Credenced Database: A Guaranteed Hardware Based Database with Privacy & Data Integrity

Linu A Thomas & Amritha B

P.G. Students, Dept. of Computer science and Engg, Lourdes Matha College of Science & Tech., Kerala, India

Abstract-

We know that data need to be encrypted before outsourcing to an external sever provider since confidentiality become important. But the fact is that most of the software based cryptographic constructs that used, for server side query processing on the encrypted data naturally limit query expressiveness. In this paper a new database named Credenced Database is introduced. Credenced Database is an outsourced database prototype that allow client to execute sql queries with privacy. Here data is encrypted using combining three algorithm. Credenced database also provide platform to check the integrity of stored data to ensure that the stored data are not modified by any one. Even though their is some cost overhead and performance limitations, we can ensure that the cost per query are in magnitude cheaper than any existing system.

Keywords- Security, Database, Credenced hardware

1. INTRODUCTION

We all are much concern about our stored data, whether they are protected from data leaks and they are not modified by any anyone etc. Although there are so many benefits of outsourcing and cloud significant challenges yet lie in the path of large scale adoption since such services often require their customers to inherently trust the provider with full access to the outsourced datasets. Numerous instances of data leaks or illicit insider behaviour have left clients reluctant to place sensitive data under the control of a remote third party provider, without practical assurances of privacy and confidentiality, especially in business, healthcare and government frameworks. Moreover today’s privacy guarantees for such services are at best declarative and subject customers to unreasonable fineprint clauses. E.g., allowing the server operator to use customer behaviour and content for commercial profiling or governmental surveillance purposes. In order to store the data safely we are encrypting the data and stored in secure coprocessor database. But the fact is that processing encrypting data is very complex. Once encrypted inherent limitations in the types of operation performed on encrypted data lead to fundamental expressiveness and practical constraints.

Recent theoretical cryptography results provide encryption mechanisms that allow computation of arbitrary functions .One such encryption mechanism was proposed by marten van djik in [1] which allow computation of arbitrary function without decrypting the inputs. Unfortunately actual instances of such mechanisms seem to decades away from being practical an example of this was proposed by Rosario gennaro in [2]. Murat Kantarcioglu and Chris Clifton proposed to leverage tamperproof hardware to privately process data server side in [3]. Common wisdom so far has been that credenced hardware is generally impractical due to its performance limitations and higher acquisition costs. As a result with very exceptions [3], these efforts have stopped short of proposing or building full-fledged database processing engines. In Privacy-Preserving Data Publishing: A Survey of Recent Developments [4], Benjamin C. M. Fung provides methods and tools for publishing useful information while preserving data privacy.

This paper mainly focus on privacy criteria that provides safety guarantee. General objective of this paper is to transform the original data into some anonymous form to prevent from inferring its record owners sensitive information. Amanjot kaur proposed Hybrid Encryption For Cloud Database Security [5] which suggests a technique to enhance the security of cloud database. Here three algorithms that is RSA, Triple DES and Random Number generator algorithms are combined to provide flexible multilevel and hybrid security. The main disadvantage in this method is that it creates an additional overhead on the query performance and computational time is much high.

2. PROBLEM DEFINITION

Data must be encrypted to make data confidential. But the fact is that cryptographic overheads even for processing encrypted data by the server are extremely high even for small operation. At the same time we must provide integrity. Most of the system does not provide provision for integrity checking. It is important to confirm that even though data is encrypted it is not modified by anyone.

3. PROPOSED WORK

Here a new system consisting of credenced database is proposed. The system mainly consists of number of client, server and one credenced database. It is shown in fig: 1. Credenced database is actually a secure co-processor (SCPU). It is having high computational ability and high memory capacity. All the computation will be done on SCPU. That is encryption, decryption and processing all are done on SCPU.

A full-fledged secure database is built with leveraging server side trusted hardware. The main advantage of using secure co-processor is that computation inside SCPU is cheaper than any equivalent cryptographic operation performed on the providers unsecured common server hardware [6] . All the processing related to encrypted data will be done on secure co-processor and result will be stored in server. This will help the server to increase the speed for normal functions.

