Journal of Network and Computer Applications 172 (2020) 102835

Contents lists available at ScienceDirect

Journal of Network and Computer Applications

journal homepage: www.elsevier.com/locate/jnca

DNA computing and table based data accessing in the cloud environment

Suyel Namasudra [a,*], Suraj Sharma [b], Ganesh Chandra Deka [c], Pascal Lorenz [d]

[a] Department of Computer Science and Engineering, National Institute of Technology Patna, Bihar, India
[b] Department of Computer Science and Engineering, IIIT Bhubaneswar, Odisha, India
[c] Ministry of Skill Development and Entrepreneurship, Government of India, New Delhi, India
[d] University of Haute-Alsace, Colmar, France

ARTICLE INFO

Keywords:
Cloud service provider
DNA computing
DNA-EXOR rule
ASCII value
CloudSim.

ABSTRACT

The attractive benefits of cloud computing environment motivate Information Technology (IT) companies to move their IT infrastructure on the cloud server. In a cloud environment, companies must depend on the untrusted Cloud Service Provider (CSP), which raises data security issues due to the presence of numerous hackers and malicious users. Apart from data security, high searching time of data owner, high data accessing time and high system overheard are other issues during data accessing from the cloud environment. To solve all these issues, in this paper, a novel DNA based fast and secure data access control model has been introduced for the cloud environment. Here, the cloud service provider maintains a table for fast and efficient data accessing. In this proposed scheme, a long 1024-bit Deoxyribonucleic Acid (DNA) based password or secret key is used to encrypt users' confidential or personal data. Experimental results along with theoretical analysis prove the efficiency of the proposed model over the existing models.

1. Introduction

Nowadays, cloud computing is one of the advanced fields in IT industries. There are numerous interconnected parallel and distributed computers in a cloud environment, which are combined by different technologies, namely utility computing, network system, virtualization and distributed processing. These technologies allow various services like a server, pay-per-use, network, etc. (Huth and Chebula, 2011). There are mainly three entities in a cloud computing environment: (i) Cloud Service Provider (ii) Data Owner (DO), and (iii) user. The CSP plays the role of a central authority or overall administrator of any type of cloud computing environment. DOs are responsible for storing their normal or confidential or any type of data in the cloud environment, whereas users are the authorized entities, who can access data or file (Namasudra et al., 2014; Li et al., 2019; Namasudra and Roy, 2017a). During data accessfrom the cloud server, there are several issues like high time for searching a DO, high time for data accessing and high system overhead. When the user sends a data access request to the cloud server, the CSP may check the entire database to find one DO for providing the Public Key of the DO (PCKDO) to the authorized user. So, in many cases, the searching time is high. The user can send a request to the DO by using the corresponding PCKDO for getting the secret key and access right (certificate). If the CSP takes much time to find the PCKDO, the user must wait for a long time, and as result, the accessing time of data can also be high. Therefore, the customer or user needs to pay more for using the cloud service. High system overhead is another issue in the existing scheme (Gao et al., 2012) as the DO must be always online during the entire data accessing or communication process.

Many schemes are already developed to solve these problems (Namasudra, 2017, 2020; Sarkar et al., 2015; Namasudra et al., 2017a, 2017b, 2018a, 2018b, 2018c, 2018d, 2020a, 2020b, 2020c, 2020d; Assis and Bittencourt, 2020; Ma et al., 2020; Fu et al., 2018; Sajid et al., 2016; Alguliyev et al., 2020; Namasudra and Roy, 2016, 2017b, 2018; Thaseen et al., 2020; Wang et al., 2019a; Kumar et al., 2020; Namasudra and Deka, 2018; Hossain and Muhammad, 2016; Lojka et al., 2016; Tripura and Roy, 2017; Tripura et al., 2018, 2020; Zhao et al., 2019; Devi et al., 2020). Role Based Access Control (RBAC) model has been proposed by Ferraiolo and Kuhn (1992). RBAC is based on the role of the job, where data accessing by the user is mainly controlled based on the job role. However, this scheme is not much secured to protect user's data. Attributes of a ciphertext are utilized in Key Policy based Attribute-Based Encryption (KPABE) (Goyal et al., 2006). In KPABE, the DOs must depend on the key generator and they are not allowed to control the access policies. Activity Based Access Control (AtBAC)

* Corresponding author.
E-mail address: suyelnamasudra@gmail.com (S. Namasudra).

https://doi.org/10.1016/j.jnca.2020.102835
Received 25 April 2020; Received in revised form 30 July 2020; Accepted 12 September 2020
Available online 24 September 2020

(Ajgaonkar et al., 2015) model suggested that the user's Access Right (AR) for any data in the respective organization can be assigned on the basis of the designation of the user. But, AtBAC increases the system overhead. Another model was introduced based on the substring index generation process known as Index Generation Based Access Control (IGBAC) model (Raghavendra et al., 2016). Here, the key generation time is high. A novel technique for a multitenant cloud computing environment has been proposed by Almutairi et al. (2018), where the notion of sensitivity in the cloud data centre has been introduced. However, this scheme does not support fast data accessing. So, users must pay more for using cloud services. All these existing schemes experience the high searching time of DO, high data accessing time, high system overhead and data security issue.

Data security of any cloud computing environment is another critical challenge due to the presence of many attackers, who always attempt to hack confidential data of the DO. Sometimes, they replace DO's confidential data. So, the CSP must provide a strong security technique for protecting confidential data against unauthorized accesses. Nowadays, DNA computing is used in many fields to improve data security. In DNA computing, DNA sequence, biochemistry and hardware are used for encoding the genetic information in a computer. DNA computing is used in DNA cryptography for data encryption, so that the unauthorized users and attackers are unable to read the data content. 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). These bases are combined to form a DNA sequence. Adleman (1994) first used DNA for computation. When concerning data security in a cloud computing environment, DNA computing has got wide attention because of the complexity in its structure. A novel Public Key System by using DNA (PKSDNA) was proposed by Tanaka et al. (2005) in which one-way function was applied. But, the drawback of PKSDNA is that it increases the data accessing time and this scheme is not secured against the password guessing attack. In 2016, a novel Probabilistic Secret Sharing Scheme (PSSS) (Tuncer and Avci, 2016) has been proposed by using DNA-Exclusive OR (DNA-EXOR) operation. PSSS is vulnerable against the side channel attack. Wang et al. (2017) have suggested a DNA based reversible data hiding scheme for users' confidential or sensitive data. This scheme does not offer a strong security technique for protecting the user's data. A novel encryption scheme was proposed by Wang et al. (2018) using coupled map lattice and DNA sequence. However, the encryption process of this scheme takes much time.

A novel data Access Control Model (ACM) has been proposed in this paper, namely DNA based Fast and Secure (DNAFS) ACM for the cloud computing environment. DNAFS can solve the problems of the existing schemes. In DNAFS, the CSP manages a temporary table based on the file or data size and data type of the DO. Here, whenever a user or customer sends a data access request, the CSP initiates a query in the table based on the requested size and type of the data to find the DO. Therefore, the CSP does not search the whole database to find the DO and can easily provide the PCKDO to the user in less time. Thus, the searching time of PCKDO can be decreased. The accessing time of data can be repeatedly decreased in DNAFS since the searching of PCKDO is minimized and the user can pay less. To improve data security, a novel data encryption technique has been used. In the proposed data encryption technique, at first, the DO generates a long 1024-bit DNA based Secret Key (DNASK). The key is randomly generated by using the user's characteristics and other secret information. Then, the secret key is used to encrypt the data by using a novel DNA computing based data encryption technique. The DO shares the DNA based password and all the credentials only with the authorized users after encryption. Thus, data security can be improved in the proposed encryption technique. This proposed work has the below mentioned contributions:

1) A novel fast and secure data access control model (DNAFS) has been proposed in this paper. In DNAFS, the CSP maintains a table for fast data accessing. The searching time of DO and the accessing time of data can be minimized using this table.
2) A DNA computing based data encryption technique has been proposed for improving data security in which a long 1024-bit randomly generated DNASK is used.
3) In this paper, security analysis of DNAFS and experimental results have been presented.

The rest of the paper consists of several parts. Section 2 discusses related works. Background studies of the proposed model are presented in section 3. In section 4, the detailed discussion of the proposed model has been presented. Section 5 and section 6 deal with the security analysis and performance analysis of the proposed scheme, respectively. Section 7 presents the future work directions to enhance the proposed scheme. At last, the paper is concluded in section 8.

2. Related works

In this section, many access control models and DNA based encryption schemes are briefly discussed.

2.1. Access control models

Many researchers have proposed many ACMs for the cloud environment for efficient and fast data accessing.

In RBAC (Ferraiolo and Kuhn, 1992), every user or customer gets a separate role during data access. Here, the role is used for assigning access rights, and users cannot give access rights to other users. In RBAC, users have to satisfy few rules: rule authorization, rule assignment and transaction authorization to access any data. Transaction authorization guarantees that only the valid users are able to execute a transaction by confirming both the rule assignment and rule authorization. In RBAC, data are encrypted only once that may increase security issues.

In 2012, to get fine grained, flexible and scalable access control, Wan et al. (2012) have proposed a Hierarchical Attribute Set Based Encryption (HASBE) scheme that extends the concept of Attribute Set Based Encryption (ASBE) with a hierarchical structure. In this scheme, the user's access rights are expired after a predefined time. This scheme increases overhead because the workload of higher-level trusted authority is transferred to lower-level domain authority.

Ajgaonkar et al. (2015) have used the concept of RBAC to propose an activity-based ACM. This scheme supports accessing of any data by the designation of the user in a company s/he works. Here, the data decryption key is encrypted by using Rivest-Shamir-Adleman (RSA) algorithm to improve data security. This model comprises of some components: roles, tasks, users, permissions, session, user assignment, security clearance, security level and data. Therefore, the chances to increase the system overhead are high.

