A Survey on Encryption and Deception based Approaches for Data Security on Cloud

R. Ramachandran[1], N. Arunachalam[2], K. Deebiga[3], R. Surekha[4]

[1,2]*Associate Professor, Department of Information Technology, Sri Manakula Vinayagar EngineeringCollege, Puducherry, India*

[3,4]*Student, Department of Information Technology, Sri Manakula Vinayagar EngineeringCollege, Puducherry, India*

ramachandiran@smvec.ac.in, [2]narunachalam85@gmail.com , [3]deepsdeebi07@gmail.com , [4]surekharajendiran@gmail.com

***Abstract*— Cloud computing has been conceptualized as the upcoming generation paradigm in computation. With growing demand for cloud storage, computing power and infrastructure, cloud computing has turn into more extensive in today's age. With the advantages of Cloud computing, the consumers can have easy association to storage, infrastructure, and cloud facilities over the internet. Cloud Computing Security, especially Data Security is the predominant obstacle for authorizing Cloud Computing services. A cloud storage system acquires an extensive number of data in its storage server. Since the data is accumulated for a long-term over the internet it does not support the data confidentiality and make the hackers to divert the data furnished in the storage system and even while the data is delivered to cloud environment, it lacks data integrity and cause the cloud user unfulfilled. The extended attack surface and the lack of persuasive security, privacy, and protection measures are still one of the obstruction of widely redistributing applications on the cloud infrastructure. This survey paper illustrates about different encryption and deception techniques to protect the cloud storage environment from intruders. It concisely covers few of the existing approaches that are used to improve the security in the cloud environment.**

Keywords — Data Security, Encryption Technique, Deception Technique, Cloud Security

I. INTRODUCTION

In the cloud computing environment, both resources and applications are conveyed on demand over the Internet as services. Cloud is an environment of the hardware and software resources in the data centers that afford varied services over the network or the Internet to fascinate user requirements [1]. Recent security events on public cloud data storage has risen concerns on cloud data security. Cloud computing facilitates users to maintain their information in the cloud so as to utilize scalable on-demand services. Primarily for small and moderate-sized organization with limited budgets, enabling to achieve high cost savings and enhancements of productivity by utilizing cloud-based services for managing projects, to accomplish collaborations among each other [2]. Cloud Service Providers (CSPs), use to concentrate on sensitive data, establish probable security and privacy bottlenecks which are not available in the similar trusted domains of enterprise users.

In-order to protect the user sensitive data confidentiality from unauthorized CSPs, a naive way is to use cryptographic methodologies, by revealing decryption keys making available to accredit users. However, sensitive data is outsourced for distribution on cloud servers by organization users; the encryption system enforced will also afford increased performance, delegation, and scalability, in addition to supporting fine-grained access control. Thus, this is proficient so as to best distribute the requirements for accessing data anytime irrespective of location, negotiating within the organization.

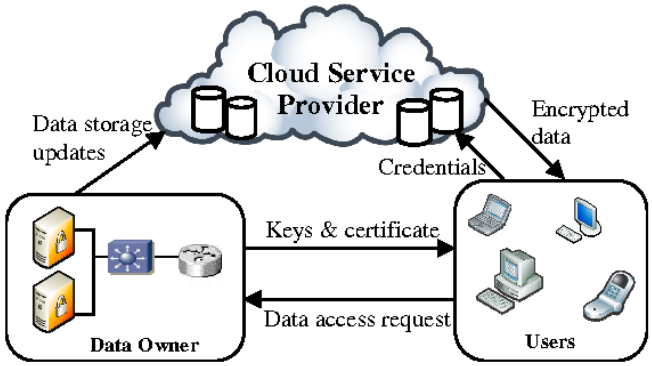

Fig 1.1 Cloud Service Provider

The fundamental solution to contract with this troublesome circumstance is to utilize the cryptographic methods in cloud environment [3]. Cryptographic Techniques is widely classified into two divisions: Asymmetric key Cryptography and Symmetric key Cryptography. The cryptographic methodologies is enforced to assure the data used in the cloud and prevent information from being leaked and to ensure that the privacy has been maintained. The cryptographic techniques were used to establish security for the data hoarded in the cloud.

