Novel cryptographic approach to enhance cloud data security

Novel cryptographic approach to enhance cloud data security

Kanika Tyagi[1], S.K Yadav[2], Mayank Singh[3]

[1]Noida International University, Greater Noida, U.P,India
[2]Noida International University, Greater Noida, U.P,India
[3]ConsillioIntelligence Reserach Lab, Noida,U.P,India

Email: kanikat374@gmail.com

Abstract. In today's era , cloud computing has become the more promising business concept which impacted almost every section of our lives and business. Cloud computing for sure has grown rapidly to become one of the major areas of research. As it provides an on demand access to a shared pool of resources and makes easier for the organizations to use their data at any place and at anytime without considering hardware devices along them but cloud security is still the most crucial and considerable issue in each organization .So there is need for some secure authentication so that data on clouds remain safe and secure. In this context passwords are the basic form of authentication .so there is need of some mechanism which provides password security. Weak and poor passwords management leads to breach in cloud data. Passwords to gain access to secret data should be so as strong to prevent dictionary attacks and brute force attacks. In our proposed system a mechanism is presented to secure the data on cloud using combination of some algorithms viz: PBKDF2, Argon 2, AEs-256 and IDA. In this paper we proposed a method to generate the most secured cryptographic keys using the blend of two key derivation functionsPBKDF2 and Argon 2. In our proposed model there would be no need to store and send key for encryption and decryption. Advanced Encryption Standard (AES) is used for encryption for encryption. Information Dispersal algorithm is used to prevent data breaching situations on clouds in financial institutions and provide better confidentiality, availability and integrity of data.

Keywords: PBKDF2, AES, SHA, IDA

1. Introduction

In IT society, a new buzzword "going cloud" is impacting the nearly every section of business. As cloud computing allows storage of huge amount of data for future use which enables the organizations not to buy special hardware for storing their data. Organizations take on lease any cloud server from various cloud vendors and pay as per they used that cloud. Attackers try to breach the data on cloud through various methods as brute force attacks and dictionary attacks. Regardless of the fact that cloud computing is providing many features as flexibility, cost effectiveness, broad network access and

resource pooling [1], the most prevalent issue associated with it is Data security. Cloud data security is the main concern which comes as an obstacle in the adoption of cloud computing. Being a multi user and having distributed service oriented architecture, cloud computing is more prone to security threats.

While someone thinks about security, then passwords become the basic form of authentication [2]. An authentication consists of user name and the password is the most common method to authenticate user on internet but weak password management practices are leading to breach in data over the cloud. Attacks on data harming to both user and the vendors. So to achieve security password should be strong enough to prevent various attacks as dictionary attacks and brute force attacks [3].Brute force attack is a kind of attack where multiple combinations of username and passwords tried again and again until the correct one is found. [4]

Mostly a password choose by user is very short, simple and lack enough entropy [5][6][7]. That is why user chosen password cannot be used directly to secure cryptographic system. Various studies have shown the threat of choosing weak passwords by humans for security systems [8][9][10].One possible solution to this is use of key derivation functions. One possible solution to this is use of key derivation function. Key derivation function takes some initial random keying material and transfers it into one or more cryptographic keys [11].These cryptographic keys are used in cryptographic algorithms to protect the data during transmission. The key derivation function which takes user chosen password as an input known as password based key derivation function (PBKDF2).

Motivation: Poor password management leads to unauthenticated access of data that allows cyber criminals to break password security. Although there are various security methods and encryption techniques used for password security, passwords are still prone to multiple attacks. The main aim behind this proposal is to blend two or more different cryptographic approaches to increase the complexity of passwords and makes it very difficult to crack them**.**

The structure of this paper is organized as follows:

Section I: Introduction will introduce the topic and motivation of research. Its subsections A. Password Based Key Derivation Function explains the PBKDF2 and its specification along with the explanation of algorithm. B. Argon 2 states the specification and algorithm of Argon 2 and C. Information Dispersal Algorithm throws light on Information dispersal Algorithm. Section II: Related work will present the literature review of research with subsection A. Password protection using hashing algorithms will throw light on work done on different hashing algorithms, then B. Password protection using encryption algorithms will explain the comparative study of various encryption algorithms and features of AES algorithm and after that C. Previous work on hybrid algorithms which will explain the various hybrid models. Section III. Proposed Work explains the proposed model of this research. In the next Section IV. Methodology of this model is described with the help of an algorithm. Then Section V. Implementation shows the full working of this model. After that, Section VI. Evaluation discusses the result of this research with graphs and tables. The final section VI. Conclusion and Future scope summarizes the research and explains the future scope.

