Secure Login of Statistical Data With Two Parties

Qasim Uddin[1], Mohamed Badruddin M[2], S.Brintha Rajakumari[3]
[1,2]UG Student, *qasimuddin2014@gmail.com*, Department of CSE, Bharath University, Chennai.
[3]Assistant Professor, *brintha.ramesh@gmail.com*,Department of CSE, Bharath University, Chennai.

***Abstract*—** Privacy-containing data publishing shows the problem of releasing sensitive data while the mining of useful information. Among present privacy models, SHA privacy algorithm provides more security and privacy model. In this paper, we address the problem of released private data, where different dataset for the same set of user are held by two parties. Here, we present an algorithm for sensitive private data released on web in the form of statistical data. After this, we propose a SHA algorithm that releases differentially private data in a secure way during the privacy computation. Experimental results on real-scenario suggest that the proposed algorithm can effectively preserve information during mining of private information.

Keyword- *Classification Analysis, Secure Data Integration, Statistical Data.*

*****

I. INTRODUCTION

Large database is used due to rapid demand on communication and storing sensitive data. Each database used by a particular user, for example census data by agency, data related to medicine used by hospitals, financial by banks, etc. Beyond this, the uses of these paradigms are more concerned in cloud computing which increases the distribution of data between multiple parties. These data distribution can be manipulated for better data mining to gain better decision support and services. For example data can be integrated to research on medical data, homeland services, and customer service. However data integration can be in such a way that no sensitive data or information can be exposed during mining of private dat. In this paper we propose an algorithm to securely combine user-specific private data from two parties. This integrated data contain the sensitive private information for data mining task.

SHA-1(Secure Hash Algorithm 1) is a 160-bit hash function which resembles the earlier MD5algorithm. This was designed by the National Security Agency (NSA) to be part of authentication system. There was some weakness in Cryptographic which were discovered in SHA-1. Cryptographic hash is like a signature for a text or a data file. Here it should keep in mind that once this password hash is generated and stored in database, you cannot convert it back to stored data as password. When each user login into application, the password hash is generated again, and match with hash stored in database over server. In this way if user forgot his/her password, It should be send him a temporary password and ask him to change it with his new one. It's common security techniques now-a-days that every released data should be encrypted. For this we also included the MD5 Message-Digest Algorithm to achieve this. The basic idea of md5 algorithm is to map data sets of variable length to data sets of a fixed length. In order to do this, the input message is split into chunks of 512-bit data block. Addition of a padding is done which is the end so that it's length can be divided by 512. After this, blocks are implemented by the MD5 algorithm and the result will be a 128-bit hash value. This algorithm generates hash is a 32-digit hexadecimal number used in encryption.

SHA1 is message-digest algorithm, which takes an input message of any length less than 2^64 bits and produces a 160-bit output as the message digest. The SHA-1 is called secure based on the SHA1 RFC document because it is computationally infeasible to find a message which corresponds to a given message digest, or to find two different messages which produce the same message digest. Any change to a message in transit will, with very high probability, result in a different message digest, and the signature will fail to verify.

The original specification of the algorithm was published in 1993 as the Secure Hash Standard, FIPS PUB 180, by US government standards agency NIST (National Institute of Standards and Technology). This version is now often referred to as "SHA0". it was withdrawn by the NSA shortly after publication and was superseded by the revised version, published in 1995 in FIPS PUB 180-1 and commonly referred to as "SHA1".

This paper is organized as follows. Session 1 describe the SHA algorithm implementation. Session 2 is described about related work. Session 3 describe about data integration and encryption techniques. Implementation of each module is described in session 4. Session 5 is about conclusion of these techniques.

 ISSN: 2321-8169
1480 - 1483

II. RELATED WORK

We briefly summarize the most recent and closely connected work, which is classic in long-term archival storage systems. This trouble is first measured under a single server. A main restriction of the above schemes is that they are designed for a single-server. If the server is fully controlled, then the above schemes can only provide detection of released data, but cannot recuperate the unique data. This lead to the propose of resourceful data checking schemes [3][4] in a multi server. By striping unnecessary data across multiple servers, the original files can still be recovered from a subset of servers even if some servers are down. Efficient data integrity checking has been proposed for different security level, there are keys in difference with our effort. First, their drawing extends the single-server. SHA [2][3] security model is used by web server and database system. They dispute that striping client data across many providers can allow customers to avoid retailer lock-in, decrease the rate of switch provider, and better bear source outages or failure. SHA-1[2][4] can reduce the cost of encryption for a large organization. SHA [1][4]is a spread cryptographic system that permits a set of servers to prove to a client that a encrypted file is intact and retrievable. This paper is described to ensure the security system and release the data over server to publish publically. During this process [6][7] they have add the encryption process in 256 bit. Efficiency of this is it generates checksum for defined data.

