Securing Multimedia by Using DNA-Based Encryption in the Cloud Computing Environment

SUYEL NAMASUDRA, National Institute of Technology Patna
RUPAK CHAKRABORTY, Guru Nanak Institute of Technology
ABHISHEK MAJUMDER, Tripura University
NAGESWARA RAO MOPARTHI, Koneru Lakshmaiah Education Foundation

Today, the size of a multimedia file is increasing day by day from gigabytes to terabytes or even petabytes, mainly because of the evolution of a large amount of real-time data. As most of the multimedia files are transmitted through the internet, hackers and attackers try to access the users' personal and confidential data without any authorization. Thus, maintaining a strong security technique has become a significant concerned to protect the personal information. Deoxyribonucleic Acid (DNA) computing is an advanced field for improving security, which is based on the biological concept of DNA. A novel DNA-based encryption scheme is proposed in this article for protecting multimedia files in the cloud computing environment. Here, a 1024-bit secret key is generated based on DNA computing and the user's attributes and password to encrypt any multimedia file. To generate the secret key, the decimal encoding rule, American Standard Code for Information Interchange value, DNA reference key, and complementary rule are used, which enable the system to protect the multimedia file against many security attacks. Experimental results, as well as theoretical analyses, show the efficiency of the proposed scheme over some well-known existing schemes.

CCS Concepts: • **Security and privacy → Key management**;

Additional Key Words and Phrases: Cloud computing, DNA computing, complementary rule, American Standard Code for Information Interchange, decimal encoding rule, CloudSim

ACM Reference format:
Suyel Namasudra, Rupak Chakraborty, Abhishek Majumder, and Nageswara Rao Moparthi. 2020. Securing Multimedia by Using DNA-Based Encryption in the Cloud Computing Environment. *ACM Trans. Multimedia Comput. Commun. Appl.* 16, 3s, Article 99 (December 2020), 19 pages.
https://doi.org/10.1145/3392665

1 INTRODUCTION

Multimedia can be defined as the combination of several forms of media that include text, audio, graphics, and video, among others. It can be displayed, played, interacted with, and recorded

Authors' addresses: S. Namasudra (corresponding author), Department of Computer Science and Engineering, National Institute of Technology Patna, Patna, Bihar, India, PIN: 800005; email: suyelnamasudra@gmail.com; R. Chakraborty, Department of Computer Science and Engineering, Guru Nanak Institute of Technology, Kolkata, West Bengal, India, PIN: 700114; email: rupak.jis@gmail.com; A. Majumder, Department of Computer Science and Engineering, Tripura University (A Central University), Suryamaninagar, Tripura, India, PIN: 799022; email: abhi2012@gmail.com; N. R. Moparthi, Department of Computer Science and Engineering, Koneru Lakshmaiah Education Foundation, Vaddeswaram, AP, India, PIN: 522502; email: mnrphd@gmail.com.

1551-6857/2020/12-ART99 $15.00
https://doi.org/10.1145/3392665

by using any computerized or electronic device. Multimedia involves advanced techniques and methods for information extraction and data transformation.

Today, cloud computing is one of the benchmarks to store multimedia files, where a huge number of distributed and parallel computers are interconnected. Here, many technologies like utility computing, virtualization, the network system, and distributed processing are combined to allow many services, such as a server, space, network hardware, and pay-per-use, and many more [1]. The term *cloud* signifies the capability of providing the cloud services remotely. In 2007, cloud computing received popularity, when the collaboration was made between two major IT companies: Google and IBM. There are three main entities in a cloud environment: (i) Data Owner (DO), (ii) Cloud Service Provider (CSP), and (iii) user. The DOs and users use the cloud services provided by the CSP [2–4]. Here, many DOs store their confidential multimedia files, and users access multimedia files from the cloud server. Managing multimedia files in the cloud environment is one of the challenges of top concern due to the presence of many attackers and malicious users. Since the CSP is the central authority or administrator that manages all tasks of a cloud server, it should provide a strong security technique of the user's confidential or sensitive multimedia data or file.

Recently, many researchers have proposed numerous schemes to improve the security of multimedia files and to improve the performance of the cloud environment [5–23]. Shamir [24] proposed a novel Identity-Based Encryption technique, where the sender of any multimedia data or message specifies a unique identity that should match at the receiver's end for decrypting the message. In 1992, the Role-Based Access Control [25] model was proposed based on the job role. Here, based on the job role, the user's accesses are controlled. However, Role-Based Access Control is not secured. The ciphertext is assigned with a number of attributes in Key Policy Attribute-Based Encryption [26], in which the DOs cannot control the access policies, and they must trust the key generator. In Cipher Policy Attribute-Based Encryption [27], an advanced model of Key Policy Attribute-Based Encryption is used in which each ciphertext is associated with an access policy. In Cipher Policy Attribute-Based Encryption, a user's attributes are used to generate the Private Key of the User (USRPrK). Mian et al. [28] proposed a new Provisioning Data Analytic Workload model for the cloud environment; however, this model does not consider the security issue of the confidential multimedia file. In the Activity-Based Access Control model [29], a user's access right to any multimedia file is provided based on the designation of the user in a particular organization. Since there are several components in Activity-Based Access Control model, the system overhead can be increased. The Index Generation–Based Access Control [30] model was introduced by using substring index generation process. Here, key generation time is high. In all of these existing schemes, security of the multimedia data content is a major issue.

Today, DNA computing is widely using in the field of cryptography. Here, DNA computing is used for data encryption, so that the unauthorized, malicious users and attackers are unable to read the data content. In the traditional encryption algorithms, plaintext of any data is converted into its ciphertext by combining 0 and 1. However, in DNA-based encryption, instead of using 0 and 1, data are encrypted by using human DNA bases, namely Adenine (A), Guanine (G), Cytosine (C), and Thymine (T). The concept of DNA has received much attention in the field of information security due to its complex structure. DNA can also be used to store a multimedia file or message. One gram of DNA can store about 700 TB of multimedia files [31]. That means a few grams of DNA can store all multimedia files of the world. The Hiding Messages in Microdots [32] scheme is the first method to use DNA computing for data hiding. However, this scheme is very slow. A novel Public Key System by using DNA (PKSDNA) computing and one-way function was proposed by Tanaka et al. [33]. PKSDNA increases the accessing time of multimedia files. The Chaos-Based Image Encryption (CBIE) [34] scheme was proposed to encrypt images. CBIE does not support strong security. Wang et al. [35] suggested a DNA-based Reversible Data Hiding Scheme (RDHS) for users' sensitive multimedia files. This scheme takes a lot of time to encrypt any multimedia file. A Double-Layer Data Hiding (DLDH) [36] scheme was proposed based on a recombinant DNA technique. DLDH is secured against the password guessing attack. All of these schemes face a security issue. In addition, these schemes take a lot of time for secret key generation, secret key retrieval, multimedia encryption, and multimedia decryption.

To solve the problems of the existing schemes, in this article a multimedia encryption technique is introduced for the cloud computing environment on the basis of DNA computing: the DNA-Based Multimedia Security (DNABMS) model. In the proposed scheme, the DOs encrypt a multimedia file by utilizing a DNA sequence unlike the traditional approaches (0 and 1). Here, the DO randomly generates a long 1024-bit DNA-based Secret Key (DNASK) based on the user's attributes and a password provided by the user. The DO uses the same key to encrypt the multimedia file by using a novel DNA-based encryption process. After encrypting by the DNASK, the DO encrypts the multimedia file using the DO's Private Key (DOPrK). Then, the DO again encrypts the encrypted multimedia file by the Public Key of the CSP (CSPPuK). The DO stores the encrypted multimedia data content on the cloud database, and only the authorized users are able to access the encrypted multimedia file from the cloud server. This proposed scheme can take less time for secret key generation, secret key retrieval, multimedia encryption, and multimedia decryption. The main contributions of this article are summarized as follows:

1) A novel 1024-bit DNASK generation technique is introduced in this work by using DNA computing.
2) To improve the security of a multimedia file, a novel DNA-based encryption technique is proposed by using the 1024-bit DNASK, DOPrK, and CSP's public key. The proposed encryption technique can resist many security attacks.
3) The proposed scheme takes less time for secret key generation, secret key retrieval, multimedia encryption, and multimedia decryption.
4) Experimental results and performance analysis of DNABMS are presented in this work to prove its efficiency over the existing techniques.

