RESEARCH ARTICLE OPEN ACCESS

Enabling Data Privacy and Data Confidentiality on Database

Vinodh Kumar.J[1] , Sivamohan.S[2]
[1]Department of CSE, [2]Department of IT,
SRM University,Chennai, India,

--************************----------------------------

ABSTRACT:

Organizations outsource their database to service provides and there are concerns raised on data confidentiality and privacy of customer data. To avoid such apprehension, data are encrypted before it is farm out. Any software-based cryptographic methods deployed on server-side query to process encrypted data limits query expressiveness. Here a setup is initiated called Trusted DB which is an outsourced model that allows users to run their query with privacy and under regulatory norms by minimizing server load in significant query execution phase. Trusted DB does not limit query performance and the cost per query is orders of degree lower than any existing software-only mechanisms available.

Keywords: **Secure Co-Processor (SCPU) , tamper-proof , Trusted DB , homomorphism**

--************************----------------------------

1. INTRODUCTION

Almost all organizations have their database outsourced and cloud storage is maintained by third party vendor's who offer a database service of some kind as part of their overall solution. The customers need to essentially trust the provider with full access to the outsourced data sets. But several occasions of prohibited insider behavior or data leaks have left clients unwilling to place sensitive data under the control of service provider, without practical guarantee of privacy and confidentiality especially in banking, trade, healthcare and government.

There are many existing research to deal with such outsourcing security aspect by encrypting data before it is outsourced. Once encrypted however, restrictions in the types of primitive operation performed on encrypted data lead to fundamental expressiveness and practicality constraint. Modern speculative cryptography results provide hope by proving the being of universal homomorphism, i.e. encryption methods that allow computation of arbitrary function without decrypting the inputs. But instances of such method seem to be decades away from being realistic.

Furthermore, the sensitive nature of the data makes it desirable to protect data from both tampering and accidental corruption. Therefore, the data should be stored in a scalable and trusted database system. There where ideas proposed to control tamper-proof hardware to privately process data server-side. Although there is a common perception that trusted hardware is generally unrealistic due to its performance limitations and higher purchase costs.

There are recent insights on the cost and performance tradeoff that suggest approach somewhat differently. In particular computation on secure processors is order of degree lower than any cryptographic operations performed on encrypted data on service provider's unsecured server hardware. This is due to cryptography expenses that allow processing on encrypted data even for simple operations extremely high. Therefore, we speculate that a complete privacy enabling secure database leveraging server-side trusted hardware can be built and run at a fraction on common server hardware. This is confirmed by building TrustedDB, an SQL database processing engine that makes use of tamper-proof cryptographic coprocessors to the outsourced data.

Designing a secure coprocessor tamper resistant with computational ability and memory capacity is very challenging. Trusted DB achieves this by accessing external storage while preserving data confidentiality with on-the-fly encryption and preprocessing client queries to identify sensitive components to run into the secure coprocessor. Non-trusted components are divest to host server which improves performance and the transaction cost.

2. RELATED WORK

The Work presented by H. Hacigumus, B. Iyer, C. Li, and S. Mehrotra [20] propose dissection of data into undisclosed partitions and rewriting of queries over the original data in terms of the partition identifiers which balances a tradeoff between client and server-side processing.

According to H Wang and L.V.S. Lakshmanan use of Order preserving encryption for querying encrypted XML databases. Another technique referred to as splitting and scaling is used to decrease the frequency distribution of encrypted data from that of the plaintext data. Here, each plaintext value is encrypted using multiple distinct keys. Then, corresponding values are simulated to ensure that all encrypted values occur with the same frequency thereby discomforting any frequency-based attacks.

T. Ge and S. Zdonik [16] propose an encryption scheme in a trusted-server model to ensure privacy of data residing on disk. The FCE scheme designed here is equivalently secured as a block cipher, however, with increased efficiency. In [42] they have proposed executing aggregation queries with confidentiality on an untrusted server.

The work by L. Bouganim and P. Pucheral [25] uses a smart card for query processing and for enforcing access rights. The client query is split such that the server performs majority of the computation. The solution is limited by the fact that the client query executing within the smart card cannot generate any intermediate results since there is no storage available on the card.

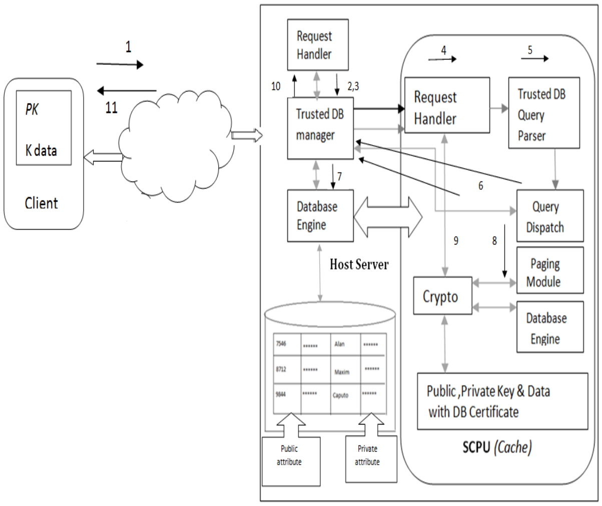


Figure. 1 Trusted DB Architecture

Work proposed by B.I.H. Hacigumus and S. Mehrotra [19] suggest decryption performed within the secure confinements of the SCPU, thereby processing is done on plaintext data. Also limitations on the nature of predicates that can now be employed on encrypted attributes including arbitrary user are used in defined functions.