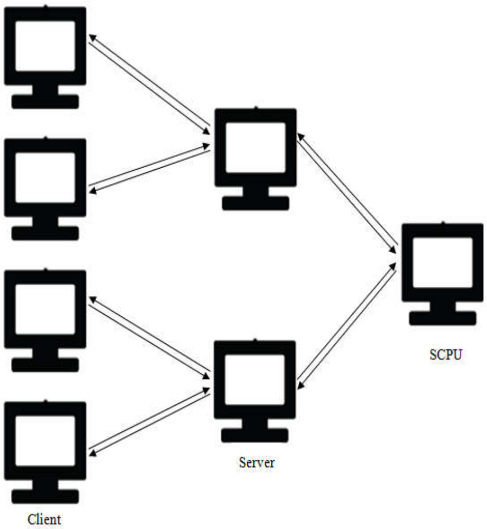

Fig: 1

Query execution in SCPU entails a set of stages [6]. (1) Initially a client defines a database schema and partially populates it. Sensitive attributes are marked using the SENSITIVE keyword. (2) Then a client send a query request to the host server through a standard SQL interface. The query is transparently encrypted. (3) Form there query is send to the scpu request Handler. (4) Then Query is decrypted at the request handler and given to query parser to construct set of plans. (5) Query Optimizer then estimates the best plan based on cost of execution. (6) Finally the result send to the client.

In this system encryption is done in secure co-processor. Each user is having various privileges, and based on this privileges each user is treated. Privileges of each user is encrypted and stored in secure co-processor. Only the admin who is handling the SCPU can see the privileges of each user. For encrypting three algorithm SHA256, base 64 and Md5 are combined together . The main advantage of this encryption method is that for even though we are encrypting same data result will be different for different time. So that even an outside see the encrypted data he cannot identify whether any two user is having same privileges or not. All the encryption, Decryption and processing are done on secure co-processor.

Another advantages is that the client can upload any document to secure coprocessor. It will be encrypted and saved in secure coprocessor. So that client and save its memory space. When ever needed the client can decrypted and download the document from secure coprocessor. Even though the document is in SCPU it is confidential, since it is encrypted using client private key. Here we are XOR encryption . This system provide a provision for the client to check the integrity of his stored document . Integrity is checked using the hash value. Not only the client but also the admin can upload and save his document in secure coprocessor. Admin's document are double encrypted. AES and Blowfish algorithms are used for encrypting admin's document. This system also provide admin to check the integrity of all the documents.

4. RELATED WORK

Damini E proposed [7] tuple-level encryption and indexes on the encrypted tuple. Range queries are processed by encrypting individual B+ -tree nodes and retrieve a desired encrypted B+ -tree node from the server, decrypt and process it. However, this leads to minimal utilization of server resources thereby undermining the benefits of outsourcing. Moreover, transfer of entire B+ -tree nodes to the client results in significant network costs.

Vignesh ganapathy in [8] proposed vertical partitioning of relations amongst multiple untrusted servers. The privacy goal of this paper is to prevent access of a subset of attributes by any single server. Gagan Aggarwal in [9] also uses vertical partitioning in a similar manner and for the same privacy goal. But there is a difference in partitioning and optimization algorithms. [10] Introduces the concepts of logical fragments to achieve the same partitioning effect in [8], [9] on a single server.

[11] propose an encryption scheme in a trusted- server model to ensure privacy of data residing on disk. In [12] Raluca proposed a system similar to [11] which ensure only privacy of data residing on disk. In order to increase query functionality a layered encryption scheme is used and then dynamically adjusted according to client queries. Credenced DB on the other hand operates in an un-Trusted server model. In [13] Mustafa proposed a new query optimizer that takes into account both performance and disclosure risk for sensitive data. Individual data pages are encrypted by secret keys that are managed by a trusted hardware module. The decryption of the data pages and subsequent processing is done in server memory.

