Preserving Privacy of Data in Distributed Systems Using Homomorphic Encryption

P. Kalyani(✉), M. Masooda, and P. Namrata

SIES Graduate School of Technology, Nerul, Navi Mumbai, India
{kalyani.pampattiwar,masooda.modak,namrata.patel}@siesgst.ac.in

Abstract. Distributed systems like cloud platforms are being used widely in recent times. However, such platforms face a major issue of data storage on cloud in terms of security. If data on the cloud is not encrypted, it can be accessed by unauthorized members. This violates the confidentiality of the data. If data is stored in an encrypted format, every authorized party member will have to decrypt the data first in order to perform operations on it and then encrypt it back to upload it on the distributed platform. This has to be done every time, and every member performs operations on data. Needles to say, it complicates the entire procedure of sharing data and a lot of time is wasted in encrypting and decrypting data even for small operations like searching and sorting. To avoid these complications, this paper suggests a way to store data on cloud by homomorphically encrypting it. Homomorphic encryption allows user to compute on encrypted data without the need of decrypting it. In this paper, elliptic curve cryptography (ECC) is used for homomorphic encryption of data. The size of the ciphertext generated by ECC is smaller than the ciphertext generated by other encryption schemes. As cloud is mostly used for storing databases, this paper further employs searching and sorting techniques on the encrypted data.

Keywords: Data Mining · Privacy preservation · Homomorphic Encryption · Distributed Systems

1 Introduction

Distributed computing is being used on a wide-scale basis by almost all big and small organizations to store and access shared data. However, the security issues possessed by cloud computing become a deterrent when it comes to the wholehearted acceptance of the computing model. Data in cloud needs to be secure throughout its whole life cycle that is from creation to destruction of the data. This paper stresses on the confidentiality of data stored on cloud when it is at rest.

If data is stored on cloud in plain text format, it is vulnerable to attacks [1]. Hence, data on cloud is usually stored in encrypted format. However while performing operations on this data, it becomes necessary to convert it back into plain text. The client has to convert the encrypted data into plain text every time before operating on it. This defeats

C. R. Panigrahi et al. (eds.), *Progress in Advanced Computing and Intelligent Engineering*, Advances in Intelligent Systems and Computing 1198, https://doi.org/10.1007/978-981-15-6584-7_30

the purpose of cloud as the client is doing the majority of the work. Also, a lot of time is wasted in encrypting and decrypting data each time.

This is where homomorphic encryption comes into play. Homomorphic encryption allows computation on encrypted data. The results yielded by this are the same as the results obtained from performing the same operation on plain text. Thus, the client can now store data on cloud in an encrypted form. Any operations required by the client are performed on the encrypted data by the cloud provider, and the results are sent back to the client where the client has to only decrypt the result and not the whole dataset. This saves time and provides security to the data. The client is also relieved of computing on the data.

In this paper, elliptic curve cryptography (ECC) with ElGamal is used. Further, we apply searching and sorting techniques on the encrypted data. The data to be searched is sent by the client to the distributed server in an encrypted format. The searching is carried out on the server, and encrypted result is sent back to the client where it is decrypted and the final result is obtained. Sorting is done in a similar way where encrypted data is sorted by the cloud server and sent to the client where it is decrypted and displayed.

The paper is organized as follows: Sect. 1 presents an introduction to the area of homomorphic encryption, and Sect. 2 gives an insight into the related work done. Section 3 is proposing our system; Sect. 4 gives a detailing on experimentations and result, while Sect. 5 concludes the paper.

2 Related Work

Gentry [2] proposed the first fully homomorphic encryption (FHE) scheme. Also, the arithmetic operations over integers using homomorphic encryption were proposed by Dijk, Gentry, Halevi and Vaikuntanathan. In the research work [3, 4], the authors have proposed ECC-ElGamal scheme to implement homomorphic encryption over plain text [5].

