

Biometric Inspired Homomorphic Encryption Algorithm for Secured Cloud Computing

Yogesh Bala and Amita Malik

Abstract Cloud computing widely uses resource sharing and computing framework over the Internet. Data security is the key objective while sharing data over untrusted environment. This paper presents a novel biometric inspired homomorphic encryption algorithm (BIHEA) for secured data/files transmission over hybrid cloud environment. The proposed algorithm encrypts the user data at run-time by providing the authorized user biometric-feature-based one time password. Every time a user is authenticated by a totally different one time password. The BIHEA provides a good solution to commonly identified theft seen in cloud environment like phishing, shoulder surfing.

Keywords One time password • Homomorphic encryption • Biometric • Cloud computing

1 Introduction

Cloud computing has recently drawn extensive attention to the organization and business community. Cloud computing provides the platform over which the sharing of resources over the Internet can be done efficiently [1]. There are various cloud models that have been developed, and we can describe these models in terms of ‘Z as Utility’ where Z may represent as hardware infrastructure, application software or storage infrastructure etc. Amazon Microsoft Azure and Salesforce.com are the successfully implemented cloud computing platforms over which the available resources are shared ubiquitously at low costs. Cloud computing implementation in

Y. Bala (✉) · A. Malik
Deenbandhu Chhotu Ram University of Science and Technology,
Murthal, Sonapat, Haryana, India
e-mail: joinyogeshbala@gmail.com

A. Malik
e-mail: amitamalik.cse@dcrustm.org

real time has many challenges. As we know that the major concern of any business organization is the security and confidentiality of its information and data.

In the growing era of cloud computing, organizations are placing their data on the cloud to achieve the benefits provided by the cloud such as service flexibility, multitenancy and configurable computing resources which help them to expand their business with minimum effort, time and cost. But the privacy and data security still remains the key concern for the organizations in adoption of clouds. As the data is in the hand of third party, many encryption algorithms have been proposed by researchers to provide the security to stored data. Uma Somani et al. [1] have implemented the concept of RSA encryption along with the digital signature that results in enhancing the data security of cloud in cloud computing. Yu et al. [2] have described a cryptographic method that improves the security and confidentiality of prioritized information on cloud server. Tirthani and Ganesan [3] have presented Diffie-Hellman Key Exchange algorithm using elliptic curve cryptography for efficient transfer of encrypted data. In this paper, authors have used a traditional one-tier authentication which is vulnerable to security attacks. Arasu et al. [4] have proposed the method that concatenates message, hash function and key which helps in ensuring the authenticate message delivery. The method implements a single-tier authentication and hence not a suitably strong for cloud environment. Rivest et al. [5] introduced the concept of homomorphic encryption which enables the computation of encrypted data without using the secret key. Thus, it facilitates to perform operations on the encrypted data without decrypting it. Before the introduction of homomorphic encryption, it was not possible to perform operation on encrypted data, so we have to decrypt the data on the cloud server before performing any calculation on the data. So, the homomorphic encryption allows the cloud provider to perform the operations on encrypted data without decrypting it.

In this paper, we address this open issue and propose a two-tier bio-inspired homomorphic encryption algorithm that provides a secured data access scheme at two layers over unreliable cloud computing media. Our proposed system is based on the knowledge that in real-time scenarios, all the information/message can be encrypted by defining a key component. The same secret keys are shared among users which allow a user to decrypt the encrypted data, only if the key matches with the generated key. Key is generated using the biometric-feature-based algorithm in which multiple keys are created that are dependent on the biometric feature of user in real-time scenario. At a time, one key is passed to the other user as one time password for authentication and to decrypt the stored data at cloud servers [6–10]. Such a design also brings about confidentiality, security and authorization of data access on cloud. Only the data owner can grant the permission to access the data, without any such permission the user will not be able to access the data.

The rest of the paper is organized as follows. Section 2 discusses the homomorphic encryption algorithm. Section 3 presents the proposed BIHE algorithm. In Sect. 4, we analyse our proposed system in terms of its security features and time complexity. We conclude this paper in Sect. 5.

