

HOMOMORPHIC ENCRYPTION TECHNIQUE FOR CLOUD COMPUTING SECURITY OF CLOUD DATA

Amruta Patil¹, Apurva Kirve², Sayali Nandeshwar³, Swati Ture⁴, Prof. N.R. Shikalgar⁵

¹⁻⁴B.E. Student, Zeal Engineering College, Narhe, Pune, India

⁵Professor, IT Department, Zeal Engineering College, Narhe, Pune, India

Abstract - Cloud computing is deemed together of the foremost powerful innovations within the computing world, but its usage continues to be hindered by security considerations. Many sorts of secret writing algorithms were applied within the cloud for securing the information. Data will be saved on within the cloud in associate with encrypted condition. So, a newest system referred to as homomorphic encryption is comes up that enables to use specific operations on the encrypted informtion. The paper presents a summary of security problems in cloud computing and utilization of the absolutely homomorphic encryption technique has drawbacks of huge key size and low calculation potency, and it's not sensible for the secure cloud computing. We have a tendency to build up a homomorphic encryption system supported the load balancing (data in multiple file in encrypted form), in cloud computing and RSA formula that is increasing homomorphic.

Key Words: Homomorphic Encryption, Load Balancing, Cloud Computing, RSA, Data Security, Public Key, Private Key.

I. INTRODUCTION:

It has been a typical observe for organizations to spread their on-line business logics to net hosting service suppliers for over ten years. Generally, databases and additionally the business logics of a company are hosted by a 3rd party to avoid wasting the IT management time and price. The cloud computing any pushes forward this paradigm. There were so many cloud information centers that store a really great storage of data from sources and support data encryption in computation. Security may be a significant problem for such information centers once the information they need are touchy. An information center is also attacked, compromised associate degreed an addition the aptitude of corporate executive attacks .The security issues with the outsourced database may be solved if the important data are encrypted. Naturally it ends up in the matter of however {the data |the info |the information} center will perform computation on encrypted data. Homomorphic encoding schemes, offer an answer for this stalemate: Such schemes provide functions to gauge encrypted data, the results of that continues to be encrypted info, however may be decrypted into the results of the (logically) same perform applied to the data.

II. HOMOMORPHIC ENCRYPTION

Homomorphic encoding systems have the aptitude of playacting computation on encrypted info while not knowing the personal key. These calculations, produce associate degree outcome that is itself encrypted. The results of any computations on the encrypted knowledge is that the same as within the case of data.

Mathematically, we tend to aforementioned that system is Homomorphic encoding if:

HE (A) and HE (B) will calculate HE (C (A, B)),

Where C will be: Add, multiplication, X-or

If the customer desires to perform computations on its data within the cloud, the personal key ought to be shared with the user to decipher the info. Sharing the key would permit access to the info from a cloud supplier. Therefore to unravel this drawback, the homomorphic encoding used. The shopper permits for cloud suppliers to reckon the info while not decrypting it. The result are came back to the customer aspect and it's still encrypted. So, since the user is that the solely holder of the personal key, nobody else is ready to decipher neither knowledge nor results.

HOMOMORPHIC ENCRYPTION TECHNIQUES

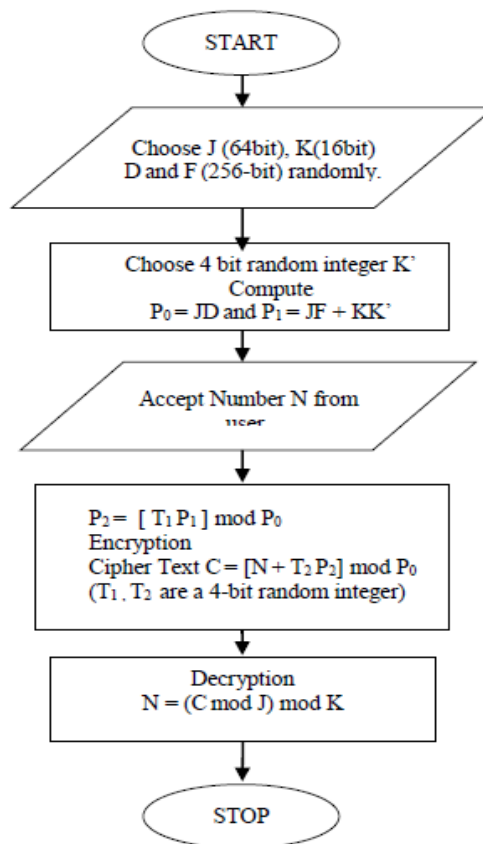


Fig. 1 Flowchart of Homomorphic Encryption Techniques.