The rest of the article is structured as follows. Section 2 discusses the literature review. Background of the proposed scheme is presented in Section 3. The proposed scheme is discussed in Section 4. Section 5 and Section 6 deal with an example of the DNASK generation process and security analysis, respectively. Results and discussions are presented in Section 7. Our conclusion and future work are presented in Section 8.

2 LITERATURE REVIEW

Many researchers have proposed numerous advanced encryption techniques by using DNA computing to protect the users' confidential multimedia files.

In 1999, Clelland et al. [32] proposed the first method for data hiding by using DNA computing, namely hiding messages in microdots. Here, *microdot* can be defined by the process for concealing the messages known as steganography. In this scheme, a DNA-encoded message is first hidden within the massive complex human genomic DNA, then the encoded message is further concealed by restricting this sample to a microdot. However, this scheme is very slow.

In 2000, Leier et al. [37] proposed the Cryptography with DNA Binary Strands technique to improve data security. This scheme consists of two approaches. In the first approach, DNA strands are used for information hiding, and the second approach is based on the graphical subtraction, which helps to strengthen the security of the first approach. Cryptography with DNA Binary Strands does not support fast data access.

Tanaka et al. [33] proposed a public key system by using DNA computing and one-way function. This scheme is mainly developed for solving the key distribution issue between the sender and receiver. Here, messages are hidden in the dummy DNA sequences and restored by using polymerase chain reaction amplification and sequencing. PKSDNA increases multimedia accessing time and is not secured against the password guessing attack.

In 2007, MingXin et al. [38] proposed a symmetric cryptosystem, namely DNA Symmetric-key Cryptosystem (DNASC), for improving data security. Here, decryption and encryption keys are generated by using DNA computing. In DNASC, a huge number of DNA probes are hybridized and identified at the same time, and thus, the decryption processes are executed in a parallel way. This scheme takes a lot of time to encrypt and decrypt any data.

Enayatifar et al. [34] proposed a CBIE scheme based on the genetic algorithm, logistic map, and DNA masking. The main goal of CBIE is to find the best DNA mask for image encryption. In the first phase of CBIE, many DNA masks are created by using the DNA sequences and logistic map. In the next phase, the genetic algorithm is used to find the best DNA mask. This scheme does not provide a strong security mechanism.

In 2017, Wang et al. [35] proposed RDHS by using the DNA Exclusive OR (EXOR) rule for the users' confidential or sensitive multimedia files. In RDHS, information is embedded, and after the extraction process, information is exactly restored as the original host signal. RDHS recovers the original image without any restoration. However, if an attacker gets the binary encoding rule and the DNA EXOR rule, the attacker can get the original content of the data.

A DLDH scheme was proposed by Wang et al. [36] based on a recombinant DNA technique and DNA sequence. In this scheme, a message or data is hidden in a fake DNA sequence, which is generated by using the mapping rule table, and it is further concealed in a living organism (DNA plasmid) by using the recombinant DNA technique. DLDH is secured against the password guessing attack.

3 BACKGROUND OF THE PROPOSED SCHEME

This section deals with two aspects of the proposed scheme: (i) the system model and (ii) the system requirements and design goals.

3.1 System Model

There are three entities in the proposed DNABMS:

1) *Cloud service provider*: This entity is the overall administrator or central authority of any cloud environment that supports infrastructure and provides cloud services by utilizing numerous servers having much power and adequate memory space [39].
2) *Data owner*: DOs are the entities who store their confidential or normal data and multimedia files on the database of the cloud server and depend on the CSP to manage the data.
3) *User*: Users can be considered as the authorized entities or parties who desire to get multimedia file content or any service from the cloud server. Figure 1 depicts the system model of the proposed scheme.

3.2 System Requirements and Design Goals

System requirements and design goals of the proposed scheme include the following:

1) *Fine-grained access control*: The CSP must give a guarantee that only the authorized users can access multimedia files from the cloud server. Users must not access multimedia files by their wishes. The DO has to assign access right to his or her multimedia files, and the users are unable to access any multimedia file if they do not satisfy the access right. The

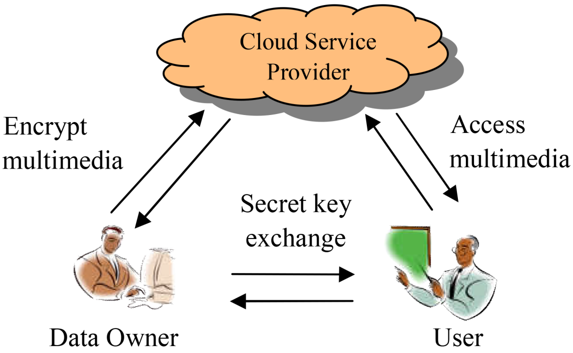

Fig. 1. System model of DNABMS.

main design goal of the proposed scheme is to provide the access right to multimedia files by generating a long 1024-bit password or secret key based on DNA computing.

2) *Security*: Security is one of the major concerns for any cloud environment. There are numerous hackers and attackers in the cloud environment due to its internet-based processes or services. Hackers always want to access users' sensitive and confidential multimedia data content. Therefore, the CSP must support a strong security technique for protecting users' sensitive multimedia files. Achieving an efficient and scalable multimedia storage scheme for the cloud environment is another design goal to provide strong security to the users' confidential information.
3) *Reduce the system's overhead*: The DOs must be online through the complete multimedia communication phase in the existing schemes, which increases system overhead. An effort is made here in DNABMS for minimizing the overhead.

4 PROPOSED SCHEME

A 1024-bit DNA-based password or secret key is used in this proposed scheme to encrypt the users' confidential multimedia file or message, and the key is generated through several processes, which improves the security. The proposed scheme consists of mainly five phases: system setup, registration, login, DNA-based multimedia storage, and multimedia access. Figure 2 shows the entire workflow of DNABMS.

4.1 System Setup

In this system setup phase, a big prime number (p) is chosen by the service provider to identify a multiplicative group Z_p^*. The CSP selects his or her own public and private key pair from Z_p^*. Then, the public and private key pairs are chosen from Z_p^* by the CSP for all of the users. These key pairs are given to the respective users at the time of their registration. The users' public keys are publicly available. The DOs' key pairs (public and private key) are also selected by the CSP, and only the authorized users know the Public Key of the DO (DOPuK). Only the authorized entities know the CSPPuK. Most importantly, an individual entity's private key is kept secret.

