Automated Animal/Human Sperm Analysis System

Dr. Ketki P. Kshirsagar

Vishwakarma Institute of Information Technology, Pune, Maharashtra, India

Dr. RajeshreeAmitkumarShinde

LokmanyaTilak College of Engineering, Koperkhairne, Mumbai, India

ABSTRACT

Sperm analysis (SA) technology was developed in the late 1980s for analyzing sperm movement characteristics or kinematics and has been highly successful in enabling this field of research. The technology has also been used with great success for measuring semen characteristics such as sperm concentration and proportions of progressive motility in many animal species, including wide application in domesticated animal production laboratories and reproductive toxicology. However, attempts to use SA for human clinical semen analysis have largely met with poor success due to the inherent difficulties presented by many human semen samples caused by sperm clumping and heavy background debris that, until now, have precluded accurate digital image analysis. Specific requirements for validating SA technology as a semi-automated system for human semen analysis are also provided, with particular reference to the accuracy and uncertainty of measurement expected of a robust medical laboratory test for implementation in clinical laboratories operating according to modern accreditation standards.

This paper is developedAutomated Sperm Analysis (ASA) system. ASA system that has progressed to solve many of the above mentioned image analysis problems. The evolution system comprises four modules (concentration and motility, morphology, vitality and fragmentation) that are fully integrated and associated with an extensive database that will allow results import and export in many of the formats used by hospitals. It offers several platforms from human and veterinary to research and several others including automatic stage, data share and capture stations far away from the analysis system.

ASA system is address sperm morphology analysis, as well as incorporating tests of sperm function into the platforms. Future ASA systems could provide improved clinical relevance for semen analysis by integrating the automated analysis of semen parameters and sperm functional assessments, and perhaps even incorporating molecular biological aspects of sperm analysis, into a database with specialized statistical analytical capabilities. This type of approach could also be adopted for the domestic animal industry to better predict the relative fertility of sires.

The proposed ASA system shows betterment in existing analysis technologies. The proposed system improve sperm feature issues, based on the cost-effective availability of higher resolution digitizers and greater computing power combined with software features such as automated and/or interactive illumination control, advanced brownian motion filtering, drift filtering, tail detection, smart tracking through collisions, adaptive smoothing to derive the average path, and the introduction of fractals as a kinematic measure, have given us ASA instruments that are certainly more user-friendly in both the human clinical and domestic animal fields.

Homomorphic Map Based Salt Encryption and Decryption for Secured Cloud Data Storage and Accessibility

M.Gomathe, S.Prasanna
VISTAS, Pallavaram, Chennai.

ABSTRACT

Cloud data storage and access has always been an important issue for online data sharing in cloud environment. Recently, how to deal with the key pair for signature generation and verification in the settings of cloud data access security has been proposed and studied. To address the challenge, existing solutions all requires to generate the pairing-based cryptography, which may inevitably bring in new local burdens to the client, with less focus made on security aspect. In this paper, focus is made on data access security and proposes a new paradigm called identity-based data access security with cryptographic techniques. The proposed method is called as IDentity based Boneh–Franklin Cryptographic Data Accessing (ID-BFCDA) for secured cloud data storage and accessibility. In this paradigm, a cloud user ID (i.e. public key) is generated by the Secret Key Generator (SKG) using Row Modulus Setup and thus the key update burden on the client cloud user is said to be kept minimal. The SKG in our work play the role of authentic party and make it in charge of both the data storage and providing public key for key-exposure resistance through Homomorphism Map-based Extraction. Here, cloud user only generate encrypted data and sends it along with the public key to the cloud server via Salt Homomorphic Hash Encryption. In order to make it more laborious to reveal a hash, salting is performed where an extra bit is appended to the password before we hash it. In addition, the design also supplies secured data access by including authentication mechanism by means of Homomorphic Hash Decryption. Extensive security and performance analysis show the proposed method is provably secure and highly efficient in terms of both data storage and accessibility. Experiments conducted on Amazon EC2 moreover signify the fast performance of the design.

