A Review: Reading of Text Hidden in Image Using Steganography

Purnima [1], Lalitkumar P Bhaiya [2], Ghanshyam Sahu [2]

[1]*Student, Dept Of CSE, BCET, Durg, India.*

[2]*Assistant Professor, Dept Of CSE, BCET, Durg, India.*

Corresponding Author: i4usmriti@gmail.com

Abstract: - We occasionally hear in the news about a breach or attack on certain well-known firms as though it were just another piece of news, but in reality, it is a serious problem because it involves the personal information of people, their money in trade, and the management of their businesses and projects. The honey encryption planner is going to be the topic of conversation for the duration of this essay. Honey Encryption is an encryption method that provides plain acceptable text in order to provide flexibility against brute-force attacks. Honey Encryption does this by encoding data in a way that is not easily broken. It is difficult to build a convincing message trap that is flawless enough to fool the striker even when he feels that he possesses the message in its original form. This is due to the fact that for each key that is used by a trespasser to decode a message, two key regions are open. The typo problem is the second difficulty, and it happens when a lawful user pushes the wrong key by accident. This causes the user to see what appears to be real false plain text even though the user did nothing wrong. Our goal is to come up with more foolproof ruses that are clever enough to prevent an intruder's attempts to discern the genuine meaning of what we are attempting to say through our communication. We also need new security methods because the attackers are looking for new ways to attack the systems, so we proposed a new way to protect messages and passwords well and make them difficult to break and take all of the possibilities of attack, including the brute-force, and then the data is hidden in an image with a public secret key. This solution was developed because the attackers are looking for new ways to attack the systems. This is due to the fact that the attackers are continuously searching for new ways to launch attacks on the systems.

Key Words: — ***Honey encryption, ElGamal algorithm, steganography, brute force attack, typo problem, honeyword, DTE.***

I. INTRODUCTION

There is, of course, a downside to all this progress: the password that grants access to our private lives and data [1]. However, every facet of our life now relies on either a single number or a string of characters. We handpick it and craft it to keep vigil over the entry point to our data and information. In this post, we'll take a look at the honey encryption planner and see how it stacks up.

The honey algorithm [2] is a type of cryptographic system that offers resistance to brute-force attacks [3] by presenting a convincing but phoney plain text for each blank key that an intruder attempts to decode a message with. [2] This stops the intruder from employing brute force to decode the data. The first major challenge is coming up with a convincing trap message that is good enough to fool an attacker into thinking they got the real message. The typo problem arises when a legitimate user accidentally clicks the void key [4], resulting in the display of genuine trap plain text.

To this end, we're working to develop better disguised approaches that can successfully foil an adversary's decoding attempts. Due to the fact that hackers are always trying to find new ways to breach security, we've developed a novel method to protect messages and passwords that is both difficult to break and takes into consideration all conceivable kinds of attack, including brute force. One method to increase data security is to encrypt the message using the ElGamal algorithm, which was created by Taher Al-Jamal in 1985 and is used in public key encryption. This is done by combining more than one data security technique (cryptography and steganography) to reap the benefits of one and overcome the drawbacks of the others. To do this, we apply the Diffie-Hellman key exchange principle

Manuscript revised July 01, 2023; accepted July 02, 2023. Date of publication July 03, 2023.

This paper available online at www.ijprse.com

ISSN (Online): 2582-7898; SJIF: 5.59

to a mathematical problem known as the discrete logarithm problem [5] [6].

Then the honey algorithm is used to encrypt the message with the password, and the result is 100. However, if the user enters an incorrect password that is not present in the Distribution Transform Encoding (DTE) and inverse table, the password will appear to be incorrect; however, if the attacker enters an incorrect password that is already in use, the message will be decrypted and the plain text will be displayed. If the DTE tables record a password that is not valid, the administrator will receive a warning that the user database has been compromised. After then, the encrypted message, public key, private key, and all of the tables and variables of the honey algorithm, as well as the DTE and inverted table, will be hidden, along with any other information related to the ElGamal technique. To conceal information, a cover picture with a public secret key only has to reveal the location of the public key, as in the case of a one-way transmission from a sender to a receiver. This is due to the fact that the cover art has been encrypted using a publicly available secret key.