1.1. Password Based Key Derivation Function (PBKDF2)

PBKDF2 is a password based key derivation function, that takes user entered password as an input and generate fixed length of secured derived key. Its CPU intensive operations [12] protect the data from brute force attacks. Brute force attacks are caused by weak user's passwords. PBKDF2 uses pseudorandom functions (PRF) ,and add salt to the password which decreases threats to pre computed

hashes. As number of rounds increases, it gets harder for attacker to breach in data. Using PBKDF2 one can slow down the brute force attacks and dictionary attacks.[13]

Algorithm

The PBKDF2 takes pseudorandom functions PRF, user generated password u_p, a random salt value s, count of no. of iterations c, and required length of derived key d_k_len for cryptography key. Then it produces derived key C_d_key.

C_d_key=PBKDF2 (PRF, u_p,s,c,d_k_len)

Cryptographic derived key is computed as follows:

C_d_key=$S_1 \| S_2 \| \ldots\ldots\ldots S_{dk\_len}$

Where S_i is the block of secret key

as

S_1=Function (u_p,s,c,1);

S_2=Function (u_p,s,c,2);

.

.

.

.

S_3=Function (u_p,s,c,d_k_len);

Now each secret key block S_i is computed as follows:

S_i=Function (u_p,s,c,d_k_len);

$S_i=T_1\ T_2\ \ldots\ldots\ldots\ldots Tc$

Where

T_1=PRF(u_p,S_i);

T_2=PRF(u_p,T_1);

.

.

.

$T_c=PRF(u_p,T_{c-1})$;

This way PBKDF2 make harder for intruders for attack.

1.2. Argon2

Argon2 is the next generation of memory hard functions [14].Argon 2 is suitable for crypto currencies, password based key derivation functions and password hashing. It was declared winner of the PHC [15]. It has two flavors i.e, Argon 2d and Argon 2i.Argon2d is data dependent access of memory and it is much faster , which enables it to suitable for the applications where side channels attacks does not take place. Argon2i based on data-independent memory access which makes it enable for password hashing and password based key derivation function.[16] . As it makes more passes over the memory so it is best in protecting from trade off attacks.[17]

Inputs

Argon2 uses two types of inputs: primary inputs and secondary inputs [17].Primary inputs are always given by user as

- Message M of any length from 0 to 2^{32}-1 bytes
- Nonce N of any length from 8 to 2^{32}-1 bytes

Secondary inputs are as follows:

- Degree of parallelism p (integer value from 1-64)
- Tag length l (integer no . of bytes from 4 to 2^{32}-1)
- Memory size m(integer no. of kilobytes from 8p to 2^{32}-1)
- Number of iterations count c (any integer from 1 to 2^{32}-1)
- Secret value k (length from 0-32 bytes)
- Associated data X (length from 0 to 2^{32}-1)bytes
- Version number *v*

In Argon 2 , user entered password and salt, with other parameters are hashed .It uses Blake2bHash function H and compression function C. Argon 2 first takes user's message M and nonce to extract entropy. All the other secondary inputs are also there.

$H_0=H(p,l,m,c,v<M>,M,<N>,N,<K>,K,<X>,X)$.

Here H_0 is 32 byte value and variable M,N,K,X are preplaced with their length.

1.3 IDA (Information Dispersal Algorithm)

IDA is an algorithm which split the file and data packet into small pieces and makes them unrecognizable. These small pieces of data stored on various different clouds across the network. The sliced data can be reassembled at other end using the right key. According to [18], this method breaks the file F of size length L into m pieces F_i and each length $F_i=L/m$.IDA provides good confidentiality and integrity of data.