4.2 User Registration

The user must send a request to the service provider for registering in the cloud server. When the CSP receives the request, it collects the personal or secret information of the user, such as first school name, date of birth, first teacher name, and address, and generates a user's profile. Then, a secure communication medium (i.e., secure socket layer) is established to deliver the key

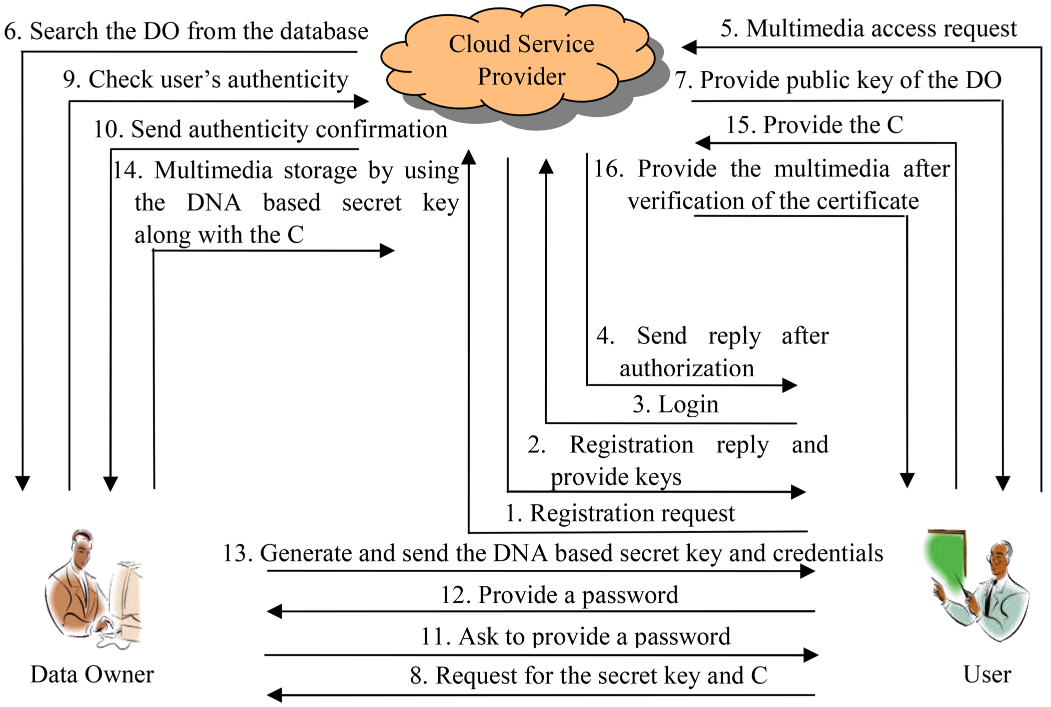
Cloud Service Provider
6. Search the DO from the database
9. Check user's authenticity
10. Send authenticity confirmation
14. Multimedia storage by using the DNA based secret key along with the C
5. Multimedia access request
7. Provide public key of the DO
15. Provide the C
16. Provide the multimedia after verification of the certificate
4. Send reply after authorization
3. Login
2. Registration reply and provide keys
1. Registration request
13. Generate and send the DNA based secret key and credentials
12. Provide a password
11. Ask to provide a password
8. Request for the secret key and C
Data Owner
User

Fig. 2. Workflow of the proposed DNABMS.

pair, registration reply, and duplicate identity. The particular registered user can only be able to fetch the real identity from the duplicate identity. Attackers or hackers cannot get the actual user's identity from that duplicate identity. The same process is followed for the DO's registration in the cloud server.

4.3 User Login

The authorized user or customer is only permitted to log into the system or server after the registration phase. The CSP replies an acknowledgment to the user after a login process. After the login process, a request is sent to the service provider by the user for accessing multimedia files. The CSP sends DOPuK to the user or customer in the encrypted form to get the certificate or access right, as well as the secret key or password, from the respective DO. The user can only request to the corresponding DO after getting the respective DOPuK. After confirming the user's authenticity from the CSP, the DO provides the certificate and secret key of the requested multimedia file to the authorized user in the encrypted form.

4.4 DNA-Based Multimedia Storage

DNA-based multimedia storage process is divided into two phases: DNASK generation and DNA-based multimedia encryption.

4.4.1 DNASK Generation. The DO carries the secret key generation process and generates the DNASK only for the authorized or valid users. The DO asks the user to give a password and fetches essential user attributes, if the user is a valid one.

Then, the decimal encoding rule (Table 1) is applied to the password and attributes' values to make the information in the same format. The decimal encoding rule supports randomness as the

Table 1. Decimal Encoding Rule

Digits/Symbols/ Alphabets	Decimal Value	Digits/Symbols/ Alphabets	Decimal Value	Digits/Symbols/ Alphabets	Decimal Value
1	01	A	11	′	37
2	02	B	12	~	38
3	03	C	13	!	39
4	04	D	14	@	40
.	.	.	.	.	.
.	.	.	.	.	.
.	.	.	.	.	.
8	08	X	34	(	48
9	09	Y	35	)	49
0	10	Z	36	Blank space	50

Table 2. ASCII Values

8-Bit Binary Number	ASCII
00000000	199
00000001	5
00000010	223
.	.
.	.
.	.
11111111	100

DO can apply any decimal encoding rule to improve the security of multimedia files. The resultant decimal number is transformed to its corresponding 8-bit binary numbers, and these 8-bit binary numbers are transformed into their corresponding American Standard Code for Information Interchange (ASCII) values as per Table 2. The main aim of this approach is to generate a random decimal number that corresponds to each 8-bit binary number. These ASCII values are then again converted to their real 8-bit binary number. The length of the resultant binary number may be less or more than 1024-bit. If it is less than 1024-bit, then extra zeros are added to its right side to make it 1024-bit. If not, extra bits are deleted from the left side. The generated 1024-bit binary number is grouped into four parts (i.e., 256-bit in each group). Each group (256-bit) is termed randomly by using the DNA bases, namely A (Adenine), C (Cytosine), G (Guanine), and T (Thymine). The DO then selects a random DNA reference key from the publicly available DNA bases and adds it at the middle of the DNA bases. The complementary rule is then applied to the DNA sequence to make it complex. The main aim of using the complementary rule is to make the DNA sequence more complex by converting it into another DNA sequence. This process improves the security of multimedia files against hackers and malicious users. The complementary rule can be represented as $V(C) = W$, where $C \neq W$. Here, the following complementary rule is used in DNABMS:

(CA) (AT) (TG) (GC) that means $V(C) = A$, and so on.

The resulting DNA sequence is provided to the user or customer as the DNASK. The secret key generation process consists of the following steps:

Table 3. DNA Bases of 2-Bit Binary Numbers

00	01	10	11
G	T	A	C
G	C	A	T
T	C	A	G
T	G	A	C
C	T	A	G
.	.	.	.
.	.	.	.
.	.	.	.
A	C	G	T
A	T	G	C

Step 1: At first, to get the secret key and certificate, a request is sent to the DO by the user.

Step 2: In the 2nd step, the DO verifies the user's authorization, and if the user is a valid one, the next process is executed. Otherwise, the user's request is declined.

Step 3: In the 3rd step, the DO asks the user to give a password.

Step 4: The password and necessary attributes of the user are gathered.

Step 5: The decimal encoding rule is applied to the resultant of step 4 by using Table 1.

Step 6: In the 6th step, individual numbers are transformed into their 8-bit binary values.

Step 7: Here, 8-bit binary numbers are transformed into their corresponding ASCII values by using Table 2.

Step 8: ASCII values are again transformed into their real 8-bit binary values.

Step 9: In the 9th step, bits are deleted or added to make it 1024-bit.

Step 10: Group the binary sequence into four parts (256-bit block) and assign the DNA bases to each part.

Step 11: In the 11th step, a DNA reference key is randomly selected and added at the middle of the resultant DNA bases of the preceding step.

Step 12: Apply the complementary rule on the resultant of step 11 to make it the final DNASK.

