DNA Cryptography Based User Level Security for Cloud Computing and Applications

Prasanna Balaji Narasingapuram, M. Ponnavaikko

Abstract: This paper proposed a novel cryptography method for enhancing the user level security to avoid malicious user entering into cloud applications. Existing research works have been proposed various cryptographic methods, algorithms and techniques for validating the user for accessing data or operating cloud applications. But still the malicious user activity like sybil, sinkhole, Denial of Service, Distributed Denial of Service, Economic Denial of Sustainability, selective forwarding and so on, is increasing day by day in cloud. This paper taken this problem as a major problem and motivated to provide a better solution which can eliminate the malicious user activity in cloud. To do this, this paper used DNA cryptography method for generating a strong key for user and data encryption – decryption process. User information is converted into human deoxyribonucleic acid form for generating strong key and data encryption. The implementation of the proposed approach is carried out in DOTNET framework and the experimental results are verified. Based on the results the performance is evaluated by comparing with the existing results.

Keywords: Cryptography, DNA Cryptography, Cloud Security, User level security, Strong Key Generation, Data Encryption, Data Decryption.

I. INTRODUCTION

Security services comprises of confidentiality, authentication and data integrity, and digital signature. In case, a person “A” wants to send a message to another person “B”, secretly they need to follow the above said security service mechanisms. In confidentiality the data security is provided using symmetric or asymmetric method in two different ways such as block cipher and stream cipher. Symmetric method uses a single key [K] for both encryption and decryption, whereas in asymmetric method uses a pair of keys [KU, KR] for encryption and decryption process separately. Data Encryption Standard [1-4] and Advanced Encryption Standard [1-4] algorithms belong to symmetric method. Rivest–Shamir–Adleman algorithm belongs to asymmetric method and it uses public key private key for encryption – decryption process. Authentication is a process where it validates the data/message sender. For integrity a fixed length value is used for converting the plain text in to unreadable format. That appending the fixed length value into the plain text. In order to obtain the fixed length value Message Digest [5], Secure Hash Algorithm -512 [6] and keyed-hash message authentication code [7] algorithms are used. Generally, integrity is, the receiver should trust the data is not modified by anyone in the network. In the digital signature algorithm, the sender used his own private key for encrypting the data and send to destination. It can be done by digital signature algorithm.

![Figure-1. Taxonomy of Cryptographic Algorithms (Existing)](image)

In term of security applications, the Figure-2 shows the taxonomy of algorithms used for security purpose. According to the applications the selected algorithms are selected to provide various kinds of security in any kind of network applications.

Revised Manuscript Received on January 20, 2020

Prasanna Balaji Narasingapuram, Research Scholar, Computer Science Engineering, Information Technology, Bharath Institute of Higher Education and Research (BIHER), Chennai, India. prasannabalaji.narasingapuram@gmail.com

Dr. M. Ponnavaikko, Provost, Bharath University, Chennai, India. ponnava@gmail.com
DNA Cryptography Based User Level Security for Cloud Computing and Applications

Most of the people in internet/network likes to transmit their data or message securely. One of the methods is converting the data into unreadable format. Author in [8] used set of alphabets to replace each character in the message transmitted from source node to destination node. For example, each letter ‘A’ is replaced by D, ‘B’ by ‘E’ and so on. Some of the methods verifies and authorize only certain people to send and receive the data in the network. In this kind of scenario, some of the authorized people can be converted as a sinkhole (one who holds the entire data by himself) [9]. In recent days IT, various techniques, methods and approaches are used for secured data transmission. One of the cryptographic methods used for secured communication & data transmission for protecting the information is Deoxyribonucleic acid (DNA) cryptography [11-12]. DNA cryptography is introduced by author in [10]. It is the procedure of changing an ordinary plain-text into unintelligible or unreadable text and vice versa. It is the ancient art originated from the Egyptian recorders who used non- standard secret symbol used in epitaph or carving. In internet applications, protecting and providing security for data while moving and at rest is more difficult and imposes challenge for organizations. Cryptography is concerned with Confidentiality, Integrity and Availability which is known as Confidentiality, integrity and availability (CIA) triad. Modern cryptography also consists of an additional characteristic called non repudiation. Cryptography provides the mathematical expression and techniques for the terms related to information/data security such as,

- Confidentiality [13],
- Data Integrity,
- Entity Authentication And
- Data Origin Authentication.

