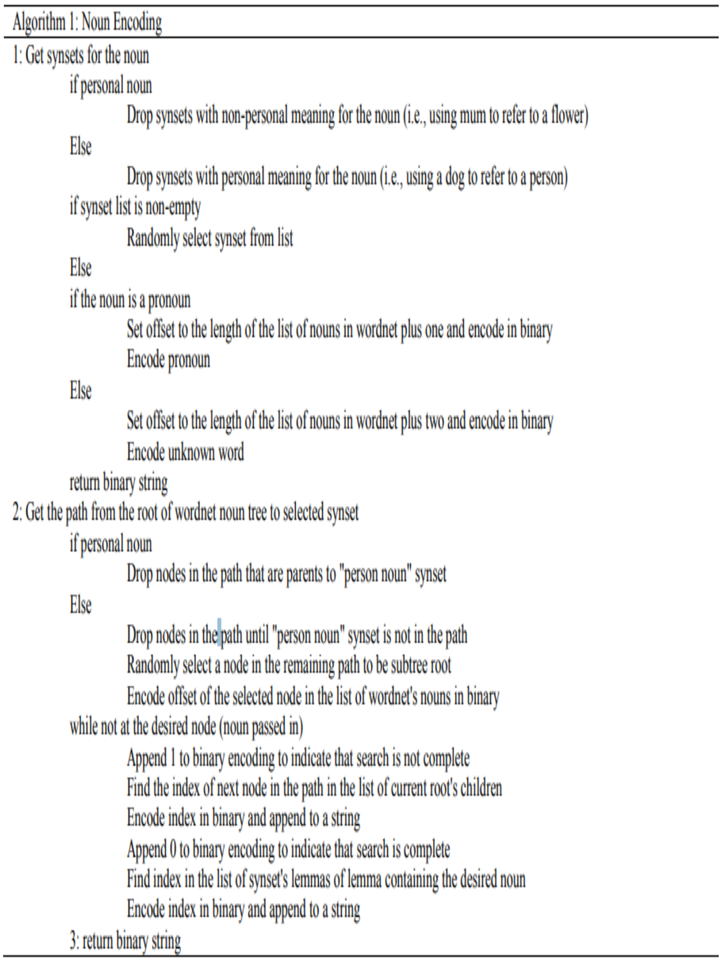
Revised Manuscript Received on August 05, 2019.

Mrinal Paliwal, Department of Computer Science and Engineering, Sanskriti University, Uttar Pradesh, India. (E-mail: sanpubip@gmail.com)

Kamouflage are stored in the code sequence. The similarity between Kamouflage and honey encryption is simple to see as both have the same concept of misleading and stopping attackers from acknowledging the right information from deliberately false information (e.g. false lists of passwords). In addition, another system called organizational routing scheme [5] has been released when Honey Encryption was introduced. It was also a technique for cryptography attacks, using the theory that all pass key could be used and it would give complaints which might mislead the malicious user, in the same way as honey encryption. Another structural code was introduced [6] but there is no official mathematical evidence. However, syntactic safety and natural linguistic capacity are provided. In an attempt to ensure assured and verifiable security in the sector, structural engineering requires more studies, as is Honey encoding. Another theory was proposed on natural language encoder by [7] for safeguarding passcode vaults. The main focus was on producing false but faithful looking vault to the malicious user. The mischievous user cannot decide whether it’s the initial or false vault. The scheme also requires the assailant to go internet to trace and prohibit its operations. This works well for keyword configurations, but cannot be expanded to help big human-text situations. An enhanced conventional honey encoding scheme [8] was introduced in the year 2015 to promote genomic information injection. This disclosed method is for ensuring DNA products as well as for shielding genomes against mauling by an unlimited intruder.

