Tokenization as a Service over Cloud

Riya Banerjee[1] Rashmi Sharma[2] Deepak Sharma[3] Ms. Rekha Jadhav[4] Ms. Sonali Sonawane[5]
[1,2,3]Student [4]H.O.D [5]Lecturer
[1,2,3,4,5]Department of Information Technology
[1,2,3,4,5]G. H. Raisoni Institute of Engineering and Technology Affiliated to Savitribai Phule Pune University Pune, Maharashtra, India

Abstract— All organizations have some kind of sensitive data. This data can be personally identifiable information (PII), account details, credit card numbers, Electronic Health Records (EHR) etc. Organizations mostly choose to store their data on cloud as cloud provides many benefits like less cost and ease of data access. Protecting sensitive data can be difficult as well as expensive. Banking and financial services, healthcare, retail and government entities must follow strict guidelines when handling sensitive data in cloud that include PCI DSS, ITAR, FERPA, HIPAA and HITECH. Tokenization solution replaces sensitive data with unique identification strings to make it harder to steal sensitive data and also meets the guidelines given by PCI DSS, HIPAA and HITECH. In this paper we have shown how cloud based tokenization service provides a way to secure sensitive data at a lower cost and a higher capacity.

Key words: Tokenization, Cloud

I. INTRODUCTION

Cloud is nothing but a cluster of remote servers and various other software resources. Cloud gives the benefit of a centralized database and also access to resources and services online. Using the cloud helps remove the upfront costs, including software and hardware procurement. It also removes the burden of capacity management due to which one can store unlimited data without worrying about space. Cloud enables data to be stored for as long as desired. Implementation, monitoring and management are all handled by the cloud providers.

Clouds can be further classified as public, private and hybrid. Public cloud is the one which renders its services over a network that is open for public use. Private cloud is the one that is operated and managed solely for a single organization. This can be done either by the organization itself or by some third party. Hybrid cloud is a composition of two or more clouds.

As more and more organizations are choosing to use the facilities of a cloud, security concerns for cloud providers are increasing day by day. There are many technologies that provide security to cloud data. There is a new technology called tokenization which is the process of replacing a sensitive data field with a surrogate value. This is called a token. The token has no meaning or value. It is not related to the original data in any way. Only the tokenization server can tokenize data to create tokens, or retrieve original data from corresponding tokens under strict security controls.

In our system tokens will be generated by using SHA-1 algorithm. The size of each token generated will be same i.e. of 512 bits. This will further be encrypted using AES algorithm to provide more security.

Tokenization process may be used to safeguard sensitive data involving, for example, bank accounts, financial statements, medical records, criminal records, driver's licenses, loan applications, stock and other types of personally identifiable information (PII).

II. LITERATURE SURVEY

A. *Third Party Auditor*

In this, the data is stored on the cloud. The security tasks are performed by the trusted service provider and data is stored on some other semi-trusted machine. While sending the data from the client to the third party over the public cloud, channel encryption is required for safe transfer of data. After that, the encryption of data is done. The overhead of repeated encryption make this technique very complex. Third Party Auditor verifies the data to be stored on the cloud. Fig. 1 gives its block diagram. [1]