The binary number generated in step 9 is used during the multimedia encryption, and the resultant DNA string (secret key) of step 12 is provided to the user after encrypting by the DOPrK and User's Public Key (USRPuK). The DO also sends Table 1, Table 2, Table 3, the DNA reference key, and the complementary rule in the same encrypted message. The corresponding user can decrypt the encrypted message only by utilizing the USRPrK and DOPuK. The user can quickly recover the original key because she or he has all credentials provided by the DO.

4.4.2 Multimedia Encryption by Using the DNA Sequence. In the DNA-based multimedia encryption phase, the DO splits the plaintext into 1024-bit blocks. Here, each 1024-bit block is then grouped into four equal blocks (i.e., 256-bit in each block). After dividing the plaintext into 256-bit blocks, it is converted into the corresponding ASCII values based on an 8-bit binary number as per Table 2. These ASCII values are then yet again converted to their real 8-bit binary number. Then, the EXOR operation is executed between the 256-bit block of the encrypted plaintext and the 256-bit block of the key. Thus, there are four EXOR operations for each 1024-bit block of plaintext. Each 2-bit binary number is then transformed into the corresponding DNA bases by using Table 3. Since there are four DNA bases, there can be a maximum of 24 combinations of the DNA bases, and the DO can choose any one out of 24 combinations to transform the encrypted binary form of plaintext into the DNA bases.

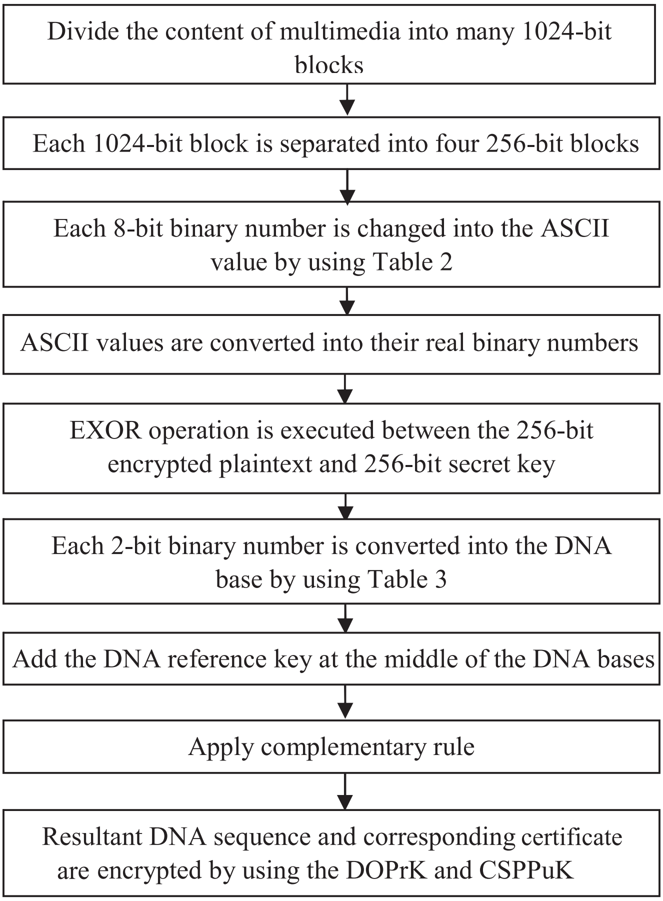
Divide the content of multimedia into many 1024-bit blocks
Each 1024-bit block is separated into four 256-bit blocks
Each 8-bit binary number is changed into the ASCII value by using Table 2
ASCII values are converted into their real binary numbers
EXOR operation is executed between the 256-bit encrypted plaintext and 256-bit secret key
Each 2-bit binary number is converted into the DNA base by using Table 3
Add the DNA reference key at the middle of the DNA bases
Apply complementary rule
Resultant DNA sequence and corresponding certificate are encrypted by using the DOPrK and CSPPuK

Fig. 3. DNA-based multimedia encryption.

To improve the security, the DO adds the DNA reference key at the middle of the achieved DNA bases. Here, the DNA reference key is the same as that used in the secret key generation phase. The DO then uses the same complementary rule used in the previous Section 4.4.1 on the resultant DNA bases.

At last, the achieved DNA sequence is encrypted by using the DOPrK, and it is again encrypted by the CSPPuK. The DO also encrypts the corresponding access right or certificate along with the multimedia data content and saves the entire encrypted information on the cloud database. The CSP cannot decrypt the encrypted multimedia file because the secret key and other credentials are shared only with the authorized user. The CSP only decrypts the encrypted multimedia file or message by using the Private Key of the CSP (CSPPrK) followed by the DOPuK and saves it on the cloud database along with the certificate. Figure 3 shows the DNA computing–based multimedia encryption process.

4.5 Multimedia Access

Only authorized users can access multimedia files from the cloud server. There are several steps for any multimedia file accessing:

Step 1: In the 1st step, the user has to register in the cloud server (CSP).

Step 2: In the 2nd step, the CSP gives an acknowledgment to the user.

Step 3: In this step, the user logs into the cloud server.

Step 4: The valid or authorized users' login request is accepted by the CSP, and the CSP sends a reply or acknowledgment to the user when a successful authorization process is completed.

Step 5: A request is then sent to the CSP by the user in the encrypted form to access the multimedia file.

Step 6: Here, the CSP finds the DOPuK of the corresponding requested multimedia file from the cloud database.

Step 7: In the 7th step, the CSP provides the DOPuK to the respective user after making it ciphertext by using the CSPPrK and USRPuK. The user retrieves the DOPuK after decrypting by using the USRPrK and CSPPuK.

Step 8: Now, the user initiates an encrypted request to the respective DO by using the USRPrK and DOPuK to give the secret key and certificate.

Step 9: In the 9th step, the DO verifies the user's authenticity from the CSP.

Step 10: In this step, the CSP sends a reply to the DO regarding authentication of the user.

Step 11: In the 11th step, the DO asks the user to give a password.

Step 12: The user provides a password to the DO.

Step 13: The DO generates a DNASK by utilizing the information of the password and user's attributes, then sends all credentials to the user encrypted by using the DOPrK and USRPuK.

Step 14: In the 14th step, the DO stores the ciphertext of the multimedia file and the corresponding certificate on the cloud server encrypted by using the DOPrK and CSPPuK. The CSP merely decrypts the information by using the CSPPrK and DOPuK.

Step 15: Then, the user presents the certificate to the CSP encrypted by using the USRPrK and CSPPuK. The encrypted message is decrypted by the CSP for getting the certificate.

Step 16: In this step, the CSP searches the requested multimedia file from the server and verifies the user's certificate with respect to the certificate provided by the DO. When the certificate is correct or valid, the rest of the processes are performed.

The CSP encrypts the user's requested multimedia file by utilizing the CSPPrK and USRPuK before sending it to the user (i.e., $\mathrm{EK}_{\mathrm{USRPuK}}[\mathrm{EK}_{\mathrm{CSPPrK}}\{\mathrm{EK}_{\mathrm{DNASK}}(\text{multimedia})\}]$). Here, $\mathrm{EK}_{\mathrm{DNASK}}(\text{multimedia})$ means the encrypted multimedia file or message by using the DNASK.

The user gets the encrypted message and decrypts it by the USRPrK, then the CSPPuK is used for decryption. At last, the decryption process is again executed by using the DNASK. Since the user has all of the credentials, which have been used during encryption by the DO, the user can effortlessly get the original data content.

5 EXAMPLE OF THE DNASK OR PASSWORD GENERATION PROCESS

In this section, an example is presented to generate the DNASK:

Step 1: In the first step, to get the secret key and certificate, a request is sent to the DO by the user.

Step 2: The DO verifies the authenticity of the user.