The flowchart in fig. 1 shows the proposed scheme to perform homomorphic encryption techniques algorithm. This techniques is simplified and efficient version of applied in AWS public cloud for security of client data. (J, K) represent a secret Key and (P0, P1) forms a public key. Number N to be encrypted is accepted as user input.

For Example-

Input is given as J=14883982794894487223, K=43321 and number to be encrypted N=9

Then D and F are calculated as

D=70677186543966147614195862042065680704217811307170938823680817972460078770747 and

F=73039047329961611877474622320644292204439326844747783070676806904287578243639

Consider four bit number K' = 12 then compute

P0=1051958028511940305929320607565533427835574146640991376691781227504638919782237324865124737665581.

And

P1=1087111923814632766481581241697004087337750199678917794234945876512572829144575038603913867044349

Perform Encryption and get

C=351538953026924605522606341314706595021760530379264175431646490079339093623377137387891293787689

Decryption is performed and get back plain text N=9

III. SYSTEM WORKING AND IMPLEMENTATION

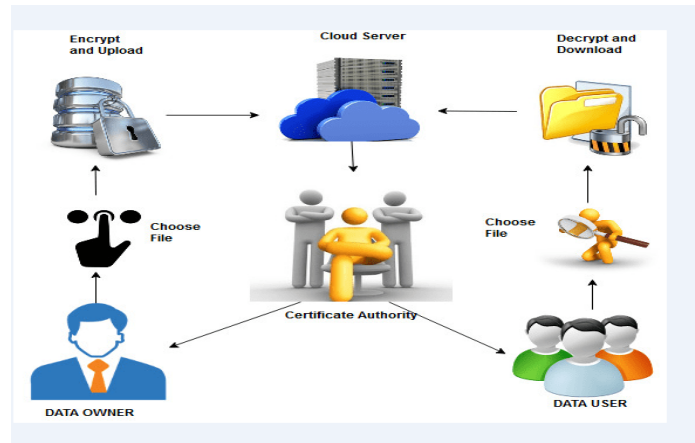


Fig. 2 System Architecture

When the data send to the Cloud server (laptop as a cloud) we use standard encryption methods to secure the operations and the storage of the data. Our basic concept is to encrypt the data before send it to the Cloud provider. But the last one needs to decrypt data at every operation. The data owner upload the file to cloud server with the help of private key given by service provider to encrypt the data at the time of uploading file, user have permission to give access permission to file for sharing activity as well as user can set priority to file folder as public or private for confidential data. This techniques execute an application of a method to conduct operations on encrypted information without decrypting them, which will reinforce the same output after mathematical expression as if we have worked directly on the raw information.

Homomorphic Encryption system used to operate encrypted data without decryption them, the data owner is the only holder of the OTP generated by system to decrypt the encrypted file. When we decrypt the result of any operation, it is the same as if we had carried out the calculation on the raw data. This allows the user to login based on his credentials and then the user can perform operations on their data based o requirements. Once user is done with all the tasks, it can opt to exit the system.

The following are the steps of program implementation:

Step 1: Install java (Latest JDK).

Step 2: Install MySQL workbench, Tomcat 8.5, MySQL installer, Eclipse oxygen.

Step 3: Open eclipse and import the file name as HomomorphicFileEncryption.

Step 4: Create Database for Tables generation with proper schema as name in hybernet.cfg.xml file.

Step 5: Add required library to execute code or make error free like server runtime jar file and jre system to configure project

Step 6: After the installation of Tomcat on Eclipse framework the user is available with all the needed packages.