The above said methods need to be improved enormously in recent years with technological advancements and growing computing power. In cryptography Encryption is a term which is defined as a process of encoding a message or information which is identified only by authorized users. It is used to make the information hidden. The unauthorized users cannot access the data. The word encrypt refers make the data is secret and it can be written as,

\[
\text{En} \rightarrow \text{"to make"}
\]

\[
\text{Crypt} \rightarrow \text{"secret"}
\]

\[
\text{Encrypt} \rightarrow \text{"to make secret"}
\]

To read an encrypted file, one must have access to a secret key which is used to decrypt the data. The process of encryption can be obtained using two different algorithms are:

- Asymmetric or Public Key Cryptography [14-17].
- Symmetric or Secret key cryptography [14-17].

II. CHALLENGES IN TRADITIONAL CRYPTOGRAPHY

Modern computers store data using a binary format. The size of the keys used in recent cryptographic applications is too big. It is very much difficult to crack a key when a billion calculations perform at a second as the combination to crack the key is larger and takes more time. Quantum computation is a new phenomenon which stores data using quantum bits. This performs calculations faster and hence the codes which take more time to break can be cracked speedily. Some of the challenges of traditional cryptographic methods are, in which infrastructure it is executed, key size, and the quality of the algorithm. While thinking about the infrastructure or platform, various traditional algorithm has been used for solving security issues. Recent days cloud computing and all other networking applications need information security for protecting the data and user validation. User validation, validates the user and authenticate them after validity. As traditional encryption algorithm has severe security problems. The field of information security give importance to the new way of protecting the data. The DNA based cryptography has identified as new way of secure data in the form DNA molecules which uses DNA strands to hide the information. The main objective of DNA cryptography is to provide confidentiality when the persons sends data over a network. This paper discusses about DNA Cryptography, difference between traditional cryptography and DNA Cryptography, various works done in the field of DNA Cryptography

The similarity between information computing and DNA computing are illustrated in Figure-3. In computing industry, the user input like numerals, alphabets and alphanumeric are converted as binary numbers can be identified only by the processor. Similar to this, biological information is represented as DNA molecules and it is coded using AGCT (A-Adenine, C-Cytosine, G-Guanine, T-Thymine letters).
This paper focused on using DNA computing, where it is used to create novel materials for next generation microprocessors. In 1994, Dr. Adleman [18] solved HDPP (Hamiltonian Directed Path Problem) using DNA. DNA is not directly used in computation, rather it is acting as a massive memory. He described that the solution of molecular combinations can be used to solve any combinatorial problems. It is done by experimenting the DNA-computational system as a simulation model for the combinatorial problems. Adleman proved that DNA computing is suitable for a greater number of combinatorial problems, where this paper also trying to use DNA computing for providing user level security in cloud applications.

![Figure-3. DNA Computing](image)

Figure-3. DNA Computing

From various experimental explanations and tutorials of Adleman, it is understanding that DNA computing [19-20] is used to store large volume of data using re-combinative characteristics of DNA. A small size of DNA can provide millions to billions of parallel interactions speedily. It is linear processing of parallel processing. AND, OR, NOR and NOT operations are mainly used for linking, cutting, pasting and other operations suitable with DNA. One of the function complementarities makes the DNA as unique. It can be used for unique key development or in error correction. The speed and memory occupation of DNA computing comparing with other computers is given in Table-1, it shows the DNA is highly suitable for high speed parallel process over large volume of data at high speed.