2 Homomorphic Encryption Algorithm

Homomorphic encryption is the cryptography technique that enables to communicate multiple number of parties in cooperation to generate the ciphertext without the knowledge of plain text. Thus, homomorphic encryption applies the algebraic operations on the ciphertext, without deciphering it to plain text. The homomorphic encryption technique can be expressed as follows.

Consider $E(x)$ be the function defined for performing encryption and m_1 and m_2 be two plain texts. C_1 and C_2 are the ciphertexts given as in Eqs. 1 and 2.

$$c = E(m)_{11} \quad (1)$$

$$c = E(m)_{22} \quad (2)$$

$$c \times c = E(m + m)_{1212} \quad (3)$$

Equation 3 performs the algebraic operations x on C_1 and C_2 as shown in Eq. 3. Here $+$ is also the mathematical operation applied over messages. In this paper, homomorphic encryption is chosen as an encryption method to encrypt the user data over cloud.

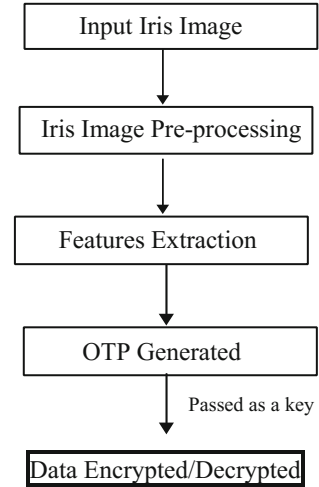
3 Biometric Inspired Homomorphic Encryption Algorithm

This section presents a BIHEA. The BIHEA allows the data to be encrypted using key generated as one time password (OTP) from iris of the registered user over the cloud, which is used as a key to decrypt ciphertext only in one time process.

Figure 1 describes the system architecture of BIHEA algorithm. In the first step, the scanned retinal image of cloud user is taken as an input to BIHEA. The reason for choosing the iris in proposed algorithm is that retinal vessels of a specific cloud user have unique feature which have the least chance of matching with other.

Thereafter, the input image is resized into 505×598 to make the choice of the data pool get powerful enough that it can generate highly random data. We can either use RGB or greyscale image. RGB image is used in this work. The input RGB image is converted into greyscale image. Grey images are used to extract the edges (retinal vessels) of a retinal image. Retinal vessels are often used for authentication purpose. Fovea is fixed in retinal image, but optical disc can move. All the blood vessels are connected to the optical disc. Feature points are collected from the edge extraction process of a retinal image. So, extracted feature points from retinal image can be different for a same person. These different feature points are useful to create OTPs. In this proposed approach, random numbers of variable length are generated using the retinal feature points.

Fig. 1 System architecture of BIHEA encryption and decryption



This random number of variable length is used as OTP. OTP is valid only for a single session. Every time a user wants to enter the system, a new OTP is generated.

Both encryption of data and decryption of data are based on homomorphic encryption method. The BIHEA algorithm is described as follows:

Step-1: Generation of OTP

In this, we take the first component which is the scanned image of the registered user as input. Retina is a powerful biometric factor to generate the OTP. After that we resize the image according to the requirement. In our work, we have resized the image to 505×598 . Now as we have taken the RGB image, we convert the image into greyscale. The reason for converting into greyscale image is that RGB image has three channels whereas greyscale image has only one channel, thus it eases in the computation of Euclidean distance. Next step is to extract the edges of greyscale retinal image. Now, the intensity value of a point in the image is either 0 or 1. Edges (blood vessels) gone through the points have intensity value 1, and rest of the image points have intensity value 0. The points (x, y) which have intensity (I) value 1 are taken for further use. Now, distance is calculated from $(0, 0)$ to each point. We have used 'Sobel' method to extract the edges of a retinal image. We store those distance value in the matrix that are multiple of 7, i.e. $D\%7 == 0$. This calculation is done to confuse the hacker. They do not know which numbers are used in the system to generate the OTP. There may be possibility of duplicity in the matrix. Thus, duplicate number checking process is used to filter the duplicate number from D matrix, and it is stored in final matrix, M . A random number (N) is selected from the range $4 \leq N \leq 7$. Now, N numbers are selected randomly from the final matrix M . N is taken randomly to create variable length number. Finally, permutation of N numbers is done to generate the variable length random number called OTP.