Step 7: Run Java Code

Step 8: Exit Java Code after completion of project

IV. RESULT:

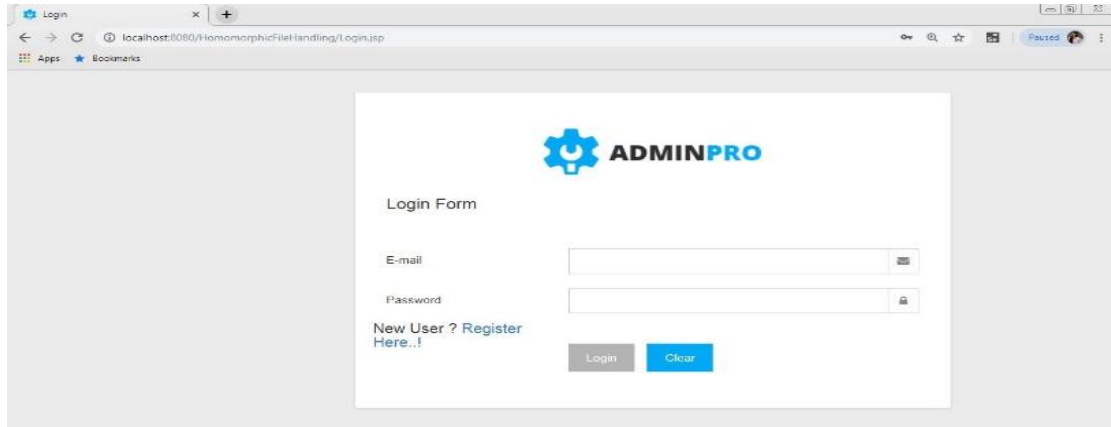


Fig. 3 Login Form

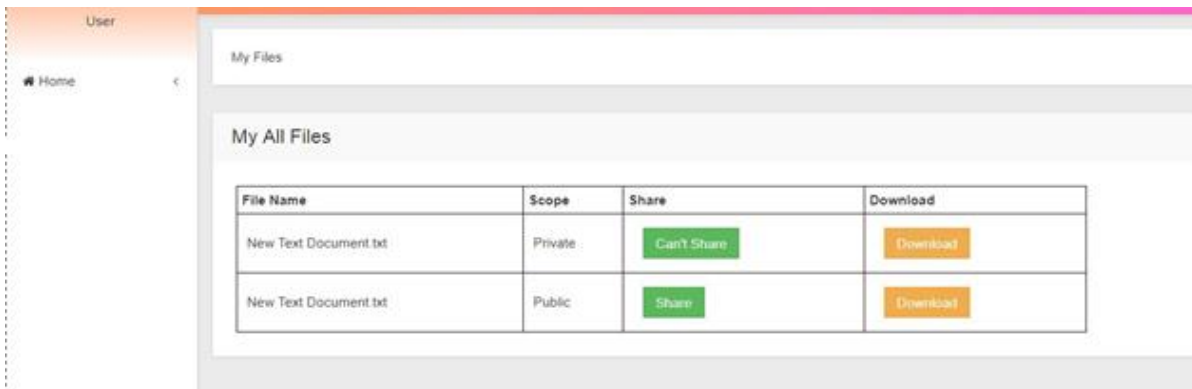


Fig. 4 File Information

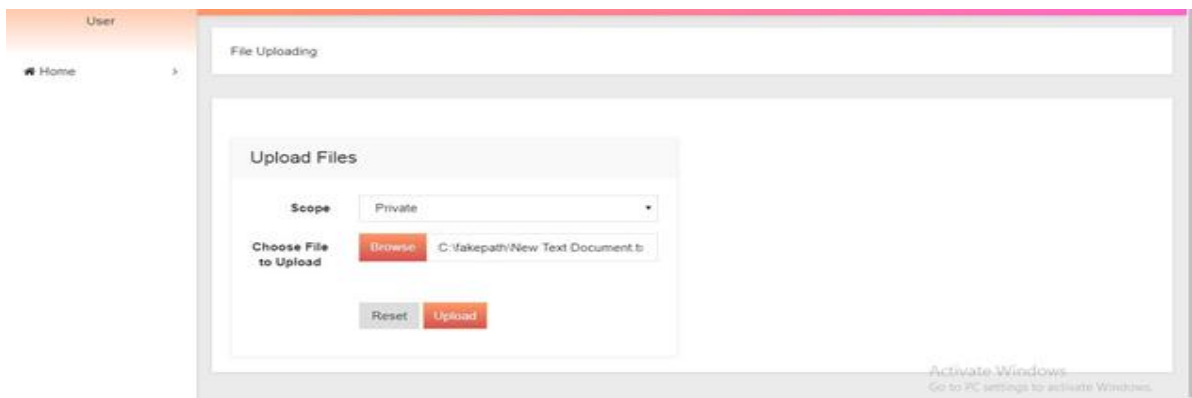
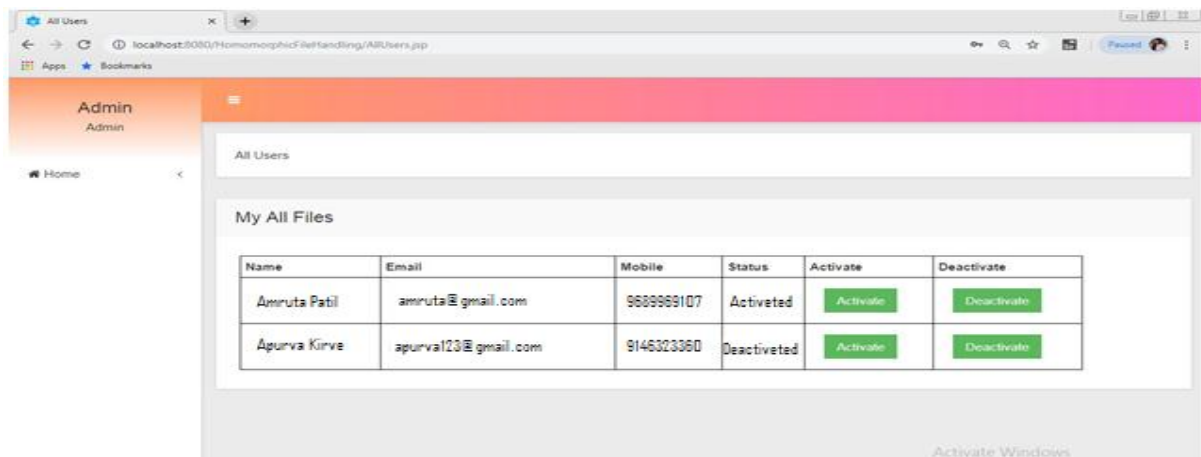