Step 3: Let the user give a password as 12AB!, and the DO fetches the following mentioned attributes:

NAME: SUYEL
EMPLOYEE ID: 202002140419
ADDRESS: NIT PATNA

Step 4: 12AB! SUYEL 202002140419 NIT PATNA [Password and attributes are gathered]

Step 5: $\underbrace{01}\ \underbrace{02}\ \underbrace{11}\ \underbrace{12}\ \underbrace{39}\ \dots\dots\dots\ \underbrace{26}\ \underbrace{11}\ \underbrace{30}\ \underbrace{24}\ \underbrace{11}$

[Apply the decimal encoding rule by using Table 1]

Step 6: $\underbrace{00000001}\,\underbrace{00000010}\,\underbrace{00001011}\,\underbrace{00001100}\dots\dots$

$\dots\dots\dots\dots\,\underbrace{00011010}\,\underbrace{00001011}\,\underbrace{00011110}\,\underbrace{00011000}\,\underbrace{00001011}$

[Transform into 8-bit binary values]

Step 7: $\underbrace{5}\ \underbrace{223}\ \underbrace{45}\ \underbrace{13}\ \dots\dots\dots\dots\ \underbrace{58}\ \underbrace{45}\ \underbrace{34}\ \underbrace{116}\ \underbrace{45}$

[Transform into ASCII values by using Table 2]

Step 8: $\underbrace{00000101}\,\underbrace{11011111}\,\underbrace{00101101}\,\underbrace{00001101}\dots\dots\dots$

$\dots\dots\,\underbrace{00111010}\,\underbrace{00101101}\,\underbrace{00100010}\,\underbrace{01110100}\,\underbrace{00101101}$

[Transform into 8-bit binary values]

Step 9: Delete or add bits to make it 1024-bit.

Step 10: $\underbrace{\underbrace{00000101}\,\underbrace{11011111}\,\underbrace{00101101}\,\underbrace{00001101}\dots}_{A}\ \underbrace{\dots\dots\dots\dots\dots}_{T}$

$\underbrace{\dots\dots\dots\dots\dots}_{G}\ \underbrace{\dots\,\underbrace{00111010}\,\underbrace{00101101}\,\underbrace{00100010}\,\underbrace{01110100}\,\underbrace{00101101}}_{C}$

[Assign a DNA base to each 256-bit]

Step 11: ATGCTAATCGGC [Select and add a DNA reference key (GCTAATCG)]

Step 12: TGCAGTTGACCA [Apply the complementary rule]

6 SECURITY ANALYSIS

The proposed DNA-based multimedia encryption scheme is secured against many attacks. Security analyses of DNABMS are presented in this section.

6.1 Stolen Verifier Attack

To improve the security of multimedia files, the server or the administrator does not save the secret password in the plaintext manner in many applications but instead saves the secret password in the form of a hashed password. In the stolen verifier attack, the hackers or malicious user get the hashed password [40]. In DNABMS, the DO generates the DNASK and uses it to encrypt the multimedia file. The encrypted multimedia file is again encrypted by using the DOPrK and CSPPuK before storing it on the cloud database. The DNASK is only provided to the authorized users encrypted by using the DOPrK and USRPuK, and the DO generates the secret key after verifying the authenticity of the user from the CSP. The respective user is only able to get the secret key when she or he decrypts by using the USRPrK and DOPuK. The DO does not store the DNASK anywhere, not even on the cloud server. Therefore, DNABMS can be treated as secure and protected against this attack.

6.2 Collision Attack

In a collision attack, the attacker or malicious user finds two different inputs, which give the same hash value [41]. In DNABMS, the secret key is generated based on the password provided by the user and the user's attributes like ID, voter ID number, address, and date of birth. Thus, the attributes of two users cannot be the same at all, which are also secret. Moreover, the password is based on the user's choice. In DNABMS, the DO, as well as the CSP, does not disclose any sensitive information to others. If a hacker tries to combine the details of two different inputs of two different users, then the hacker or malicious user must have all secret details of two authorized users as well as Table 1, Table 2, Table 3, the DNA reference key, and the complementary rule. Thus, getting all information about two different users is almost impossible. Therefore, the proposed DNABMS can be treated as protected against this attack.

6.3 Phishing Attack

In a phishing attack, the malicious user or hacker gets the authorized user's sensitive information, such as user ID, voter ID number, and password, and many more [42]. Then, the attacker utilizes this authorized information for getting any unauthorized service from the cloud server. In DNABMS, the CSP assembles all of the details or information of the user after getting a registration request from the user. Then, the CSP provides the public and private key pair through the secure socket layer. The CSP also follows the same approach for registering the DOs. When the user sends an access request for multimedia or data, the service provider only gives the DOPuK of the respective DO to the user in the ciphertext form. The DO only provides the DNASK after confirming the user's authenticity. Both the DO and CSP do not expose any sensitive or confidential information of themselves to even an authorized user. Thus, the hackers or malicious users cannot get the information of the authorized users, and DNABMS can resist the phishing attack.

6.4 Password Guessing Attack

In the password guessing attack, the hackers or malicious users try all possible combinations of the password to get a valid password [43]. In the proposed scheme, the DO creates the DNASK based on the password and user's attributes by using Table 1, Table 2, the DNA reference key, and the complementary rule. The DO also uses Table 2, Table 3, the DNA reference key, and the complementary rule during the multimedia encryption phase. Since the DNASK is generated by using the user's secret information, it is challenging for the hackers or unauthorized users to get all of the secret information. If a hacker gets all of the secret information of an authorized user in a worst case, she or he also must have Table 1, Table 2, the DNA reference key, and the complementary rule to retrieve the secret key. The DO shares these credentials only with the authorized or valid user in the encrypted structure by using the DOPrK and USRPuK, and only the respective valid user can decrypt it by using his or her USRPrK and DOPuK, and guessing all of this information is almost impossible for any attacker or malicious user. Therefore, DNABMS can be treated as secure against this password guessing attack.

6.5 Masquerade Attack

In a masquerade attack, an unauthorized user uses any fake information or identity for getting an unauthorized access [44]. In the proposed scheme, all users need to be registered at the CSP, and they need to log in after a successful registration process for accessing any service. Thus, unauthorized or malicious users cannot log into the cloud server by using a fake identity, and they cannot access any services from the cloud server without logging into the cloud server. In a worst case, if a hacker logs into the cloud server and requests to get any multimedia file, the CSP provides the DOPuK for getting the secret key and certificate from the respective DO. The DO confirms the user's authenticity from the CSP before generating and sending the DNASK and certificate. If the hacker uses the fake identity for accessing any multimedia file, then, his or her authenticity is not verified from the CSP's end, and the DO rejects the request to send the secret key and certificate. Thus, an attacker or hacker cannot get any unauthorized access from the cloud server by using the fake identity, and DNABMS can be treated as secure against this attack. Table 4 shows comparisons among many existing schemes and the proposed scheme.