***KEYWORDS**: Salt, Homomorphism Map, Encryption, Decryption, Row Modulus, Secret Key Generator.*

INTRODUCTION

With the swift development of cloud computing, several clients store their data to public cloud servers. Hence, new security problems have to be addressed to assist more client’s process their data in public cloud. A new proxy-oriented data uploading and remote data integrity checking model was introduced in [1], called, Identity-based Proxy-oriented data Uploading and remote data Integrity checking in public Cloud (ID-PUIC). A new architecture, called, Dropping Activation Outputs with Localized First-layer Deep Network (DAO-LFDN) was constructed in [2] for deep network where users do not provide their original data, therefore ensuring user privacy and data security. The entire encryption process followed by the users was performed by combining both feed-forward propagation and data encryption into single process.

Identity-Based Encryption with Equality Test (IBEET) scheme has in the present identified as a promising solution where the users have the potentiality to delegate trapdoor to the power system control server. The power system control server then searched on the encrypted data to identify whether two different ciphertexts were encryptions of the same plaintext. With the application of IBEET scheme using bilinear pairing, it was found that the time consuming Hash to Point Function was said to be reduced in [3].

RELATED WORKS

Secure and efficient data storage and accessing through physical devices proven to be authenticated are considered to be challenging due to the different set of devices required to access the services and data. In [4], the prime requirement of the combined use of encryption and watermarking mechanisms in the database was studied. An oblivious similarity based searching for encrypting the data that has been outsourced to outside domain using bloom filter and probabilistic homomorhpic mechanism was designed in [5], resulting in the reduction of search space. With this aspect, a new concept in cloud computing called, data provenance was investigated in [6] that in turn minimized the inclusion of third party. In [7], computing models designed on the basis of providing resources in a voluntarily manner with the objective of creasing ad hoc clouds and harnessing computing at the edge of the network was designed. Division and replication of data was investigated in [8] based on fragmentation and dispersal towards optimal performance and security.

In summary, works on security and privacy issues on cloud related to storage and access have mostly been focused on identifying private remote integrity. However, it does not provide any guideline on how such requirements should be checked for ensuring both secured cloud storage and access in cloud environment. Also, very limited works have been taken in place for analyzing data integrity and data confidentiality. Our work intends to fill the gap of the current state of the art by presenting a novel method that combines data integrity and data confidentiality with the secured cloud data storage requirements so that appropriate cloud deployment model could be selected to support data access security.