2. RELATED WORK

2.1. Password Security using hashing Algorithms

Sriramya, Karthika [18] proposed that hashing is better than encryption as plain text cannot be obtained from generated hash value because hashing is a one way transformation process. But sometimes when two different passwords have the same hash value, due to collision effect [19], the hash value can be obtained. That's why hashing algorithms as MD5, SHA1 and SHA 256 are not very much secure and vulnerable to attacks.

[19] explained that there are many software like Hashcat [20] and John the Ripper [21] are available ,which makes traditional hashing algorithm useless as they can obtain actual password after cracking the hashing algorithms.

To solve this problem a technique called" key Stretching "can be used [22].In this techniques a value called "salt" (random bit of string) is generated using pseudorandom functions and added to the passwords before hashing. It generates a different hash value every time even if the same password is entered. [19] stated that randomness of hash makes it enable to prevent rainbow table and lookup tables attacks but cannot prevent data from brute force attacks and dictionary attacks. So PBKDF2 must be used in this situation as it follows key stretching method. In key stretching computations are added in key generation process and slow down the algorithms. As fast algorithms can easily break than slow algorithms. PBKDF2 is a slow algorithm and besides using salt, there is also iteration count which makes it more secure algorithm [23].

[24] states that user chosen passwords cannot be directly used as secured cryptographic keys as they lack in entropy and have weak randomness. But there are some situation in which password is the only way to authenticate the data for cryptographic algorithms. So pbkdf2 is the first choice to use in this situation as it derives the cryptographic key based on user entered password.

[19] proposed that Scrypt is password based key derivation function that takes user's plain text as input and derives a hash value. It is a memory hard function and highly secure and makes almost impossible for an attacker to crack the system.

[25] proved that among all hashing algorithm, Argon2 is the most secure and best hashing algorithm which give best result with AES. So our proposed model is using blend of two best and highly secured algorithm for password hashing. But they are still not fully prone to all types of attacks so along with these algorithms cryptography should be used for the protection of password.

2.2. Password protection using Encryption algorithms

The last section describes various hashing algorithms used for security of passwords, but all had their some weaknesses against attacks.

This section presents the cryptographic algorithms like encryption, as AES which may achieve required level of data security.

[25] presented an approach of using encryption algorithms like AES, DES, IDEA, RC4,RSA, DSA and ECC that can be used for security of data.

[26] states that password management is the most crucial task when we use hashing to protect the password as brute force attack can easily break the password. So we need a mechanism which can slow down brute force attack.

[27] stated that NIST used Rijndael algorithm as Advanced Encryption Standards (AES) because AES has high security and efficiency. DES did not provide effective data security so it is replaced by AES. Although 3 DES came as an extension of DES but it was very slow. On the other hand AES is a block –Cipher algorithm that used the concept of permutation box and substitution box and it has varying key sizes as AES-128, AES-256 .AES in one of the best encryption algorithm.

[28] has done the comparative study of various symmetric algorithms like AES, DES and 3 DES and asymmetric algorithms RSA and concluded that AES is the most efficient in terms of time, throughput and encryption/decryption speed.

[29] also performed comparative analysis of algorithms like AES, DES and RSA with LSB substitution technique and found out that AES is the more efficient not only in terms of time, throughput and encryption/decryption speed but in memory usage also.

[30] also compared the various symmetric algorithms based on various factors as speed ,block size, key length and security and concludes that AES was much efficient in terms of encryption/decryption time ,throughput and faster too.

Table:1 Comparative study of various encryption algorithms [30]

Characteristics	**AES**	**DES**	**3DES**	**IDES**	**Blowfish**	**RC5**
Key Length	128, 192, 256	56	112, 168	128	32 - 448	MAX 2040
Block Size	128, 192, 256	64	64	64	64	32, 64, 128
Speed	Very Fast	Very Slow	Slow	Slow	Fast	Slow
Security	Considered Secure	Proven Inadequate	Considered Secure	Proven Inadequate	Considered Secure	Considered Secure
Cryptanalysis Resistance	Very strong against differential, truncated, differential, linear, interpolation and square attack.	Vulnerable to differential and linear cryptanalysis, weak substitution table.	Strong against differential, truncated differential, linear, interpolation and square attack.	Vulnerable to differential and linear cryptanalysis.	Strong against the standard differential and linear cryptanalysis.	Vulnerable to differential, truncated differential, linear, interpolation and square attack.