The authors in [6, 7] in their research paper titled "homomorphic encryption scheme based on elliptic curve cryptography for privacy protection of cloud computing" proposed ECC-based homomorphic encryption schemes to solve the following issues in the execution of fully homomorphic encryption scheme: the public key is too large, the expansion rate of the ciphertext is large and the calculation of the ciphertext is too time-consuming. In [8], the authors investigate how different operations can be defined on FHE cloud data. Thus from the literature review, it can be seen that there are homomorphic encryptions which are used in the context of security of the data [7]. In [9, 10], authors have used additive homomorphic encryption to compute secure sum computation in SMC environment to make it applicable for insecure networks. The protocol is suitable for semi-honest parties who follow the steps in the protocol but also curious to learn data of the other parties.

Our work is extending this security to allow the operations like searching and sorting to be performed on this encrypted data in a distributed environment, thereby relieving the owner of performing these operations.

3 Proposed System

In this system, we propose a banking application where the database containing the account information of the client is stored on a cloud server. The client, in this case the bank employee, would insert data to be put into the database through the application. This data will be homomorphically encrypted at the client-side and then uploaded in the database in the cloud server, where it stays in the encrypted format. Now the client can ask the server to perform searching or sorting operations on the encrypted data and send it back to the client. On the client end, the data is decrypted and results are displayed. The system consists of three major steps:

1. Encryption of data using ECC-ElGamal Scheme using Koblitz's method.
2. Storing the encrypted database on a distributed server.
3. Performing searching and sorting of the encrypted data.

In ECC the plaintext is converted into integers. These integers are then plotted as points on the elliptic curve.

$$y^2 = x^3 + ax + b \tag{1}$$

Now these points on the elliptic curve are added using the ElGamal scheme. Koblitz's method is used to encode the data. This is explained in detail in the flow diagram of ECC-ElGamal shown in Fig. 1.

This encrypted data is stored on the server in a database. Multiple clients can access this server. Every client can sort data as well search for a particular entry on the server. The sorting is performed using queries on the database on the server side. This sorted data is then sent to the client where it is decrypted. When a client searches for a particular item in the database, the client will provide a primary or candidate key whose corresponding entry will be fetched.

In the banking application, the entries were sorted based on the bank balance of the entries. Searching was performed on the bank's database using two parameters, account number and the branch name. These parameters served as the candidate key.

4 Implementation and Evaluations

To implement the proposed homomorphic encryption, we design a banking system where information is entered through the client side, encrypted, and stored in the server in encrypted format. Later searching and sorting operations are performed on the encrypted data, and the results are sent to the client where it is decrypted and shown to the client in plain text.

The client side of the banking system shows an interface to make an entry into the database table according to the account number and branch. Entry made is acknowledged as shown in Fig. 2.

These entries are encrypted and stored on the server in an encrypted format itself. We can see as shown in Fig. 3 that the confidentiality of the data is preserved as the data