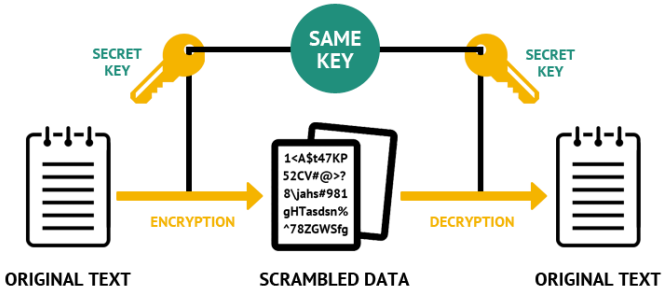

Fig1.1Symmetric Cryptography

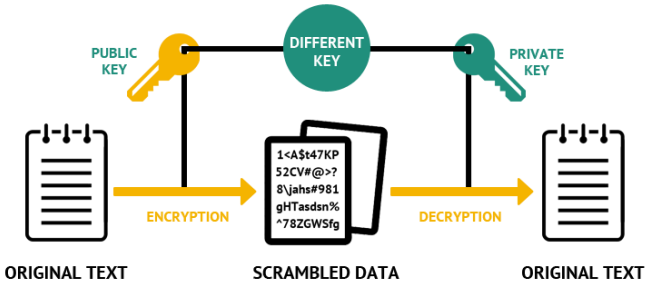

Fig 1.2 Asymmetric Cryptography

The immense increase in data breach over time signifies that the prevailing measures for acquiring data may be inadequate and requires extensive analysis. The advent of high speed and supercharged parallel and distributed systems paved the way for analyzing, gathering and processing huge chunks of data generally referred to as big data. However, this enormous advancement lay cryptosystem at a disadvantages as attackers influence on the great computational power, to accomplish brute-force attack [4]. Even though traditional encryption patterns continue to pledge security by increasing the size of the key of computational inaccessibility of searching for the key, cryptosystems fail to withstand cryptanalysis attack specifically the brute-force attack.

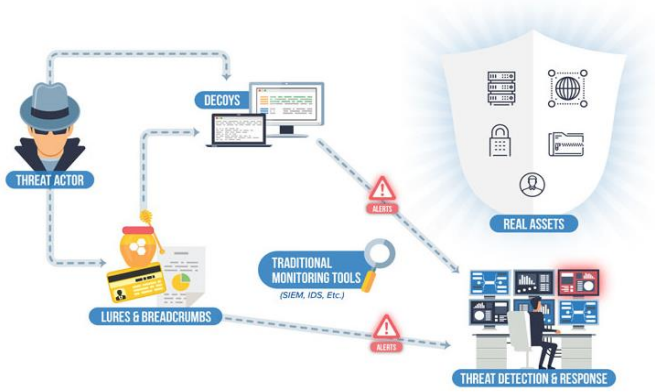

Fig 1.3 Deception Technique

A deception-based approach is required to address the faults of password-based encryption schemes and is employed in password based system for securing the data [5]. The deception technique is pivotal in the event of confronting an adversary. It permits an environment where an adversary is cornered into taking actions that consume his resources. Employing deception and decoy techniques in network systems assist to detect, trace, monitor and deter the activities of an adversary. It is organized to make the adversary's life ambitious where a false reality is projected as reality to him.

II. Related Works

In the past decades, the varsity number of approaches are illustrated to secure the data from intruders, of which a few of the encryption and deception based techniques are examined to determine best solution for security.

A. Encryption-based Techniques:

An extensive amount of research has been published on utilizing encryption-based methods for securing the confidential data on cloud. Most of the frameworks with encryption use password-based encryption (PBE) [6]. These systems are susceptible to brute force guessing attacks. Password-based encryption is a well-known method of creating powerful cryptographic keys. PBE cryptography is based on the hashing mechanism. A password and salt will be combined so that it will generate random data through the application function process and will be processed by the iteration count so that when the mixing process has completed it will offend the data in the assemble of cipher text. It allows the user to discover powerful secret key based on passwords provided by the users. The produced key bytes are assumed to be as random and unreliable as possible. PBE methods are susceptible in brute- force attacks because of user generated weak or repeated passwords. It is not a secured storage of data and is fragile in preventing the attacks**.**

Data Encryption Standard encryption and decryption technique using secret key has been examined [7]. DES is the block cipher algorithms used as standard symmetric encryption algorithm. It will generate 8 blocks of ciphers linked into one cipher text, but the cipher text are fragile against brute force attacks. Multiple Diffusion and Confusion rounds are used to boost the level of difficulty of performing a cryptanalysis on the cipher text. It has high security level related to a small key used for encryption and decryption. Triple DES is best encryption and decryption algorithm than DES that is used to encrypt the file and documents uploaded by the user [8]. Triple DES was discovered back when DES was obtaining a bit weaker than user were comfortable with. As an outcome, they needed an easy way to get more strength. If a system reliant on DES, making a composite function out of numerous DES is likely to be accessible than scampering in a new cipher and sidesteps the political issue of disputing that the new cipher is better than DES. Triple DES is much more secure than single DES and any other cryptographic algorithms. It is used for securing the information and it is often used in the cloud security. It is applied three times than a normal DES algorithm.