The remainder of the paper is organised as follows: In the first part, we discuss the brute force attack that required us to take countermeasures. The core ideas and components of the honey algorithm are defined in the second section. In the third section, we'll look back at the primary obstacles faced by the honey algorithm, including the creation of the honey message and typos, and we'll evaluate the pros and cons of several approaches proposed by researchers for addressing the latter. After a brief summary in the fourth section, the results and suggestions are presented in the last section.

II. RELATED WORK

In [33], a multi-modal biometric system uses both fingerprints and iris scans to identify individuals. When fusing the biometric data, the gradient pyramid approach is used as the method of choice. For the purpose of encrypting the concatenated template, the honey encryption approach was utilised. The results of the tests indicate that the proposed algorithm DWTSVDGOA generates an NC value of 1, a PSNR value of 90.75, and an SSIM value of 0.99. Both the performance and the evaluation of the technique have demonstrated that it is superior to other ways of photo watermarking that are currently in use. This conclusion was reached after the performance of the methodology was evaluated.

Honey encryption was utilised as part of [34]'s cryptographic science for the purpose of protecting communications networks. This was achieved by presenting the hacker with a series of fake keys that gave the impression of being authentic. The writers of this research work present a more effective strategy that, when used in conjunction with honey encryption, can give satisfactory outcomes. This method is described in this paper.

In order to strengthen the data protection offered by Wireless Sensor Networks (WSN), the scope of this study encompasses both the source encryption and channel encryption of input data sets. The number [35] denotes the reference that should be used for this publication. It is the implementation of honey encryption for the information bits as source encryption, and it includes Gaussian Frequency Shift Keying (GFSK) for the data that has been honey encrypted so that Frequency Hopping Spread Spectrum (FHSS) may be performed as channel encryption. Honey encryption is also known as asymmetric encryption. Within a WSN, the output of the FHSS is sent with the assistance of the Frequency Hopping Multiple Access (FHMA) protocol. Honey encryption refers to the process of applying honey encryption to the information bits so that they can be encrypted as a source. As a consequence of this, it is difficult for hackers to breach via channels, and there is also no prospect of identifying or decoding the information using a brute-force assault since honey encryption prohibits that option from occurring. As a result, it is impossible for hackers to get access. It safeguards the data by utilising two distinct types of security simultaneously.

The authors of this study [36] have devised a solution to the problem of ensuring the safety of data while it is being stored in Hadoop, and the answer may be located in their work. The authors came up with Attribute-Based Honey Encryption (abbreviated as ABHE), which is a mix of attribute-based encryption and honey encryption on Hadoop. It is denoted by the acronym ABHE. This method allows for the usage of encoded files that have previously been decoded by the Mapper after being stored in HDFS. In addition, the authors tested the innovative ABHE technique by encrypting and decrypting a variety of file sizes with it. They compared the results of their tests with those of other methods already on the market, such as AES and AES with OTP. When it comes to both encrypting and decrypting data, the ABHE approach demonstrates a significant and easily observable performance advantage.

Using a honey-encryption cryptographic technique, the authors of this study [37] suggest a safe privacy protection migration for data that is outsourced to the cloud. This migration would

preserve the data's privacy during the process. In addition, while we are moving data from our current server storage system to the cloud server storage system, we are following a migration process, which assures that both the integrity and confidentiality of the data will be preserved during the transfer. The author of this work focuses their emphasis in [38] on plaintexts, which are also known as non-numerical informative transmissions. We need to capture both the empirical and contextual aspects of the language in order to trick the attacker into thinking the decoy message came from a certain source. This will allow us to successfully fool the attacker. That is to say, there should be no differential in the language used between legitimate and fraudulent communications without compromising the structure of the authentic message. This is because there is no way to verify the authenticity of the sender.Natural language processing and extended differential privacy are two of the methods that he use in order to be successful in overcoming this challenge. When it comes to modelling privacy for text documents, the major focus should be on machine learning approaches such as word embeddings, bags-of-words, word embeddings, word extraction, and transformers for text processing. These are just some of the techniques that may be utilised. Using e-differential privacy is the next stage, which will allow us to determine whether or not this approach is secure.