1. METHODOLOGY

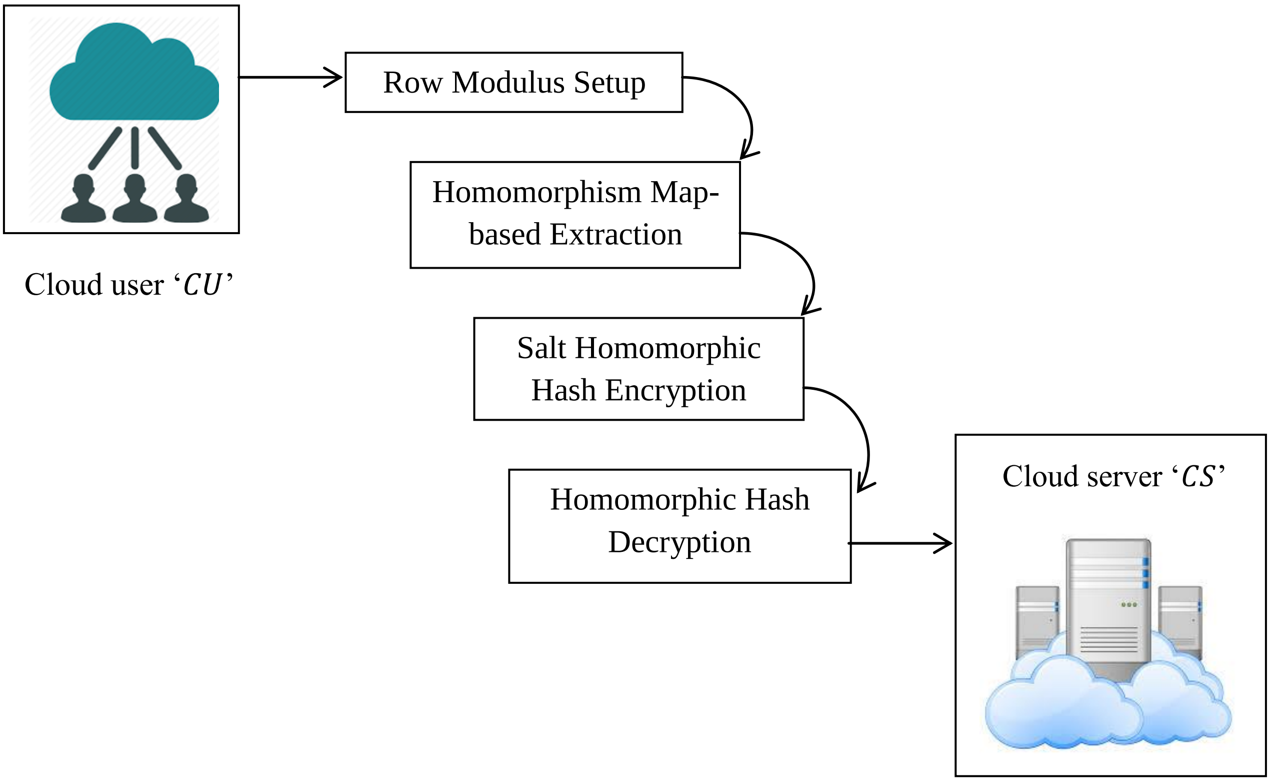

Row Modulus Setup
Homomorphism Map-based Extraction
Salt Homomorphic Hash Encryption
Homomorphic Hash Decryption
Cloud user ‘CU’
Cloud server ‘CS’

Fig 1: Block diagram of IDentity based Boneh–Franklin Cryptographic Data Accessing

With the swift development of computing and communication method, numerous data have to be dealt with. To improve the data storage and access security in cloud environment, IDentity based Boneh–Franklin Cryptographic Data Accessing (ID-BFCDA) method is designed. The main objective of ID-BFCDA method is to ensure secured cloud data storage and access with lesser time consumption and higher security level.

2.1 Row Modulus Setup

The process starts with the initial cloud user registration with the cloud server (CS). Upon successful registration, the setup or key generation process comes into foreground using Row Modulus Setup. Here, the Secret Key Generator (SKG) generates masker key 'MK', public key 'PK', i.e. cloud user ID '$CU(ID)$', along with the system parameter '$PARAMS$' as output for every registered user to store and access the data present in the cloud server using row modulus hash function.