III. DATA INTEGRATION AND ENCRYPTION TECHNIQUES

Here encryption security system is described which provides better security level to sensitive private data. When private is released on the web it is not secure to distribute between two parties. Because hackers are keep track eyes on these data. If these data is not encrypted it is easily manipulate by hackers. For this we are providing the strongest security named as SHA algorithm. In existing system this privacy model was not used. The SHA is a family of cryptographic hash functions. It is very similar to MD5 except it generates more strong hashes. However these hashes are not always unique and it means that for two different inputs we could have equal hashes. When this happens it's called a "collision". A chance of collision in SHA is less than MD5. But, do not worry about these collisions because they are really very rare. Here this figure 1 describes the functionality of SHA encryption techniques.

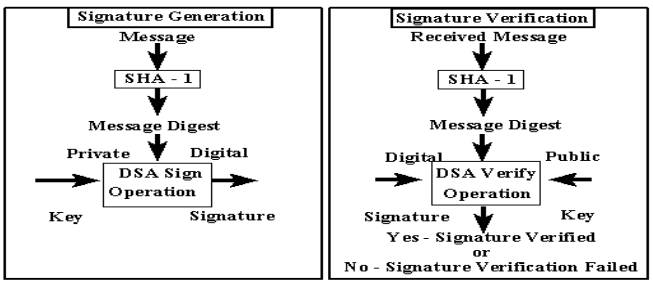

Figure 1.SHA implementation diagram.

Data Admin is the Person who is going to integrate the data in the web Server. To integrate the data on the web server, the Data Admin has to be registered on Server. Once the Data holder registers in Web server it will be encrypted.

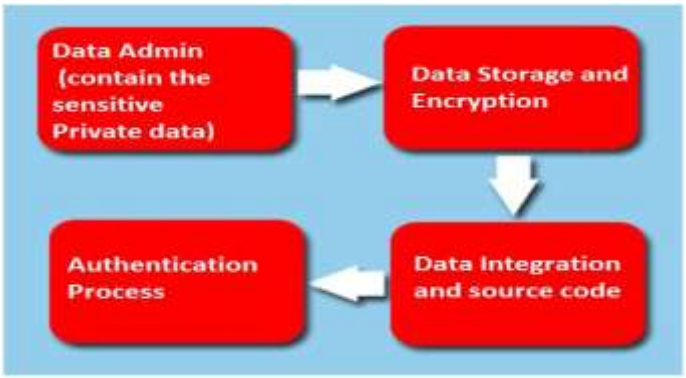

Figure 2: Step for SHA

In Data Storage and Encryption once the data was stored into the web server; the Cloud server will split the data into many parts and store all the data in the separate data servers. This security level wasn't used in existing system so that there might be a chance of hacking the entire data. Avoid the hacking process, we splitting the data and store those data in equivalent data server. We're also encrypting the data segment before storing into the data server.

In Data Integration and Source Code once the data are stored in the equivalent data servers and the keys are stored in the key servers. Encryption provides the facility to add the parity added bits, and then the data will be given to the Trusted Parity auditor. The Trusted Parity Auditor will generate the signature using change and response method. The data will be audited in this module, if any changes occur it will provide the intimation regarding the changes.

Finally we'll maintain the separate WEB server. It describes the security and authentication process. If suppose the data in the data server was misplaced, then the Main WEB server will make contact with the duplication WEB server and get the data from the duplication WEB Server. By using this

 ISSN: 2321-8169
1480 - 1483

notion, we can obtain the data if any data defeats occur. This is described in figure 3.

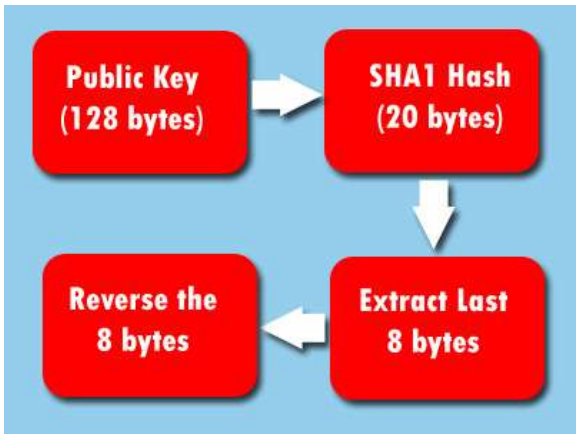

Figure 3: Encryption Process

IV. IMPLEMENTATION OF SHA

In this paper we have used the md5 algorithm and SHA algorithm to repair and reconstructed the data on the web.