Honey encryption is the foundation of a security solution that we develop in [39] to protect smart card password authentication from brute force and denial of service attacks. In this research, we establish this security solution to defend smart card password authentication. In particular, we utilise honey encryption as a defence mechanism against brute force assaults. This research details an enhanced honey encryption (HE) technique in [40]. The goal of this research is to increase the security of instant messaging networks while at the same time squandering the time and resources of antagonistic users. This study contributes to the enhancement of the HE schemes by making use of natural language processing techniques to produce chat messages that are semantically reasonable but entirely fake. Specifically, this is accomplished through the employment of a chatbot. These messages are meant for the opponent to use while he is carrying out his assaults, and they are directed to them specifically. An opponent is unable to discern decoy messages from plaintext when the encryption is conducted using the erroneous key, which is one of the conclusions of the evaluation, which indicates that the one-of-a-kind system is resistant to eavesdropping.

GenoGuard is a security system that is described in reference number [41]. Its mission is to provide an unprecedented level of protection for genetic data. Honey encryption (HE), a game-changing theoretical paradigm for encryption, is implemented into GenoGuard. HE is an acronym for "honey encryption." GenoGuard is the answer to the unanswered question of how to apply HE methods to the extremely non-uniform probability distributions that are typical of genetic data sequences. As a direct result of this, the issue at hand has been resolved. In addition to proving that decryption with any key will result in a convincing genome sequence, GenoGuard offers a guarantee of information-theoretic security against message recovery attacks. This is accomplished by showing that any key will provide the same result. To accomplish this goal, it is necessary to demonstrate that it is possible to obtain a believable genome sequence with any key. In addition to this, we also study information attacks that take place in the background. In conclusion, we offer a method of implementing the GenoGuard software that is not only successful but also parallelized.

This study contributes to the development of the honey encryption methods, which are described in [42]. The Chinese identity numbers, mobile phone numbers, and debit card passwords are the three separate sorts of private data that are then utilised with these credentials. You may read more about this research in [42]. Our system's effectiveness is analysed, and a solution to the problem of excessive overhead expenditures is put up for consideration. Additionally, the insights that were learned during the process of creating, implementing, and assessing the honey encryption approach are shared here.

Definition of the typo-based solution offered by various authors

	Authors	*Characterization*	*Advantage and Disadvantage*
1	[53]	- A typo-tolerant tester that fits reasonably well with the current password authentication scheme is introduced in this proposal. - The plan found out that if the typo-tolerant system was introduced, at least a minute would have been spared for 20 percent of consumers.	For the current password-based authentication method, this approach is appropriate because it incorporates a caps lock corrector, a first-case flip corrector, and an additional character at the end corrector to improve usability, but this approach is not suitable for dealing with the HE scheme's typo issue.
2	[54]	In an offline and online environment, the scheme offeredtwo forms of typo-safety to cope with numerous typo problems while also ensuring message recovery in a traditional HE scheme.	The kind A protocol is simpler to enforce since it needs just a server but themain downside is that the size of the key is limited and also there is the difficulty of finding typos in certain environments. kind B is an upgrade over kind A since a user can quickly find typos if he recollect his pin, but a key point here is that to check his address, the user needs to recall the pin.
3	[55]	- A customizable typo resistant password verification is presented in this plan. - This thesis suggests a straightforward blacklisting method in which it is forbidden for a limited number of dangerous typos to be admissible in the typo cache.	This thesis is an expansion of existing typo forgiving password programs but is not meant to function on typos on a decoy device. It can be changed for HE, however.
4	[4]	- rewrite the password - honey checker - honey word –service	- Help the user to retype a second password We give the attacker a bigger chance - How do we know who the attacker is from the legitimate user
5	[27]	- "online verification of plaintexts" - Honey checker	- The legitimate user enables not to make a mistake because he will check directly upon entry and have a clear idea of the structure of the entered data or dealing with it such as (SSN, E-mail, credit card number) - Where the user misspells the way that they create a proper credit card, but not the one they have, online investigation will not mark it as a misspelling and let it advance - A honey checker must be used, it will mark an error and send an alarm. This is the reason why honey words are needed. To find out if the database has been compromised - Since the verification service works with structured data, it is not a practical solution to user typos when the data consists of natural language passwords, as it is usually limited by length and some ASCII characters. This makes the online investigation service useful for PBE technologies in general and even with data with a clear architecture does not offer a