Although AES proved quite promising algorithms against various cryptanalytic attacks but some improvements suggested to enhance its security features [31]. Use of 3 cipher key may increase the security of AES. Increment in key size leads to increment in encryption/decryption time which makes difficult for a cracker to break the password.

[32] proposed that being the very popular and successful algorithm AES is not fully secure from some attacks like brute force attacks and side channel attacks. Although AES is faster, efficient and very secure than other encryption algorithms like DES, 3DES,RSA, IDEA and hashing algorithms like MD5,SHA1 and Scrypt under some situation . It prevents various attacks lie GPU, parallel processing ,linear and algebraic attacks as it has memory acceleration feature but still some key based attacks like side –channel attack, brute force attacks and reverse engineering attacks can break AES.

Therefore a hybrid approach with different combination of hashing and encryption algorithms can be used to make password more secure and protected.

2.3. Literature review of Hybrid algorithms

The last two sections represented strong techniques of hashing (PBKDF2 and Argon) and encryption algorithm(AES), both the algorithms were individually immune against many attacks .But had some weaknesses against high computation attacks .

[33] proposed an approach of using hybrid model of PBKDF2 along with AES in CTR mode and SHA3 as hashing algorithm . SHA is used to generate the key for AES. This model was found to be faster than hashing algorithm like Scrypt.

[34] presented a hybrid model of BCrypt and AES to secure financial information from brute force attacks by securing the passwords. It was analyzed on the basis of encryption/decryption time and throughput.

[35] proposed a hybrid algorithm AES (GCM), BLAKE2 and Scrypt . It was based on key derivation function and use of secured key for AES and found to be fast and efficient.

With the help of above literature review it can be concluded that combination of PBKDF2 and Argon2 is the strongest key derivation function mechanism based on memory hard functions and it can be used against multiple attacks like dictionary attacks and brute force attacks but still prone to GPU attacks .In this situation AEs encryption can be used to overcome this weakness as its hardware acceleration feature works against custom hardware attacks. Therefore this research proposes on hybrid combination of PBKDF2 and Argon2, AES and IDA (Information Dispersal Algorithm) and analyzes the performance of proposed algorithm.

3. Proposed Work

This section presents overview of proposed work for password protection from brute force attack. The main objective is to use password to generate cryptographic key and then use this key as an input to AES algorithm. So this presented mechanism enable the organizations to put their data on cloud without any hesitation.

The proposed methodology contains the following steps:

- Cryptographic key generation using PBKDF2 and Argon2
- Encryption using AES algorithm
- Data Slicing using IDA(Information Dispersal algorithm)

- Data assembling using IDA algorithm
- Data decryption using AES along with key generated using PBKDF2 and Argon2

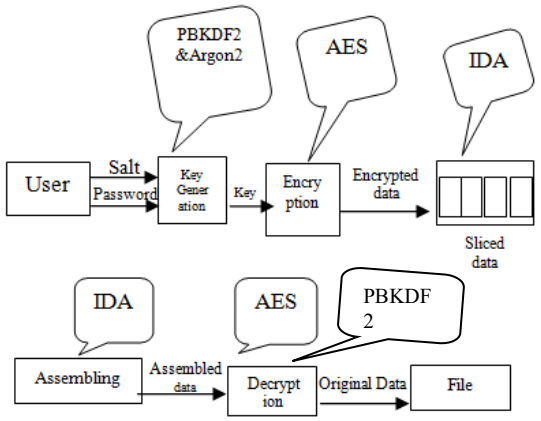

Figure:1 Block diagram of proposed work

4. Methodology

This proposed work uses PBKDF2 and Argon2 for the generation of secured derive key. Then this key will be feed to AES as an cryptographic key for encryption, then this encrypted data will be sliced over multiple clouds using Information Dispersal Algorithm, after that data assembling will be done using IDA and again using AES algorithm data will be decrypted and securely obtained.

Algorithm for the given approach is as follows:

Step: 1 Input user chosen password and file name by the user.