Table-1. DNA Computing Ability Comparing with other Computing

| Processor                  | Capacity               |
|---------------------------|------------------------|
| Desktop                   | 10⁶ operation / second |
| Super Computer            | 10¹⁵ operation / second |
| 1 μmol of DNA             | 10¹⁵ operation / second |
| 1 Joule of Energy         | 2 x 10¹⁵ operations    |
| Memory Capacity           | 1 bit / cubic nanometre|

The DNA computer the data is represented using a sequence of four nucleotides “A-Adenine”, “C-Cytosine”, “G-Guanine”, and “T-Thymine”. Electrical impulses are replaced by chemical properties of the molecules. It is used to analyze the patterns of data combination or data string. For example, data is converted into ACGT form, and into binary or decimal number as an unbreakable password for user identity and validation. It gives improved security for user-level security and data-level security. But this paper focuses on using user-level security alone.

III. PROPOSED SYSTEM

The proposed system generates a DNA based key for user authentication key to get entry or data access permission in the network/cloud applications. A new proposed encryption method is used based on random number generation for creating a DNA pattern. The entire algorithm comprises of three stages such a key generation, random key generation and encryption-decryption. Initially the input data is encrypted and feed as input into the next level. Second, a random number is generated for example, P_k, is used for encryption in the next level. Finally, the decryption process is applied. The input data is a plain text, having set of characters. Else each single element is considered as a character and changed into relevant American Standard Code for Information Interchange (ASCII) form. The ASCII character is converted into binary form. The entire process of DNA encoding is illustrated in Figure-4. In this encryption process, an input message considered as M and transmitted to the receiver after encryption.

Encryption Process

The encryption process is explained in the following steps as:

**Step-1:** Original text is converted into ASCII (decimal Form)

**Step-2:** All decimal values are considered as blocks

**Step-3:** ASCII message into binary form (0’s and 1’s)

![Figure-4. DNA Based Data Encryption](image)

Figure-4. DNA Based Data Encryption
The encryption process is described using a numerical illustration as an example: the input data is BALA, the process is expressed as:

**Step 1:** ASCII value of “BALA” is taken from ASCII table given in Table-2.
- B = 66
- A = 65
- L = 76
- A = 65

**Step -2:** for each ASCII code the equivalent binary data is converted as

- B = 66 = 00010010 = 00 | 01 | 00 | 10 = ATAG
- A = 65 = 00010001 = 00 | 01 | 00 | 01 = ATAT
- L = 76 = 00011100 = 00 | 01 | 11 | 00 = ATCA
- A = 65 = 00010001 = 00 | 01 | 00 | 01 = ATAT

**Table-2. ASCII – Values**

| Char | ASCII | Decimal | Bits | Char | ASCII | Decimal | Bits |
|------|-------|---------|------|------|-------|---------|------|
| 0    | 48    | 000000  | f    | 70   | 010110 | d      | 100  |
| 1    | 49    | 000001  | g    | 71   | 010111 | e      | 101  |
| 2    | 50    | 000010  | h    | 72   | 011000 | f      | 102  |
| 3    | 51    | 000011  | i    | 73   | 011001 | g      | 103  |
| 4    | 52    | 000100  | j    | 74   | 011010 | h      | 104  |
| 5    | 53    | 000101  | k    | 75   | 011011 | i      | 105  |
| 6    | 54    | 000110  | l    | 76   | 011100 | j      | 106  |
| 7    | 55    | 000111  | m    | 77   | 011101 | k      | 107  |
| 8    | 56    | 001000  | n    | 78   | 011110 | l      | 108  |
| 9    | 57    | 001001  | o    | 79   | 011111 | m      | 109  |
| :    | 58    | 001010  | p    | 80   | 100000 | n      | 110  |
| ;    | 59    | 001011  | q    | 81   | 100001 | o      | 111  |
| <    | 60    | 001100  | r    | 82   | 100010 | p      | 112  |
| =    | 61    | 001101  | s    | 83   | 100011 | q      | 113  |
| >    | 62    | 001110  | t    | 84   | 100100 | r      | 114  |
| ?    | 63    | 001111  | u    | 85   | 100101 | s      | 115  |
| @    | 64    | 010000  | v    | 86   | 100110 | t      | 116  |
| A    | 65    | 010001  | w    | 87   | 100111 | u      | 117  |
| B    | 66    | 010010  | x    | 96   | 101000 | v      | 118  |
| C    | 67    | 010011  | y    | 97   | 101001 | w      | 119  |
| D    | 68    | 010100  | z    | 98   | 101010 | x      | 120  |
| E    | 69    | 010101  |     | 99   | 101011 | y      | 121  |

**Step 4:** Hence the code is 147-148-149-148 and it is the final encrypted password. This kind of password can be generated for Bank Account number, credit / debit card number generation.