			complete solution.
6	[10]	- error-detecting codes - checksums, - online verification of plaintexts	- - It helps the user to discover and reduce errors - But it helps to reduce the size of the key area, and therefore it causesa security deterioration. Therefore, careful construction and application must be done - Fake passwords/honey tokens are suggested to be shared frankness between password vault apps and providers of services. Applying debugging codes to regular texts in HE can generate honey tags without explicit participation. - Since this technique reduces message space, it degrades protection marginally and should be used with caution. However, it does provide an intriguing way to link HE security to online security tests.
7	[13]	- Honey tokens without explicit sharing. - False passwords/honey tokens are directly exchanged between password vault apps and serviceproviders.	The bogus passwords/honey tokens were proposed to be directly exchanged between password vault applications and service providers. Honey tags can becreated without explicit involvement by the application of debugging codes tostandard texts in HE. By reducing message space, this method degrades protection slightly, and it should be implemented with caution. However, it provides an intriguing comparison of HE security to online security controls.
8	[2]	- tail-tweaking methods - error-detection code	- When tail-tweaking, it can be beneficial if the password tail is distinct from the honey word tails, so that a typing error would not convert the password to a honey word. - The honey word tails can also be very distinct from one another so that the password does not stand out like the sweet word that is "the most distinctive"from the others. Typos can be detected using an error-detection code (as for ISBN book codes). - This property enables the identification of a single-digit replacement or a transposition of two neighboring digits in the tail.
9	[56]	- The first suggestion is to add any detail to the plaintext that is exclusive and verifiable by the patient but useless to the adversary. We propose adding a stringof bits similar to the seed. - -Before encryption, the device should have a poolof N validation images from which the user can select. The validation images do not need to be used in the ciphertext. - is based upon the concealment of a biometrictemplate among decoys.	- This alternative works best if the PIN is a uniformly random string; otherwise, since the PIN is no longer consistent, it will compromise protection. - By merely signaling that a shown image is familiar, the user can validate proper decryption. - With the age of the patient, and if he has a disease on his finger, or ifhe works in difficult work or difficult manual work, it may damage the fingerprint, and thus it may prevent him from entering and verifying the password
10	[57]	- Each person will create a text using a limited number of characters that are unrelated to some of thehealth attributes of his or her health record.	- When a user enters a password, a validation text appears, and the user can check to see whether it is the text he or she entered. Otherwise, the recipient must retype his or her email.
11	[44]	the use of honeywords made up of other people's passwords	The user is unable to forget the passwords of other users.
12	[50]	- Type scheme is designed for a conventionalclient-server model. - - Type scheme is designed for an extended system model with an additional database manager.	- In terms of false-positive score, it has a disadvantage. Even if a user does not make any typos in the password, he or she will see that there are certain typos inthe password. - Users are burdened by having to memorize side detail in addition to passwords. - The type scheme was created with a traditional client-server model in mind.Even among the schemes, it is the easiest and most efficient. - The type scheme is intended for a framework model that includes an external database manager. It solves the first form of accuracy problem. By using additional side details, it provides high precision in detecting typos (e.g., PIN).

III. CONCLUSIONS

We got to the conclusion that the honey algorithm has proved its efficacy against the attack that relies on brute force after performing an intensive study on the honey algorithm, mentioning and examining the work of academics from a number of sectors, and coming to the conclusion that the honey algorithm has demonstrated its efficacy against the assault that relies on brute force. Furthermore, when it is paired with hash functions in the process of constructing DTE and salting, it becomes even more strong and is able to ward against other sorts of assaults, such as the rainbow attack and the dictionary attack. This is because hash functions are used in both processes of building DTE and salting. And other attacks, taking into consideration the challenges that are faced, the most important of which is the creation of honey words, typos, and the quality of DTE construction, and the honey algorithm can be used or combined with other algorithms to be stronger and more effective and to overcome some of the gaps that the honey algorithm suffers from, such as the chosen-ciphertext attack (CCA). And other attacks, taking into consideration the challenges that are faced, the most important of which is the creation of honey words. And other assaults, bearing in mind the difficulties that must be overcome, the most significant of which is the production of honey words. Finding several solutions to the problem of typographical mistakes made by users and going into detail on the most significant benefits and downsides of each strategy, as well as the qualities of the honey algorithm that are most remarkable for their contribution to the following, are both things that need to be done in order to address the issue of typographical errors made by users:

- Verify that the password you used for authentication is accurate.
- Check to see that the data's integrity is preserved by the use of a valid password.
- It is compatible with the PIN, the RSA, and the PW.
- Needs to:
    - The use of bees is not the most efficient approach for honey production.
    - What is the most effective strategy for developing an optimisation of DTE?
    - What is the mathematical equation that may be used to calculate the possibility that a message will be received?

REFERENCES

[1]. Noorunnisa, N. S., & Afreen, D. K. R. (2016). Review on Honey Encryption Technique. International Journal of Science and Research (IJSR) ISSN (Online), 2319-7064.

[2]. Juels, A., & Rivest, R. L. (2013, November). Honeywords: Making password-cracking detectable. In Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security (pp. 145-160).

[3]. Juels, A. (2014, June). A bodyguard of lies: the use of honey objects in information security. In Proceedings of the 19th ACM symposium on Access control models and technologies (pp. 1-4).

[4]. Lindholm, R. (2019). Honey Encryption: implementation challenges and solutions.

[5]. ElGamal, T. (1985). A public key cryptosystem and a signature scheme based on discrete logarithms. IEEE transactions on information theory, 31(4), 469-472.

[6]. Meier, A. V. (2005, June). The elgamal cryptosystem. In Joint Advanced Students Seminar.

[7]. Oppliger, R. (2011). Contemporary cryptography. Artech House.

[8]. Taneski, V., Heričko, M., & Brumen, B. (2014). Password security—No change in 35 years?In 2014 37th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO) (pp. 1360–1365). IEEE.

[9]. Abdalla, N. A. A. (2019). Preservation of Data Confidentiality Using Honey Encryption (Doctoral dissertation, Sudan University of Science and Technology).

[10]. Juels, A., & Ristenpart, T. (2014, May). Honey encryption: Security beyond the brute-force bound.