In 2017, Cross Tenant Access Control (CTAC) scheme was proposed by Alam et al. (2017). In this scheme, the cloud service provider plays the activity of the trusted third party. CTAC supports sharing of the resources between two different tenants. There are four algorithms in CTAC for delegation mechanism and permission activation between tenants: activation, forward revocation, delegation and backward revocation. CTAC does not support fast data accessing. Table 1 summarizes the key findings of the access control models.

In a cloud computing environment, data leakage issue is increasing because of sharing resources. To solve this issue, a novel mechanism was proposed by Almutairi et al. (2018) for the multitenant cloud environment, namely Notion Based ACM (NBACM). NBACM introduces the notion of sensitivity at the data centre's end. NBACM is based on RBAC, and it is designed to minimize the risk of assigning the Virtual Machine (VM) for each task. However, this scheme does not support fast data accessing.

Modified Hierarchical Attribute-Based Encryption (MHABE) (Xie et al., 2019) model has been proposed by using attribute-based

Table 1
Key findings of access control models.

ACM	Key findings	Disadvantages
RBAC (Ferraiolo and Kuhn, 1992)	RBAC is based on the role of the users.	In RBAC, data are not secured.
HASBE (Wan et al., 2012)	Access rights are expired after a predefined time in this scheme.	HASBE increases the system overhead.
AtBAC (Ajgaonkar et al., 2015)	AtBAC is based on the designation of the user in a company.	Security is a major issue of AtBAC.
CTAC (Alam et al., 2017)	It allows resource sharing between tenants.	CTAC is slow for resource sharing.
NBACM (Almutairi et al., 2018)	NBACM solves data leakage issue.	NBACM takes high time to provide data.
MHABE (Xie et al., 2019)	MHABE has been proposed for data processing, data accessing and data storing.	In MHABE, the searching time of DO is high.

encryption and Hierarchical Identity-Based Encryption (HIBE). MHABE is designed for focusing data processing, data accessing and data storing in the multi-user cloud environment. Nevertheless, the searching time of DO is high in MHABE, so users need to pay more for using cloud services.

2.2. DNA based encryption schemes

Nowadays, DNA computing is used to encrypt users' confidential or sensitive data.

In 1999, Hiding Messages in Microdots (HMM) scheme was proposed by Clelland et al. (1999). HMM is the first method that uses DNA computing for data hiding. Here, microdot can be referred to as the procedure to conceal the messages or data known as steganography. In HMM, a DNA encoded message is first hidden within a massive complex human genomic DNA, and then, the encoded message is further concealed by restricting this sample to a microdot. However, this scheme is very slow.

A public key cryptosystem has been proposed in 2005 by using a one-way function and DNA computing (Tanaka et al., 2005). This scheme was developed to solve the key distribution problem between the receiver and the sender. Here, messages are hidden in the dummy DNA sequences and restored by using the Polymerase Chain Reaction (PCR) amplification and sequencing. The data accessing time is increased in PKSDNA and this scheme is insecure against the password guessing attack.

A novel probabilistic secret sharing scheme has been proposed by Tuncer and Avci (2016) based on the secret sharing by using DNA-EXOR operation. In PSSS, a truth table is used based on DNA-EXOR, and data are divided into three secret shares, where each part of the data is embedded into the red, green and blue channels of the cover image, respectively. PSSS is not secured against the side channel attack since if an attacker gets the truth table, s/he can retrieve the secret message.

In 2017, Sheela et al. (2017) have proposed a Novel Cryptosystem based on DNA (NCDNA) by using hybrid chaotic shift transform, chaotic maps and DNA encoding rules. They have used chaotic maps as a standard map and Two-Dimension Modified Henon Map (2D-MHM), which has special chaotic behavior for a wide range of control parameters. To improve security, DNA encoding rule is used in this scheme. However, this scheme is not secured against the insider attack.

Wang et al. (2018) have proposed a novel encryption technique to encrypt an image, namely Coupled Map Lattice based ACM (CML). It is based on coupled map lattice and DNA sequences. Here, the given keys and SHA-256 algorithm are combined to generate initial values of coupled map lattice and other parameters. Then, the original image is encoded to a DNA matrix, and this DNA matrix is scrambled by using a cyclic shift. Finally, the scrambled DNA matrix is diffused by using the DNA rules and resultant DNA sequence of the coupled map lattice system. However, this encryption process is slow because of the presence of many operations. Table 2 summarizes the key findings of the DNA based encryption schemes.

A Double Layer Data Hiding (DLDH) scheme has been proposed by Wang et al. (2019b) based on the recombinant DNA technique and DNA sequence. In this scheme, a message or data is hided in a fake DNA sequence, which is generated by using the mapping rule, 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 studies of the proposed model

System model of the proposed scheme, system requirements and deign goals have been dealt in this section.

3.1. System model

The proposed DNAFS is composed of three entities:

1) *Cloud service provider:* This entity has the overall power of any cloud environment. The CSP uses numerous servers to provide cloud infrastructure and services to both the user and DO (Namasudra, 2018).
2) *Data owner:* The DO is one special kind of user, who stores any type of data (confidential or normal) on the cloud server in the encrypted form. They depend on the service provider for data security.
3) *User:* Users are the customers or entities, who wish to get any data or file or any kind of service from a cloud environment. They are the authorized entities, who also depend on the CSP for IT infrastructure or any kind of service. Fig. 1 highlights the system model of DNAFS.

3.2. System requirements

There are few system requirements as mentioned below:

1) *Fine-grained access control:* The CSP must assure that only the authorized users can be able to access data from the cloud server and they cannot access data by their wishes. The DO has to assign access

Table 2
Key findings of DNA based encryption schemes.

Scheme	Key findings	Disadvantages
HMM (Clelland et al., 1999)	HMM is the first scheme that uses DNA computing to hide any data.	HMM is very slow.
PKSDNA (Tanaka et al., 2005)	PKSDNA mainly solves key distribution problem.	This scheme takes high time to provide data.
PSSS (Tuncer and Avci, 2016)	In PSSS, a DNA based truth table is used.	PSSS is not secured against side channel attack.
NCDNA (Sheela et al., 2017)	DNA encoding rule is used in NCDNA.	NCDNA is not secured against insider attack.
CML (Wang et al., 2018)	It is based on coupled map lattice and DNA sequences.	CML is slow.
DLDH (Wang et al., 2019b)	DLDH hides data in a fake DNA sequence.	This scheme increases the system overhead.

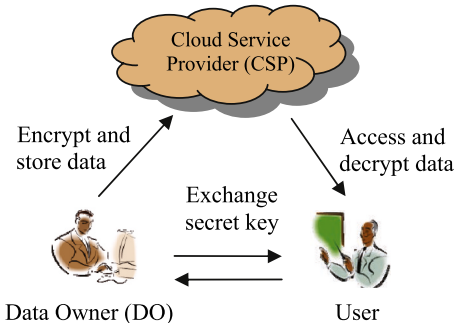

Fig. 1. System model of DNAFS.

right for the users to access DO's data, and the users cannot access the data, if the access right is invalid.

2) *Minimizing the data accessing time:* In a cloud environment, the users always pay money based on pay-per-use. If the data accessing time is high, they must pay more. Reducing the data accessing time in the cloud environment is always a major requirement for the users.
3) *Security:* In a cloud environment, DO's sensitive data are usually outsourced to the geographically distributed server, so the security issue of data increases because the CSP can illegally use data. In addition, hackers always want to access the sensitive and confidential data. Therefore, there must be a strong security technique for protecting the sensitive data.
4) *Reducing the overhead of the* system: The overhead of the cloud environment becomes high, when the DOs remain online during the entire data communication process. Therefore, minimization of the overhead can be highly beneficial.

3.3. Design goals

The main design goals of DNAFS have been briefly discussed in this sub-section.

1) To achieve an effective and robust ACM for the cloud environment for minimizing the searching time of a DO and the data accessing time.
2) To develop a strong data encryption technique for the confidential or sensitive data for the cloud computing environment.
3) To provide the data access right by creating a long 1024-bit secret key or password on the basis of DNA computing aided with a corresponding Access Certificate (AC).
4) To minimize the system overhead of the cloud environment by not keeping the DO always online during the entire data communication process.

4. Proposed model

In the proposed DNAFS, the CSP maintains a CSP Table (CSP-TAB) with six attributes: Group Identity (ID) (GP_ID), Data Type (Dt), Data Size (Sz), Data Owner's ID (DO_ID), DO's Time & Date (DO_T&D) and Group's Time & Date (GP_T&D). GP_ID attribute maintains the identity number of the group. Sz and Dt fields keep the range of data size and data type of the corresponding group, respectively. The range of data size and data type are determined by the CSP. The DOs, who share the same type and same size of data are kept in a particular DO_ID. DO_T&D holds the corresponding date and time of the DOs about when they have sent the requested data. GP_T&D field holds the creation details of any group. This CSP-TAB is mainly used to minimize the searching time of DO and data accessing time.

A long 1024-bit DNASK is used in the proposed DNAFS to encrypt DO's sensitive and confidential data. The data owner asks the user to give a secret password and uses other confidential information of the user like Media Access Control (MAC) address, Date of Birth (DOB) and many more to generate the DNASK through several processes, which improve security of the proposed scheme. The proposed scheme consists of mainly six phases: (i) system setup (ii) registration (iii) login (iv) processes on the CSP-TAB (v) DNA based data encryption, and (vi) data access. The entire workflow of DNAFS has been shown in Fig. 2.