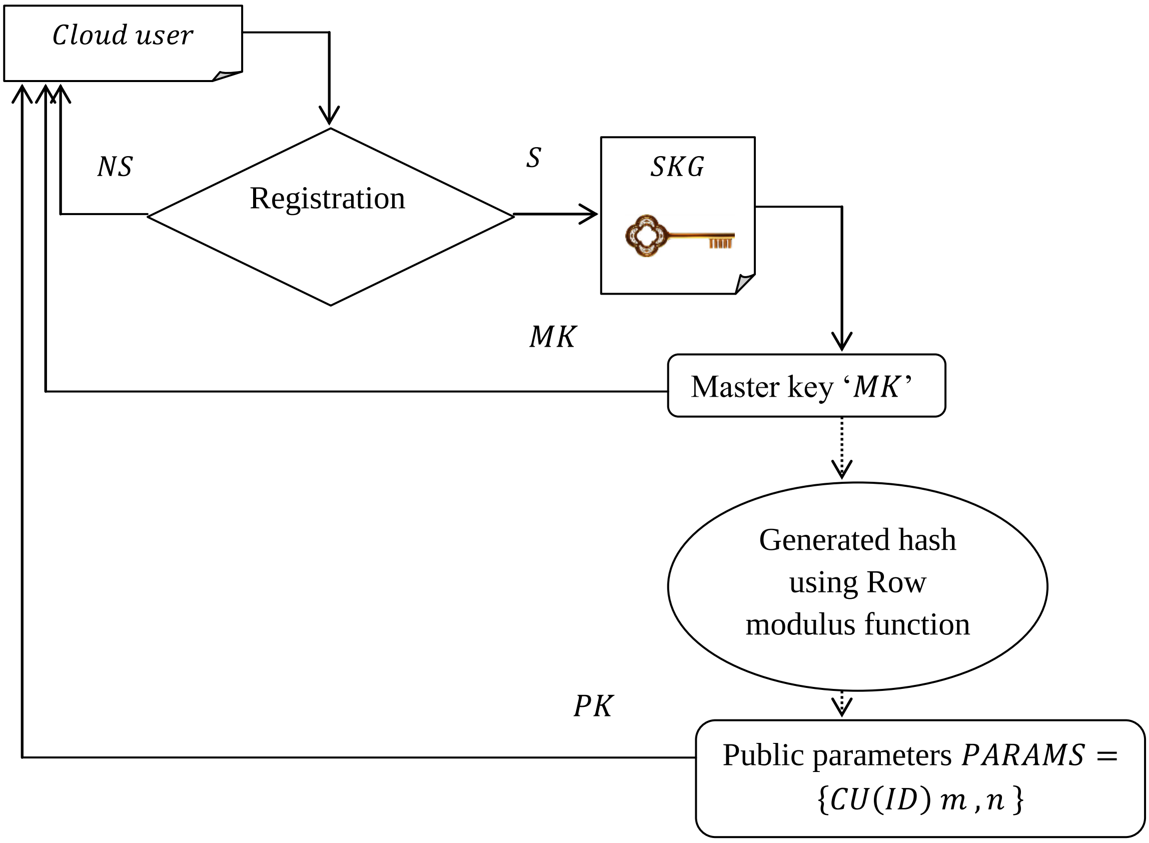

Cloud user
Registration
NS
S
SKG
MK
Master key 'MK'
Generated hash using Row modulus function
PK
Public parameters PARAMS = {CU(ID) m ,n }

Fig: 2 Block diagram of Row Modulus Setup

Here, the master key is represented in the form of matrix. Besides, the secret key generator selects two positive integer 'm' and 'n' as the row count and column count of the matrix. The matrix is represented with Master Key Matrix or 'MKM'. The Master Key Matrix or 'MKM' is represented as given below.

$$MKM = \begin{bmatrix} r_{11} & r_{12} & \ldots.. & r_{1n} \\ r_{21} & r_{22} & \ldots.. & r_{2n} \\ \ldots.. & \ldots.. & \ldots.. & \ldots.. \\ r_{m1} & r_{m2} & \ldots.. & r_{mn} \end{bmatrix} \qquad (1)$$

$$PPM = MKM * h \qquad (2)$$

$$\begin{bmatrix} Q_{11} & Q_{12} & \ldots.. & Q_{1n} \\ Q_{21} & Q_{22} & \ldots.. & Q_{2n} \\ \ldots.. & \ldots.. & \ldots.. & \ldots.. \\ Q_{m1} & Q_{m2} & \ldots.. & Q_{mn} \end{bmatrix} \qquad (3)$$

$$PPM = \begin{bmatrix} r_{11} * h & r_{12} * h & \ldots.. & r_{1n} * h \\ r_{21} * h & r_{22} * h & \ldots.. & r_{2n} * h \\ \ldots.. & \ldots.. & \ldots.. & \ldots.. \\ r_{m1} * h & r_{m2} * h & \ldots.. & r_{mn} * h \end{bmatrix} \qquad (4)$$

$$h \leftarrow CU_i \,\%\, m \qquad (5)$$

The pseudo code representation of Row Modulus Key Generation (RMKG) algorithm is given below.

Input: Cloud Users '$CU = CU1, CU2, \ldots, CUn$', Cloud Server '$CS$', Security Parameter '$SP = SP_1, SP_2, \ldots, SP_n$'