Step: 2 Take password and generate hash using PBKDF2. Set this as the C_d key.

Step: 3 Pass the key produced in Step: 2 as input to Argon2 and after specified no. of iterations most secured key will be generated free from almost each attack. Set this as C_k key.

Step: 4 Pass the key produced in step 3 as input to AES -256 encryption function.

Step: 5 The encrypted file will be dispersed at multiple clouds (m slices) using Information dispersal algorithm

Step: 6 sliced data will be concatenate using IDA after verify the hash value stored in database and get the reconstructed encrypted data E_d

Step: 7 Take the encrypted data E_d and again enter the password to generate a key using PBKDF2 to decrypt the data by AES decryption. As there is no need to store key anywhere in the database and every time a different derived key is generate even for the same password.

Step: 8 Thus using secured password data on the cloud will be safe

5. Implementation

This section describes the implementation of the proposed model. We have developed a web application in Jquery by using PHP and Azax. The application is locally over XAMPP server.

The homepage of the web application has two input boxes prompting for file name and password. As user selects the file and enters the password to generate the key ,the password is passes through PBKDF2 and Argon2 .(Key derivation phase).As already explained that if a user enter same password many times , the web application generates different hash value every time because of random salt.

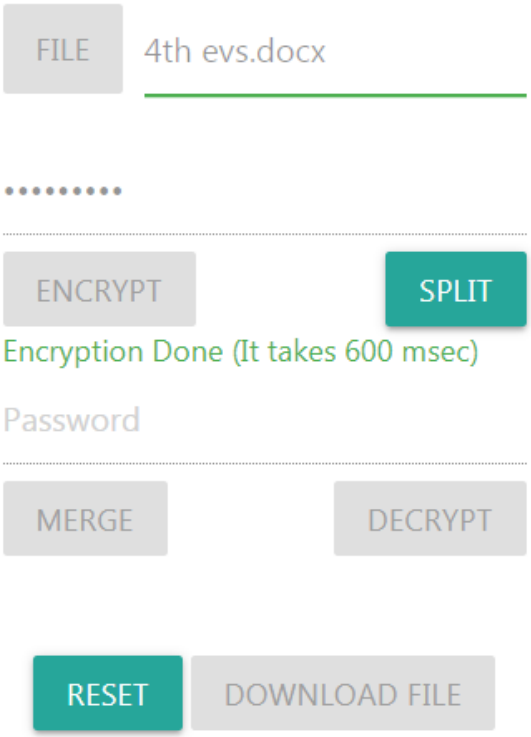

Figure: 2 Home page of web application

This hash value will be treated as private key for AES encryption. Now the derived key C_d encrypt the data and split the file at multiple clouds .Then again after entering the password pbkdf2 generate the key and dispersed files are merge using IDA and generated key is used to decrypt the file AES decryption. Then the decrypted file can be downloaded can data can be access only by the authenticated user.

6. Evaluation

Encryption time is the total time taken to encrypt the plain text, i.e total time taken for PBKDF2 along with Argon2 hashing and AES encryption together. The efficiency of a cryptographic algorithm is inversely proportional to encryption time.[36]

On the other hand, the performance of an algorithm is measured by its throughput. It can be calculated by dividing the size of plain text by encryption time.

Results of encryption time taken by Bcrypt with AES, and (PBKDF2+Argon2 with AES +IDA) for same users are compared. The combination of BCrypt and AEs has already examined by [37].

Table:2 Analysis of Encryption time

Data	**Bcrypt+AES (in seconds)**	**PBKDF2+Argon2+AES**
Data 1	0.487353239	0.470456361
Data 2	0.408241016	0.398463434
Data 3	0.502834321	0.486647516
Data 4	0.405210823	0.382048292
Data 5	0.392928871	0.254532654
Average Time	0.43931366	0.39842965