In [14] Alexander SCPU are used to retrieve X509 certificates from a database. However, this only supports key based lookup. Each record has a unique key and a client can query for a record by specifying the key. [15] uses multiple SCPUs to provide key based search. The entire database is scanned by the SCPUs to return matching records. Rakesh in [16] implements arbitrary joins by reading the entire database through the SCPU, Such as approach is clearly not practical for real implementations since it is lower bounded by server-SCPU bandwidth

5. CONCLUSION AND FUTURE WORK

This paper mainly focused to establish a secure full fledged privacy enabling credenced database which provide confidentiality and data integrity. This paper also help to minimise query execution cost and time. Since computation inside secure coprocessor is cheaper than any other method used in provider’s unsecure hardware. This work can extended to find out some mechanism to prevent modifications in encrypted data.

REFERENCES

[1] Marten van Dijk, Craig Gentry, Shai Halevi, and Vinod Vaikuntanathan. Fully homomorphic encryption over the integers. In Henri Gilbert, editor, EUROCRYPT, volume 6110 of Lecture Notes in Computer Science, pages 24–43. Springer, 2010

[2] Rosario Gennaro, Craig Gentry, and Bryan Parno. Non-interactive verifiable computing: Outsourcing computation to untrusted workers. In Tal Rabin, editor, CRYPTO, volume 6223 of Lecture Notes in Computer Science, pages 465–482. Springer, 2010.

[3] Luc Bouganim and Philippe Pucheral. Chip-secured data access: confidential data on untrusted server. In Proceedings of VLDB, pages 131–141. VLDB Endowment, 2002.

[4]BENJAMIN C. M. FUNG,KE WANG,RUI CHEN,and PHILIP S. YU Privacy-Preserving Data Publishing: A Survey of Recent Developments. In ACM Computing Surveys, Vol. 42, No. 4, Article 14, Publication date: June 2010

[5]Amanjot Kaur, Hybrid encryption for cloud database security In International journal of engineering science & advanced technology ,Volume-2,2012

[6]Sumeet Bajaj and Radu Sion TrustedDB: A Trusted Hardware based Database with Privacy and Data Confidentiality. In IEEE Transactions on knowledge and data engineering, VOL. 26, NO. 3, MARCH 2014

[7] Hakan Hacigumus, Bala Iyer, Chen Li and Sharad Mehrotra. Executing SQL over Encrypted Data in the Database-Service- Provider Model. In Proceedings of SIGMOD, pages 216–227, 2002.

[8]Vignesh Ganapathy, Dilys Thomas, Tomas Feder, Hector Garcia- Molina, and Rajeev Motwani. Distributing data for secure database services. In Proceedings of PAIS, pages 8:1–8:10, New York, NY, USA, 2011. ACM.

[9]Gagan Aggarwal, Mayank Bawa, Prasanna Ganesan, Hector Garcia-Molina, Krishnaram Kenthapadi, Rajeev Motwani, Utkarsh Srivastava, Dilys Thomas, and Ying Xu 0002. Two can keep a secret: A distributed architecture for secure database services. In CIDR, pages 186–199, 2005.

[10] Valentina Ciriani, Sabrina De Capitani Di Vimercati, Sara Foresti,Sushil Jajodia, Stefano Paraboschi, and Pierangela Samarati. Combining fragmentation and encryption to protect privacy in data storage. ACM Trans. Inf. Syst. Secur., 13(3):22:1–22:33, July 2010.

[11] Tingjian Ge and Stan Zdonik. Fast, secure encryption for indexing in a column-oriented dbms. In ICDE, 2007.

[12] Raluca Ada Popa, Catherine Redfield, and Nickolai Zeldovich. Cryptdb: protecting confidentiality with encrypted query processing. In Proceedings of SOSP, 2011.

[13] Mustafa Canim, Murat Kantarcioglu, Bijit Hore, and Sharad Mehrotra. Building disclosure risk aware query optimizers for relational databases. Proc. VLDB Endow., 3(1-2):13–24, September 2010.

[14] Alexander Iliev and Sean WSmith. Protecting Client Privacy with Trusted Computing at the Server. IEEE, Security and Privacy, 3(2), Apr 2005.

[15] S. W. Smith and D. Safford. Practical server privacy with secure coprocessors. IBM SYSTEMS JOURNAL, 40(3), 2001.

[16] Rakesh Agrawal, Dmitri Asonov, Murat Kantarcioglu, Yaping Li. Sovereign Joins. In Proceedings of the 22nd International Conference on Data Engineering, page 26. IEEE Computer Society, 2006.