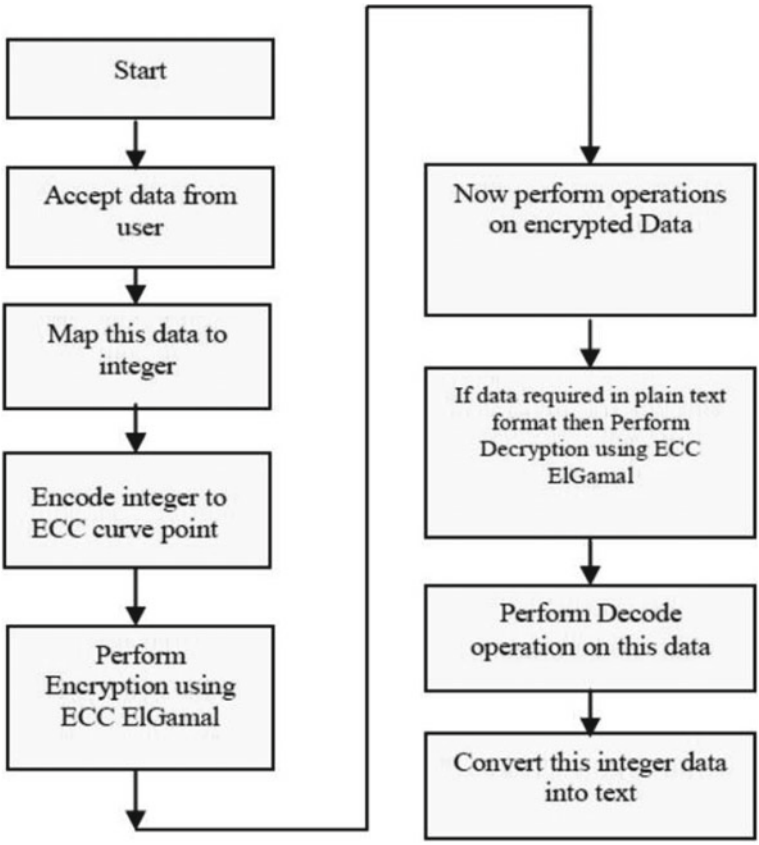

Fig. 1. Flow diagram of ECC-ElGamal scheme

are not stored on the server in plain text format. Now operations can be performed on this data.

In this system, we provide the client with the functionality of sorting the data as well as searching the details of a particular account holder by querying the account number and branch name. The interface for the same can be shown in Fig. 4. The searching and sorting operation are shown in Figs. 4 and 5.

Thus, we achieve the objective of encrypting the data and storing it on a remote server in encrypted format. We also perform searching and sorting operation on the encrypted data which is later sent to the client where it is decrypted, and the results are shown in plain text. Hence, we achieve the basic objective of homomorphic encryption of providing confidentiality to data stored on distributed systems like cloud along with the liberty to perform operations on encrypted data.