III. PROPOSED WORK

The strength of the proposed strategy is used to convince the assailant that it has the initial signal, and to confuse the malicious user. “Python 3.63” and “Natural Language Processing” (NLP) databases are used for the assessment of the proposed scheme. The natural language processing tool reveals how words relate and how significant phrases are combined to shape an expressive sequence of words. The decoy signal is entirely distinct from the initial text (plaintext) but has semantic and relative significance. Each transmitted signal is handled like a phrase sequence and computed in the English Language with linguistic features. Each term is categorized by its lexical functions according to the part of the speech. The primary components of the English language are names, pronouns, verbs, adverbs, adjectives, prepositions, interjection and conjunctions. Algorithm 1 shows how nouns are processed and encrypted and the entire working of the proposed system is exemplified in Figure 1.



```
Algorithm 1: Noun Encoding
1: Get synsets for the noun
    if personal noun
        Drop synsets with non-personal meaning for the noun (i.e., using mum to refer to a flower)
    Else
        Drop synsets with personal meaning for the noun (i.e., using a dog to refer to a person)
    if synset list is non-empty
        Randomly select synset from list
    Else
    if the noun is a pronoun
        Set offset to the length of the list of nouns in wordnet plus one and encode in binary
        Encode pronoun
    Else
        Set offset to the length of the list of nouns in wordnet plus two and encode in binary
        Encode unknown word
    return binary string
2: Get the path from the root of wordnet noun tree to selected synset
    if personal noun
        Drop nodes in the path that are parents to "person noun" synset
    Else
        Drop nodes in the path until "person noun" synset is not in the path
        Randomly select a node in the remaining path to be subtree root
        Encode offset of the selected node in the list of wordnet's nouns in binary
    while not at the desired node (noun passed in)
        Append 1 to binary encoding to indicate that search is not complete
        Find the index of next node in the path in the list of current root's children
        Encode index in binary and append to a string
        Append 0 to binary encoding to indicate that search is complete
        Find index in the list of synset's lemmas of lemma containing the desired noun
        Encode index in binary and append to a string
    3: return binary string
```

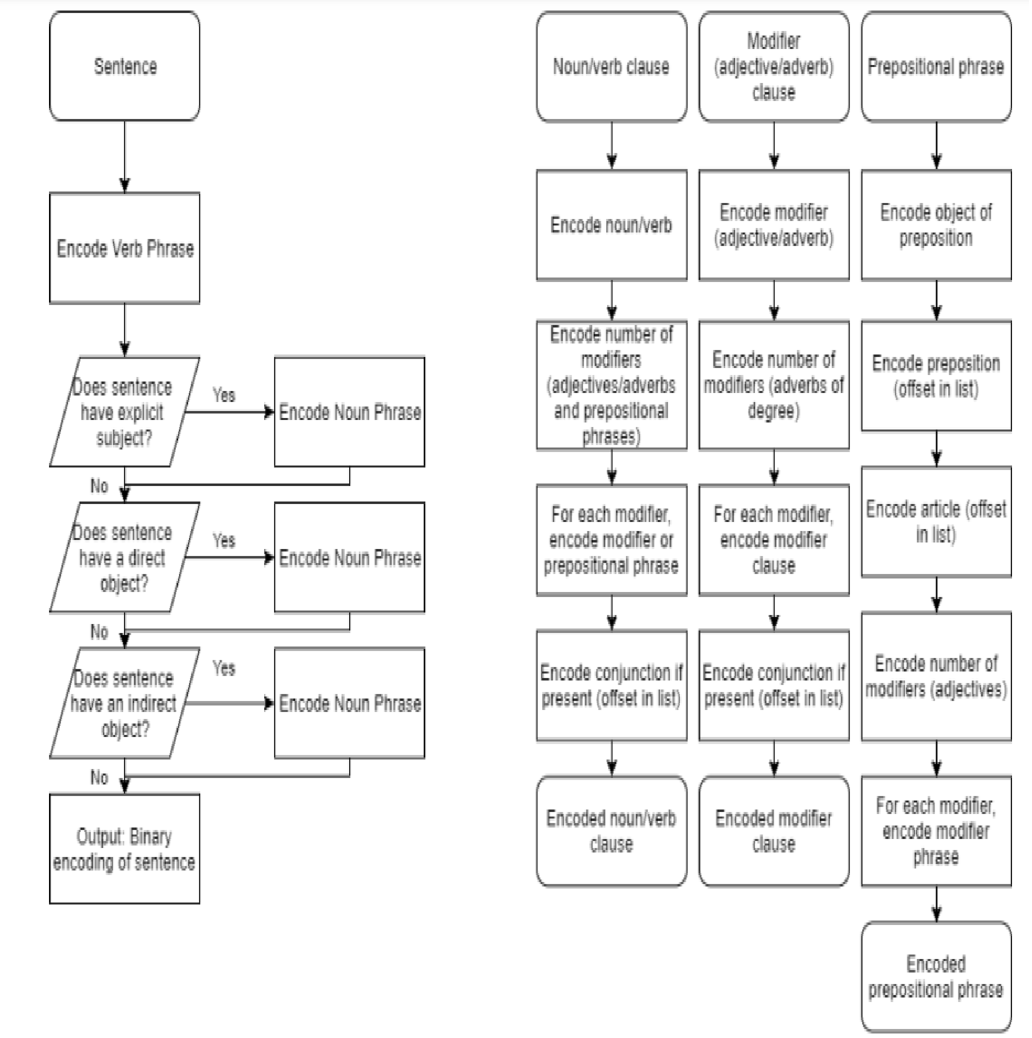
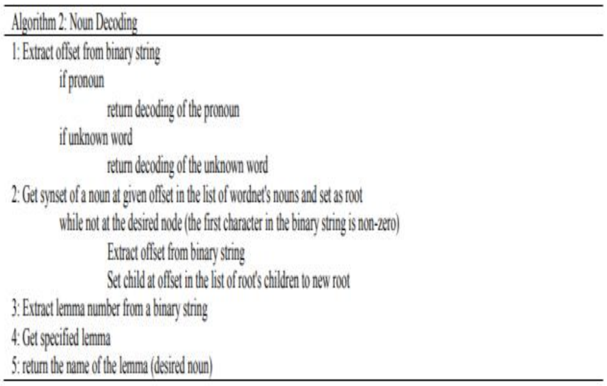