A public key cryptosystem, Rivest Shamir Adleman (RSA) algorithm has been examined [9] to furnish security for data stored in cloud. The development of the Public key cryptography is greatest and possibly it provides a radical departure. The algorithm consists of Public Key and Private Key. Public Key is familiar to all cloud users and Private-Key is familiar only to the user who formerly possess the data. Encryption is accomplished by the Cloud service provider and decryption is transported out by the Cloud user. Once the data is encrypted with the Public Key, it can be decrypted with the corresponding Private Key. It is also known as the Asymmetric algorithm due to the use of two key along with secret key. The framework structure of RSA algorithm depends on the number of inferences. It is the most security framework in the key frameworks. An outsider can't break the private key in light of factorization of bigger numbers. Thus, RSA encryption algorithm is an attainable solution for secure communication in cloud computing.

Homomorphic Encryption to protect data on cloud has been analyzed [10]. It allows user to operate encrypted data directly without decryption. It can be either fully or partial homomorphic encryption. A fully homomorphic encryption (FHE) supports arbitrary number of both operations addition as well as multiplication on ciphertext. Partial Homomorphic encryption scheme such as RSA cryptosystem and Paillier cryptosystem are insufficient for cloud computing. Partial HE is limited to number of operations. Partial HE can be either additive HE or multiplicative HE. RSA cryptosystem is vulnerable to common modules attack. Traditional encryption techniques are not sufficient to secure data in processing state. It appears that homomorphic encryption methods detract the problem of data security in cloud computing. But currently both

fully as well as partial homomorphic methods are not feasible and not so easy to implement for cloud computing. The fully homomorphic method took long time to perform on encrypted data and partial homomorphic methods are also not efficient as it is able to perform either addition or multiplication. The homomorphic encryption based solutions for data security are not practical in real world.

Blowfish algorithm, a symmetric key approach has been examined [11]. It uses a key of variable length of 32-448 bits to encrypt data clocks of 64 bits. It achieves the first-rate performance amongst all selected algorithms in terms of changing packet size. No attack has yet been successful against this algorithm. In terms of converting information kind such as text-to-image conversion, Blowfish is time consuming. Blowfish is not suitable for applications where the key changes often, like packet switching. It is appropriate where the key does not change repeatedly, like Communications link encryption. The throughput for encryption and decryption of Blowfish is better in encryption and decryption process than the other algorithms. BE scheme offers security as well as fast encryption & decryption in which Blowfish algorithm is used to encrypt/decrypt the data because this symmetric key algorithm is so fast, so BE system uses symmetric key algorithm to provide security and reduce the overall process time. The blowfish algorithm can also be used for large size of data for fast encryption and decryption, so it will be helpful for improving the speed and security of the big size data.

Twofish algorithm of 128 bit block encryption analyzed extensively [12]. It is one of the efficient algorithm because of the performance on both Hardware and Software platforms with strong keys. Encryption algorithm would not be of much use if it is very much secure but slow in performance. The security and performance of encryption algorithms must be balanced. Twofish users block ciphering with single key of any length up to 256bits.It allows to implement to trade of encryption and speed key and setup time and code size to balance performance from the key dependent S-boxes it can withstand unknown attacks come next and resist known attacks. The performance of the encryption algorithm is also balanced. The data loss will be reduced and increased security using Twofish algorithm's private key along with signature. It can also be extended to the future work in encryption and decryption of large size of text files, images, audio files and video files.