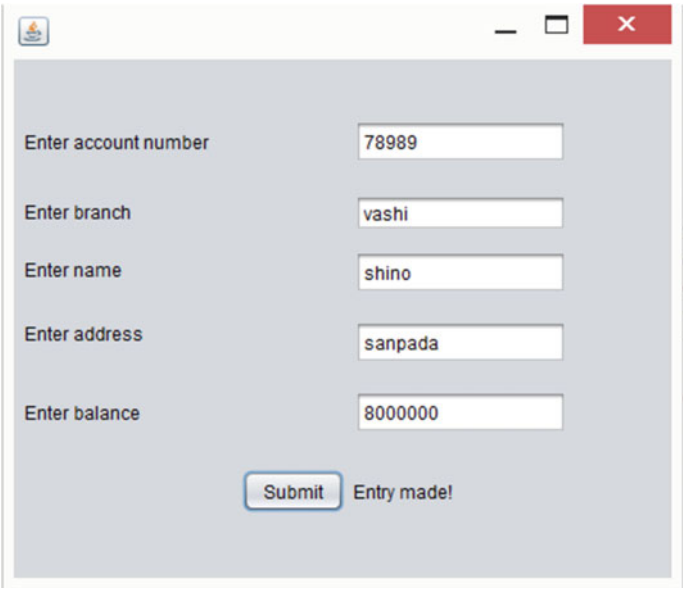

Fig. 2. Client interface for account entry

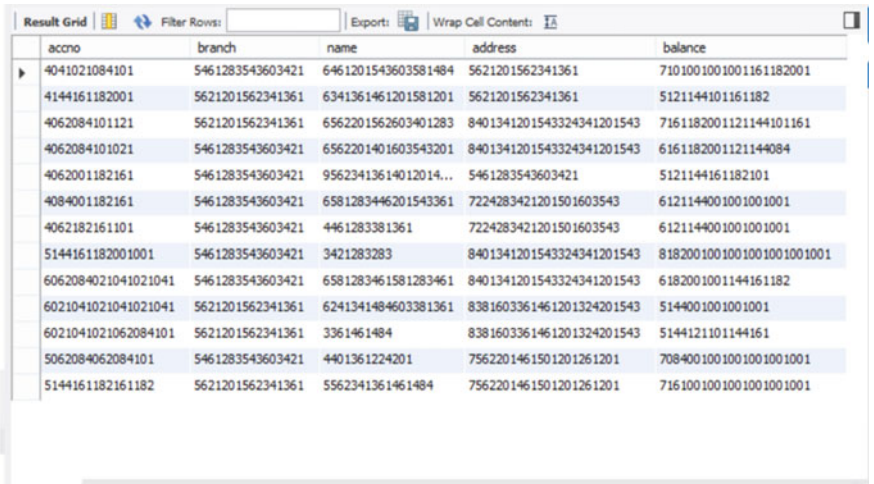

Fig. 3. Encrypted data on server

5 Conclusions

The banking system shows the implementation of homomorphic encryption by storing data on the distributed platform in encrypted format and performing operations like searching and sorting on it. Using ECC-ElGamal encryption scheme considerably small ciphertext was obtained which in turn takes relatively less space for storage. Searching

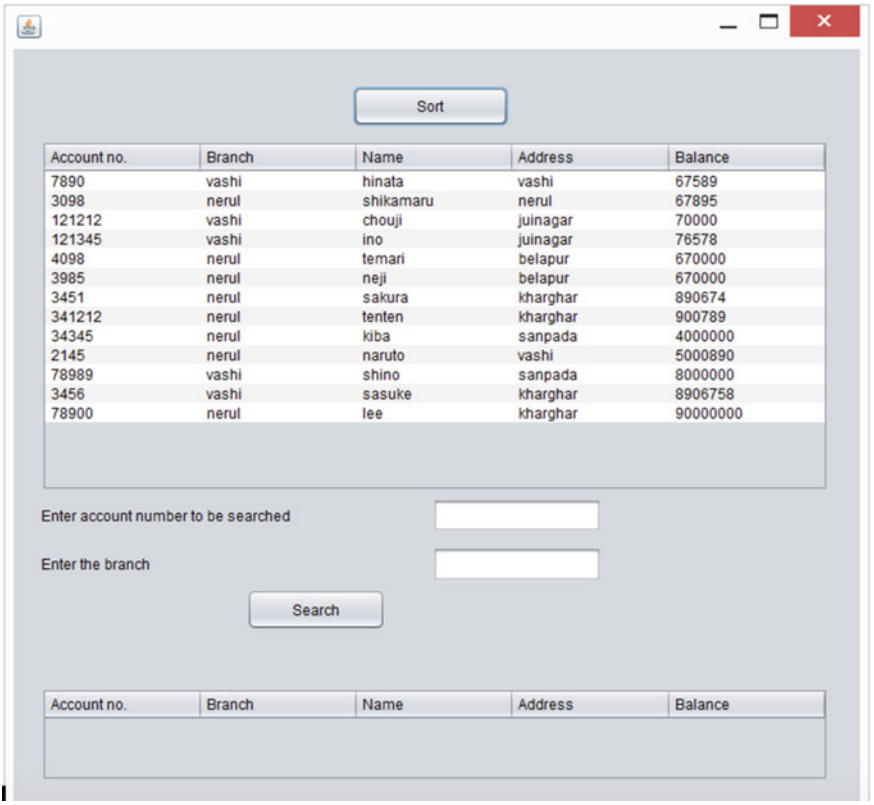

Fig. 4. Client interface for searching and sorting

and sorting of the data were successfully carried out. However, the need to reduce the size of the ciphertexts even further persists because of storage restrictions. Also, more efficient ways of searching and sorting on the encrypted data need to be researched.