Step-2: BIHEA Encryption Technique Using OTP as a Key

Input the data of cloud users and converts the data into its respective ASCII value. After the conversion into ASCII, we have to make equal length data of both users as we have to perform the mathematical operation on it. If the length is not same, then firstly we select the data whose length is small and append it with white space. Thereafter, data of both cloud users is converted into 16-bit binary data format. Now in order to perform the operation of encryption, we firstly randomly generate a number between 1 and 100 and then add integer obtained by multiplying the OTP generated in previous step with 19, in each bit of ASCII data. The arithmetic + operation is performed over resulted data of cloud users and stored on the cloud server.

Step-3: BIHEA Decryption Technique Using OTP as a Key

In cloud when any registered user needs the data of other registered cloud user, the request is sent by the user to the cloud server to retrieve the data. Thereafter, request is accepted by cloud server, and it delivers the stored data to the requisitioned. After receiving the data from the cloud, it will decrypt the received data using the same biometric inspired OTP key. User converts both data, i.e. its own data and decrypted data, into 16-bit binary format. They perform the exclusive OR operation on these two 16-bit binary data, and the result obtained is converted into the decimal value. Finally, we convert decimal value, i.e. respective ASCII value, into the corresponding ASCII characters. Hence, we get the data of the other registered cloud user.

4 Results and Discussion

The proposed BIHEA scheme tries to mitigate the security attacks such as unauthorized access of data, information disclosure during sharing, accessing and sharing the data of one user with other users without the permission and acknowledgement of data owner.

A. Breach of Data Access

The proposed BIHEA system explained in Sect. 3 which give permission to the user having an OTP is authorized to access the data. Only the data owner is authorized to issuing of OTP scan. The data cannot be accessed either by the Cloud Storage Provider or by the users, if they do not have the OTP. The imposition of the access control policy is guaranteed even if the Cloud Storage Server is not within the reach of the data owner or if it is malicious and untrusted as the access to the data or information depends on the OTP generated by the data owner. Breach of data access can happen in two possible situations.

- (1) The OTP with which the data can be decrypted is acquired by the unauthorized user or attacker, without the any knowledge or help by the Cloud Storage Provider. To access such an OTP, the attacker will have to know (a) iris of the

registered user, (b) random number generated, and (c) randomly chosen integer, e.g. '19', which is multiplied with the OTP used as a key. The knowledge of these three secrets is impossible. So, it is hardly possible for an attacker to access such an OTP without any help from the Cloud Storage Provider.

- (2) The other possible situation is that the OTP with which the data can be decrypted is acquired by the unauthorized user or attacker, with the knowledge or help by the Cloud Storage Provider. To access such an OTP, the iris of the registered user or the knowledge of randomly chosen value for the OTP must be known to the attacker. As OTP is delivered to user in the form of short message service (SMS), it is not possible for the attacker to calculate OTP from SMS. The SMS message is kept in secret and private by the user, so the attacker could not access the key. In brief, it is impossible for the attacker to access the key even with if Cloud Storage Provider helps the attacker.

B. Data Disclosure during Sharing

The data in the cloud environment is always in its encrypted form whether it is shared or any computation is done on the data; however, it may be encrypted with different keys, at different stages. Thus, the data is not decrypted at any point of instant before delivering the computational results to the requested user, who is authorized to access the data. Hence, it guarantees that the entire process of sharing does not allow leaking of any part of the information to unauthorized user.