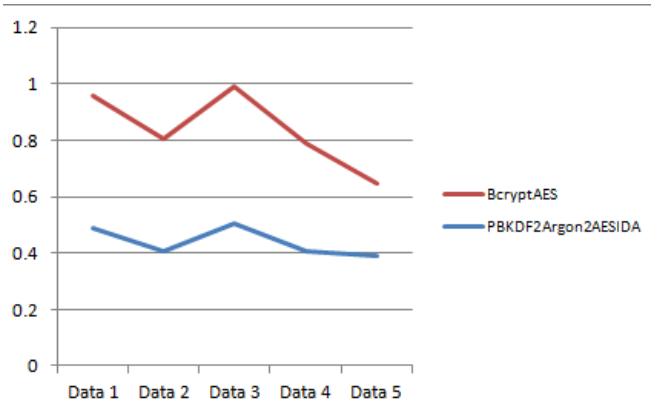

Figure: 3 Analysis of Encryption time

7. Discussion

As discussed earlier in literature review, BCrypt was the strongest KDF algorithm before Argon 2. The result from the above comparison shows that AES with PBKDF2 and Argon 2 has very low encryption time compared to others. Therefore, for this research, the proposed hybrid model of PBKDF2, Argon2 and AES proved to be more secure than others against brute force attacks.

8. Conclusion and Future Scope

The main aim of proposed solution was to evaluate whether security of data can be enhanced by the combination of hybrid hashing and encryption algorithms and check their vulnerability against brute force attack. As PBKDF2 and Argon2 proved to be very strong combination of key derivation function as complexity of password is increased by passing it to PBKDF2 and Argon2 is based on memory hard function which makes it secure against brute force attacks and data dictionary attacks. Then encryption through AES algorithm made it more secure against GPU attacks as AES proven to be the most strong encryption algorithm. Then Information dispersal algorithm secure the encrypted data as if any breach found during the transmission of data then security can be break. So IDA helps to prevent this problem by slicing the encrypted data over multiple cloud and make proposed system more secure and safe.

The future scope of this research could be to check whether there are other combinations of algorithms which can enhance the security of password with AES or with KDF. Other different combination of hybrid algorithms can also be considered for this model.

References

[1] https://csrc.nist.gov/publications/detail/sp/800- …….145/final
[2] George Hatzivasilis," Password –Hashing Status", Journal Cryptography 1020010.
[3] Farcasin, M.; Chan-tin, E. Why we hate IT: Two surveys on pre-generated and expiring passwords in anacademic setting. InSecurity and Communication Networks; Wiley: Hoboken, NJ, USA, 2017.
[4] Cavusoglu, H.; Cavusoglu, H.; Raghunathan, S. Efficiency of Vulnerability Disclosure Mechanisms toDisseminate Vulnerability Knowledge.IEEE Trans. Softw. Eng.2007,33, 171–185
[5] Fruhwirth,C.:Newmethodsinharddiskencryption(2005),http://clemens.endorphin.org/nmihde/nmihde- A4- ds.pdf
[6] Shannon,C.E.:Predictionandentropyofprintedenglish.Bellsystemtechnicaljournal30(1),5064(1951)
[7] NIST:SP800-63-2Version2:Electronicauthenticationguideline(2013)
[8] Ertaul, L., Kaur, M. and Gudise, V. A. K. R. (2016). Implementation and performanceanalysis of pbkdf2, bcrypt, scrypt algorithms,Proceedings of the International Confer-ence on Wireless Networks (ICWN), The Steering Committee of The World Congressin Computer Science, Computer , p. 66.Gore
[9] Hellman, M. A Cryptanalytic Time-memory Trade-off.IEEE Trans. Inf. Theory2006,26, 401–406,doi:10.1109/TIT.1980.1056220.
[10] Provos, N.; Mazieres, D. A Future-Adaptable Password Scheme. In Proceedings of the 1999 USENIX AnnualTechnical Conference, FREENIX Track, Berkeley, CA, USA, 23–26 August 1999; pp. 81–9
[11] Bonneau, J.; Schechter, S. Towards reliable storage of 56-bit secrets in human memory. In Proceedings ofthe 23rd USENIX Conference on Security Symposium (SEC'14), San Diego, CA, USA, 20–22 August 2014;pp. 607–623.
[12] Chen, L.; Lim, H.W.; Yang, G. Cross-Domain Password-Based Authenticated Key Exchange Revisited.ACM Trans. Inf. Syst. Secur.2014,16, 15
[13] NIST: Recommendation for Password-Based Key Derivation, NIST Special Publication 800-132.December 2010.Availableonline: http://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-132.pdf(accessed on 28 January 2017).
[14] Password Hashing Competition (PHC)-https://password-hashing.net
[15] A. Biryukov et al., Argon2: the memory-hard function for password hashing and other applications, https://password-hashing.net/argon2-specs.pd
[16] M. Broz, "Phc benchmarks," 2015, https://github.com/mbroz/PHCtest/blob/master/output/phcround2.pdf.
[17] Rabin, M.O. Efficient dispersal of information for security, load balancing, and fault tolerance.J. ACM1989,36, 335–348.
[18] Sriramya, P. and Karthika, R. (2015). Providing password security by salted passwordhashing using bcrypt algorithm,ARPN Journal of Engineering and Applied Sciences10(13): 5551–5556
[19] Ertaul, L., Kaur, M. and Gudise, V. A. K. R. (2016). Implementation and performanceanalysis of pbkdf2, bcrypt, scrypt algorithms,Proceedings of the International Confer-ence on