4.1. System setup

The cloud service provider selects a large prime number (p) to calculate a multiplicative group Z_p^*. The public-private key pair of the CSP has been chosen from this multiplicative group. The CSP also uses Z_p^* for choosing the key pairs for both the DO and users, which are provided to them during their registration. In DNAFS, the Public Key of the User

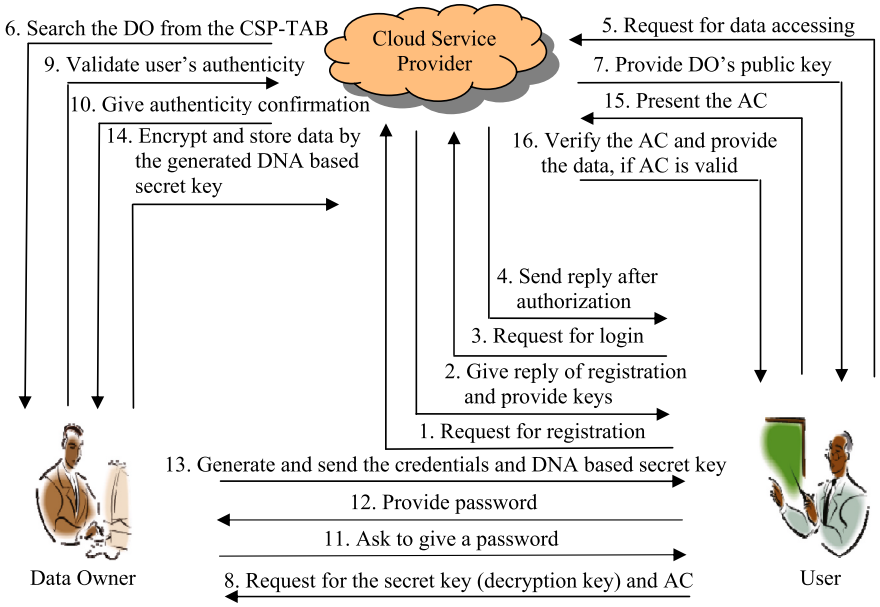

Fig. 2. Workflow of the proposed DNAFS.

(PCKUSR) is available for all the authorized entities, and only the registered user knows the respective PCKDO for data accessing. The Public Key of the CSP (PCKCSP) is known to all the registered or authorized entities and the private key of every registered entity is kept secret from other entities.

4.2. User registration

To register in the cloud server, the user must send a registration request to the cloud service provider. Then, the CSP generates a user's profile by using the confidential information of the user, such as DOB, MAC address, first school name and first teacher name. Then, Secure Socket Layer (SSL) is developed to provide the public key, private key and registration reply to the user. The CSP follows the same process to register the DOs.

4.3. User login

The CSP allows only the authorized or registered users for logging into the cloud server after the registration phase. A reply is sent to the user by the CSP after the login process. The user can send a data access request after a successful login process. The CSP provides the PCKDO to the user to get the AC and secret key (decryption key) from the respective DO. This secret key (DNASK) is required during the data decryption phase and the user provides the AC to the CSP for verification purpose.

4.4. DNA based data encryption

In this phase, at first, a long 1024-bit DNASK has been generated by the DO based on DNA computing, and then, the DO uses this DNASK to encrypt the data. There are mainly two sub-phases in the DNA based data encryption phase.

DNA Based Secret Key Generation Algorithm

Input: MAC address and attributes of an authorized user.
Output: 1024-bit DNA based secret key.

```
if D == Authorized user's ID
    COLLECT password, DOB, MAC address, first teacher name and first school name of the user
    APPLY decimal encoding rule
    CONVERT into 8-bit binary number
    DIVIDE into two equal parts
    PERFORM EXOR operation between two equal parts
    ADD password to the left end
    ADD MAC address to the right end
    APPLY decimal encoding rule
      CONVERT into 8-bit binary number or value
      if length of binary value > 1024-bit
        DELETE extra bit from the left end
      else
        ADD zero to the right end to make it 1024-bit
      DIVIDE the binary number into four blocks
      ASSIGN DNA base for each block
      DIVIDE into two equal parts
      PERFORM DNA-EXOR operation between two equal parts
      APPLY complementary rule
else
      STOP
```

4.4.1. DNA based key generation

In the DNA based key generation phase, when the DO receives a request for providing the password or secret key and AC from a user, the DO checks whether the user is valid or not. The DO generates the DNASK for the valid user after getting the authenticity confirmation from the CSP. The DO then asks the user to provide a password and collects some confidential information of the user, such as DOB, MAC address, the first teacher name and the first school name. The decimal encoding rule is applied to the password and all the secret information by using Table 3. Then, the DO converts decimal values into their corresponding binary values (8-bit), and then, divides into two equal parts. EXOR operation has been performed between both the parts. The DO adds the password and MAC address to the left side and right side of the binary number, respectively. Then, the entire value is converted into 8-bit binary value after applying the decimal encoding rule of Table 3. The length or size of the resultant binary number can be more or less than 1024-bit. Extra zeros are added to the right end for making it 1024-bit, when it is less than 1024-bit. If not, bits are deleted from its left end to make 1024-bit. In the next step, the 1024-bit binary number has been divided into 4 equal size blocks, so in each block, there is 256-bit. Then, the DO randomly assigns the DNA bases for each block that implies the DO has 24 choices to select the DNA bases for the blocks. After assigning the DNA bases, the DO divides the DNA sequence into two equal parts and applies DNA-EXOR operation between both the parts by using Table 4. At last, the DO applies the complementary rule to the resultant DNA sequence for making it more complex. The complementary rule can be represented as $V(W) = C$, where $W \neq C$. In DNAFS, the following complementary rule has been used:

(AG) (TC) (GA) (CT) that means $V(A) = G$ and so on.

The resultant complex DNA sequence has been sent to the user as DNASK encrypted by the Private Key of the DO (PRKDO) and PCKUSR. The DO also sends Table 3, Table 4, Table 5, Table 6, complementary rule and AC in the same message. The user uses his/her confidential attributes and retrieves the DNASK after decrypting by using the Private Key of the User (PRKUSR) and PCKDO.

Table 3
Rule of decimal encoding.

Alphabet/symbol	Decimal value	Alphabet/symbol	Decimal value
A	01	'	27
B	02	~	28
C	03	!	29
D	04	@	30
.	.	.	.
.	.	.	.
.	.	.	.
X	24	(	48
Y	25	)	49
Z	26	Blankspace	50

4.4.2. Data encryption

In the proposed data encryption technique, the DO splits the plaintext of any data or file into equal 1024-bit blocks. If the last block is not 1024-bit, extra bit i.e. zero is added to the right end of the last block to make it 1024-bit. The DO executes EXOR operation between plaintext of the 1024-bit block and 1024-bit secret key. Each 8-bit binary number is then converted into American Standard Code for Information Interchange (ASCII) values by using Table 5. Table 5 is randomly generated by the DO in which the ASCII value of 00000000 can be 47 or 34 or whatsoever. Thus, it provides randomness. ASCII values are then converted into their corresponding actual binary number. Then, each 2-bit binary number is converted into DNA base by using randomly generated Table 6. To change the resultant DNA sequence or bases into another random DNA sequence, the DO uses the complementary rule on the DNA bases, which is used in the above key generation phase. Then, the DO divides the DNA sequence into two equal parts and uses DNA-EXOR between both the parts by using Table 4.

At last, the DO uses the PRKDO and PCKCSP to encrypt the final DNA

Table 4
DNA-EXOR rule.

DNA-EXOR	A	T	G	C
A	A	T	G	C
T	T	A	C	G
G	G	C	A	T
C	C	G	T	A

Table 5
ASCII value.

8-bit binary value or number	ASCII
00000000	47
00000001	42
00000010	1
.	.
.	.
.	.
11111111	64

Table 6
DNA base corresponding to 2-bit binary value.

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

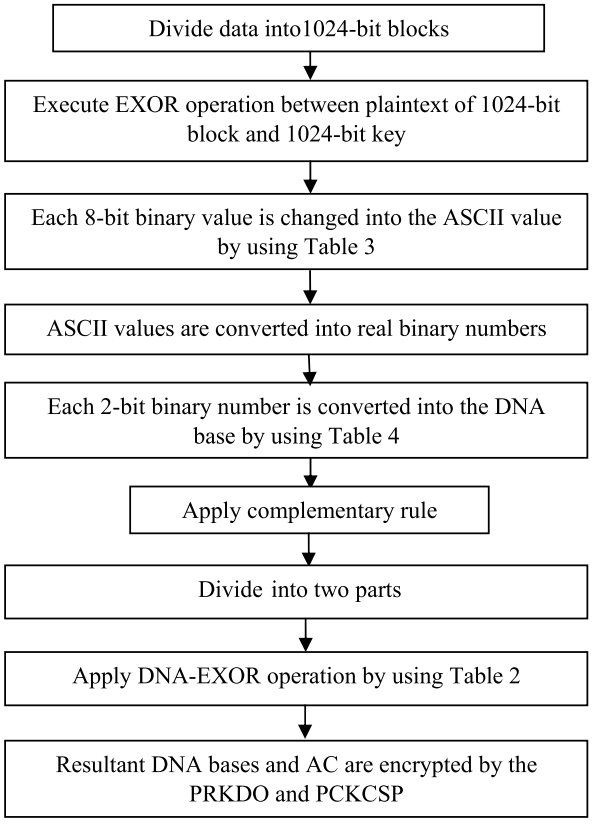

Fig. 3. DNA based data encryption.

sequence and corresponding AC. The DO then sends the encrypted message to the CSP. After receiving the encrypted message from the DO, the CSP uses its own (CSP) Private Key (PRKCSP) and PCKDO to partially decrypt the message and stores the encrypted data (encrypted by using the DNASK) on the cloud server. DNA computing-based data encryption process has been depicted in Fig. 3.

Table 7
CSP table (CSP-TAB).