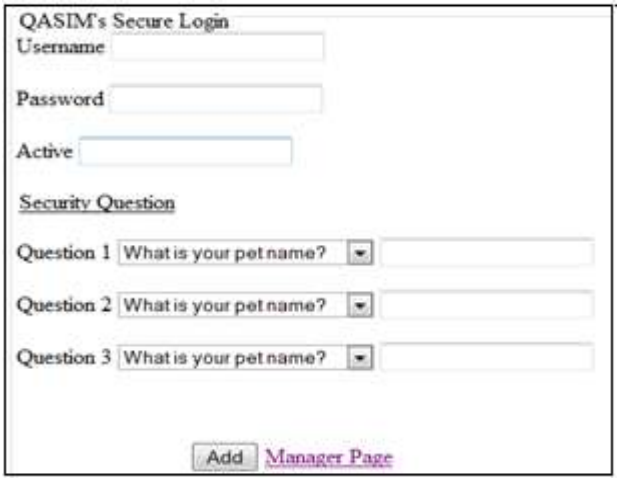

Figure 3: Registration detail

We have used windows 7 operating system to accomplish the project; ECLIPSE has been used as an included development environment. We use ECLIPSE as frontage end and SQL server 2008 as rear end. The language used for coding is JAVA and the processor to execute the project should be minimum with Pentium Dual Core 2.00GHZ and hard disk of minimum 500GB.

Here, in Figure 3 the user has to first register the account. It takes the required data and stores in database with a unique identity with different privilege. When it is retrieved it checks the security level and permission given to that data.

Figure 4: Login information

In Figure 4 the registered user has to login in their account and after clicking on login it validate the data with data stored in database. If data is matched with the data it gives control over that account.

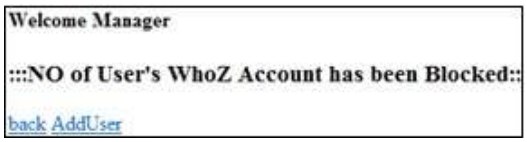

Figure 5: Manager Page will show blocked user

Here this page contains the information of the user which is unauthorized by manager. It contains the information of that party which crosses the limitation of resubmitting information when the user forgets the password and username.

V. CONCLUSION

To achieve the Data integration for two parties on the same authentication process the previously used differentially algorithm was not appropriate so we used Md5 algorithm and SHA algorithm to secure the sensitive data integration. This provides basic hash technique which encrypts the data into highly secured way. It generates checksum also which is a 160-bit hash function. However the security of those statistical data is more concern. To achieve this distributed data integration is more important and it is achieved by above methods. Different type of Keys is assigned to limit the authentication over private data.

REFERENCES

[1] R. Agrawal, A. Evfimievski, and R. Srikant, "Information Sharing Across Private Databases," Proc. ACM Int'l Conf. Management of Data, 2003.

[2] B. Barak, K. Chaudhuri, C. Dwork, S. Kale, F. McSherry, and K. Talwar, "Privacy Accuracy and Consistency Too: A Holistic Solution to Contingency Table Release," Proc. ACM Symp. Principles of Database Systems (PODS '07), 2007.

ISSN: 2321-8169
1480 - 1483

[3] R.J. Bayardo and R. Agrawal, "Data Privacy through Optimal k-Anonymization," Proc. IEEE Int'l Conf. Data Eng. (ICDE '05), 2005.

[4] R. Bhaskar, S. Laxman, A. Smith, and A. Thakurta, "Discovering Frequent Patterns in Sensitive Data," Proc. ACM Int'l Conf. Knowledge Discovery and Data Mining (SIGKDD '10), 2010.

[5] A. Blum, K. Ligett, and A. Roth, "A Learning Theory Approach to Non-Interactive Database Privacy," Proc. ACM Symp. Theory of Computing (STOC '08), 2008.

[6] J. Brickell and V. Shmatikov, "Privacy-Preserving Classifier Learning," Proc. Int'l Conf. Financial Cryptography and Data Security, 2009.

[7] [7] P. Bunn and R. Ostrovsky, "Secure Two-Party K-Means Clustering," Proc. ACM Conf. Computer and Comm. Security (CCS '07), 2007.