Sentence
Encode Verb Phrase
Does sentence have explicit subject?
Yes
Encode Noun Phrase
No
Does sentence have a direct object?
Yes
Encode Noun Phrase
No
Does sentence have an indirect object?
Yes
Encode Noun Phrase
No
Output: Binary encoding of sentence
Noun/verb clause
Encode noun/verb
Encode number of modifiers (adjectives/adverbs and prepositional phrases)
For each modifier, encode modifier or prepositional phrase
Encode conjunction if present (offset in list)
Encoded noun/verb clause
Modifier (adjective/adverb) clause
Encode modifier (adjective/adverb)
Encode number of modifiers (adverbs of degree)
For each modifier, encode modifier clause
Encode conjunction if present (offset in list)
Encoded modifier clause
Prepositional phrase
Encode object of preposition
Encode preposition (offset in list)
Encode article (offset in list)
Encode number of modifiers (adjectives)
For each modifier, encode modifier phrase
Encoded prepositional phrase

Figure 1 Flow chart of the proposed system

Algorithm 2 depicts the processing and decryption of substantives, and figure 2 shows an outline of the system in its entirety with decryption. The framework is been proposed which allows the binary encoding of a phrase. The encoding approach produces a syntactically right phrase for binary strings.



```
Algorithm 2: Noun Decoding
1: Extract offset from binary string
        if pronoun
                return decoding of the pronoun
        if unknown word
                return decoding of the unknown word
2: Get synset of a noun at given offset in the list of wordnet's nouns and set as root
        while not at the desired node (the first character in the binary string is non-zero)
                Extract offset from binary string
                Set child at offset in the list of root's children to new root
3: Extract lemma number from a binary string
4: Get specified lemma
5: return the name of the lemma (desired noun)
```