Group ID (GP_ID)	Data Owner's ID (DO_ID)	Data Size (Sz) in GB	Data Type (Dt)	DO's Time & Date (DO_T&D)	Group's Time & Date (GP_T&D)
1	198	$2^{-33} < Sz \leq 2^{-30}$	MP3	06:08 16-09-2015	05:59 22-12-2011
	033			11:32 29-05-2015	
	612			17:46 03-07-2014	
	561			20:16 28-11-2012	
	687			05:59 22-12-2011	
2	071	$2^{-30} < Sz \leq 2^{-25}$	DOCX	03:36 13-11-2014	23:49 13-12-2007
	003			01:19 21-08-2014	
	712			14:09 27-10-2012	
	149			23:46 09-07-2012	
	342			23:49 13-12-2007	
.	.	.	.	.	.
	.			.	
	.			.	
	.			.	
	.			.	
50	546	$1 < Sz \leq 2$	ZIP	16:23 18-03-2013	09:30 22-07-2010
	613			17:41 03-02-2013	
	738			10:42 12-06-2012	
	069			11:57 25-11-2010	
	001			09:30 22-07-2010	

4.5. Processes on the CSP-TAB

In the CSP-TAB, DOs' IDs are placed based on the principle of the Least Recently Used (LRU) concept. The DO's ID is kept at the top position, when the DO provides the data to the user. There are 3 main operations on the CSP-TAB: (i) inserting a DO's ID (ii) deleting a DO's ID and deleting an entire group, and (iii) searching a PCKDO. Let us assume that GP_ID and DO_ID have been defined by the two sets $S = \{UG_1, UG_2, UG_3, \ldots, UG_P\}$ and $S' = \{ug_1, ug_2, ug_3, \ldots, ug_Q\}$, respectively. Here, UG_x and ug_x represent x^{th} GP_ID and x^{th} DO's ID, respectively. P and Q denote the maximum number of GP_ID and the maximum number of DOs' IDs per DO_ID, respectively in the CSP-TAB. Table 7 shows an example of CSP-TAB.

4.5.1. Insertion

The insertion algorithm focuses to insert the ID of a DO in the CSP-TAB. When a DO wants to share any data on the cloud server, at first, DO's authenticity is checked by the CSP. Then, if the data size of the DO and data type exist in the CSP table and the respective DO_ID field is not full, the CSP adds the DO in the CSP-TAB at the corresponding position. Otherwise, a new group is added in the CSP-TAB, if the CSP-TAB is not

full. In the new group, the CSP keeps the details of the DO. Table 8 shows all the notations used in this paper.

Algorithm to Insert the ID of a DO in the CSP-TAB

Input: ID of an authorized DO.
Output: Updated CSP-TAB.

```
1  if I == Authorized DO's ID
2    c0←P, c1←Q
3      if c0 ≠ FULL
4        for all UG_x ∈ S
5          if I_Dt == GP_IDx_Dt and I_Sz = = GP_IDx_Sz
6            for all ug_x ∈ S' in UG_x
7              if c1 ≠ FULL
8                  GP_IDx_DO_ID←GP_IDx_DO_ID ∪ I
9              else
10                 DEL_DO_ID (GP_ID)
11                 GP_IDx_DO_ID←GP_IDx_DO_ID ∪ I
12           end for
13       end for
14     else
15       r←DEL_GP_ID()
16       GP_IDr_DO_ID←I
17       GP_IDr_Dt←I_Dt
18       GP_IDr_Sz←I_Sz
19       GP_IDr_DO_T&D←I_T&D
20       GP_IDr_GP_T&D←I_T&D
21       UPDATE CSP-TAB
22 else
23   STOP
```

4.5.2. Deletion

A DO's ID can be deleted from the DO_ID or a whole GP_ID can be deleted from the CSP-TAB by the CSP based on the requirement. Here, both the deletions are the parts of the insertion algorithm of a DO's ID in the CSP-TAB. So, an authorized DO's ID is the input of both the deletion algorithms.

1) *DO's ID deletion algorithm from the CSP-TAB* (***DEL_DO_ID*** (***GP_ID***)): The ID of a DO from the DO_ID can be deleted after traversing GP_ID field and the respective DO_ID field. After finding the LRU DO's ID, it has been assigned to m, and finally, deleted from DO_ID field. Then, the CSP-TAB has been updated.
2) *Whole GP_ID deletion algorithm from the CSP-TAB* (***DEL_GP_ID***()): A long time unused GP_ID can be deleted by the CSP based on the LRU concept. At first, the LRU GP_ID has been searched from the CSP-TAB, then, the CSP deletes all the entries of x^{th} GP_ID and updates the table.

Algorithm (***DEL_DO_ID*** (***GP_ID***)) **to Delete the ID of a DO from the CSP-TAB**

Input: ID of an authorized DO.
Output: Updated CSP-TAB.

```
1  for i = 1 to P
2    for j = 1 to Q
3      m←SEARCH_LRU (DO_ID_Y)
4      DO_ID_NEW←DO_ID_PREV − m
5    end for
6  end for
7  UPDATE CSP-TAB
8  STOP
```

Algorithm (***DEL_GP_ID***()) **to Delete a GP_ID from the CSP-TAB**

Input: ID of an authorized DO.
Output: Updated CSP-TAB.

```
1  SEARCH LRU GP_ID
2  DELETE entries of x^th GP_ID
3  UPDATE CSP-TAB
4  STOP
```

4.5.3. Searching a public key of the DO

When a user requests for data accessing, at first, the CSP checks the user's authenticity before providing the PCKDO to the user. The user can only send a request to the data owner to provide the secret key and AC after getting the PCKDO from the CSP. Here, the CSP matches the D_{Dt} and D_{Sz} against the GP_IDx_{Dt} and GP_IDx_{Sz}. Then, the CSP searches the PCKDO from the corresponding position of the CSP-TAB and provides the PCKDO to the requested user. Otherwise, the CSP shows invalid data size and data type.

Algorithm to Search a PCKDO from the CSP-TAB

Input: ID of an authorized user.
Output: Search PCKDO and provide it to the user.

```
1  if D == Authorized user's ID
2    for j = 1 to P
3      if D_Dt == GP_IDx_Dt and D_Sz = = GP_IDx_Sz
5        SEARCH DO from GP_IDx_DO_ID
6        PROVIDE PCKDO to D
7      else
8        Display INVALID data type and data size
9    end for
10 else
11   STOP
```

4.6. Data access

When a user wants to access any data or file from the cloud server, s/he must send a data access request to the CSP. The CSP only allows the authorized or registered users to access data from the cloud server. There are the following steps to access any data from the cloud server:

Step 1: The user must register in the cloud server by sending a registration request.

Step 2: The CSP collects all the information about the user and sends a registration reply after a successful registration process. The CSP also sends the public-private key pair along with this reply.

Step 3: The user can login into the cloud server after getting the registration reply.

Step 4: In the fourth step, the CSP accepts login request only from the

Table 8
Notations and descriptions.

Notation	Description	Notation	Description
EK_{DNASK}	Encryption by using DNASK	GP_IDx_{Dt}	Data type of x^{th} GP_ID
DK_{DNASK}	Decryption by using DNASK	GP_IDx_{Sz}	Data size of x^{th} GP_ID
EK_{PRKDO}	Encryption by using PRKDO	D_{Dt}	Requested data type of D
EK_{PCKDO}	Encryption by using PCKDO	D_{Sz}	Requested data size of D
DK_{PRKDO}	Decryption by using PRKDO	$GP\_IDx_{DO\_ID}$	DO_ID of x^{th} GP_ID
DK_{PCKDO}	Decryption by using PCKDO	$GP\_IDr_{DO\_ID}$	DO_ID of r^{th} GP_ID
EK_{PCKCSP}	Encryption by using PCKCSP	GP_IDr_{Dt}	Data type of r^{th} GP_ID
EK_{PRKCSP}	Encryption by using PRKCSP	GP_IDr_{Sz}	Data size of r^{th} GP_ID
DK_{PRKCSP}	Decryption by using PRKCSP	DO_ID_Y	Y^{th} DO's ID
DK_{PCKCSP}	Decryption by using PCKCSP	DO_ID_{NEW}	DO_ID after processing
EK_{PRKUSR}	Encryption by using PRKUSR	DO_ID_{PREV}	DO_ID before processing
EK_{PCKUSR}	Encryption by using PCKUSR	I_{Sz}	Data size of DO having identity I
DK_{PRKUSR}	Decryption by using PRKUSR	I_{Dt}	Data type of DO having identity I
DK_{PCKUSR}	Decryption by using PCKUSR	$I_{T\&D}$	Data and time of DO having ID I
$GP\_IDr_{GP\_T\&D}$	$GP\_T\&D$ of r^{th} GP_ID	$GP\_IDr_{DO\_T\&D}$	$DO\_T\&D$ of r^{th} GP_ID

authorized user and sends an acknowledgment to the user.

Step 5: The user then can send a data access request to the service provider in the encrypted form by using the PRKUSR and PCKCSP.

Step 6: The CSP searches the respective PCKDO from the CSP-TAB in the sixth step.

Step 7: The CSP sends the PCKDO to the user or customer encrypted by the PRKCSP and PCKUSR. The user gets the PCKDO after decrypting the message.

$$PCKDO \rightarrow EK_{PRKCSP}(PCKDO) \rightarrow EK_{PCKUSR}\{EK_{PRKCSP}(PCKDO)\} \rightarrow User$$

$$PCKDO \leftarrow DK_{PCKCSP}\{EK_{PRKCSP}(PCKDO)\} \leftarrow DK_{PRKUSR}[EK_{PCKUSR}\{EK_{PRKCSP}(PCKDO)\}]$$

Step 8: In the eighth step, by using the PCKDO, the user sends a request to the corresponding DO encrypted by the PRKUSR and PCKDO to get the secret key and AC.

$$Data \rightarrow EK_{DNASK}(Data) \rightarrow EK_{PRKDO}\{EK_{DNASK}(Data) \text{ and } AC\} \rightarrow EK_{PCKCSP}[EK_{PRKDO}\{EK_{DNASK}(Data) \text{ and } AC\}] \rightarrow CSP$$

$$EK_{DNASK}(Data) \text{ and } AC \leftarrow DK_{PCKDO}[EK_{PRKDO}\{EK_{DNASK}(Data) \text{ and } AC\}] \leftarrow DK_{PRKCSP}[EK_{PCKCSP}[EK_{PRKDO}\{EK_{DNASK}(Data) \text{ and } AC\}]]$$

Step 9: The DO validates whether the user is valid or not from the CSP.