**Random Key Generation**

Next stage process of DNA cryptography is a random key generation from 1 to 256 and it is assigned as $P_k$ for encryption process. In accordance to the values of $P_k$ the generated values are assigned as index, which can relate to the combination of A, T, G and C. For example, when $P_k = 1$, the DNA code in AAAA, which is given in Table-4. The 256-index value is created using permutation of four characters A, T, G and C. If $P_k$ changed then the index table is also gets changed. The encryption process in put BALA is encrypted into ATAG-ATAT-ATCA-ATAT.

| DNA Code | Binary Code |
|----------|-------------|
| A        | 00          |
| T        | 01          |
| G        | 10          |
| C        | 11          |

Hence the input data BALA is converted into ATAG-ATAT-ATCA-ATAT

**Step-3:** Each character of BALA is represented into DNA code pattern. Now each pattern is assigned with random key generated and given in 147, 148, 149, 148, which is given in Table-4. It has random key and the DNA code.
Decryption Process

It is the process converting the encrypted data into original data back. It can be done only by the authorized user who is the owner of the data. Only the owner can do decryption since the owner only have the secret key for decryption. In the decryption process, initially the encrypted data is feed as input. Then $P_k$ is generated by block. The convert into DNA code and corresponding binary values. Then the pair of binary values is substituted by 00 for 1, 01 for T, 10 for G and 11 for G. Then the block is arranged into binary values to block. Then the binary value is converted into ASCII values. Finally, from the ASCII value is converted into original data or decrypted message. The entire process is given in the following steps and illustrated in Figure-6.

### Decryption Process

**Step 1:** Take the encrypted message 147-148-149-148

**Step 2:** Substitute random generated key at an instance

|   |   |   |   |
|---|---|---|---|
| 147 | ATAG | 00 | 01 |
| 10  | 00010010 | -66 | B |
| 148 | ATAT | 00 | 01 |
| 01  | 00010001 | -65 | A |

The above described DNA cryptographic algorithm is used for key generation (as a password) for any authorised user who can operate any application or can communicate with the other authorised user in the same network. For example, when a user entering their details into the application, like username and password, the password given by the user is taken as the input data and it is encrypted using DNA cryptographic method. The process is illustrated in Figure-6. During the user registration the user provided password is encrypted and given to the user as the encrypted password and is also stored in the application server or in the DB for further comparison. It provides a high security in various applications, like when the user enters the data, it will automatically be encrypted and authenticated without the knowledge of the user.

### Table 4. Random Key Generation for DNA code

Based on this DNA cryptography, most of the enterprise applications shared in online by a greater number of users can be authenticated by verifying their DNA key. For example, if two users A and B needs to share their data, they need to enter their password, and it is encrypted using DNA cryptographic method. After encryption, the encrypted password of both the users A and B are verified in the Data Base, whether the encrypted password is available or not. If it is available for the corresponding user name, then only they (A and B) are permitted to access the application or they can share their official information.
This functionality is illustrated in Figure-7. This process can be used in any network application or cloud application for authenticating the users. In certain cases, the encrypted key is used for encrypting the data or message transmit from one user to another user in the network. DNA cryptography can be used for both encrypting small size data to large size data files which do not affect the memory or time taken for encryption and decryption process. Comparing with other cryptographic methods, DNA is fast and easy in process. It does not make more complexity regarding computational processes. Also, it can be implemented and executed in any computer programming languages like C, C++, JAVA, DOTNET, Python, and etc. Hence it is not language dependent. The entire functionality of the encryption and decryption process of the proposed DNA cryptography method is given in the form of pseudo code above. Which can be programmed directly in any computer programming language and the efficiency can be verified. In this paper the proposed DNA cryptograhic algorithm is implemented in DOTNET based internet application and the performance is compared with the other existing approaches. It can also be implemented and experimented in Python.