Fig. 4 Uploaded File



Name	Email	Mobile	Status	Activate	Deactivate
Amiruta Patil	amiruta@gmail.com	9889989107	Activated	Activate	Deactivate
Apurva Kirve	apurva123@gmail.com	9145023360	Deactivated	Activate	Deactivate

Fig. 4 User Details

V. CONCLUSION:

The security issues are main problem in day to day life to store our personal data on server. The unauthenticated person can easily access our data if there is no security or encryption on any type on data. So, to overcome that type of problems we can developed homomorphic encryption techniques to store data on cloud securely, homomorphic encryption work when we upload any file to cloud and stored in securely. The file sharing operation is totally based on file authenticated author only. User can distribute there file as public and private where only public allocated data can be share to other users. At the time of decryption we give OTP security to file that can handled by only file owner only. Due to these techniques we can store more amount of file securely on cloud. To take a look at atmosphere, the developed system has delivered promising performance outcome as compared to alternative common solutions. Therefore the planned approach may be thought-about to be used in universe situations.

REFERENCES

- 1]. Xidan Song , Yulin Wang, "Homomorphic Cloud Computing Scheme Based on Hybrid Homomorphic Encryption," in IEEE International Conference on Computer and Communication (ICC) , Chengdu , China, 2017.
- 2]. Liangliang Xiao, Osbert Bastani, I-Ling Yen, "An Efficient Homomorphic Encryption Protocol for Multi-User Systems," IACR Cryptology ePrint Archive, p. 19, 2012.
- 3]. Sweta Agrawal, Aakanksha Choubey, "Survey of Fully Homomorphic Encryption and Its Potential to Cloud Computing Security," International Journal of Advanced Research in Computer Science and Software Engineering, vol. 4, no. 7, pp. 679 - 686, 2014.
- 4]. Abdellah EZZATI, Khalid EL MAKKAOUI, Abderrahim BENI HSSANE, "Homomorphic Encryption as a Solution of Trust Issues in Cloud," in International Conference on Big Data, Cloud and Applications, Tetuan, Morocco, 2015.
- 5]. Payal V. Parmar , Shraddha B. Padhar , Shafika N. Patel , Niyatee I. Bhatt , Rutvij H. Jhaveri, "Survey of Various Homomorphic Encryption algorithms and Schemes," International Journal of Computer Applications, vol. 91, no. 8, pp. 26 - 32, 2014.
- 6]. X. Yi et al., "chapter 2," in Homomorphic Encryption and Application, SpringerBriefs in Computer Science, 2014, pp. 27-46.
- 7]. YASMINA BENSITEL, RAHAL ROMADI, "SECURE DATA IN CLOUD COMPUTING USING HOMOMORPHIC ENCRYPTION," Journal of Theoretical and Applied Information Technology, vol. 82, no. 2, pp. 206 - 211, 2015.
- 8]. Monique Ogburna, Claude Turnerb, Pushkar Dahalc, "Homomorphic Encryption," Procedia Computer Science, vol. 20, pp. 502 - 509, 2013.