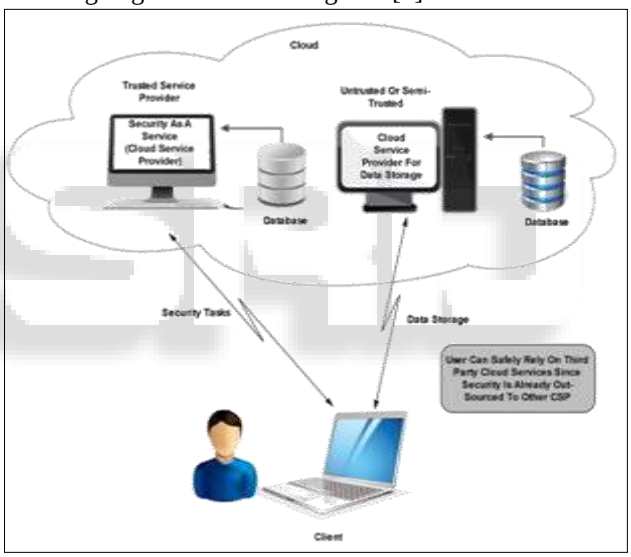


Fig. 1: Third Party Auditor Block Diagram

B. *Security as a Service Model for Cloud Environment*

Here the security is provided on the cloud itself. The security mechanisms like the encryption are done here. Since the security is provided on the cloud itself so it becomes easy for the attacker to attack or hack the data. The clients have to entrust the cloud service provider with security measures. [2]

III. PROPOSED WORK

Sensitive data like personally identifiable information (PII), account details, credit card details etc. are replaced by meaningless tokens. These tokens are then stored on the cloud. Various steps are involved in this whole process as can be seen in Fig. 2.

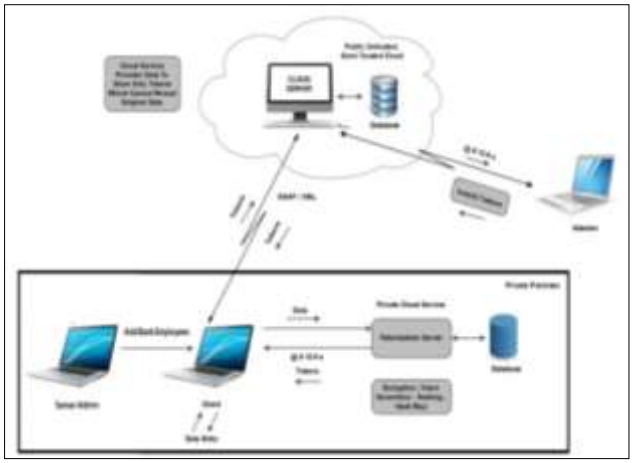


Fig. 2: System Architecture

First of all, the sensitive data are sent to the tokenization server. Here the tokens are generated for the corresponding data using the SHA-1 algorithm. Further security is provided by applying encryption using the AES algorithm. The tokens along with their corresponding data are then stored into the bank database. This is nothing but the look-up table. The tokenization server and the database are present on the private cloud. Only the tokens are then sent back to the client. The client then sends only the tokens for storage on the cloud i.e. the public cloud. Here if the attacker tries to access the data, he will get nothing but the meaningless tokens. Only authorized users have access to the real data.

In this project we have considered the scenario of a bank organization. We are going to produce tokens for the entities 'name' and 'account number'. Rest all data will remain the same. As the data is entered from the client application, the name and the account number will be sent to the tokenization server. The tokenization server will then calculate the corresponding hash values using the SHA-1 algorithm. The hash values along with their corresponding data will then be encrypted using the AES algorithm. This encrypted data will then be stored in the bank server database. Only the tokens will be sent back to the client. The client will then send the tokens for storage to the public cloud.

For retrieval of data, the tokens are then sent to the client from the public cloud. The client then sends these tokens to the tokenization server. The tokenization server, after consulting the look-up table, sends the original data back to client.

IV. MODULES

The various modules present in the proposed system are as follows:-

A. Server Admin

Used for adding new bank employees. These employees are then given the authority to manage bank users.

B. Client Application

It is the app through which client will access the services for token generation by tokenization server. Client will send the data to the tokenization server so that the data gets converted to tokens.

C. Tokenization Server

Used for generating the tokens. SHA-1 and AES algorithm are used for this purpose. It is present on the private cloud. Here the look-up table is stored.

D. Storage Server

This is present on the public cloud. The client after receiving the tokens from the Tokenization server will send the tokens to the public cloud for storage. Here data that is stored is in tokenized form.

E. Attacker Application

Here it will be shown that if the attacker tries to access the data from the bank database which is on public cloud, then he will get only the tokens which are of no use.