Table 4. Comparisons of the Proposed Scheme (DNABMS) with the Existing Schemes

Relevancy	Existing Schemes	Proposed Scheme (DNABMS)
Attacks	The existing schemes, namely DNASC [38], CBIE [34], and RDHS [35], do not support randomness during assigning of the DNA bases. Thus, these schemes are not secured against stolen verifier attack, collision attack, masquerade attack, phishing attack, and password guessing attack.	DNABMS defines randomness in the assignment of the DNA bases in both the DNASK generation phase and DNA-based multimedia encryption phase that support security against stolen verifier attack, collision attack, masquerade attack, phishing attack, and password guessing attack.
Information exchanged	The secret key and DNA chip are exchanged between the sender and receiver of multimedia file content in DNASC [38]. Only the secret key is exchanged in CBIE [34]. In RDHS [35], the DNA coding rule, DNA EXOR rule, and secret key are exchanged between the sender and receiver.	The decimal encoding rule (Table 1), ASCII values of 8-bit binary number (Table 2), DNA base of 2-bit binary number (Table 3), DNA reference key, complementary rule, and a long 1024-bit DNASK are exchanged in DNABMS between the DO and user.
Overhead	In DNASC [38], CBIE [34], and RDHS [35], overhead is less.	In DNABMS, information is exchanged and needs to be maintained. Thus, overhead is comparatively high.
Key size	In CBIE [34], a 120-bit secret key is used to encrypt the multimedia file. However, a specific key size is not maintained in both DNASC [38] and RDHS [35].	In DNABMS, a long 1024-bit DNASK is used to encrypt the multimedia data content. This secret key defines randomness during assignment of the DNA bases.

7 PERFORMANCE ANALYSIS

Performance analysis of the proposed DNABMS is presented in this section in detail.

7.1 Simulation Environment

For evaluating the performance of DNABMS, a cloud simulation environment has been built by using CloudSim 3.0.3 [45]. There are few components in the CloudSim toolkit, namely the Cloud Information Service (CIS), datacenter, host, Virtual Machine (VM), broker, cloudlet, and VM Manager.

In CloudSim, at first, the datacenter sends a request to the CIS to register his or her resources. Each datacenter has several hosts, and each host consists of several VMs having key characteristics, such as RAM, processing elements, and bandwidth. The brokers have numerous cloudlets or tasks, which are waiting for executions. The brokers send request to the CIS to give the details of the available resources for executing the cloudlets. Then, the characteristics or attributes of the datacenter are sent to the broker. Now, VMs are created for the broker or user after getting the demand from the broker. If VMs are created, the datacenter sends an acknowledgment to the broker. After receiving the acknowledgment, the broker submits the cloudlet to the datacenter broker module to execute the task. Then, the datacenter allocates VMs for cloudlet, and an acknowledgment is sent to the broker when the cloudlet is executed.

CloudSim was installed on a Dell OPTIPLEX 9020 SFF desktop consisting of 8 GB of RAM, 1 TB of storage capacity, a 3.40-GHz Intel Core i5 processor, and a Windows 10 operating system. Apache Common Math [46] was mounted with the CloudSim, and Java 8 [47] was furthermore installed on the Dell computer system.

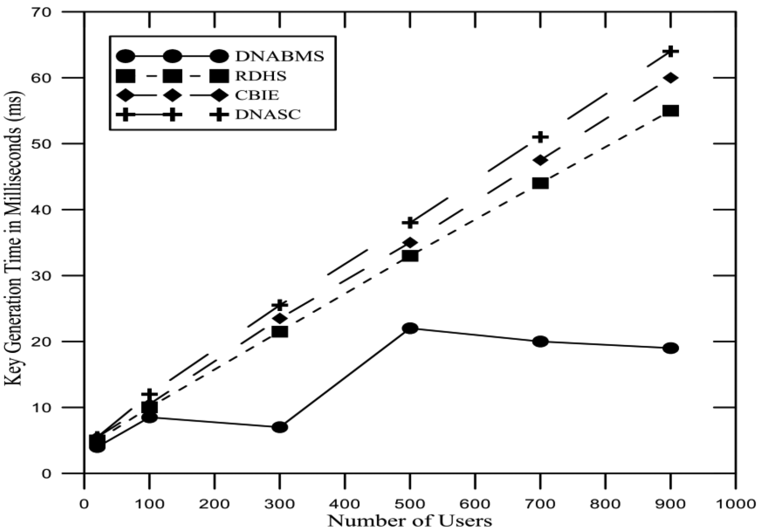

Fig. 4. Number of users vs. key generation time.

In CloudSim, 10 datacenters were considered to create a heterogeneous cloud environment that consists of 2,000 physical nodes, 1 GB/s of network bandwidth, and storage capacity of 4 GB. The following four types of VM were considered, and all of these VMs have 1 GB/s bandwidth:

1) 500 MIPS, 512 MB
2) 1,000 MIPS, 1 GB
3) 1,500 MIPS, 1.5 GB
4) 2,000 MIPS, 2 GB

7.2 Results and Discussions

Many experiments were executed for evaluating the performance of the proposed scheme with respect to the existing schemes, namely the CBIE scheme [34], RDHS [35], and DNASC [38]. All of these existing schemes focus to improve security by using DNA computing. DNA bases are used in all of these schemes. These three schemes are almost analogous to the proposed scheme in terms of working principle. Therefore, they are treated as the baseline schemes for comparisons. To calculate the secret key generation time, secret key retrieval time, and encryption and decryption times of multimedia files, 50 experiments were executed in different situations to get each result. Finally, average values were considered from those 50 results. For the simulation, datasets were collected from the publicly available CityPulse Dataset Collection [48], which provides different types of datasets, namely pollution, weather, and road traffic, among others. Here, the vehicle traffic dataset (size: 3 GB) was used for the experiments. In this section, results and discussions of DNABMS are presented.

Figure 4 depicts the results of the first experiment to calculate the key generation time. In DNABMS, all of the users have to send the requests to the respective DO for receiving the secret key and the user's access right when they wish to get any multimedia file. The DO asks the users to give a password and collects a set of secret user attributes, and uses Table 1, Table 2, the DNA reference key, and the complementary rule for generating the DNASK. Therefore, the proposed scheme takes a little bit of time to generate the DNASK. In DNABMS, once the secret key is provided to the users, they can use it in the future. Thus, for the existing or old users who already

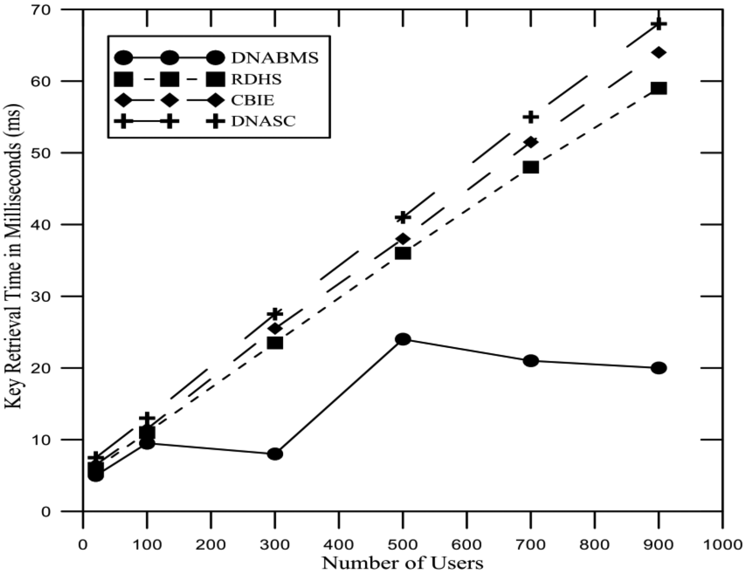

Fig. 5. Number of users vs. key retrieval time.

have the DNASK, the key generation time is reduced. Thus, DNABMS produces a zigzag curve. However, all existing schemes, namely DNASC, CBIE, and RDHS, are quite complex compared to the proposed scheme, and all users or customers must get the secret key in each accessing time of multimedia for further processes. Therefore, linear increasing curves are experienced in the case of all of these existing schemes.