[11]. In Annual international conference on the theory and applications of cryptographic techniques (pp. 293-310). Springer, Berlin, Heidelberg.
[12]. SAHU, S. (2020) Providing Information Security Using Honey Encryption.
[13]. Omolara, A. E., Jantan, A., & Abiodun, O. I. (2019). A comprehensive review of honey encryption scheme. Indonesian Journal of Electrical Engineering and Computer Science, 13(2), 649-656.
[14]. Bojinov, H., Bursztein, E., Boyen, X., & Boneh, D. (2010, September). Kamouflage: Loss-resistant password management. In European symposium on research in computer security (pp. 286-302). Springer, Berlin, Heidelberg.
[15]. Yajun, G. U. O., & Dongqi, P. U. (2019). Privacy Data Protection Based on the Honey Encryption. Netinfo Security, 19(12), 38.
[16]. Latha.K, Sheela.T (September 2019). Reducing Cloud Data Breaches and Improving Data Security using Honey Encryption Algorithm. International Journal of Innovative Technology and Exploring Engineering (IJITEE) ISSN: 2278-3075.
[17]. Rani, D. N. U., Ahmad, S. N., Reddy, C., & Lakshmi, P. J. (2018, February). Honey Maze Encryption (Home).
[18]. In 2018 IADS International Conference on Computing, Communications & Data Engineering (CCODE).
[19]. Noorunnisa, N. S., & Afreen, D. K. R. (2019). Honey Encryption based Password Manager. Journal of Emerging Technologies and Innovative Research (JETIR) ISSN: 2349-5162.
[20]. Srinivasu, N., Sahil, M., Francis, J., & Pravallika, S. (2017). Security enhanced using honey encryption for private data sharing in cloud. International Journal of Engineering & Technology, 7(1.1), 675-678.
[21]. Fun, T. S., Samsudin, A., & Zaaba, Z. F. (2017).
[22]. Enhanced security for public cloud storage with honey encryption. Advanced Science Letters, 23(5), 4232-4235.
[23]. Huang, Z., Ayday, E., Fellay, J., Hubaux, J. P., & Juels, A. (2015, May). GenoGuard: Protecting genomic data against brute-force attacks. In 2015 IEEE Symposium on Security and Privacy (pp. 447-462). IEEE.
[24]. Arun, S., & Shanker, N. R. (2006). Data Security in Cloud Storage Using Advanced Encryption Standard and Honey Cryptography.
[25]. Srilatha Komakula, V.Shobha Rani (2020).Honey Encryption With Quantum Key Distribution. International Journal for Innovative Engineering and Management Research- IJIEMR (Vol 09 Issue01, Jan 2020.
[26]. Mok, E., Samsudin, A., & Tan, S. F. (2017).
[27]. Implementing the honey encryption for securing public cloud data storage. In First EAI International Conference on Computer Science and Engineering (Vol. 10).
[28]. Tan, S. F., & Samsudin, A. (2018). Enhanced security of internet banking authentication with extended honey encryption (XHE) scheme. In Innovative Computing, Optimization and Its Applications (pp. 201-216). Springer, Cham.
[29]. Tyagi, N., Wang, J., Wen, K., & Zuo, D. (2015). Honey encryption applications. Network Security, 2015, 1-16.
[30]. Rajalakshmi, M., & Parthasarathy, C. (2006). An Implementation of Fhma for Honey Encrypted Datasets in Wireless Sensor Networks.
[31]. A. Juels and T. Ristenpart, "Honey Encryption: Encryption beyond the Brute-Force Barrier," in IEEE Security & Privacy, vol. 12, no. 4, pp. 59-62, July-Aug. 2014.
[32]. Jaeger, J., Ristenpart, T., & Tang, Q. (2016, May). Honey encryption beyond message recovery security. In Annual International Conference on the Theory and Applications of Cryptographic Techniques (pp. 758-788). Springer, Berlin, Heidelberg.
[33]. Menezes, A.J., van Oorschot, P.C., & Vanstone, S.A. (1997). Handbook of Applied Cryptography (1st ed.). CRC Press.
[34]. C. R. (2009). Vulnerability Note VU# 836068 MD5 vulnerable to collision attacks.
[35]. Sotirov, A., Stevens, M., Appelbaum, J., Lenstra, A. K., Molnar, D., Osvik, D. A., & de Weger, B. (2008). MD5 considered harmful today, creating a rogue CA certificate. In 25th Annual Chaos Communication Congress (No. CONF).
[36]. Stevens, M., Bursztein, E., Karpman, P., Albertini, A., & Markov, Y. (2017, August). The first collision for full SHA-1. In Annual International Cryptology Conference (pp. 570-596). Springer, Cham.
[37]. Devi, R., & Sujatha, P. (2020). A Hybrid Watermarking System for Securing Multi-modal Biometric Using Honey Encryption and Grasshopper Optimization Technique. In Intelligent Computing and Innovation on Data Science (pp. 725-734). Springer, Singapore.
[38]. Piyush, "Advanced Honey Encryption: An Escape-less Trap for Intruders," 2018 4th International Conference on Computing Communication and Automation (ICCCA), 2018, pp. 1-4.
[39]. Rajalakshmi, M., & Parthasarathy, C. (2006). An Implementation of Fhma for Honey Encrypted Datasets in Wireless Sensor Networks.
[40]. Kapil, G., Agrawal, A., Attaallah, A., Algarni, A., Kumar, R., & Khan, R. A. (2020). Attribute based honey encryption algorithm for securing big data: Hadoop distributed file system perspective. PeerJ Computer Science, 6, e259.
[41]. Ravindranadh, K., Kiran, M. S., Kumar, B. D. S. P., & Priyanka, D. (2018). Data migration in cloud computing using honey encryption. International Journal of Engineering & Technology, 7(2.8), 230-234.