V. ALGORITHM

A. SHA-1

SHA1 stands for "Secure Hashing Algorithm". It is a hashing algorithm designed by the United States National Security Agency and published by NIST. SHA1 is currently the most widely used SHA hash function. SHA1 outputs a 160 bit digest of any sized file or input. It uses a 512 bit block size and has a maximum message size of 2 ^ (64 - 1) bits.

These are examples of SHA-1 digests. ASCII encoding is used for all messages.

SHA1 ("The quick brown fox jumps over the lazy dog") = 2fd4e1c6 7a2d28fc ed849ee1 bb76e739 1b93eb12.
Even a small change in the message will, result in a completely different hash. For example, changing dog to cog produces a hash with different values for 81 of the 160 bits.

SHA1 ("The quick brown fox jumps over the lazy cog") = de9f2c7f d25e1b3a fad3e85a 0bd17d9b 100db4b3

B. AES

After calculating the hash values of corresponding data using SHA-1 algorithm, the hash values are then encrypted using Advanced Encryption Standards (AES) algorithm to provide extra security to the data.

AES is a symmetric-key algorithm, meaning the same key is used for both encrypting and decrypting the data. It is combination of both substitution and permutation. It has a fixed block size of 128 bits and key size of 128, 192 or 256 bits.

VI. WORKING

A. Server Admin Module

Fig. 3: Admin Login

The Server Admin will login through this page. After that he can add bank employees. These employees are then given the rights to manage user data.

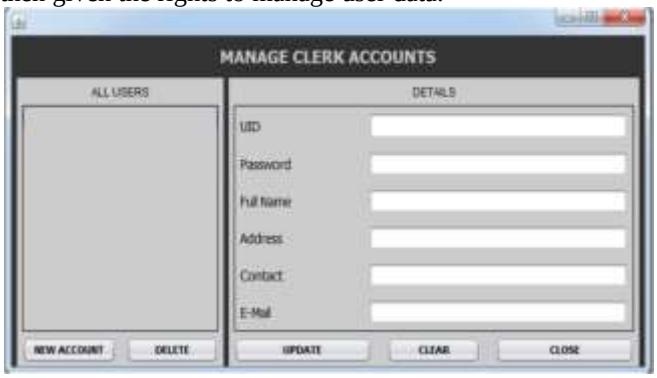


Fig. 4: Manage Clerk Account

Through this page the server can manage the bank employees.

B. Tokenization Server Module

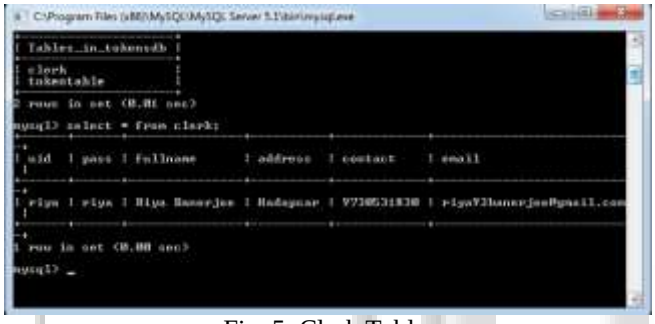


Fig. 5: Clerk Table

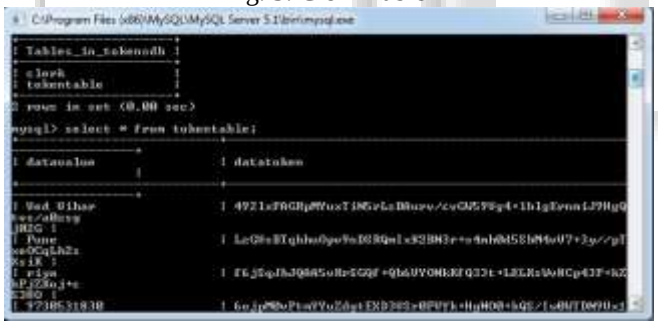


Fig. 6: Look-up Table

The tables shown in Fig. 5 and Fig. 6 are the two tables which are present on the private cloud. The Clerk table contains the bank employee details. The Look-up table consists of tokens and their corresponding data.