The second experiment validates key retrieval time. It can be easily recognized from Figure 5 that DNABMS gives much better results for retrieving the secret key by the user. In the proposed scheme, the old or existing users do not need to get the DNASK from the DO if they have already collected the secret key. Thus, they do not retrieve the DNASK for the subsequent time of multimedia access. However, in DNABMS, all users or customers have to get the DNASK from the DO at the time of first multimedia access. Therefore, they have to retrieve the key during the first access, and the key retrieving time is increased for the proposed scheme. In the existing schemes, namely DNASC, CBIE, and RDHS, whenever the users want to get any data from the cloud database, every time they have to get the secret key from the DO and have to retrieve it. Thus, the key retrieving time is high for the existing schemes, and it produces linearly increasing curves.

In Figure 6, it can be easily seen that the proposed scheme produces much better results than the existing schemes to encrypt the multimedia file content in the cloud computing environment. In the proposed scheme, after generating the DNASK, the DO uses Table 2, Table 3, the complementary rule, and the other credentials to encrypt the multimedia file, which is a bit time consuming. However, in DNASC, vast parallelism computation is required to encrypt any multimedia content. In addition, billions of DNA sequences are also hybridized in DNASC for multimedia file encryption, which also increases the time. In CBIE, chaotic maps and genetic algorithm are used to encrypt confidential data. The iterative algorithm of CBIE increases the time of multimedia file encryption since it runs until it finds the optimum results. The classical algorithm of reversible data protecting or data hiding approach is used in RDHS, where using the histogram modification technique increases the time of data encryption compared to the proposed scheme. Thus, in both CBIE and RDHS, the encryption time of the multimedia file is more than DNABMS.

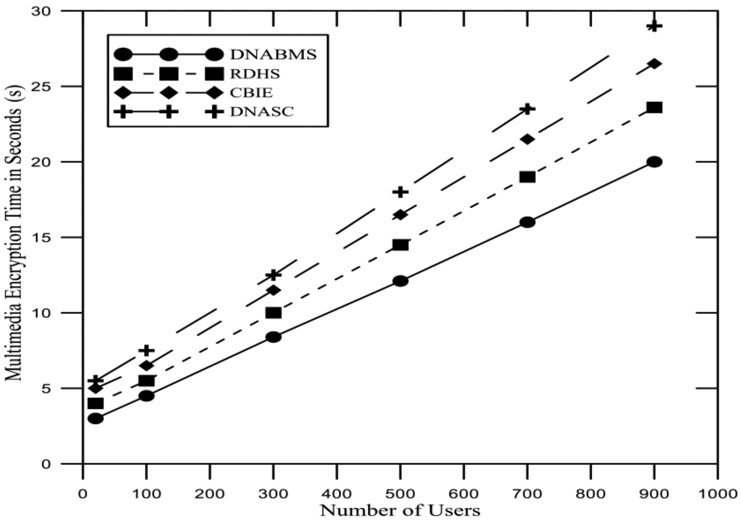

Fig. 6. Number of users vs. encryption time.

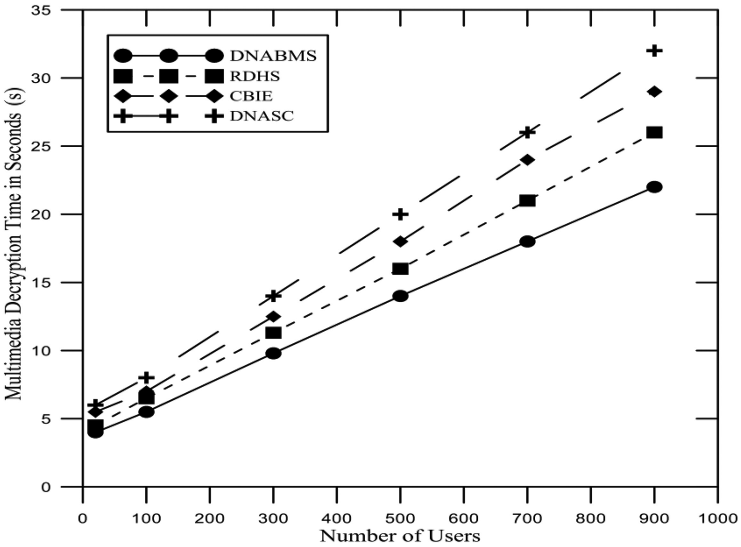

Fig. 7. Number of users vs. decryption time.

The last experiment shows the results for multimedia decryption time. In DNABMS, the DO provides all the credentials to the users at the time of transferring the secret key. The users can easily decrypt the multimedia file or information by using these credentials. In DNABMS, there are no complex operations and four EXOR operations are required for multimedia decryption, which decreases the time of decryption. However, in the existing schemes, namely DNASC, CBIE and RDHS, many complex operations and many EXOR operations are required to decrypt multimedia file, which increases the time as shown in Figure 7. Thus, the decryption time of multimedia file is high in these existing schemes compared to the proposed scheme.

8 CONCLUSION AND FUTURE WORK

The security issues of multimedia files in the cloud computing environment are increasing day by day. A long 1024-bit DNASK or password generation technique is proposed in this article by using a password and the secret attributes of the user. The same key is used in the novel multimedia encryption technique, which is based on DNA computing. The secret key is secured as it labels the randomness during assignment of the DNA bases, and the confidential credentials are only shared with the authorized users. The proposed scheme can resist many security attacks in the cloud environment. Here, the DOs can be online to provide the DNASK and other credentials, and they can go offline after delivering all of the credentials. Thus, system overhead is decreased. Experimental results prove that the proposed technique is more efficient and effective than the other well-known existing schemes. There is a considerable scope to improve the authentication process of the cloud computing environment.

REFERENCES

[1] A. Huth and J. Chebula. 2011. *The Basics of Cloud Computing*. Carnegie Mellon University, Pittsburgh, PA.

[2] S. Namasudra, S. Nath, and A. Majumder. 2014. Profile based access control model in cloud computing environment. In *Proceedings of the International Conference on Green Computing, Communication, and Electrical Engineering*. IEEE, Los Alamitos, CA, 1–5.

[3] S. Li, G. Wang, and J. Yang. 2019. Survey on cloud model based similarity measure of uncertain concepts. *CAAI Transactions on Intelligence Technology* 4, 4 (2019), 223–230.

[4] S. Namasudra, D. Devi, S. Kadry, R. Sundarasekar, and A. Shanthini. 2020. Towards DNA based data security in the cloud computing environment. *Computer Communications* 151 (2020), 539–547.

[5] S. Namasudra. 2019. An improved attribute-based encryption technique towards the data security in cloud computing. *Concurrency and Computation: Practice and Exercise* 31, 3 (2019), e4364.

[6] M. Sarkar, K. Saha, S. Namasudra, and P. Roy. 2015. An efficient and time saving web service based Android application. *SSRG International Journal of Computer Science and Engineering* 2, 8 (2015), 18–21.

[7] S. Namasudra, P. Roy, and B. Balamurugan. 2017. Cloud computing: Fundamentals and research issues. In *Proceedings of the 2nd International Conference on Recent Trends and Challenges in Computational Models*. IEEE, Los Alamitos, CA.

[8] R. M. Alguliyev, R. M. Aliguliyev, and L. V. Sukhostat. 2020. Efficient algorithm for big data clustering on single machine. *CAAI Transactions on Intelligence Technology* 5, 1 (2020), 9–14.

[9] S. Namasudra. 2018. Taxonomy of DNA-based security models. In *Advances of DNA Computing in Cryptography*, S. Namasudra and G. C. Deka (Eds.). Taylor & Francis, 53–68.

[10] S. Namasudra, P. Roy, B. Balamurugan, and P. Vijayakumar. 2017. Data accessing based on the popularity value for cloud computing. In *Proceedings of the International Conference on Innovations in Information, Embedded, and Communications Systems (ICIIECS'17)*. IEEE, Los Alamitos, CA.