Step 10: The CSP sends an authentication reply to the DO.

Step 11: In the eleventh step, the DO requests the user to provide a password for generating the DNASK.

$$Password \leftarrow DK_{PCKUSR}\{EK_{PRKUSR}(Password)\} \leftarrow DK_{PRKDO}[EK_{PCKDO}\{EK_{PRKUSR}(Password)\}]$$

Step 12: The user provides a password to the DO encrypted by using the PRKUSR and PCKDO, and the DO decrypts it by using the PRKDO and PCKUSR.

$$Password \rightarrow EK_{PRKUSR}(Password) \rightarrow EK_{PCKDO}\{EK_{PRKUSR}(Password)\} \rightarrow DO$$

$$DNASK \text{ and } Cred \rightarrow EK_{PRKDO}(DNASK \text{ and } Cred) \rightarrow EK_{PCKUSR}\{EK_{PRKDO}(DNASK \text{ and } Cred)\} \rightarrow User$$

$$DNASK \text{ and } Cred \leftarrow DK_{PCKDO}\{EK_{PRKDO}(DNASK \text{ and } Cred)\} \leftarrow DK_{PRKUSR}[EK_{PCKUSR}\{EK_{PRKDO}(DNASK \text{ and } Cred)\}]$$

Step 13: In the thirteenth step, the DO uses secret information and password of the user to generate a 1024-bit DNASK. The DO sends the DNASK and all the Credentials (Cred), namely Table 3, Table 4, Table 5, Table 6, complementary rule and AC to the user as an encrypted message by using the PRKDO and PCKUSR. The user decrypts it by using the PRKUSR and PCKDO.

$$EK_{DNASK}(Data) \rightarrow EK_{PRKCSP}\{EK_{DNASK}(Data)\} \rightarrow EK_{PCKUSR}[EK_{PRKCSP}\{EK_{DNASK}(Data)\}] \rightarrow User$$

Step 14: The DO encrypts the requested data by using the DNASK, and stores the ciphertext of data and corresponding AC on the cloud server after encrypting by the PRKDO and PCKCSP. The CSP partially decrypts the ciphertext by using the PRKCSP and PCKDO.

Step 15: In *the* fifteenth step, the AC is sent to the CSP by the user for the verification purpose encrypted by the PRKUSR and PCKCSP. The CSP decrypts the message by using the PRKCSP and PCKUSR to get the AC.

$$AC \rightarrow EK_{PRKUSR}(AC) \rightarrow EK_{PCKCSP}\{EK_{PRKUSR}(AC)\} \rightarrow CSP$$

$$AC \leftarrow DK_{PCKUSR}\{EK_{PRKUSR}(AC)\} \leftarrow DK_{PRKCSP}[EK_{PCKCSP}\{EK_{PRKUSR}(AC)\}]$$

Step 16: In the last step, the CSP searches the data from the cloud database, and then, checks whether the AC provided by the user is matched with the corresponding AC of the data provided by the DO. If the access certificate is valid, the CSP provides the ciphertext of data after encrypting by using the PRKCSP and PCKUSR. Otherwise, the data access request is declined by the CSP.

The user decrypts the encrypted message by using the PRKUSR and PCKCSP. Finally, the user uses the long DNASK to decrypt the message. The user can easily get the original data content because s/he has all the secret credentials.

$$Data \leftarrow DK_{DNASK}\{EK_{DNASK}(Data)\} \leftarrow DK_{PCKCSP}[EK_{PRKCSP}\{EK_{DNASK}(Data)\}] \leftarrow DK_{PRKUSR}[EK_{PCKUSR}[EK_{PRKCSP}\{EK_{DNASK}(Data)\}]]$$

5. Security analysis

DNAFS can resist insider attack, password guessing attack, distributed denial of service attack, side channel attack and stolen verifier attack. In this section, security analysis of the proposed DNAFS has been described.

5.1. Insider attack

An attacker or malicious user accesses any data in an insider attack by violating any kind of security policy. Any entity can be a hacker in a cloud environment because of its internet-based service, and the entity always tries to violate the access right or security policy to get unauthorized data accessing. Even the service provider also tries to hack data by an insider attack. In DNAFS, the DO generates the secret key based on the user's secret information and a password provided by the user. In addition, the DO uses complementary rule, Table 3 and Table 4 to generate the DNASK. In the proposed DNA based data encryption technique, the DO encrypts the data by using the DNASK, Table 4, Table 5, Table 6 and complementary rule. As the DNASK is generated based on the secret information of the user, it is almost impossible to guess all the secret information. The attacker also must have all the randomly generated tables, complementary rule and AC to get the original data. The authorized user only gets all the credentials encrypted by using the PRKDO and PCKUSR. The CSP or DO or any other user does not aware of all the confidential information. Thus, the proposed DNAFS is secured against the insider attack.

5.2. Password guessing attack

In a password guessing attack, the hackers try all the possible combinations of a password for getting the correct password. In the proposed DNAFS, the DO generates the secret key on the basis of the randomly given password by the user, secret information of the user, Table 3, Table 4 and complementary rule. In the proposed data encryption technique, the DO uses Table 4, Table 5, Table 6 and complementary rule. The DO randomly generates all the tables by his/her own wish to improve data security. For example, the decimal value of the alphabet "A" in Table 3 can be 01 or 10 or anything. The DO can use a different table for the same data. Guessing all the attributes by any attacker is quite difficult and all the credentials are shared in the encrypted form only with the authorized user. The authorized user can only get all the credentials after decrypting it. Therefore, DNAFS is not vulnerable to this attack.

5.3. Distributed denial of service attack

In a Distributed Denial of Service (DDoS) attack, numerous requests are sent to the cloud server by many malicious users or hackers, who are geographically distributed. These requests are sent for making cloud services inaccessible for the authorized users. In DNAFS, the DO randomly generates a long DNASK and uses the same secret key to encrypt the data. The DO can randomly assign the DNA bases in both DNA based key generation and data encryption phase to improve data security. The DO also utilizes the PRKDO and PCKCSP for encrypting the data. The CSP or any other entity does not know anything about the DNASK and other credentials. If DDoS attack has been made by many attackers in the cloud server, they must have all the credentials i.e. Table 3, Table 4, Table 5, Table 6, complementary rule and AC for getting the original data. These credentials are only shared with the authorized users, so there cannot be any data issue in the proposed scheme. Thus, DNAFS can resist the DDoS attack.

5.4. Side channel attack

In this attack, attackers or hackers place an unauthorized virtual machine on the same host, so that they can get access to any confidential data. Here, hackers attack the encryption algorithm for unauthorized access. In DNAFS, the DO generates the secret key by several processes for making it more secure and this 1024-bit DNA based key is used in the data encryption phase. The ciphertext of data is yet again encrypted by using the PRKDO and PCKCSP along with the AC. Then, it is stored on the database of the cloud server. Only the respective users get the DNASK in the encrypted form and the corresponding user can only get the DNASK after decryption. If any hacker or malicious user places any VM, the security of data cannot be compromised because the data is encrypted three times in the proposed scheme by using the DNASK, PRKDO and PCKCSP. So, the proposed scheme is secured against the side channel attack.

5.5. Stolen verifier attack

In a stolen verifier attack, the hackers or attackers try to get the hashed password. Then, they recover the original key from the hashed password, and then, data are accessed from the cloud server. Here, in the proposed DNAFS, after verifying the user's authorization, the DO generates the DNASK based on the secret information of the user and a password provided by the user. The DO then encrypts the DNASK by using the PRKDO and PCKUSR, and then, it is sent to the user. The respective user only can decrypt the message by using the PRKUSR and PCKDO to get the DNASK. The DO does not provide the DNASK to other entities and the DNASK is not stored anywhere. Thus, DNAFS can resist the stolen verifier attack in the cloud environment.

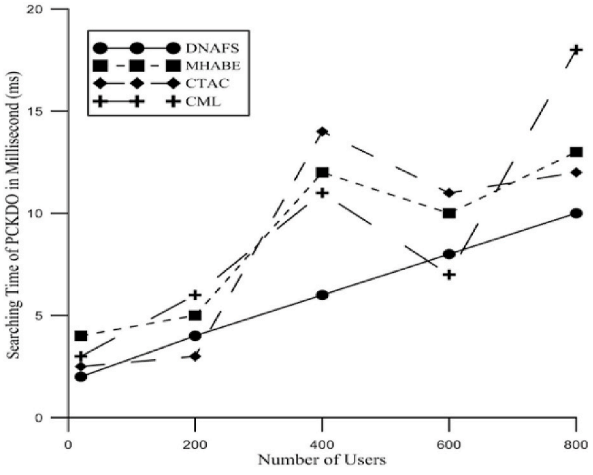

Fig. 4. Searching time to find public key of the DO vs. number of users.

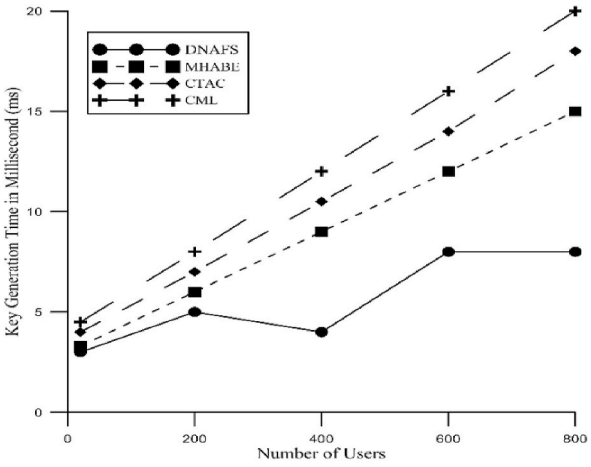

Fig. 5. Key generation time vs. number of users.