C. Client Application Module

Fig. 7: Server IP Address

Through this page the bank employee provides the IP addresses of the servers present on private and public cloud.

Fig. 8: Clerk Login

The bank employee can login using this page. After that he can manage the user data.

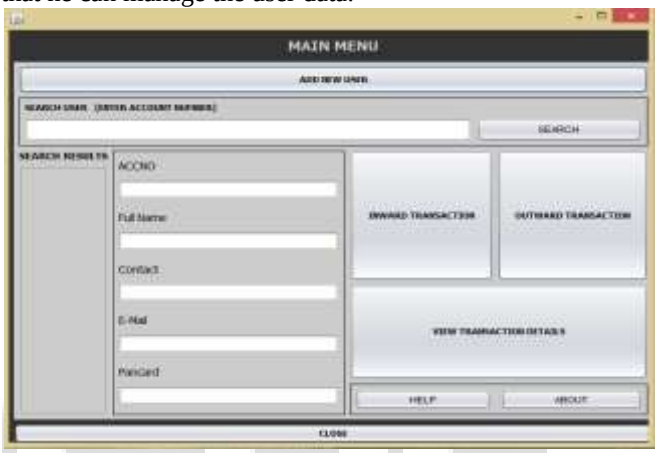
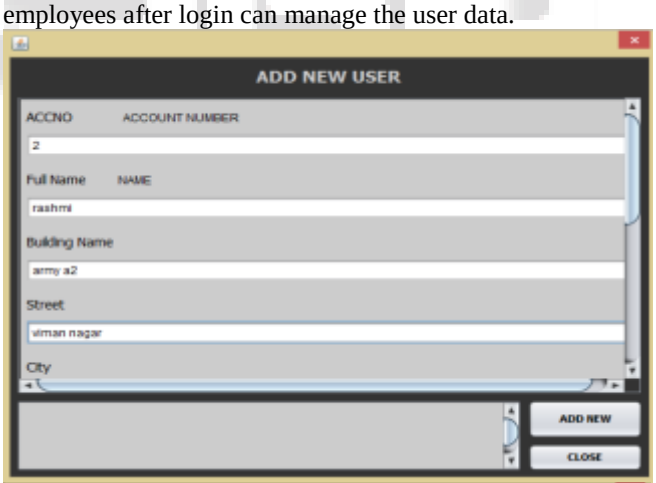


Fig. 9: Clerk Window

This is the page from where authorized bank employees after login can manage the user data.



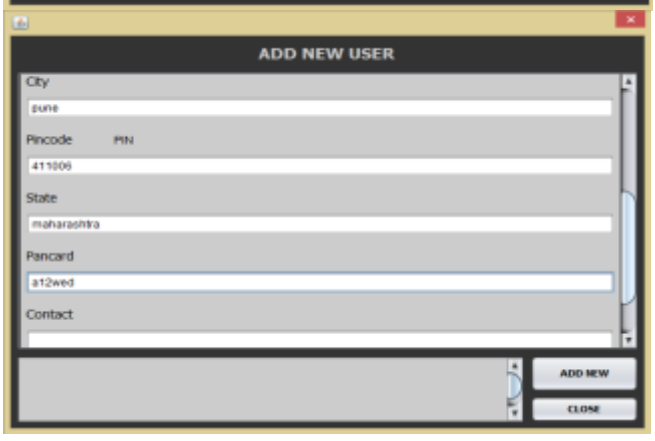


Fig. 10: Addition of new user

This way a bank employee can add a new user.

Fig. 11: Search User

The users can be searched using their account number.

Fig. 12: Inward Transactions

Fig. 13: Outward Transactions

After the respected user is searched, the transactions details for that particular user are added as shown in Fig. 12 and Fig. 13.