[11] S. Namasudra, G. C. Deka, P. Johri, M. Hosseinpour, and A. H. Gandomi. 2020. The revolution of blockchain: State-of-the-art and research challenges. *Archives of Computational Methods in Engineering*. In Press.

[12] D. Devi, S. Namasudra, and S. Kadry. 2020. A boosting-aided adaptive cluster-based undersampling approach for treatment of class imbalance problem. *International Journal of Data Warehousing and Mining* 16, 3 (2020), 1–27.

[13] S. Namasudra and P. Roy. 2016. Secure and efficient data access control in cloud computing environment: A survey. *Multiagent and Grid Systems* 12, 2 (2016), 69–90.

[14] S. Namasudra and G. C. Deka. 2018. *Advances of DNA Computing in Cryptography*. Taylor & Francis.

[15] S. Namasudra, D. Devi, S. Choudhary, R. Patan, and S. Kallam. 2018. Security, privacy, trust, and anonymity. In *Advances of DNA Computing in Cryptography*, S. Namasudra and G. C. Deka (Eds.). Taylor & Francis, 153–166.

[16] S. Namasudra and P. Roy. 2017. Time saving protocol for data accessing in cloud computing. *IET Communications* 11, 10 (2017), 1558–1565.

[17] S. Namasudra. 2020. Data access control in the cloud computing environment for bioinformatics. *International Journal of Applied Research in Bioinformatics*. In Press.

[18] G. C. Deka, M. Kathing, and D. P. Kumar. 2013. Library automation in cloud. In *Proceedings of the International Conference on Computational Intelligence and Communication Networks*.

[19] S. Namasudra and G. C. Deka. 2018. Introduction of DNA computing in cryptography. In *Advances of DNA Computing in Cryptography*, S. Namasudra and G. C. Deka (Eds.). Taylor & Francis, 27–34.

[20] S. Namasudra and P. Roy. 2018. PpBAC: Popularity based access control model for cloud computing. *Journal of Organizational and End User Computing* 30, 4 (2018), 14–31.

[21] S. Namasudra, G. C. Deka, and R. Bali. 2018. Applications and future trends of DNA computing. In *Advances of DNA Computing in Cryptography*, S. Namasudra and G. C. Deka (Eds.). Taylor & Francis, 181–192.

[22] X. Zhao, R. Li, and X. Zuo. 2019. Advances on QoS-aware web service selection and composition with nature-inspired computing. *CAAI Transactions on Intelligence Technology* 4, 3 (2019), 159–174.

[23] S. Namasudra and P. Roy. 2016. A new table based protocol for data accessing in cloud computing. *Journal of Information Science and Engineering* 33, 3 (2016), 585–609.

[24] A. Shamir. 1985. Identity-based cryptosystems and signature schemes. In *Advances in Cryptology*, G. R. Blakley and D. Chaum (Eds.). Springer, 47–53.

[25] D. F. Ferraiolo and D. R. Kuhn. 1992. Role-based access controls. In *Proceedings of the 15th National Computer Security Conference.* 554–563.

[26] V. Goyal, O. Pandey, A. Sahai, and B. Waters. 2006. Attribute-based encryption for fine-grained access control of encrypted data. In *Proceedings of the 13th ACM Conference on Computer and Communications Security.* 89–98.

[27] J. Bethencourt, A. Sahai, and B. Waters. 2007. Ciphertext-policy attribute based encryption. In *Proceedings of the IEEE Symposium on Security and Privacy.* 321–334.

[28] R. Mian, P. Martin, and J. L. V. Poletti. 2013. Provisioning data analytic workloads in a cloud. *Future Generation Computer Systems* 29, 6 (2013), 1452–1458.

[29] S. Ajgaonkar, H. Indalkar, and J. Jeswani. 2015. Activity based access control model for cloud computing. *International Journal of Current Engineering and Technology* 5, 2 (2015), 708–713.

[30] S. Raghavendra, K. Meghanaa, P. A. Doddabasappaa, C. M. Geetaa, Rajkumar Buyyab, K. R. Venugopala, S. S. Iyengarc, and L. M. Patnaik. 2016. Index generation and secure multi-user access control over an encrypted cloud data. *Procedia Computer Science* 89 (2016), 293–300.

[31] G. Cui, Y. Liu, and X. Zhang. 2006. New direction of data storage: DNA molecular storage technology. *Computer Engineering and Application* 42, 26 (2006), 29–32.

[32] C. T. Clelland, V. Risca, and C. Bancroft. 1999. Hiding messages in DNA microdots. *Nature* 399, 6736 (1999), 533–534.

[33] K. Tanaka, A. Okamoto, and I. Saito. 2005. Public-key system using DNA as a one-way function for key distribution. *Biosystems* 81, 1 (2005), 25–29.

[34] R. Enayatifar, A. H. Abdullah, and I. F. Isnin. 2014. Chaos-based image encryption using a hybrid genetic algorithm and a DNA sequence. *Optics and Lasers in Engineering* 56 (2014), 83–93.

[35] B. Wang, Y. Xie, S. Zhou, C. Zhou, and X. Zheng. 2017. Reversible data hiding based on DNA computing. *Computational Intelligence and Neuroscience* 2017 (2017), Article 7276084.

[36] Y. Wang, Q. Han, G. Cui, and J. Sun. 2019. Hiding messages based on DNA sequence and recombinant DNA technique. *IEEE Transactions on Nanotechnology* 18 (2019), 299–307.

[37] A. Leier, C. Richter, W. Banzhaf, and H. Rauhe. 2000. Cryptography with DNA binary strands. *Biosystems* 57, 1 (2000), 13–22.

[38] L. MingXin, L. XueJia, X. GuoZhen, and Q. Lei. 2007. Symmetric-key cryptosystem with DNA technology. *Science in China Series F: Information Sciences* 50, 3 (2007), 324–333.

[39] S. Namasudra. 2018. Cloud computing: A new era. *Journal of Fundamental and Applied Sciences* 10, 2 (2018), 113–135.

[40] R. Isawa and M. Morii. 2011. One-time password authentication scheme to solve stolen verifier problem. In *Proceedings of the Forum on Information Technology.* 225–228.

[41] K. Schramm, T. J. Wollinger, and C. Paar. 2003. A new class of collision attacks and its application to DES. In *Proceedings of the Fast Software Encryption Conference.* 206–222.

[42] T. N. Jagatic, N. A. Johnson, M. Jakobsson, and F. Menczer. 2007. Social phishing. *Communications of the ACM* 50, 10 (2007), 94–100.

[43] L. Gong. 1995. Optimal authentication protocols resistant to password guessing attacks. In *Proceedings of the 8th IEEE Computer Security Foundations Workshop.* IEEE, Los Alamitos, CA, 24–29.

[44] M. B. Salem and S. J. Stolfo. 2011. Decoy document deployment for effective masquerade attack detection. In *Proceedings of the International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment.* ACM, New York, NY, 35–54.

[45] R. N. Calheiros, R. Ranjan, and A. Beloglazov. 2011. CloudSim: A toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms. *Software: Practice and Experience* 41, 1 (2011), 23–50.

[46] Apache Commons Math. 2019. Home Page. Retrieved September 14, 2019 from http://commons.apache.org/proper/commons-math/download_math.cgi.

[47] Java. 2019. Home Page. Retrieved September 14, 2019 from http://java.com/en/download/index.jsp.

[48] CityPulse Dataset Collection. 2019. Home Page. Retrieved September 15, 2020 from http://iot.ee.surrey.ac.uk:8080/datasets.html.

Received January 2020; revised March 2020; accepted April 2020