Advanced Encryption Standard (AES) algorithm has been analyzed [13] which is regarded as the most popular symmetric cryptographic algorithm. It is very significant to develop high performance. AES algorithm has a high level of security because 128, 192 or 256-bit keys are used in this algorithm. It exhibits resistance against a different types of attacks such as square attack, key attack, key recovery attack and differential attack. The encryption time of AES is less than DES. So, it means that AES performance is much better than the DES. Till date there is no proof to crack this algorithm. It is a highly secure encryption algorithm. Data can also be protected against future attacks such as smash attacks. AES encryption algorithm has minimal storage space and high performance without any weakness and limitation while other symmetric algorithms have weaknesses and differences in performance and storage space. As AES encryption technique is used for data transfer, it excludes the likelihood of the system to be unavailable at times during the arrival of enormous data. Since denial of access to the third party is done, possibility of intruders to mask as the third party and intrude into the network is avoided. The implementation of Advanced Encryption Standard for securing data bestows benefits of less computation time and less memory consumption in divergence to other algorithms.

Diffie hellman algorithm for information liability and security in cloud is incrediblily analysed [14]. It is a secure algorithm that offers high performance, allowing two computers to publicly exchange a shared value without using data encryption. This exchanged information is preserved with a hash function. Diffie hellman protocols for authenticated key exchange (AKE) are outlined to contribute a pool of players with shared secret key which is used later to achieve multicast message integrity. This technique is used in a public network for exchanging cryptographic keys securely. A secure channel is provided with the help of Diffie Hellman protocol such that no one except the data owner can access the data without his permission. Cloud computing has great and immense scope. Triple data scheme of encryption can be considered for maintaining security and liability of the data. Homomorphic encryption seems to be very effective but needs further study and consideration. It basically tries to eliminate the insecurity faced by the data owner for his data is on cloud and under the control of the cloud provider.

The password guessing attack susceptibility using Strong Diffie Hellman-DSA Key Exchange algorithm has been proposed [15]. Diffie-Hellman is a famous algorithm that enables two users to establish a secret key, securely and without any need to exchange the secret key. But this protocol is insecure against the man-in-the-middle attack. This problem is solved by providing authentication to the Diffie-Hellman key exchange using the Digital Signature Algorithm (DSA). The algorithm works by first of all making choice of encryption parameters which will be shared across several users of the system and then in the second stage private and public key is created for single user. The signature is discovered with the private key by the sender and then it's verified and authenticated by the receiver using the public key. Digital Signature Algorithm (DSA) has a key size of 3072 bits. Original DSA algorithm has its own security limitations, that is, only one key is used. So in order to improve its security, more than one key is used due to which the difficulty of deciphering key increases. The use of biometric data affords a non-repudiation mechanism that solves the limitation of password-based authentication, such as the tendency of users to choose a simple password. The security of this system can protect from several attacks in cloud system.

ElGamal Encryption has been analyzed [16] which is an asymmetric key encryption algorithm that relies on the Diffie Hellman key algorithm. ElGamal cryptosystem affords better user authentication and performance with respect to the security accomplishment against attacks. ElGamal encryption has three components: the key generator, the encryption algorithm, and the decryption algorithm. The average encryption time required by RSA is 2655 milliseconds and that of ElGamal is 60235 milliseconds. The average decryption time required by RSA is 72671 milliseconds and that of ElGamal is 37724 milliseconds. Thus it is observed that RSA requires less time for encryption as compared to ElGamal, but requires more time for decryption.

It proves to be faster in decryption as compared to RSA even for larger file size. It improvises the randomization for key generation, encryption, and decryption from the ElGamal cryptosystem. Consequently, for the proposed algorithm, key generation is a time-consuming one, since it will be done periodically, it is tolerable. It provides data confidentiality and ElGamal security relies on the difficulty of randomness and the discrete logarithm problem. This algorithm gives an additional layer of security by asymmetrically encrypting keys which were properly used for symmetric text encryption.

B. Deception-based Techniques:

A hashing method using salting and differential masking that can provide the better security of the password with the faster time. Although this method is better method for the passwords with better processing time, it can't strongly protect the brute force attack. Above the description, hashing algorithm that uses for increasing computation time and producing false plaintext messages are not fully protect the various attacks especially brute force attack. So, the deception-based strategies have been enforced as countermeasures to delay, detect and confuse an adversary attempting to divert data on a network. A Honey encryption (HE) is introduced [17] which is a general approach to encrypting messages using low min-entropy keys such as passwords. HE is outlined to generate a cipher text which, when decrypted with any of a number of incorrect keys, provides plausible-looking but fake plaintexts called honey messages. A key advantage of HE is that it affords security in cases where limited entropy is accessible to withstand brute-force attacks that try every key. It contributes extensive security beyond conventional brute-force bounds. HE can also provide a hedge against partial disclosure of high min-entropy keys. It provides more security than existing PBE scheme, however it cannot effectively protect normal HTTPS certificate keys.