Pseudocode _ DNA_Encryption()

{Key ← random();
  User ← key;
  Data1 ← user.data;
  Data2 ← Ascii(data1);
  Data3 ← DNAcode(Data2);
  Pk ← Choose number from 1 to 256
  Each DNAcode ← Pk
  Edata ← encrypt(msg, DNAcode)
  Return Edata;)

Pseudocode _ DNA_Decryption()

{Input pk, Edata;
  Data1 ← Permutate(Edata, DNAcode, Pk mod 256)
  Data2 ← binary(Data1)
  Data 3 ← Ascii(Data2)
  Data ← Decrypt(Data3, DNAcode)
  Return Data;)

Figure-6. User Assigned Password Converted into DNA Crypted Password

Figure-7. Two Users Can Share Their Data After DNA Crypted PWD Based Authentication
IV. EXPERIMENTAL RESULTS AND DISCUSSION

The proposed DNA cryptographic algorithm is implemented in C#.NET language of DOTNET Frameworks over Windows-10. The algorithm is considered as a symmetric key encryption algorithm, because it uses single key for encrypting the data character by character. One of the features used in the proposed system is, it does not require DNA chromosome or any other data same to DNA sequence for data processing. In order to simplify the process, the private key $P_k$ range is taken from 1 to 256 for easy conversion. In order to obtain the encryption, there are two values such as the input data and $P_k$ values need to be transmitted. To evaluate the performance of the proposed DNA cryptography method the execution time complexity is calculated and compared with other existing algorithms. The comparison results are given in the following Figure-8. From the results, it is decided that the security performance is good and satisfactory. Also, it is concluded that the proposed DNA cryptography is highly suitable for any network / cloud applications in terms of authentication.

![Figure-8. Time Complexity Comparison](image)

The time complexity is compared with the existing Asymmetric DNA algorithm discussed in [13], and it is proved that the asymmetric DNA is compared with the DEX, TDES, Blowfish, and AES. Since it has been stated that ADNA is better than the other approaches, this proposed DNA is compared to prove the betterness.

Comparing with the other approaches our proposed DNA

![Figure-9. Time Complexity of Different Stages of the Framework](image)
DNA Cryptography Based User Level Security for Cloud Computing and Applications

does not provide more complexity in terms of time, where it can also reduce the cost complexity. Another factor which determines the performance of the proposed algorithm is time taken for parameter / key generation. In this paper the key generation is the main stage and important process.

Since, the time taken for the key generation process is calculated and compared with the existing approach Electrical Curve Cryptographic (ECC) and Hyperelliptic Curve Cryptosystem (HECC) described in [21]. The comparison result is given in Figure-9 and it shows that the proposed DNA obtained very less time and computational complexity than the other ECC and HECC approaches. Also, the performance of DNA is evaluated by changing the key size and the time complexity is verified. The key size calculated in the experiment is 32, 52, 64 and 128 bits. Finally, for 52 bits the results are compared and given in Figure-9. From the results, it is found that the proposed DNA method obtained less time complexity than the other existing ECC, HECC methods which did the similar kind of research work. The proposed DNA obtained 189ms, 245ms, and 212ms for password generation, user registration and authentication process respectively and it is highly small when comparing with the other existing approaches, given in Figure-9. Hence the proposed DNA is considered as an efficient method for cloud / network applications.

V. CONCLUSION

The main objective of this research work is to design and implement a novel security algorithm for tightening the user-level security. It is an authentication model for any kind of network or cloud applications which has user authentication process. User authentication process is one of the main process and it is very essential process in a secured data transmission application, validates the user as authorised or malicious user and ensure that the particular user can access the data or not. In order to do that authentication process is used as the main process and it is carried out as the initial stage of the research work. The security (user validation) is provided by DNA cryptography based key generation, assignment and verification to the user for authentication. The merits of the DNA cryptographic method are explained in detail and experimented. The results are compared with the existing approach results and proved that the proposed DNA is better than the other cryptographic approaches in terms of time and computational complexity.