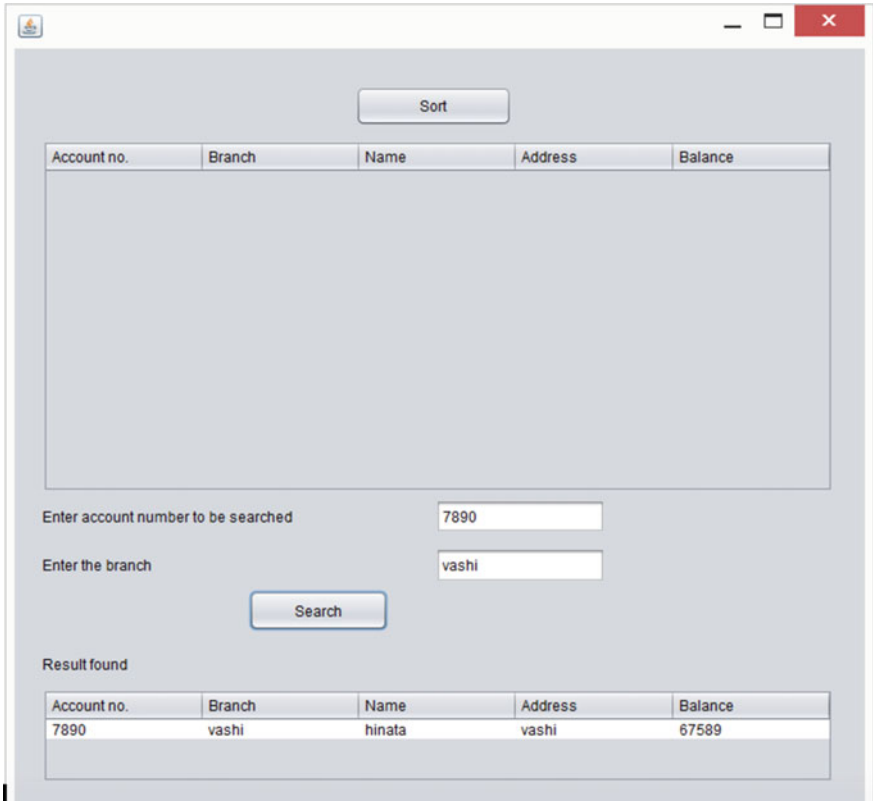

Fig. 5. Result of searching operation

References

1. Vigila SMC, Muneeswaran K (2009) Implementation of text based cryptosystem using elliptic curve cryptography. In: 2009 first international conference on advanced computing. IEEE (2009)
2. Hankerson Darrel, Menezes Alfred J, Vanstone Scott (2005) Guide to elliptic curve cryptography. Comput Rev 46(1):13
3. Meier AV (2005) The elgamal cryptosystem
4. Potey MM, Dhote CA, Sharma DH (2016) Efficient homomorphic encryption using ECC-ElGamal scheme for cloud data
5. Raju GVS, Akbani R (2003) Elliptic curve cryptosystem and its applications. In: SMC'03 conference proceedings. 2003 IEEE international conference on systems, man and cybernetics. Conference theme-system security and assurance (Cat. No. 03CH37483), vol 2. IEEE
6. Hong M, Wang PY, Zhao WB (2016) Homomorphic encryption scheme based on elliptic curve cryptography for privacy protection of cloud computing. In: 2016 IEEE 2nd international conference on big data security on cloud (big data security)
7. Kumari M, Sharma P (2014) Privacy preserving using homomorphic encryption. Int J Adv Res Comput Sci Softw Eng ISSN 2277
8. Chatterjee A, Sengupta I (2015) Searching and sorting of fully homomorphic encrypted data on cloud. IACR Cryptol ePrint Arch 2015:981

9. Gentry C, Halevi S (2011) Implementing gentry's fully-homomorphic encryption scheme. In: Annual international conference on the theory and applications of cryptographic techniques. Springer, Berlin, Heidelberg
10. Sheikh R, Mishra DK (2019) Secure sum computation using homomorphic encryption. In: Data science and big data analytics. Springer, Singapore, pp 357–363