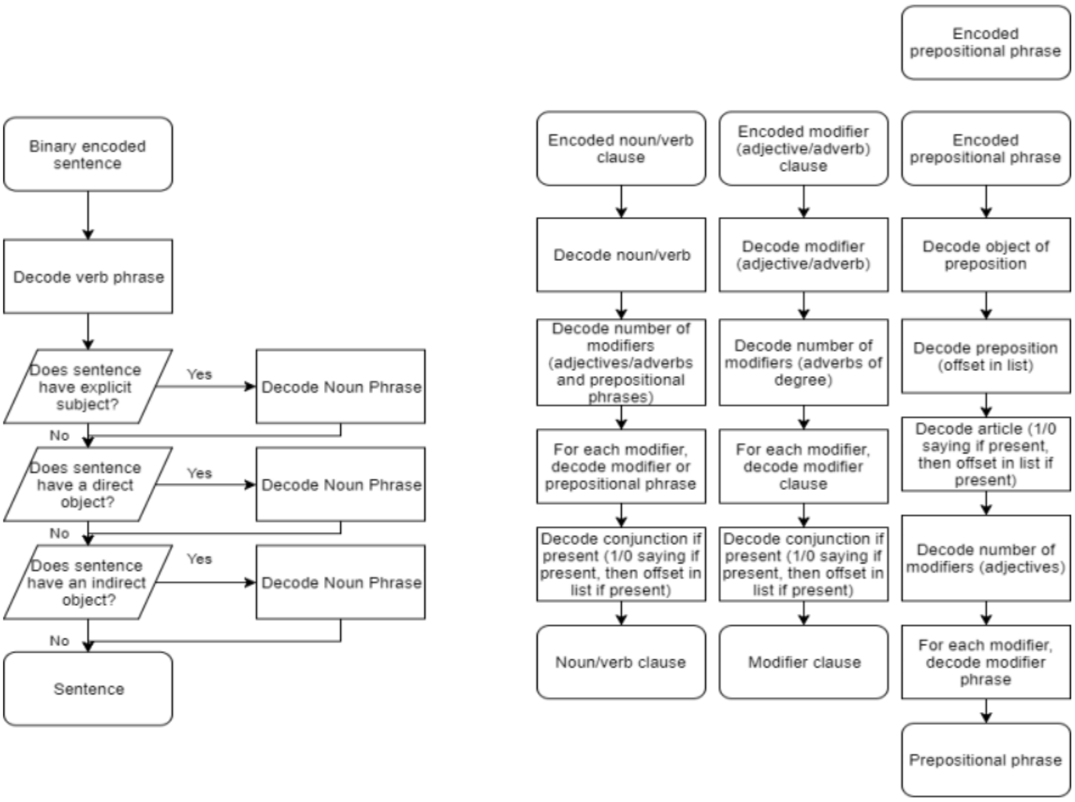

Binary encoded sentence
Decode verb phrase
Does sentence have explicit subject?
Yes
Decode Noun Phrase
No
Does sentence have a direct object?
Yes
Decode Noun Phrase
No
Does sentence have an indirect object?
Yes
Decode Noun Phrase
No
Sentence
Encoded noun/verb clause
Decode noun/verb
Decode number of modifiers (adjectives/adverbs and prepositional phrases)
For each modifier, decode modifier or prepositional phrase
Decode conjunction if present (1/0 saying if present, then offset in list if present)
Noun/verb clause
Encoded modifier (adjective/adverb) clause
Decode modifier (adjective/adverb)
Decode number of modifiers (adverbs of degree)
For each modifier, decode modifier clause
Decode conjunction if present (1/0 saying if present, then offset in list if present)
Modifier clause
Encoded prepositional phrase
Encoded prepositional phrase
Decode object of preposition
Decode preposition (offset in list)
Decode article (1/0 saying if present, then offset in list if present)
Decode number of modifiers (adjectives)
For each modifier, decode modifier phrase
Prepositional phrase

Figure 2 Overview of proposed system with decryption

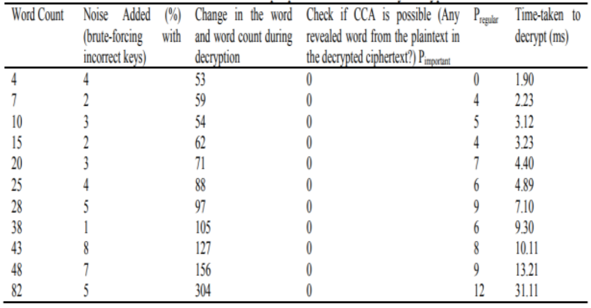
Any standard cipher is used to merge the suggested algorithm with a particular use. For example, the sender encrypts the binary string by means of a chipper. A wrongly labeled code is produced with an incorrect key that should interpret a semantically right signal. Any improper pass key will correctly decode to the correct message of a completely different length and structure. The proposed approach is entirely dissimilar from the prevailing approach as no part of the plaintext is revealed during decryption with an incorrect key. The opponent can't carry out a chosen-cipher text attack with this proposed strategy.