In order to improve the safety of the hashed passwords, a new method called honeywords generation method has been suggested [18]. They maintain the honeywords with the real password for each user's account by creating honeywords. The attacker cannot classify which password is real password if he gets inversion file of hashed password or honeywords. The auxiliary server or honey checker can classify the real password and honeyword for login process and will set off an alarm to categorize between the honeywords and real password. However, this method can cause typing mistake of users during entering of password because honeywords generation method create the similar honeyword with the real password. The main weakness of this method is storage overhead problem. The use of honeywords may be very beneficial in the current environment, and is simple to implement. The fact that it works for every user account is its big advantage over the related technique of honeypot accounts. Honeywords are a simple-to-deploy and powerful new line of defense for existing password systems. We hope that the security community will benefit from their use.

Honey encryption is also associated to Format-Preserving Encryption (FPE) and Format Transforming Encryption (FTE) [19]. In cryptography, format-preserving encryption (FPE), indicate to encrypting in such a way that the output, ciphertext is in the same format as the input i.e. plaintext. Honeyword based techniques are getting popular as it provides various benefits over traditional password based schemes. Although honey encryption has been enforced to multiple number of applications, due to the variety of message formats and probability characteristics, the message space outline needs to vary for new types of applications. In FPE, the plain text message space is the same as the cipher text message space. In FTE, the cipher text message space is peculiar from the message space. Honey encryption maps a plaintext message to a seed range in the seed space. Since the message space and the seed space are different, the cipher text message space is different from the message space. It is appropriate for a small, not large, message space as the overhead of processing a large message space is very high. The capability of protecting sensitive private data provided by HE varies for different applications.

The honey encryption concept has been applied to MANETs [20] to forbid ad hoc networks from the brute-force attack, the honey encryption technique is adopted to conserve credit card numbers and a simplified version of text messaging. Most of these data in are from consistently distributed message spaces. However, genomic data ordinarily has profoundly non-uniform probability distributions. The GenoGuard mechanism consolidates a new theoretical framework for encryption called honey encryption (HE). It can furnish information-theoretic confidentiality guarantees for encrypted data. GenoGuard addresses the open problem of handling HE techniques to the extensively non-uniform probability distributions that characterize sequences of genetic data. In GenoGuard, a probable adversary can endeavor exhaustively to guess keys or passwords and decrypt via a brute-force attack. The decryption under any key will prove to earn a plausible genome sequence, and that GenoGuard affords an information-theoretic security guarantee against message-recovery attacks.

Due to the expansion of graphical processing unit (GPU), the adversary can solve the hashing password files [21]. So, the new method by keeping the decoy password and real password into the database is analyzed. In this work, the honeychecker for testing entering the false passwords to the system was described. The use of a honeychecker thus imposes an adversary to either risk logging in with a high chance of causing the detection of the compromise of the password-hash file. Although getting the password file and converting the hash code into the password, he can't enter the system without passing the honeychecker. Therefore, this method can against the attacks of attackers, it meets the storage overhead problem. For making less the existing problem of honeywords generation method, the honey circular list algorithm was examined. So, the circular list for storage of hashing password was created. When the attacker gets the password file, he cannot estimate the distance of this password and he cannot login into the system. The honey circular list method can make less the storage problem of earlier existing algorithms. Honeywords are temporarily advocate to turn into acceptable passwords. This forbids denial-of-service attacks resulting from attack on the honeychecker or the communications between the system and the honeychecker. However, this method cannot almost solve the problem of honeywords producing methods.

A new encoding and decoding scheme called a distribution transforming encoder (DTE) are used by Honey Encryption [22]

and it has the limitation for assigning the plaintext messages. The key challenges are improvement of applicable precedent of a new type of randomized message encoding strategy called a distribution - transforming encoder (DTE), and analyses of the expected maximum loading of bins in various kinds of balls-and-bins games. The DTE in that system has limitation for placing plaintext message into the seed space. Distribution transforming encoder (DTE) process is the essential key of honey encryption algorithm that assigns the message space M to the binary bits string of seed space SDES involves 16 rounds of identical operations. HE automatically advances security in a several number of practical settings. When the attacker tries to get decrypted data by using brute force attack, that technique can deceive the attacker and can give bogus meaningful messages. DTE process uses cumulative distribution function to map the message space into the seed space and it meets the message space limitation problem. DES can be broken easily as it has known vulnerabilities and moreover, the security is not guaranteed.