In the next level of the research work, infrastructure level security is provided by membrane computing method which is suitable for cloud security.

REFERENCES
1. Gurpreet Kaur, Dr. Gagandeep Jagdev, (2017), “Implementation of DES and AES Cryptographic Algorithms in Accordance with Cloud Computing”, International Journal of Research Studies in Computer Science and Engineering, Vol. 4, No. 4, PP. 1-14.
2. Priyadarshini Patil et al., “A comprehensive Evaluation of Cryptographic Algorithms: DES,3DES, AES, RSA and Blowfish”, 2015, Volume 78, 2016, Pages 617-624.
3. M.Meeen et al., “A study and comparative analysis of cryptographic algorithms for various file formats”, IJSR, 2013, ISSN:2339-7064.
4. Miss. Shakeeba et al., “Cloud Security based on Multilevel Encryption Algorithms”, IJARCE, 2016, ISSN (online):2278-1021.
5. Rivest, Ronald, The MD5 Message Digest Algorithm (online), Internet Engineering Task Force, 1992, Available at: http://tools.ietf.org/html/rfc1321 Accessed on: 2013-25-01.
6. Eastlake, Donald E. 3rd. and Jones, Peter, US Secure Hash Algorithm 1 (SHA1) (online), Internet Engineering Task Force, 2001, Available at: tools.ietf.org/html/rfc3174 Accessed on: 2013-26-01.
7. Bellare, Mihir, Canetti, Ran, and Krawczyk, Hugo, Keying hash functions for message authentication (online), University of California San Diego, Computer Science and Engineering, 1996, Available at: http://cseweb.ucsd.edu/~mihir/papers/kmd5.pdf Accessed on 2013-07-03.
8. "The Basics of Cryptography-Fisher College of Business". [Online] Available: https://fisher.osu.edu/~muhanna.1/pdf/crypt.pdf.
9. "Cryptography Just for Beginners". [Online] Available: https://www.tutorialspoint.com/cryptography/cryptography_tutorial.pdf.
10. A.Gehani, T. LaBean, and J. Reif. DNA-Based Cryptography. Lecture Notes in Computer Science, Springer. 2004.
11. Tornea, O., and Borda, M.E., DNA Cryptographic Algorithms. IFMBE Proceedings. 26:223–226, 2009.
12. Borda M. & Tornea O. DNA secret writing techniques [C]. In COMM (2010), Chengdu: IEEE, June 10-12, 2010: 451–456.
13. Stallings, W., Network security essentials, Prentice Hall, Fourth edition, 2011.
14. Anurag Roy and Asoke Nath, “DNA Encryption Algorithms: Scope and Challenges in Symmetric Key Cryptography”, IJRAE 2016.
15. John H Reif, Michael Hauser, Michael Pirrung and Thomas LaBean, “Application of Biomolecular Computing to Medical Science: A Biomolecular Database System forStorage, Processing & Retrieval of Genetic Information & Material”, Duke University, 2006.
16. Junling Sun, “Sequence Splicing Techniques and Their Applications For Information Encryption”, International Conference on Advanced Mechatronic Systems, Tokyo, Japan, September 1S-21, 2012.
17. V. M. M. Shyam, N. Kiran, “A novel encryption scheme based on DNA computing,” In 14th IEEE International Conference, Tia, India, Dec. 2007.
18. Leonard. M. Adleman, (1994), “Molecular computation of solutions to combinatorial problems”, (1994).
19. Xing Wang and Qiang Zhang, “DNA computing-based cryptography”, in the IEEE proceeding of Fourth International Conference on Bio-Inspired Computing, pp.1 – 3, Oct 2009.
20. Vijayakumar P., Vijayalakshmi V. and Zayaraz G., “DNA Computing based Elliptic Curve Cryptography” in the International Journal of Computer Applications, vol.36, no.4, pp.18-21, Dec. 2011.
21. Radu Terec, Mircea-Florin Vaida, Lenuța Alboae, Lgia Chiorea, (2011), “DNA Security using Symmetric and Asymmetric Cryptography”, International Journal on New Computer Architectures and Their Applications, Vol.1, No. 1, pp. 34-51.