To access the decrypted information during the sharing process, an unauthorized user must have the key or knowledge of iris pattern of registered user and method of generation of OTP using that pattern with the random generated numbers. From the above analysis, it is determined that the data cannot be decrypted by the unauthorized user. To decrypt message, the attacker needs the key used.

C. Time Complexity

Finally, the time complexity of BIHEA with the existing RSA, DES and AES encryption is done. It is clearly evident from the bar graphs shown in Figs. 2, 3, 4 and 5

Fig. 2 Computation time comparison of DES, AES, RSA and BIHEA

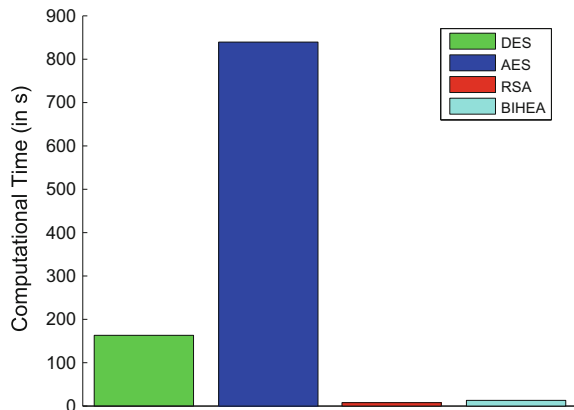


Fig. 3 Bar chart of DES algorithm in comparison to BIHEA

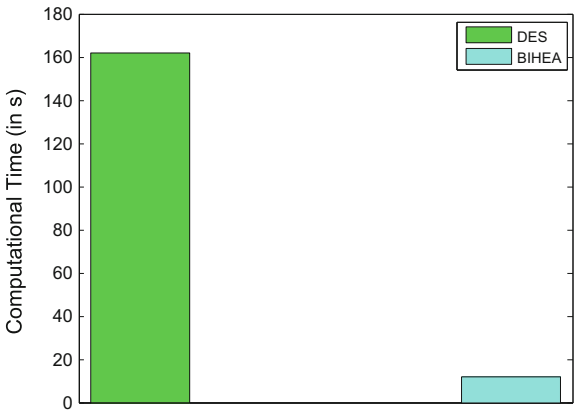


Fig. 4 Bar chart of AES algorithm in comparison to BIHEA

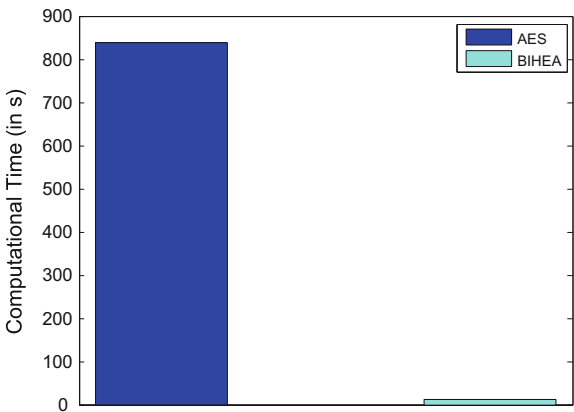
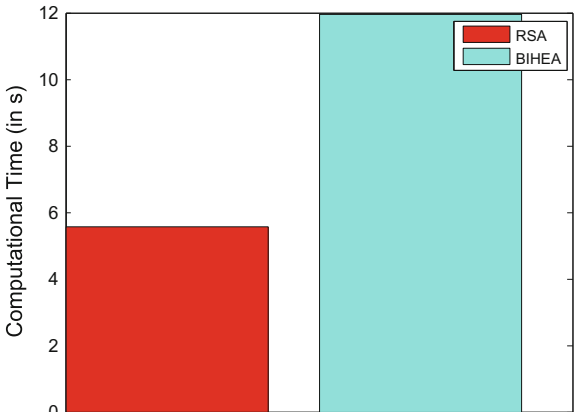


Fig. 5 Bar chart of RSA algorithms in comparison to BIHEA



that BIHEA takes less time in comparison to AES and DES encryption algorithm, but takes more time in comparison to RSA algorithm. RSA algorithm takes least time for execution, but it is more vulnerable to attacks. Thus, researchers proposed AES and DES encryption algorithms that reduce the effects of attacks, but on the other hand, the algorithm complexity increases, hence takes more time to execute. The proposed BIHEA offers both the advantages, i.e. it takes less time to execute and is more resistant to the attacks due to its two-tier architecture.