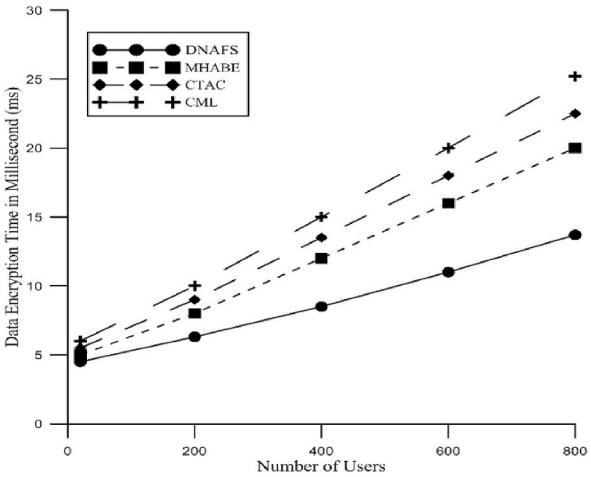

Fig. 7. Data encryption time vs. number of users.

6. Performance analysis

Performance analysis of the proposed DNAFS has been presented in this section in detail.

6.1. Simulation environment

A cloud computing simulation environment is built to evaluate the performance of the proposed DNAFS. Here, in this paper, CloudSim 3.0.3 is used for evaluation purpose (Calheiros et al., 2011). In CloudSim, the datacenter sends a request to the Cloud Information Service (CIS) for registering the resources. All the datacenters have numerous hosts and each host has numerous VMs with RAM, processing elements and bandwidth. The brokers or users have numerous cloudlets i.e. tasks and brokers send request to the CIS for getting the available resources to run all the cloudlets. The characteristics or attributes of the datacenter are then sent to the broker. When the broker sends a request for the VMs, VMs are created by the datacenter and a reply is sent to the broker. Then, the broker submits the cloudlet to the module, namely datacenter broker to execute. The datacenter allocates VMs for cloudlet, and an acknowledgment is sent to the broker, when the cloudlet is executed.

Dell Optiplex 7050 MT desktop is used to install CloudSim with the following configurations:

1) Storage capacity: 2 TB
2) RAM: 8 GB
3) Operating system: Windows 10
4) Processor: Corei7

Apache Commons Math 3.6.1 (Apache commons math, 2019) is configured with the CloudSim tool and Java version 8 (Java, 2019) is furthermore installed on the Dell Optiplex 7050 MT desktop. Twenty datacenters are considered to build a heterogeneous cloud computing environment in which there are 3000 physical nodes and memory capacity is 4 GB. In the simulation, 4 types of VMs are considered with 1 GB/s bandwidth:

1) 1500 MIPS, 2 GB
2) 2000 MIPS, 2.5 GB
3) 2500 MIPS, 3 GB
4) 3000 MIPS, 3.5 GB

6.2. Results and discussions

The proposed technique is compared with three other existing schemes, namely coupled map lattice-based ACM (Wang et al., 2018), cross tenant access control (Alam et al., 2017) and modified hierarchical attribute-based encryption (Xie et al., 2019). In different scenario, 100 experiments are executed to get the searching time of PCKDO, key generation time, key retrieval time, data encryption time, data decryption time and data accessing time. Average values are then calculated from 100 results. Dataset is collected from CityPulse Dataset Collection

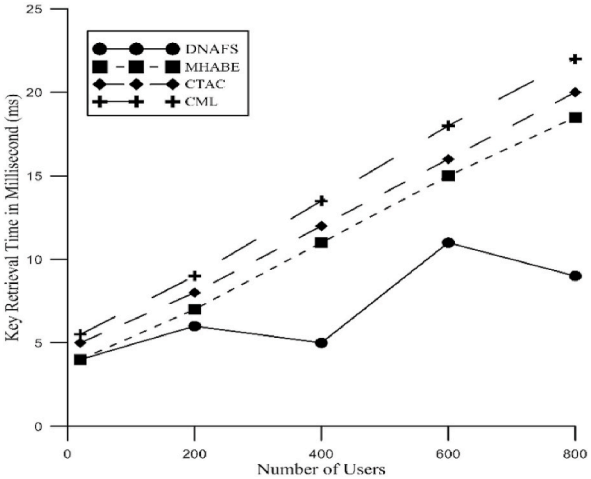

Fig. 6. Key retrieval time vs. number of users.

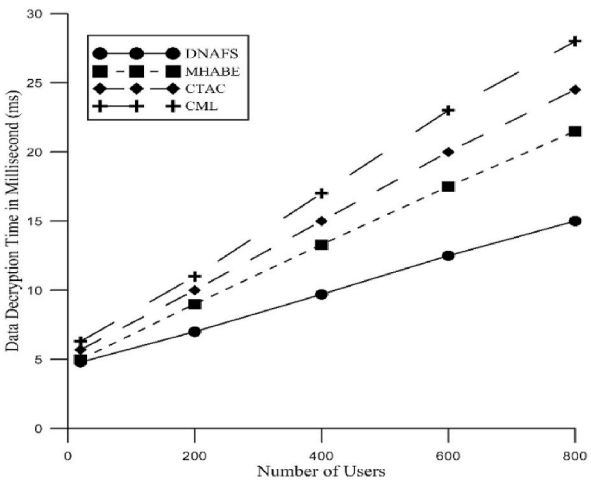

Fig. 8. Data decryption time vs. number of users.

(CityPulse dataset collection, 2019). Here, the weather dataset is used in the experiments.

The first experiment is executed for searching the PCKDO from the cloud server. It can be clearly seen in Fig. 4 that the proposed DNAFS gives a much better result than the other existing schemes, namely MHABE, CTAC and CML to deliver the PCKDO to the users. In DNAFS, after receiving a data access request from the user or customer, the CSP searches the data owner of the requested data or file from the CSP-TAB. Here, the service provider (CSP) searches the DO based on the requested file or data type and data size. So, the CSP directly finds the position of the DO from the CSP-TAB and does not need to search the entire cloud database for a single DO. The CSP can easily provide the PCKDO of the requested data in less time after finding the DO. If the number of users increases, load on the CSP also increases and many users send data access requests to the CSP. Then, the CSP needs to search the PCKDO for each data access request, and as result, it consumes more time and produces a linearly increasing curve for the proposed DNAFS. In the existing schemes, namely MHABE, CTAC and CML, the CSP stores the details of the DO in the scattered matter. These schemes do not maintain CSP-TAB or any other technique to search the DOs in less time. Sometimes, the CSP searches the DO in less time, and sometimes, the CSP searches the entire database to find a DO. Thus, zigzag curves are experienced for these existing schemes.

Fig. 5 illustrates the results of the second experiment for calculating the secret key generation time. When the users want to access data in the proposed DNAFS from the cloud environment, they must give request to the DO for the secret key and access certificate. The DO requests to the users to provide a password and collects some secret information of the user for secret key generation. Then, the DO uses complementary rule, Table 3 and Table 4 to generate the DNASK. Thus, DNAFS takes time to generate the secret key. In a cloud environment, there can be new users and old users. In DNAFS, the DO generates the DNASK for all the new users, who have not accessed data from the cloud server. So, the key generation time is high. However, the DO does not generate the DNASK for the old users, who have already accessed data from the cloud server because once the secret key or password is sent to the existing users, the key can be used in the future. It is very difficult to predict whether a user is new or existing in any cloud computing environment. Existing users sometimes send the data access requests, and sometimes, new users send the data access requests. So, the zigzag curve is produced for DNAFS. However, the existing models are more complex than the proposed scheme, and both new and old users must collect the DNASK from the DO. So, the DO needs to generate the secret key for all the users in the existing schemes and linearly increasing curves have been generated.

The result of the next experiment verifies the secret key retrieval time. Fig. 6 shows that DNAFS produces better results than other existing schemes for retrieving the secret key. In a cloud environment, both new users and old users send data access requests. In DNAFS, when a user sends a data access request to the CSP and if the user has already accessed the data from the cloud server i.e. the user is an old user, the CSP directly provides the data to the user. Here, the DO does not generate the secret key for the particular user. Therefore, the user does not retrieve the secret key for the data access request, and thus, the key retrieval time is minimized in the proposed DNAFS. However, in the existing schemes, the DO needs to generate the secret key for all the users, and after receiving the secret key from the DO, all the customers or users must retrieve the key for each data accessing process. Therefore, the key retrieval time is high for the existing schemes, namely MHABE, CTAC and CML.

In Fig. 7, it can be easily understood that DNAFS gives much better results than MHABE, CTAC and CML to encrypt data in the cloud environment. In DNAFS, the DO uses DNASK, Table 4, Table 5, Table 6 and complementary rule to encrypt data. There are two EXOR operations in the proposed data encryption technique that consume a little bit of time at the DO's end. In MHABE, attribute-based encryption and HIBE have been combined to provide efficient data access control in the cloud

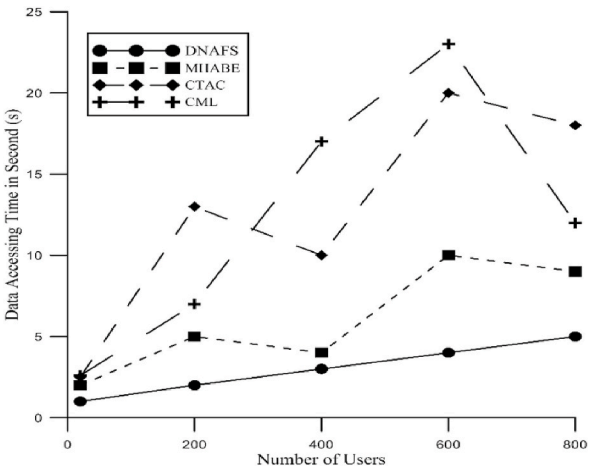

Fig. 9. Data accessing time vs. number of users.

environment. Because of combining both the schemes, MHABE is much complex and many EXOR operations are executed. The CSP manages all the tasks in CTAC, and this scheme consists of many operations to encrypt the data. In CML, DNA sequence and coupled map lattice are used for data encryption. The user must give a key in CML, and this key is combined with SHA-256 algorithm that takes time to encrypt the data. Therefore, the encryption time of data in these existing schemes is more than the proposed DNAFS.