IV. RESULTS

For the evaluation process, testing with a controlled quantity of unwanted data is embedded to a cipher test and associating the rationality of outcome result. The number of words encrypted is known as Word Count. Amount of flipped casual bits is known as Noise Added. If the opponent attempts to decrypt the passkey, the rates of production of true words generated are called Change in the Word Count. The amount of significant words occurs in the decrypted signal from the fundamental plaintext when the opponent attempts the wrong passkey. PINTER is key phrases in the plaintext that must not occur with incorrect pass keys in case of decryption (words, pronoun, noun, adverb, adjective) is termed as P-Important. The decrypted signal displays periodic phrases in the fundamental plaintext when the opponent attempts wrong pass keys. This is important, as frequent phrases assist in shaping the phrase so that it can be used to retrieve the plaintext in every electronic mail. The time taken to decrypt by employing the proposed technique is termed as "Time-Taken".

Table 1 below depicts the performance of the proposed scheme.

Table 1 Performance of Proposed scheme



Word Count	Noise Added (%) with (brute-forcing incorrect keys)	Change in the word and word count during decryption	Check if CCA is possible (Any revealed word from the plaintext in the decrypted ciphertext?) $P_{important}$	$P_{regular}$	Time-taken to decrypt (ms)
4	4	53	0	0	1.90
7	2	59	0	4	2.23
10	3	54	0	5	3.12
15	2	62	0	4	3.23
20	3	71	0	7	4.40
25	4	88	0	6	4.89
28	5	97	0	9	7.10
38	1	105	0	6	9.30
43	8	127	0	8	10.11
48	7	156	0	9	13.21
82	5	304	0	12	31.11

V. CONCLUSION

Messages were programmed in American standard Code for Information Interchange which were making them susceptible to a crude attack, in the traditional encryption strategy, an attempt to decrypt a cipher text by simply removing the true messages from false messages, based on the standardized allocation of characters. Honey encryption provides a corrective measure to the traditional encryption scheme by offering a blockade against brute-force attacks. Honey encoding is a decoy-based encoding system that gives meaningful, unsecured plaintexts. Therefore, the enemy must deliberate that the enemy has plaintext when the malicious user has a message of the decoy.In this paper, a modified honey encryption technique is proposed for promoting human-generated text identification through the use of an artificial languages method. The proposed strategy plans the link between phrases to capture language patterns, which are syntactically and semantically significant, as the base for creating fake texts as decoys, so that an opponent uses an invalid key to decode the code text. The text design and text duration stored in the fundamental text is held confidential and unsuccessful encryption generates radically distinct messages from the initial text, such as decoys, which can effectively hide and protect highly confidential messages and human-written records.