5 Conclusion

The lack of infrastructure ownership in cloud computing results in lack of user interest for storing its valuable data over the cloud. Thus, it becomes essential to develop the user's trust in cloud for sharing its data over the cloud environment. In this paper, we have proposed a biometric inspired homomorphic encryption algorithm (BIHEA) which is successfully implemented. The proposed encryption system has two-tier mechanism that means user data is encrypted using a secret key that is generated using iris of the user and generates the ciphertext which can be decrypted with a singular decryption key obtained as OTP from the registered user. This system allows the re-encryption of the user data, by altering the encryption key without decrypting the data. Thus, BIHEA provides a good system for sharing the user data on the cloud securely. The proposed system protects user data from unauthorized access and allowing enforcing the sharing policies as stated by the data owner.

We have performed thorough study and analysis of various security schemes before finalizing the proposed system and proof that the system allows user to securely share the data over untrusted cloud servers. The security analysis of the proposed BIHEA system infers that it can prevent number of security attacks and provide strong trusted environment for sharing the user data over untrusted cloud in comparison to DES, AES and RSA schemes. In the future, replay attack can also be considered and prevented using Time Stamp in BIHEA. It is also foreseen to perform real test with distributed computing on existing cloud servers like Amazon, Salesforce.com, Hadoop along with MATLAB tool.

References

1. Somani, U., Lakhani, K., Mundra, M.: Implementing the Digital Signature with RSA Encryption algorithm to Enhance the Data Security of cloud in cloud computing 1st International Conference on Parallel Distributed and Grid Computing. IEEE (2010)
2. Yu, S., Wang, C., Ren, K., Lou, W.: Achieving Secure, Scalable, and Fine-grained Data Access Control in Cloud Computing. IEEE (2010)
3. Tirthani, N., Ganesan, R.: Data security in cloud architecture based on diffie hellman and elliptical curve cryptography IACR Cryptology, ePrint Archive 49 (2014)

4. Arasu, S.E., Gowri, B., Ananthi, S.: Privacy-preserving public auditing in cloud using HMAC algorithm,” International Journal of Recent Technology and Engineering, IJRTE **2**(1) (2013)
5. Ronald L. Rivest, Leonard Adleman, and Michael L. Dertouzos.: On Data Banks and Privacy Homomorphisms, chapter On Data Banks and Privacy Homomorphisms Academic Press. pp. 169–180 (1978)
6. Jansen W.A.: Cloud Hooks: Security and Privacy Issues in Cloud Computing 44th Hawaii International Conference on System Sciences (2011)
7. Miranda, M., Pearson, S.: A Client-Based Privacy Manager for Cloud Computing COMSWARE’09. Dublin, Ireland (2009)
8. Wang, J., Mu, S.: Security issues and countermeasures in cloud computing in 2011 IEEE International Conference on Grey Systems and Intelligent Services, pp. 843–846 (2011)
9. A. Tripathi and A. Mishra (2011) Cloud computing security considerations in 2011 IEEE International Conference on Signal Processing, Communications and Computing (ICSPCC), pp. 1–5
10. Mathisen E.: Security challenges and solutions in cloud computing in 2011 Proceedings of the 5th IEEE International Conference on Digital Ecosystems and Technologies Conference (DEST), pp. 208–212 (2011)

Nature Inspired Computing

Proceedings of CSI 2015

Panigrahi, B.K.; Hoda, M.N.; Sharma, V.; Goel, S. (Eds.)

2018, XVI, 210 p. 97 illus., 67 illus. in color., Softcover

ISBN: 978-981-10-6746-4