Inverting the hash values by performing brute force computation is one of the latest security threats on password based authentication technique. Honeyword based authentication protocol can successfully mitigate this threat by making password cracking detectable. However, the existing schemes have several limitations like Multiple System Vulnerability, Weak DoS Resistivity, Storage Overhead, etc. A new honeyword generation approach was proposed [23] , identified as Paired Distance Protocol (PDP) which overcomes almost all the drawbacks of previously proposed honeyword generation approaches. The comprehensive analysis shows that PDP not only attains a high detection rate of 97.23% but also reduces the storage cost to a great extent. Depending upon the randomness of RS, PDP provides the highest level of security standard. It has limitations such as co-relational hazard, DoS resistivity and issue regarding Typo safety. The performance of honey encryption mechanism was evaluated and implementation must be customized for different applications as the message spaces vary.

Recent security incidents on public cloud data storage had ascend concerns on cloud data security. Existing cloud data protection solutions that primarily relying on the conventional password-based encryption cannot efficiently resist password guessing and password cracking attacks. To address this issue, an eXtended Honey Encryption (XHE) strategy was proposed [24] by adding an additional protection mechanism on the encrypted data. When the attacker attempts to access these encrypted data by entering the incorrect password, instead of rejecting the access, the HE algorithm generates an indistinguishable bogus data, in which the attack could not determine whether the guessed password is working correctly or not. Therefore, increasing the complexity of password guessing and cracking attacks. The message space of the proposed scheme is pre-fixed and the complexity and the size of inverse sampling tables is growing exponentially with the increase of the file names and its extension sizes. XHE scheme can be further adapted to work with the recent advancement of cryptography algorithm such as Homomorphic Encryption for supporting the computation on encrypted data, as well as Attribute-Based Encryption (ABE). It is susceptible to password guessing attack such as brute-force, dictionary or rainbow table attack.

III. Conclusion

Cloud Computing is a cost-effective, flexible and justified distribution platform for furnishing business or consumer IT services over the Internet. Users are very suspicious about whether their data is highly secure and private. Over the most recent two decades, different bunching techniques have been proposed and in any case, each one of them have a few advantages and disadvantages as for securing the private data on cloud. The objective of this paper explores on existing methods to provide highest level of security for data stored on cloud. Various cryptographic techniques based on encryption and deception have been propounded and analyzed to protect the data against intruders. The deception based approach seems to have higher chances of success compared to traditional cryptographic approaches. However, the certain state-of-art approaches fails to achieve the throughput, Performance measure and Accuracy.

References

[1] Albugmi, A., Alassafi, M. O., Walters, R., & Wills, G. (2016). Data security in cloud computing. 2016 Fifth International Conference on Future Communication Technologies (FGCT). doi:10.1109/fgct.2016.7605062

[2] Shuijing, H. (2014). Data Security: The Challenges of Cloud Computing. 2014 Sixth International Conference on Measuring Technology and Mechatronics Automation. doi:10.1109/icmtma.2014.52

[3] Bhardwaj, A., & Som, S. (2016). Study of different cryptographic technique and challenges in future. 2016 International Conference on Innovation and Challenges in Cyber Security (ICICCS-INBUSH). doi:10.1109/iciccs.2016.7542353

[4] Izhar, S., Kaushal, A., Fatima, R., & Qadeer, M. A. (2017). Enhancement in data security using cryptography and compression. 2017 7th International Conference on Communication Systems and Network Technologies (CSNT). doi:10.1109/csnt.2017.8418539

[5] Albanese, M., Battista, E., & Jajodia, S. (2015). A deception based approach for defeating OS and service fingerprinting. 2015 IEEE Conference on Communications and Network Security (CNS). doi:10.1109/cns.2015.7346842

[6] Shuo Zhai, & Tao He. (2010). Design and implementation of password-based identity authentication system. 2010 International Conference on Computer Application and System Modeling (ICCASM 2010). doi:10.1109/iccasm.2010.5623039

[7] Maitri, P. V., & Verma, A. (2016). Secure file storage in cloud computing using hybrid cryptography algorithm. 2016 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET). doi:10.1109/wispnet.2016.7566416

[8] Semwal, P., & Sharma, M. K. (2017). Comparative study of different cryptographic algorithms for data security in cloud computing. 2017 3rd International Conference on Advances in Computing,Communication & Automation (ICACCA) (Fall). doi:10.1109/icaccaf.2017.8344738

[9] Mahalle, V. S., & Shahade, A. K. (2014). Enhancing the data security in Cloud by implementing hybrid (Rsa & Aes) encryption algorithm.