Fig. 8 shows the results of the next experiment. When a user sends a request to get the secret key and AC, after checking the user's authentication, the DO sends all the tables, complementary rule and AC along with the DNASK in the encrypted form. The user decrypts the data or message using the PRKUSR and PCKDO to get all the credentials. Then, the user can easily get the original data content in less time as there are no complex operations in the proposed DNAFS and only two EXOR operations are needed during data decryption phase. As the user executes only two EXOR operations for data decryption, these consume less time in the proposed scheme. Many EXOR operations are required for the existing schemes, namely MHABE, CTAC and CML for data decryption. In addition, these existing schemes are more complex than DNAFS because of the presence of complex operations. The data decryption time is high in MHABE, CTAC and CML because of executing many EXOR operations.

The next simulation results are to validate the data accessing time. The data accessing time has been considered from the data request time to the data decryption time. In DNAFS, the CSP can easily search the PCKDO in less time from the CSP-TAB on the basis of the requested data size and data type. So, the users do not need to wait for a long time for getting the PCKDO, and by using the respective PCKDO, the users can send the requests to the DOs to get the respective secret keys and ACs. Since the key generation time, key retrieval time, data encryption time and data decryption time are less in DNAFS, the data accessing time is also automatically less. Thus, a linearly increasing curve is experienced that can be seen in Fig. 9. CTAC consists of four algorithms for delegation mechanism, namely activation, forward revocation, delegation and backward revocation, which increase the data accessing time. In all the existing schemes, since the CSP searches the entire database to provide a PCKDO, the searching time of PCKDO is high. Moreover, the key generation time, key retrieval time, data encryption time and data decryption time are also high in MHABE, CTAC and CML. Thus, the data accessing time is much high in all these schemes.

7. Future work directions

Identity and access management in a cloud environment is one of the important aspects to manage any data. Attackers try to get any data by

using the stolen credentials of an authorized user. Identity and access management are always used together. However, both have distinct concepts. In a cloud environment, each entity needs an identity. When the CSP verifies the identity of an entity, it is known as authentication. Access management can be referred to ensure that only the valid entities can execute the tasks that they are allowed to perform. There is a huge scope to improve identity and access management techniques in a cloud environment. The traditional authentication techniques take much time to authenticate the users, which utilize more cloud service. Moreover, hackers and malicious users attack the authentication phase to get unauthorized access. Nowadays, the machine learning algorithms are used in academia and industry due to the data driven feature to achieve enhanced security. A machine learning algorithm can be developed to train a cloud computing environment for detecting sophisticated attacks, which are similar to the already defined known attacks.

In a cloud environment, the user revocation problem is one of the key issues for access management as users must not access any data or service after deregistering from the cloud server. A novel technique can be developed to address the user revocation problem of the cloud environment. In the future, the proposed scheme can be improved by developing a novel fast and efficient authentication technique for the cloud computing environment. In addition, the proposed scheme will be evaluated in a real time cloud computing environment in the future.

8. Conclusions

In the cloud environment, data accessing issue and data security issue are very critical. In this paper, a novel access control model has been proposed. Here, the CSP maintains a temporary table on the basis of the data size and data type for fast and efficient data accessing. A 1024-bit secret key is generated in the proposed technique by using DNA computing for improving data security, and in the novel DNA computing-based data encryption process, the same key is used for data encryption. In the proposed scheme, the secret key defines randomness as the data owner can randomly assign the DNA bases in both the key generation and data encryption phase. Here, the DO can be online to provide the secret key and other credentials to the user. Then, the DO can go offline. So, they do not need to be always online, which reduces the system overhead. The proposed scheme is secured against insider attack, password guessing attack, distributed denial of service attack, side channel attack and stolen verifier attack in the cloud environment. The results of the experiments show that the proposed scheme is much better than the other well-known existing schemes in the cloud computing environment. The main limitation of the proposed scheme is that the CSP may need to search the entire database for providing the requested data, if the user's requested data type and data size are not matched with the details of the CSP table. Therefore, the data accessing time can be increased and the users may need to pay more for using cloud service.

Credit author statement

Suyel Namasudra is the main and corresponding author of this paper, who has conceived the idea and discussed it with all co-authors. Suraj Sharma has performed the simulations of this paper. Ganesh Chandra Deka has performed the security analysis and write up of this work. Pascal Lorenz has supervised the entire work, evaluated the performance and security analysis and proof-read the paper.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

Adleman, L.M., 1994. Molecular computation of solutions to combinatorial problems. Science 266 (5187), 1021–1024.

Ajgaonkar, S., Indalkar, H., Jeswani, J., 2015. Activity based access control model for cloud computing. Int. J. Curr. Eng. Technol. 5 (2), 708–713.

Alam, Q., et al., 2017. A cross tenant access control (CTAC) model for cloud computing: formal specification and verification. IEEE Trans. Inf. Forensics Secur. 12 (6), 1259–1268.

Alguliyev, R.M., Aliguliyev, R.M., Sukhostat, L.V., 2020. Efficient algorithm for big data clustering on single machine. CAAI Trans. Intell. Technol. 5 (1), 9–14.

Almutairi, A., Sarfraz, M.I., Ghafoo, A., 2018. Risk-aware management of virtual resources in access controlled service-oriented cloud datacenters. IEEE Trans. Cloud Comput. 6 (1), 168–181.

Apache commons math, 2019. Available: http://commons.apache.org/proper/commons-math/download_math.cgi. (Accessed 13 January 2020).

Assis, M.R.M., Bittencourt, L.F., 2020. MultiCloud tournament: a cloud federation approach to prevent Free-Riders by encouraging resource sharing. J. Netw. Comput. Appl. 166 https://doi.org/10.1016/j.jnca.2020.102694.

Calheiros, R.N., Ranjan, R., Beloglazov, A., 2011. CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms. Software Pract. Ex. (SPE) 41 (1), 23–50.

CityPulse dataset collection, 2019. Available: http://iot.ee.surrey.ac.uk:8080/datasets.html. (Accessed 13 January 2020).

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

Devi, D., Namasudra, S., Kadry, S., 2020. A boosting-aided adaptive cluster-based undersampling approach for treatment of class imbalance problem. Int. J. Data Warehous. Min. (IJDWM) 16 (3) (in press).

Ferraiolo, D.F., Kuhn, D.R., 1992. Role-based access controls. In: Proceedings of the 15th National Computer Security Conference, Baltimore, USA, pp. 554–563.

Fu, J.S., Liu, Y., Chao, H.C., Bhargava, B.K., Zhang, Z.J., 2018. Secure data storage and searching for industrial IoT by integrating fog computing and cloud computing. IEEE Trans. Ind. Inf. 14 (10), 4519–4528.

Gao, X., Jiang, Z., Jiang, R., 2012. A novel data access scheme in cloud computing. In: Proceedings of the 2nd International Conference on Computer and Information Applications, pp. 124–127.

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

Hossain, M.S., Muhammad, G., 2016. Cloud-assisted industrial internet of things (IIoT)-enabled framework for health monitoring. Comput. Network. 101, 192–202.

Huth, A., Chebula, J., 2011. The Basics of Cloud Computing. Carnegie Mellon University.

Java, 2019. Available: http://java.com/en/download/index.jsp. (Accessed 13 January 2020).

Kumar, A., Abhishek, K., Singh, A.K., Nerurkar, P., Chandane, M., Bhirud, S., Patel, D., Busnel, Y., 2020. Multilabel classification of remote sensed satellite imagery. Trans. Emerg. Telecommun. Technol. https://doi.org/10.1002/ett.3988.

Li, S., Wang, G., Yang, J., 2019. Survey on cloud model based similarity measure of uncertain concepts. CAAI Trans. Intell. Technol. 4 (4), 223–230.

Lojka, T., Miškuf, M., Zolotová, I., 2016. Industrial IoT gateway with machine learning for smart manufacturing. In: Proceedings of the IFIP International Conference on Advances in Production Management Systems. Springer, pp. 759–766.

Ma, K., Yang, G., Xiang, Y., 2020. RCBAC: a risk-aware content-based access control model for large-scale text data. J. Netw. Comput. Appl. 167 https://doi.org/10.1016/j.jnca.2020.102733.

Namasudra, S., 2017. An improved attribute-based encryption technique towards the data security in cloud computing. Concurrency Comput. Pract. Ex. https://doi.org/10.1002/cpe.4364.

Namasudra, S., 2018. Cloud computing: a new era. J. Fund. Appl. Sci. 10 (2), 113–135.

Namasudra, S., 2020. Data access control in the cloud computing environment for bioinformatics. Int. J. Appl. Res. Bioinf. (IJARB) (in press).

Namasudra, S., Deka, G.C., 2018. Advances of DNA Computing in Cryptography. Taylor & Francis.

Namasudra, S., Roy, P., 2016. Secure and efficient data access control in cloud computing environment: a survey. Multiagent Grid Syst. Int. J. 12 (2), 69–90.

Namasudra, S., Roy, P., 2017a. A new table based protocol for data accessing in cloud computing. J. Inf. Sci. Eng. 33 (3), 585–609.

Namasudra, S., Roy, P., 2017b. Time saving protocol for data accessing in cloud computing. IET Commun. 11 (10).

Namasudra, S., Roy, P., 2018. PpBAC: popularity based access control model for cloud computing. J. Organ. End User Comput. 30 (4), 14–31.

Namasudra, S., Nath, S., Majumder, A., 2014. Profile based access control model in cloud computing environment. In: Proceedings of the International Conference on Green Computing, Communication and Electrical Engineering, IEEE, Coimbatore, India, pp. 1–5.

Namasudra, S., Roy, P., Balamurugan, B., 2017a. Cloud computing: fundamentals and research issues. In: Proceedings of the 2nd International Conference on Recent Trends and Challenges in Computational Models, IEEE, Tindivanam, India.