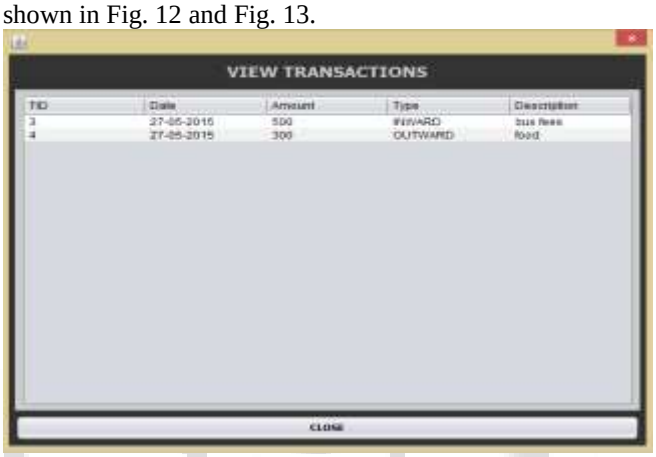


Fig. 14: View Transactions

Through this page the user can view his transaction details.

D. Storage Server Module

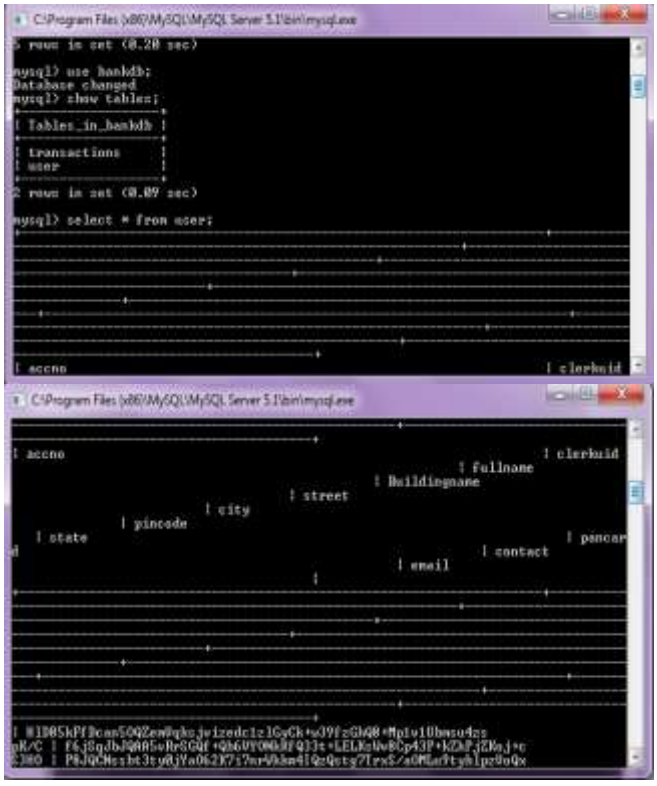


Fig. 15: User Table

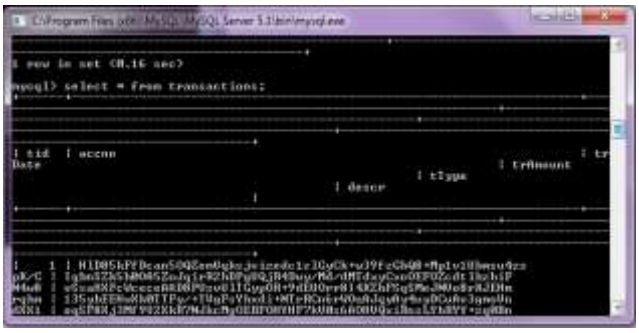


Fig. 16: Transaction Table

The tables shown in Fig. 15 and Fig. 16 are the two tables which are present on the public cloud. The User table contains the details of user in tokenized form. The Transaction table consists of transactions performed by user in tokenized form.