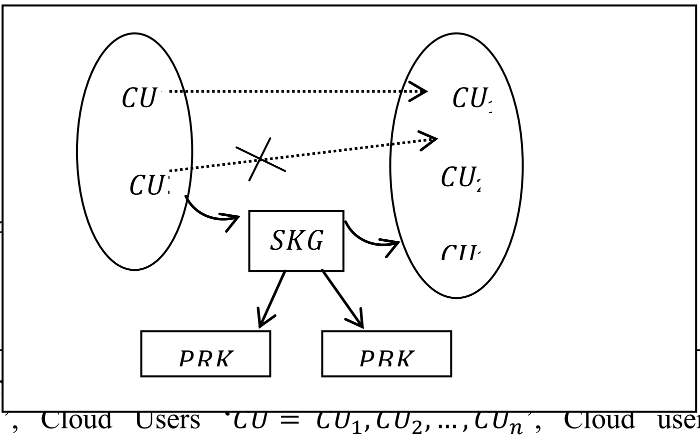
Output: System parameter '$PARAMS = \{PPM, m, n\}$', Master Key '$MK = MK_1, MK_2, \ldots, MK_n$', Public Key or Cloud user ID '$CU(ID)$'

```
Begin
    For each Cloud Users '$CU$' with Security Parameter '$SP$'
        Generate master key matrix to obtain the master key using equation (1)
        Obtain public parameter matrix using equation (2)
        Obtain the hash value using equation (4)
        Return '$PARAMS$', '$MK$', '$CU(ID)$'
    End for
End
```

Algorithm 1 Row Modulus Key Generation algorithm

2.2 Homomorphism Map-based Extraction

Upon successful generation of master key along with the cloud user ID for the registered cloud users, the Secret Key Generator (SKG) provides credentials for each of the registered cloud user by verifying the identity. This is performed via Homomorphism Map-based Extraction.



$CU_1, CU_2, \ldots\ldots, CU_n \leftarrow MAP$

$PRK = r_i CU_i$

$PBK = Q_i CU_i \qquad (8)$

Input: System par Key '$MK = MK_1, MK_2, \ldots, MK_n$', Cloud Users '$CU = CU_1, CU_2, \ldots, CU_n$', Cloud user ID '$CU(ID)$'

Output: Private Key 'PRK', public key 'PBK'

```
Begin
```

```
2:   **For** each Cloud Users '$CU$' with system parameter '$PARAMS$' and Master Key '$MK$'
3:       Obtain private key '$PRK$' using equation (7)
4:       Obtain public key '$PBK$' using equation (8)
5:       Return Private Key '$PRK$' and public key '$PBK$'
6:   **End for**
7: **End**
```

Algorithm 2 Homomorphism Map Extraction

2.3 Salt Homomorphic Hash Encryption and Decryption

After receiving the Cloud user ID 'CU_{ID}' and key, the Private Key 'PRK'and the public key 'PBK', the cloud user encrypts the data 'D' with public key and sends to cloud server for storing the data.