Namasudra, S., Roy, P., Balamurugan, B., Vijayakumar, P., 2017b. 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), IEEE, Coimbatore, India.

Namasudra, S., 2018a. Taxonomy of DNA-based security models. In: Namasudra, S., Deka, G.C. (Eds.), Advances of DNA Computing in Cryptography. Taylor & Francis, pp. 53–68.

Namasudra, S., Devi, D., Choudhary, S., Patan, R., Kallam, S., 2018b. Security, privacy, trust, and anonymity. In: Namasudra, S., Deka, G.C. (Eds.), Advances of DNA Computing in Cryptography. Taylor & Francis, pp. 153–166.
Namasudra, S., Deka, G.C., 2018c. Introduction of DNA computing in cryptography. In: Namasudra, S., Deka, G.C. (Eds.), Advances of DNA Computing in Cryptography. Taylor & Francis, pp. 27–34.
Namasudra, S., Deka, G.C., Bali, R., 2018d. Applications and future trends of DNA computing. In: Namasudra, S., Deka, G.C. (Eds.), Advances of DNA Computing in Cryptography. Taylor & Francis.
Namasudra, S., Deka, G.C., Johri, P., Hosseinpour, M., Gandomi, A.H., 2020a. The Revolution of blockchain: state-of-the-art and research challenges. Arch. Comput. Methods Eng. (in press).
Namasudra, S., Chakraborty, R., Kadry, S., Manogaran, G., Rawal, B.S., 2020b. FAST: fast accessing scheme for data transmission in cloud computing. Peer-to-Peer Network. Appl. https://doi.org/10.1007/s12083-020-00959-6.
Namasudra, S., Devi, D., Kadry, S., Sundarasekar, R., Shanthini, A., 2020c. Towards DNA based data security in the cloud computing environment. Comput. Commun. 151, 539–547.
Namasudra, S., Chakraborty, R., Majumder, A., Moparthi, N.R., 2020d. Securing multimedia by using DNA based encryption in the cloud computing environment. ACM Trans. Multimed Comput. Commun. Appl. (in press).
Raghavendra, S., et al., 2016. Index generation and secure multi-user access control over an encrypted cloud data. Procedia Comput. Sci. 89, 293–300.
Sajid, A., Abbas, H., Saleem, K., 2016. Cloud-assisted IoT-based SCADA systems security: a review of the state of the art and future challenges. IEEE Access 4, 1375–1384.
Sarkar, M., Saha, K., Namasudra, S., Roy, P., 2015. An efficient and time saving web service based android application. SSRG Int. J. Comput. Sci. Eng. 2 (8), 18–21.
Sheela, S.J., Suresh, K.V., Tandur, D., 2017. A novel audio cryptosystem using chaotic maps and DNA encoding. J. Comput. Netw. Commun. 2017, 1–12.
Tanaka, K., Okamoto, A., Saito, I., 2005. Public-key system using DNA as a one-way function for key distribution. Biosystems 81 (1), 25–29.
Thaseen, I.S., Banu, J.S., Lavanya, K., Ghalib, M.R., Abhishek, K., 2020. An integrated intrusion detection system using correlation-based attribute selection and artificial neural network. Trans. Emerg. Telecommun. Technol. https://doi.org/10.1002/ett.4014.
Tripura, J., Roy, P., 2017. Flow forecasting in multiple sections of a river system. KSCE J. Civ. Eng. 21 (2), 512–522.
Tripura, J., Roy, P., Barbhuiya, A.K., 2018. Application of RBFNNs incorporating MIMO processes for simultaneous river flow forecasting. J. Eng. Technol. Sci. 50 (3), 434–449.
Tripura, J., Roy, P., Barbhuiya, A.K., 2020. Simultaneous streamflow forecasting based on hybridized neuro-fuzzy method for a river system. Neural Comput. Appl. https://doi.org/10.1007/s00521-020-05194-x.
Tuncer, U., Avci, E., 2016. A reversible data hiding algorithm based on probabilistic DNA-XOR secret sharing scheme for color images. Displays 41, 1–8.
Wan, Z., Liu, J., Deng, R.H., 2012. HASBE: a hierarchical attribute-based solution for flexible and scalable access control in cloud computing. IEEE Trans. Inf. Forensics Secur. 7 (2), 743–754.
Wang, B., Xie, Y., Zhou, S., Zhou, C., Zheng, X., 2017. Reversible data hiding based on DNA computing. Comput. Intell. Neurosci. https://doi.org/10.1155/2017/7276084.
Wang, X., Hou, Y., Wang, S., Li, R., 2018. A new image encryption algorithm based on CML and DNA sequence. IEEE Access 6, 62272–62285.
Wang, G., Nixon, M., Boudreaux, M., 2019a. Toward cloud-assisted industrial IoT platform for large-scale continuous condition monitoring. Proc. IEEE 107 (6), 1193–1205.
Wang, Y., Han, Q., Cui, G., Sun, J., 2019b. Hiding messages based on DNA sequence and recombinant DNA technique. IEEE Trans. Nanotechnol. 18, 299–307.
Xie, Y., Wen, H., Wu, B., Jiang, Y., Meng, J., 2019. A modified hierarchical attribute-based encryption access control method for mobile cloud computing. IEEE Trans. Cloud Comput. 7 (2), 383–391.
Zhao, X., Li, R., Zuo, X., 2019. Advances on QoS-aware web service selection and composition with nature-inspired computing. CAAI Trans. Intell. Technol. 4 (3), 159–174.

Dr. Suyel Namasudra is an Assistant Professor in the Department of Computer Science and Engineering at the National Institute of Technology Patna, Bihar, India. Prior to joining the National Institute of Technology Patna, Dr. Namasudra was an Assistant Professor in the Department of Computer Science Engineering at the Bennett University, India. He has received PhD in Computer Science and Engineering from National Institute of Technology Silchar, Assam, India. His research interests include Cloud Computing, Information Security, DNA Computing and Blockchain. Dr. Namasudra has edited 1 book and 25 publications in refereed journals, book chapters and conference proceedings. He has participated in many international conferences as an Organizer and Session Chair. Dr. Namasudra is a member of the Editorial Board and Reviewer of many journals.

Dr. Suraj Sharma is currently working as an Assistant Professor in the Department of Computer Science and Engineering at International Institute of Information Technology Bhubaneswar. He has completed M.Tech. & Ph.D. from National Institute of Technology Rourkela. His research area includes IoT, Edge Computing, Cloud Computing, BlockChain, Wireless Sensor Networks, Security, Localization, Energy Modeling and QoS-based estimation.

Ganesh Chandra Deka is the Deputy Director (Training) at Regional Directorate of Skill Development and Entrepreneurship, Directorate General of Training, Ministry of Skill Development and Entrepreneurship, Government of India, New Delhi-110001, India. His research interests include NoSQL Database, Blockchain technology and Bigdata Analytics. He has authored 2 books on Cloud Computing published by LAP Lambert, Germany. He is the Co-author of 4 text books on Fundamentals of Computer Science (3 books published by Moni Manik Prakashan, Guwahati, Assam, India and 1 IGI Global, USA). Till now, he has edited 19 books (6 IGI Global, USA, 7 CRC Press, USA, 3 Elsevier and 3 Springer including 1 International Conference proceeding) on Big data, NoSQL, Blockchain Technology and Cloud Computing in general and authored 10 Book Chapters. He has published 8 research papers in various reputed journals including Elsevier (1) and IEEE (2), and already published around 47 research papers in various IEEE conferences. He is the Editor-in-Chief of the International Journal of Computing, Communications and Networking. He has published 4 Special Issues as Guest Editor in different International Journals, which are indexed in SCI and SCOPUS. Deka has organized 08 IEEE International Conferences as Technical Chair. He is the Member of the editorial board and reviewer for various International Journals.

Pascal Lorenz received his M.Sc. (1990) and Ph.D. (1994) from the University of Nancy, France. Between 1990 and 1995 he was a research engineer at WorldFIP Europe and at Alcatel-Alsthom. He is a professor at the University of Haute-Alsace, France, since 1995. His research interests include QoS, wireless networks, high-speed networks and DNA Computing. He is the author/co-author of 3 books, 3 patents and 200 international publications in refereed journals and conferences. He was Technical Editor of the IEEE Communications Magazine Editorial Board (2000–2006), IEEE Networks Magazine since 2015, IEEE Transactions on Vehicular Technology since 2017, Chair of IEEE ComSoc France (2014–2018), Financial chair of IEEE France (2017–2019), Chair of Vertical Issues in Communication Systems Technical Committee Cluster (2008–2009), Chair of the Communications Systems Integration and Modeling Technical Committee (2003–2009), Chair of the Communications Software Technical Committee (2008–2010) and Chair of the Technical Committee on Information Infrastructure and Networking (2016–2017). He has served as Co-Program Chair of IEEE WCNC′2012 and ICC′2004, Executive Vice-Chair of ICC′2017, TPC Vice Chair of Globecom'2018, Panel sessions co-chair for Globecom'16, tutorial chair of VTC′2013 Spring and WCNC′2010, track chair of PIMRC′2012 and WCNC′2014, symposium Co-Chair at Globecom 2007–2011, Globecom'2019, ICC 2008–2010, ICC′2014 and ′2016. He has served as Co-Guest Editor for special issues of IEEE Communications Magazine, Networks Magazine, Wireless Communications Magazine, Telecommunications Systems and LNCS. He is Associate Editor for International Journal of Communication Systems (IJCS-Wiley), Journal on Security and Communication Networks (SCN-Wiley) and International Journal of Business Data Communications and Networking, Journal of Network and Computer Applications (JNCA-Elsevier). He is a senior member of IEEE, IARIA Fellow and member of many international program committees. He has organized many conferences, chaired several technical sessions and gave tutorials at major international conferences. He was IEEE ComSoc Distinguished Lecturer Tour during 2013–2014.