Wireless Networks (ICWN), The Steering Committee of The World Congressin Computer Science, Computer , p. 66.
[20] https://www.openwall.com/john/
[21] https://hashcat.net/hashcat/
[22] Kartik.S, Muruganandam.A." Data Encryption and Decryption by using Triple DES and Performance Analysis of Crypto System",International Journal of Scientific Engineering and Research,Volume:2 Issue:11
[23] William Stallings," Cryptography and Network security".
[24] Turan, M. S., Barker, E., Burr, W. and Chen, L. (2010). Recommendation for password-based key derivation,NIST special publication800: 132.
[25] Hatzivasilis, G., Papaefstathiou, I. and Manifavas, C. (2015).Password hashingcompetition-survey and benchmark.,IACR Cryptology ePrint Archive2015: 265.
[26] Alvarez, R., Andrade, A. and Zamora, A. (2018). Optimizing a password hashing functionwith hardware-accelerated symmetric encryption,Symmetry10(12): 705.
[27] Widiasari, I. R. (2012). Combining advanced encryption standard (aes) and one timepad (otp) encryption for data security,International Journal of Computer Applications57(20).
[28] Singh, G. (2013). A study of encryption algorithms (rsa, des, 3des and aes) for informationsecurity,International Journal of Computer Applications67(19).
[29] Padmavathi, B. and Kumari, S. R. (2013). A survey on performance analysis of des, aesand rsa algorithm along with lsb substitution,IJSR, India
[30] Kant, D. C. and Sharma, Y. (2013). Enhanced security architecture for cloud datasecurity,International journal of advanced research in computer science and softwareengineering3(5).
[31] Sachdeva, S. and Kakkar, A. (2018). Implementation of aes-128 using multiple cipherkeys,International Conference on Futuristic Trends in Network and CommunicationTechnologies, Springer, pp. 3–16.
[32] Chhabra, S. and Lata, K. (2018). Enhancing data security using obfuscated 128-bitaes algorithm-an active hardware obfuscation approach at rtl level,2018 InternationalConference on Advances in Computing, Communications and Informatics (ICACCI),IEEE, pp. 401–406.
[33] Alvarez, R., Andrade, A. and Zamora, A. (2018). Optimizing a password hashing functionwith hardware-accelerated symmetric encryption,Symmetry10(12): 705.
[34] Kumar, N. and Chaudhary, P. (2018). Password security using bcrypt with aes encryptionalgorithm,Smart Computing and Informatics, Springer, pp. 385–392
[35] Almorabea, A. M. and Aslam, M. A. (2015). Symmetric key encryption using aes-gcmand external key derivation for smart phones, pp. 264–270.
[36] Arora, M., Sharma, S. and Engles, D. (2017). Parametric comparison of emds algorithmwith some symmetric cryptosystems,Egyptian informatics journal18(2): 141–149.
[37] Kumar, N. and Chaudhary, P. (2018). Password security using bcrypt with aes encryptionalgorithm,Smart Computing and Informatics, Springer, pp. 385–392