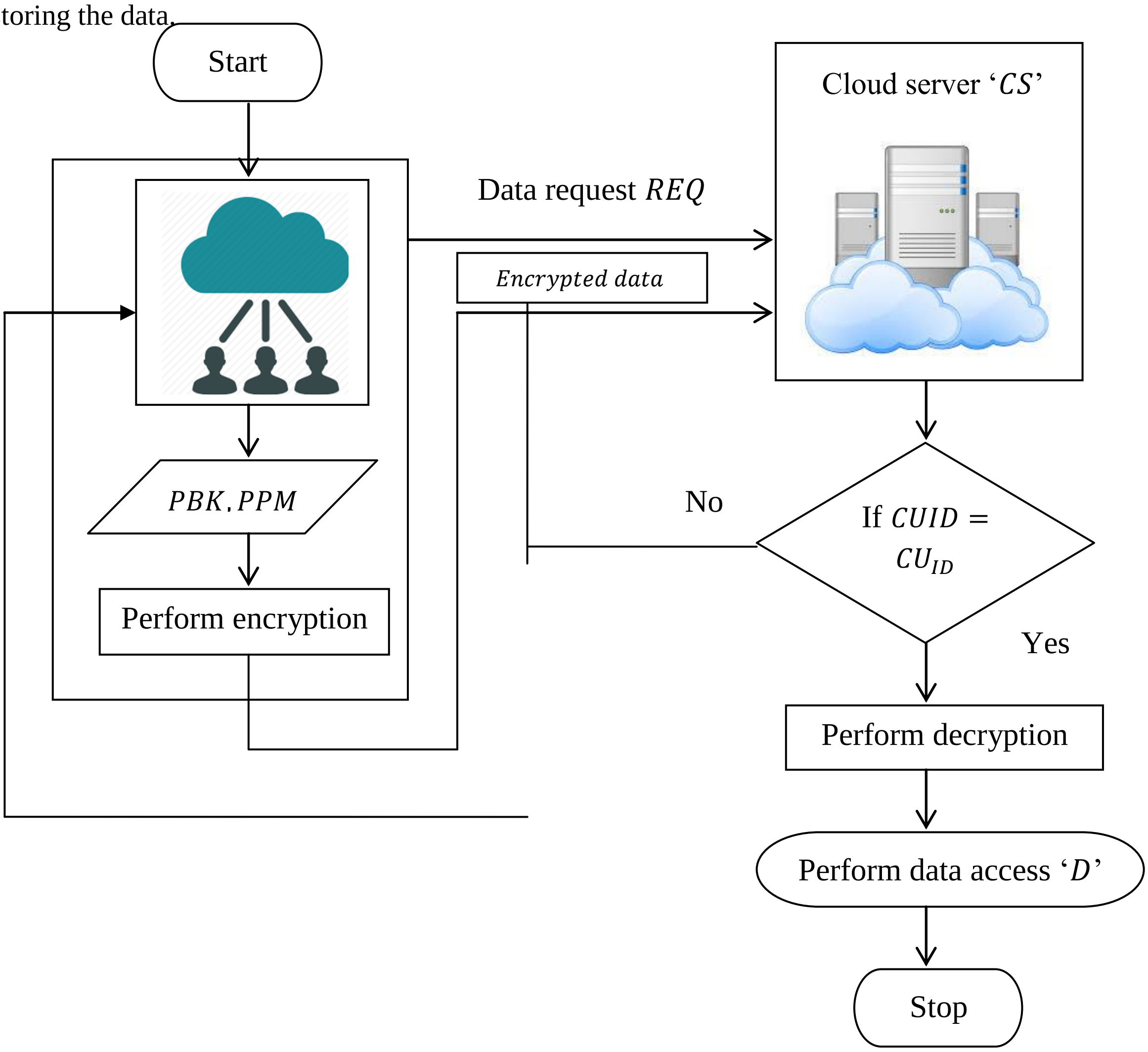


Fig: 4 Flow diagram of Salt Homomorphic Hash Encryption and Decryption

The sample cloud user ID along with the security parameter with which the salt value is generated to produce the resultant randomized number is given in the table below.

$$R = [CU_{ID}\ [SP] + SV]\ (9)$$

Table 1 Salt randomized number

Cloud user ID 'CU_{ID}'	**Security Parameter 'SP'**	**Salt value 'SV'**	**Resultant Random Number 'R'**
CU_1	$X52E$	$98123SICKWL$	$X52E98123SICKWL$
CU_2	$1RT5$	$PLKI4MBL3$	$1RT5PLKI4MBL3$
CU_3	$5IOv$	$KTICDR4512$	$5IOvKTICDR4512$

With the resultant salt randomized number, the Homomorphic Hash Encryption is performed as given below.

$$C = [R, D \oplus h(PBK)^R] \qquad (10)$$

From the above equation (10), the result of the encryption function '$ENC()$ ' is stored in the cipher text 'C' using both the public key 'PBK' with respect to the random number 'R'. The pseudo code representation of Salt Homomorphic Hash Encryption is given below.

```
Input: Cloud user ID '$CU(ID)$', Private Key '$PRK$', public key '$PBK$', Data '$D = \{D_1, D_2, ... D_n\}$', random number '$R$', Cloud Server '$CS$'
Output: Cipher text '$C = (x, y)$'
Begin
    For each Cloud user ID '$CU(ID)$' and public key '$PBK$' with random number '$R$'
        Obtain salt randomized number using equation (9)
        Perform encryption function using equation (10)
        Send the encrypted data to Cloud Server '$CS$'
    End for
End
```

Algorithm 3 Salt Homomorphic Hash Encryption

Once proven with authentication, the particular cloud user is allowed to access the data from cloud server and the data gets decrypted with private key of cloud user. This is mathematically formulated as, $D = [y \oplus h(PRK_{ID}, x)]$ (11)