2014 International Conference on Power, Automation and Communication (INPAC). doi:10.1109/inpac.2014.6981152

[10] Chauhan, K. K., Sanger, A. K. S., & Verma, A. (2015). Homomorphic Encryption for Data Security in Cloud Computing. 2015 International Conference on Information Technology (ICIT). doi:10.1109/icit.2015.39

[11] Bansal, V. P., & Singh, S. (2015). A hybrid data encryption technique using RSA and Blowfish for cloud computing on FPGAs. 2015 2nd International Conference on Recent Advances in Engineering & Computational Sciences (RAECS). doi:10.1109/raecs.2015.7453367

[12] Rane, D. D., & Ghorpade, V. R. (2015). Multi-user multi-keyword privacy preserving ranked based search over encrypted cloud data. 2015 International Conference on Pervasive Computing (ICPC). doi:10.1109/pervasive.2015.7087044

[13] Shimbre, N., & Deshpande, P. (2015). Enhancing Distributed Data Storage Security for Cloud Computing Using TPA and AES Algorithm. 2015 International Conference on Computing Communication Control and Automation. doi:10.1109/iccubea.2015.16

[14] Rewagad, P., & Pawar, Y. (2013). Use of Digital Signature with Diffie Hellman Key Exchange and AES Encryption Algorithm to Enhance Data Security in Cloud Computing. 2013 International Conference on Communication Systems and Network Technologies. doi:10.1109/csnt.2013.97

[15] Talkhaby, H. R., & Parsamehr, R. (2016). Cloud computing authentication using biometric-Kerberos scheme based on strong Diffi-Hellman-DSA key exchange. 2016 International Conference on Control, Instrumentation, Communication and Computational Technologies (ICCICCT). doi:10.1109/iccicct.2016.7987926

[16] Ara, A., Al-Rodhaan, M., Tian, Y., & Al-Dhelaan, A. (2017). A Secure Privacy-Preserving Data Aggregation Scheme Based on Bilinear ElGamal Cryptosystem for Remote Health Monitoring Systems. IEEE Access, 5, 12601–12617. doi:10.1109/access.2017.2716439

[17] S. R. Shinge and R. Patil, “An Encryption Algorithm based on ASCII Value of Data” International Journal of Computer Science and Information Technologies, vol. 5(6), 2014, pp-7232-7234.

[18] Juels, A., & Ristenpart, T. (2014). Honey Encryption: Encryption beyond the Brute-Force Barrier. IEEE Security & Privacy, 12(4), 59–62. doi:10.1109/msp.2014.67

[19] Yin, W., Indulska, J., & Zhou, H. (2017). Protecting Private Data by Honey Encryption. Security and Communication Networks, 2017, 1–9. doi:10.1155/2017/6760532

[20] Z. Huang, E. Ayday, J. Fellay, J.-P. Hubaux, and A. Juels. “Genoguard: protecting genomic data against brute-force attacks,” In 2015 IEEE Symposium on Security and Privacy (SP) , 2015, pp 447–462.

[21] Luigi Catuogno, Aniello Castiglione, Francesco Palmieri," A Honeypot System with Honeyword-driven Fake Interactive Session", IEEE 978-1-4673-7813-0 ,45,2015

[22] Ghassami, A., Cullina, D., & Kiyavash, N. (2016). Message partitioning and limited auxiliary randomness: Alternatives to Honey Encryption. 2016 IEEE International Symposium on Information Theory (ISIT).doi:10.1109/isit.2016.7541523

[23] N. Chakraborty, S. Mondal, "A new optimized honeyword generation approach for enhancing and usability",in Proc.Advances in computers and security, Springer, pp. 29s3-331 2017.

[24] K. S. M. Moe, T. Win, “Enhanced Honey Encryption Algorithm for Increasing Message Space against Brute Force Attack” in 15th Electrical Engineering/Electronics, Computer, Telecommunication and Information Technology (ECTI) Conference, Thailand, Chaing Rai, 2018.