REFERENCES

1. Cohen, F. (2006). The use of deception techniques: Honeypots and decoys. Handbook of Information Security, 3(1), 646–655.
2. Bojinov, H., Bursztein, E., Boyen, X., & Boneh, D. (2010). Kamouflage: Loss-resistant password management. In European symposium on research in computer security (pp. 286–302). Springer.
3. Wang, Ding, et al. "A Security Analysis of Honeywords." NDSS. 2018.
4. Haugum, Torstein, and Lars-Christian K. Rygh. Design, implementation and analysis of a theft-resistant password manager based on Kamouflage architecture. MS thesis. Universitetet i Agder; University of Agder, 2015.
5. Jo, H.-J., & Yoon, J. W. (2014). Poster: statistical coding scheme for the protection of cryptographic systems against brute-force attack. In Proceedings of the 35th IEEE Symposium on Security and Privacy. Retrieved from https://ieeexplore.ieee.org/document/ 6859779.
6. Jo, H.-J., & Yoon, J. W. (2015). A new countermeasure against brute-force attacks that use high performance computers for big data analysis. International Journal of Distributed Sensor Networks, 11(6), 406915.
7. R. Chatterjee, J. Bonneau., A. Juels and T. Ristenpart, “Cracking Resistant Password Vaults using Natural Language Encoders,” Proceedings – IEEE Symposium on Security and Privacy, no. 7163043, pp. 481-498, July 2015.
8. Z. Huang, E. Ayday, J. Fellay, J. Hubaux and A. Juels, “Genoguard: Protecting genomic data against brute-force attacks,” IEEE Symposium on Security and Privacy, pp. 447-462, 2015. DOI 10.1109/SP.2015.34.
9. Gamido HV, Sison AM, Medina RP., "Modified AES for Text and Image Encryption," Indonesian Journal of Electrical Engineering and Computer Science (IJEECS), vol. 11(3), pp. 27, May 2018.
10. Sultanpure KA, Gupta A, Reddy LS., “An Efficient Cloud Scheduling Algorithm for the Conservation of Energy through Broadcasting,” International Journal of Electrical and Computer Engineering (IJECE), vol. 8(1), pp.179-88, Feb 2018.
11. Singh JP, Kumar S. "Authentication and encryption in cloud computing," In Smart Technologies and Management for Computing, Communication, Controls, Energy and Materials (ICSTM), 2015 International Conference on 2015 May 6 , IEEE, pp. 216-219.
12. Jaber AN, Zolkipli MF, "Use of cryptography in cloud computing. In Control System," Computing and Engineering (ICCSCE), 2013 IEEE International Conference on 2013 Nov 29. IEEE, pp. 179-184
13. Chen D, Zhao H., "Data security and privacy protection issues in cloud computing," In Computer Science and Electronics Engineering (ICCSEE), 2012 International Conference on 2012 Mar 23. IEEE, vol. 1, pp. 647-651.
14. Omolara OE, Oludare AI, Abdulahi SE., "Developing a modified Hybrid Caesar cipher and Vigenere cipher for secure Data Communication," Computer Engineering and Intelligent Systems, vol. 5, pp. 34-46, 2014.
15. Stallings W, Brown L, Bauer MD, Bhattacharjee AK, "Computer security: principles and practice," Pearson Education 2012.
16. Lorek P, Zagórski F, Kulis M., "Strong stationary times and its use in cryptography," arXiv preprint arXiv:1709.02631, Sep 2017.
17. Jo HJ, Yoon JW, "A new countermeasure against brute-force attacks that use high-performance computers for big data analysis," International Journal of Distributed Sensor Networks. vol. 11(6), pp. 406915, Jun 2015.
18. Beunardeau M, Ferradi H, Géraud R, Naccache D., "Honey Encryption for Language," In International Conference on Cryptology in Malaysia. Springer, Cham, pp. 127-144, Dec 2016.
19. Kim JI, Yoon JW, "Honey chatting: A novel instant messaging system robust to eavesdropping over communication," In Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on 2016 Mar 20. IEEE, pp. 2184-2188.
20. Omolara AE, Jantan A, Abiodun OI, Poston HE, "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 2018, vol. 1, 2018.
21. Cousins DB, Rohloff K, Sumorok D., "Designing an FPGA-accelerated homomorphic encryption co-processor," IEEE Transactions on Emerging Topics in Computing. vol. 5(2), pp. 193-206, Apr 2017.
22. Kestur S, Davis JD, Williams O, "Blas comparison on FPGA, CPU and GPU. VLSI (ISVLSI)," 2010 IEEE computer society annual symposium on 2010 Jul 5. IEEE, pp. 288-293.
23. Sutikno T, Idris NR, Jidin A., "High-Speed Computation using FPGA for Excellent Performance of Direct Torque Control of Induction Machines," Telecommunication Computing Electronics and Control (TELKOMNIKA), vol. 14(1) pp. 1-3, Mar 2016.
24. Marks M, Jantura J, Niewiadomska-Szynkiewicz E, Strzelczyk P, Góźdź K., "Heterogeneous GPU&CPU cluster for high-performance computing in cryptography," Computer Science, vol. 13, pp. 63-79, 2012
25. Juels A, Ristenpart T., "Honey encryption: Security beyond the brute-force bound," In Annual International Conference on the Theory and Applications of Cryptographic Techniques 2014 May 11, Springer, Berlin, Heidelberg, pp. 293-310.
26. Juels A, Ristenpart T., "Honey Encryption: Encryption beyond the brute-force barrier," IEEE Security & Privacy, vol. 12(4), pp. 59-62, Jul 2014.