The pseudo code for Homomorphic Hash Decryption is given below.

```
Input: Cipher text '$C = (x, y)$', Cloud user ID '$CU(ID)$', Private Key '$PRK$', public key '$PBK$', Data '$D = \{D_1, D_2, ... D_n\}$', random number '$R$', Cloud Server '$CS$'
Output: Cipher text '$C$'
Begin
//Decryption
    For each Cloud user ID '$CU(ID)$' with Cipher text '$C = (x, y)$'
        If '$CUID = CU_{ID}$ ' then
            Cloud user is said to be authenticated user
            Perform decryption function using equation (11)
        Else
            Cloud user is not an authenticated user
        End if
    End for
End
```

Algorithm 4 Homomorphic Hash Decryption

Table 2 Tabulation for encryption time

Cloud users	**Encryption time (ms)**		
	ID-BFCDA	**ID-PUIC**	**DAO-LFDN**
100	1.352	1.858	1.653
125	1.653	1.932	1.825
150	1.135	1.245	1.753
175	1.145	1.452	1.813
200	1.59	1.325	1.568
225	1.625	1.532	1.813
250	1.893	1.925	1.752

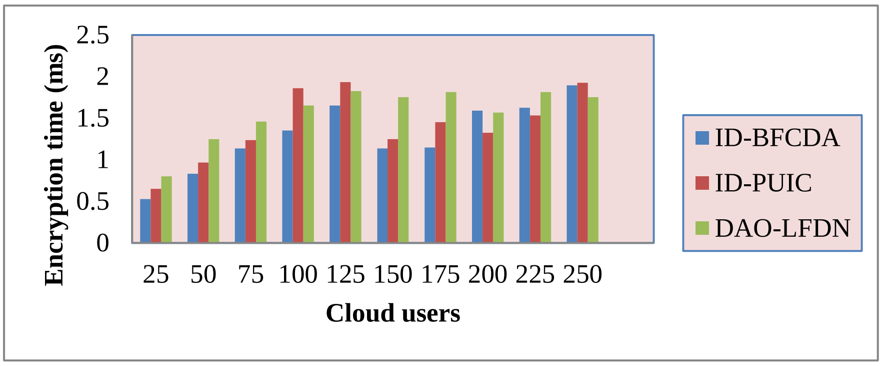


Figure 5 Graphical representation of encryption time using ID-BFCDA, ID-PUIC and DAO-LFDN

Table 3 Tabulation for cloud data integrity

Cloud users	**Cloud data integrity (%)**		
	ID-BFCDA	**ID-PUIC**	**DAO-LFDN**
100	88.13	80.14	83.25
125	85.56	78.22	85.14
150	88.32	75.32	71.32
175	89.15	79.15	75.56
200	90.33	82.35	79.13
225	92.15	84.55	85.56
250	94.5	86.16	82.13