E. Attacker Application Module

Fig. 15: Attacker Main Menu

The attacker is assumed to have attacked the storage server which is present on public cloud.

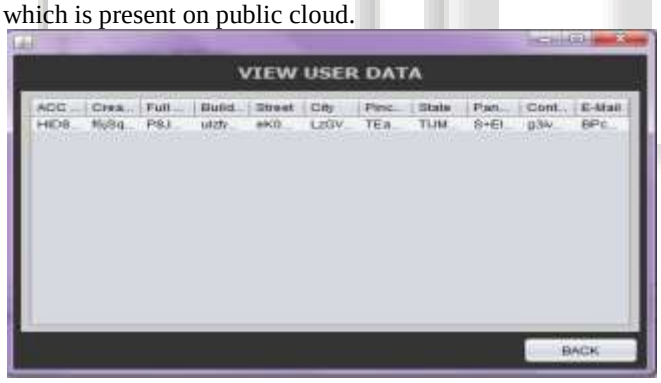


Fig. 16: List of Users as viewed by Attacker

Fig. 17: List of Transactions (performed by Users) as viewed by Attacker.

The attacker can only see tokenized data which is meaningless.

VII. Advantages

- Sensitive Data stays within the organization's control at all times.
- Strong tokens have unique security strength because they are not mathematically linked to the original value they replace.
- Significant cost savings
- Elastic capacity and unlimited data retention.
- Enhanced security over on-premise tokenization options.
- Reduced management burden.

VIII. Conclusion

The main reason for making this project is to provide security to the sensitive data over the private cloud using tokenization. Since the tokens generated will be random and of equal size, independent of the size of the word, so it will be impossible for the attacker to get the data or even guess it. If the attacker tries to access data from the cloud he will get only the tokenized data which is meaningless. Hence for this reason tokenization is considered to be the strongest way to secure data till now.

Acknowledgment

It is our immense pleasure to express our deep gratitude to Ms. Rekha Jadhav, Head of Department, Information Technology, and Ms. Sonali Sonawane, Lecturer, Information Technology for their valuable guidance, inspiration and whole-hearted involvement during every stage of project preparation. Their experience, and professional knowledge, has greatly influenced the timely and successful completion of our project preparation.

We are indebted to Dr. R. D. Kharadkar, Principal, G. H. Raisoni Institute of Engineering and Technology, Pune, for encouragement and providing us the opportunities and facilities to carry out this work. And finally we would like to thank the college for being such strength during the entire work.

References

[1] Bhavna Makhija, Vinit Kumar Gupta, Indrajit Rajput, "Enhanced Data Security in Cloud Computing with Third Party Auditor", IJARCSSE, Vol. 3, Issue 2, Feb 2013, ISSN: 2277 128X

[2] Vijay Varadharajan, Udaya Tupakula, "Security as a Service Model for Cloud Environment", IEEE, Vol. 11, No. 1, Mar 2014, pp. 1932-4537

[3] Joan Daemon, Vincent Rijmen, "AES Proposal: Rijndael", NIST, Nov 26, 2001

[4] Xiaoyun Wang, Yiqun Lisa Yin, Hongbo Yu, "Finding Collisions in the Full SHA-1", Crypto 2005, MIT.edu

[5] Abha Sachdev, Mohit Bhansali, "Enhancing Cloud Computing Security using AES Algorithm", International Journal of Computer Applications, Vol. 67, No. 9, April 2013, pp. 0975 – 8887

[6] Qian Wang, Cong Wang, Kui Ren, Wenjing Lou, Jin Li, "Enabling Public Auditability and Data Dynamics for Storage Security in Cloud Computing", IEEE Transactions, Vol. 22, No. 5, May 2011, pp. 1045-9219

[7] Hamid Banirostam, Alireza Hedayati, "A Trust Based Approach for Increasing Security in Cloud Computing Infrastructure", International Conference on Computer Modelling and Simulation in IEEE, 2013, pp. 978-0-7695-4