[42]. Panchal, K. (2020). Differential Privacy and Natural Language Processing to Generate Contextually Similar Decoy Messages in Honey Encryption Scheme.
[43]. KURNAZ, S., & Mohammed, A. H. (2020, June). Secure Pin Authentication in Java Smart Card Using Honey Encryption. In 2020 International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA) (pp. 1-4). IEEE.
[44]. Abiodun, E. O., Jantan, A., Abiodun, O. I., & Arshad, H. (2020). Reinforcing the Security of Instant Messaging Systems Using an Enhanced Honey Encryption Scheme: The Case of WhatsApp. Wireless Personal Communications, 112(4), 2533-2556.
[45]. Huang, Z., Ayday, E., Fellay, J., Hubaux, J. P., & Juels, A. (2015, May). GenoGuard: Protecting genomic data against brute-force attacks. In 2015 IEEE Symposium on Security and Privacy (pp. 447-462). IEEE.
[46]. Yin, W., Indulska, J., & Zhou, H. (2017). Protecting private data by honey encryption. Security and Communication Networks, 2017.
[47]. Yin, W., Indulska, J., & Zhou, H. (2017). Protecting private data by honey encryption. Security and Communication Networks, 2017.
[48]. Win, T., & Moe, K. S. M. Protecting Private Data using Improved Honey Encryption and Honeywords Generation Algorithm.
[49]. Sawant, S., Saptal, P., Lokhande, K., Gadhave, K., & Kaur, R. (2018). Honeywords: Making Password Cracking Detectable. International Journal of Engineering Research and Advanced Technology-IJERAT, 4(4), 01-06.
[50]. Kelley, P. G., Komanduri, S., Mazurek, M. L., Shay, R., Vidas, T., Bauer, L., ... & Lopez, J. (2012, May). Guess again (and again and again): Measuring password strength by simulating password-cracking algorithms. In 2012 IEEE symposium on security and privacy (pp. 523-537). IEEE.
[51]. A. Czeskis, M. Dietz, T. Kohno, D. Wallach, and D. Balfanz. Strengthening user authentication through opportunistic cryptographic identity assertions. In ACM CCS, pages 404–414, 2012.
[52]. O. Kharif. Innovator: Ramesh Kesanupalli's biometric passwords stored on devices. Bloomberg Businessweek, 28 March 2013.
[53]. T. Wadhwa. Why your next phone will include fingerprint, facial, and voice recognition. Forbes, 29 March 2013.
[54]. Choi, H., Jeong, J., Woo, S. S., Kang, K., & Hur, J. (2019). Password typographical error resilience in honey encryption. Computers & Security, 87, 101411.
[55]. Chatterjee, R., Bonneau, J., Juels, A., & Ristenpart, T. (2015, May). Cracking-resistant password vaults using natural language encoders. In 2015 IEEE Symposium on Security and Privacy (pp. 481-498). IEEE.
[56]. Burgess, J. (2017). Honey Encryption Review. Queen's University Belfast.
[57]. Chatterjee R, Athayle A, Akhawe D, Juels A, Ristenpart T. Password typos and how to correct them securely. In Security and Privacy (SP), 2016 IEEE Symposium on 2016 May 22 (pp. 799-818). IEEE.
[58]. Choi H, Nam H, Hur J. Password typos resilience in honey encryption. In Information Networking (ICOIN), 2017 International Conference on 2017 Jan 11 (pp. 593-598). IEEE.
[59]. Chatterjee R, Woodage J, Pnueli Y, Chowdhury A, Ristenpart T. The TypTop System: Personalized Typo-tolerant Password Checking. In Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security 2017: 329-346
[60]. Huang, Z., Ayday, E., Fellay, J., Hubaux, J. P., & Juels, A. (2015, May). GenoGuard: Protecting genomic data against brute-force attacks. In 2015 IEEE Symposium on Security and Privacy (pp. 447-462). IEEE.
[61]. Choi H, Nam H, Hur J. Password typos resilience in honey encryption. In Information Networking (ICOIN), 2017 International Conference on 2017 Jan 11 (pp. 593-598). IEEE.
[62]. Al-Qwider, W. H., & Salameh, J. N. B. (2017). Novel technique for securing data communication systems by using cryptography and steganography Jordanian Journal of Computers and Information Technology (JJCIT), 3(2), 110-130.