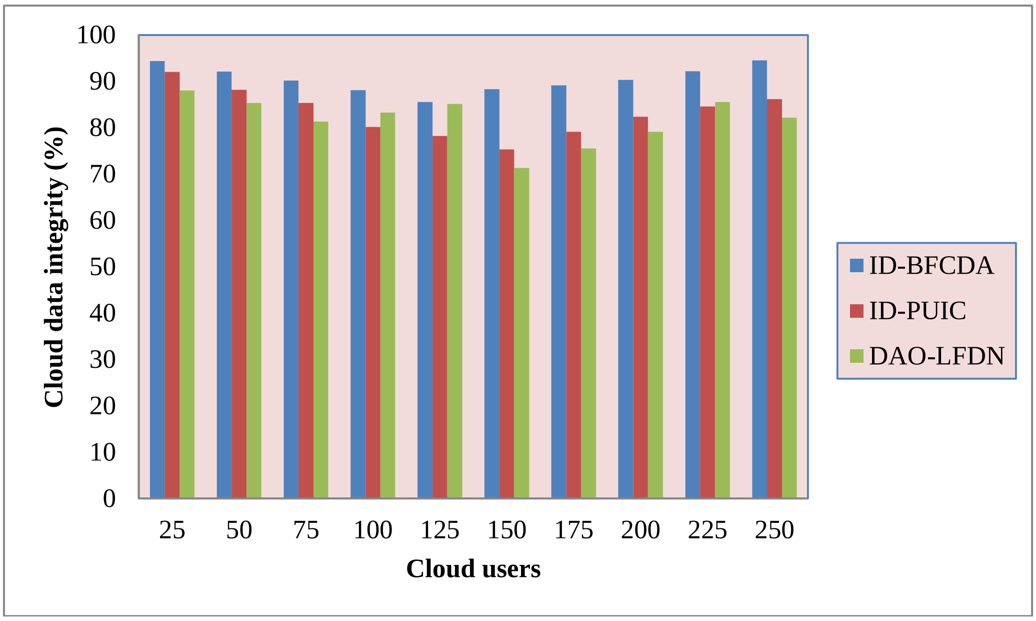


Figure 6 Graphical representation of cloud data integrity using ID-BFCDA, ID-PUIC and DAO-LFD

CONCLUSION

In this paper, we introduced the IDentity based Boneh–Franklin Cryptographic Data Accessing (ID-BFCDA) method for secured cloud data storage and accessibility in cloud environment. The ID-BFCDA method seamlessly incorporates a Homomorphism Map by applying a Homomorphism Map Extraction algorithm to data storage and accessing. ID-BFCDA not only supports data confidentiality due to Homomorphism Map-based Extraction, but also achieves data integrity because of Salt and Homomorphic Hash Encryption and Decryption. Finally, the proposed method was implemented and conducted comprehensive performance analysis and evaluation, which showed its efficiency and advantages over existing state-of-the-art works. The experimental results demonstrates that ID-BFCDA method provides better performance with an improvement of data integrity by 10% and minimizing the computational overhead by 30% when compared to the state-of-the-art works.

REFERENCES

[1] Huaqun Wang, Debiao He, Shaohua Tang, “Identity-Based Proxy-Oriented Data Uploading and Remote Data Integrity Checking in Public Cloud”, IEEE Transactions on Information Forensics and Security, Volume 11, Issue 6, June 2016, Pages 1165 – 1176

[2] Hao Dong, Chao Wu, Zhen Wei, and Yike Guo, “Dropping Activation Outputs with Localized First-Layer Deep Network for Enhancing User Privacy and Data Security”, IEEE Transactions on Information Forensics and Security, Volume 13, Issue 3, March 2018, Pages 662-670

[3] Libing Wu, Yubo Zhang, Kim-Kwang Raymond Choo and Debiao He, “Efficient identity-based encryption scheme with equality test in smart city”, IEEE Transactions on Sustainable Computing, Volume 3, Issue 1, January-March 2018, Pages 44 - 55

[4] Murat Yesilyurt and Yildiray Yalman, “New approach for ensuring cloud computing security: using data hiding methods”, Sādhanā, Springer, November 2016, Volume 41, Issue 11, November 2016, Pages 1289–1298

[5] Zeeshan Pervez , Mahmood Ahmad, Asad Masood Khattak, Naeem Ramzan, Wajahat Ali Kha, “OS2: Oblivious similarity based searching for encrypted data outsourced to an untrusted domain”, PLOS ONE | https://doi.org/10.1371/journal.pone.0179720 July 10, 2017

[6] Muhammad Imran, Helmut Hlavacs, Inam Ul Haq, Bilal Jan, Fakhri Alam Khan, Awais Ahmad, “Provenance based data integrity checking and verification in cloud environments”, PLOS ONE | https://doi.org/10.1371/journal.pone.0177576 May 17, 2017

[7] Blesson Varghese, Rajkumar Buyya, “Next generation cloud computing: New trends and research Directions”, Future Generation Computer Systems, Elsevier, Sep 2017

[8] Mazhar Ali, Kashif Bilal, Samee U. Khan Bharadwaj Veeravalli, Keqin Li, and Albert Y. Zomaya, “DROPS: Division and Replication of Data in Cloud for Optimal Performance and Security”, IEEE Transactions on Cloud Computing (Volume: 6, Issue: 2, April-June 1 2018)