3. ARCHITECTURE

3.1 Overview

To remove storage limitations in Secure Co-Processor /Cache (SCPU), the organization data is outsources and stored at the host provider's site. The query processing engines are available at both host server and in the SCPU. The database attributes are classified as public or private. Private attributes are marked as Sensitive by the client and are encrypted which can only be decrypted by the client or by the SCPU. SCPU memory limitations are not hurdled since the entire database resides outside the SCPU. The SCPU-side query processing engine access pages on demand using the Paging Module.

Query executions follow a set of stages as per *Figure 1*.

1) At first a client defines a database schema where data is populated and Sensitive attributes are marked, i.e., by deploying the "SENSITIVE" keyword where the client layer processes it by encrypting the corresponding attributes:

CREATE TABLE Employee (ID integer primary key, Name Varchar (120) SENSITIVE, Address Varchar (120) SENSITIVE);

2) The Client issues query request through SQL interface to host server and the query is encrypted using public key of SCPU at the client side.
3) Host Server forwards the query which is encrypted to SCPU request handler.
4) Request handler Inside the SCPU decrypts the query using its private key and forwards to Query Parser.

5) Query is parsed by generating set of execution plans which is constructed by rewriting the original query from the client into set of sub-queries. Sub query in the execution plan is identified as private and public.

6) Query Optimizer estimates the cost of execution plan and selects the best plan which has low cost and forwards to query dispatcher.

7) The Query dispatcher forwards the Private queries to SCPU database engine and Public queries to host server, here the maximum load run is done by host server database engine thus utilizing its cheap run cycles.

8) The end query is assembled from result returned by host server database engine which had processed public query and from SCPU database engine which processed private queries.

9) This final query is encrypted using clients public key and digitally signed by the query dispatcher in SCPU using private key if SCPU to maintain correctness assurance and forwarded to trusted DB Agent.

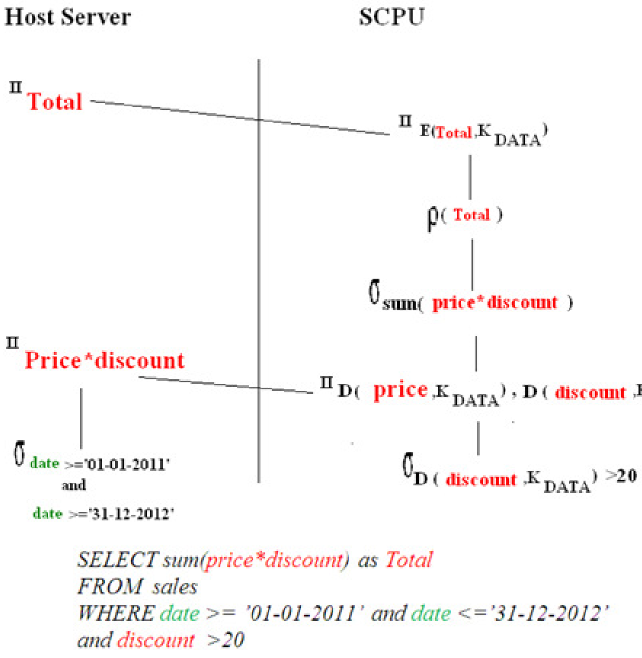
10) Trusted DB Agent forwards the net result to request handler and then to client where the data is decrypted by providing client's private key (entered by client).

3.2 Query Parsing

Confidential attributes can occur anywhere in a query like in SELECT, WHERE, GROUP clauses, aggregations, sub-queries, Joins. Query Parser job is to ensure that private attributes are to be processed only inside the SCPU. The private attributes are encrypted by shared key between client and SCPU, so host server cannot interpret these attributes. Optimized client query select with lower cost are processed by the host server which significantly enhances the performance.

To illustrate on how security is maintained, we propose a specific attribute classification into private and public types. In elaborate private attributes are encrypted and those data are confidential which preserve client's privacy especially in business, healthcare and government frameworks. Public attributes are non-encrypted data which is available for public access.

Figure 2. shows the resulting query plan which is processed by host server and SCPU, the host server first executes public query that filters all tuples that fall within the defined date range which being public attribute. Result from the public query is then processed by SCPU to filter private attribute discount. The aggregation on another private attribute price is done by decrypting inside the SCPU after the aggregate operation which is then encrypted before sending it to client.



```
SELECT sum(price*discount) as Total
FROM sales
WHERE date >= '01-01-2011' and date <='31-12-2012'
and discount >20
```

Figure 2. Query Parsing – Green indicates public and Red indicates Private attributes

3.2.2 DML Example

Data Manipulation queries (INSERT, UPDATE) also undergo a rewrite. For illustration consider the DML query

UPDATE EMPLOYEES SET BONUS =BONUS+ 2500 WHERE DEPARTMENT_ID=35.

If BONUS is a private attribute then SCPU decrypts, performs the addition and then encrypts the updated result. But If the BONUS is a public attribute then operation is performed inside host server.

3.2.3 Query Optimization

Query Optimizer Optimizes the query in the database system by first The Query Plan Generator constructs possibly multiple explain plans for the client query. For each constructed plan the Query Cost Estimator computes execution cost of that plan. The best plan i.e. one with the lowest cost is selected and passed on to the Query Plan Interpreter for execution.

To estimate the best plan the cost estimator is provided with key information such as availability of an index, possible distinct values of an attribute, average time to read a page, average transfer time between host server and SCPU, numbers of CPU cycles per execution .These information are stored in System Catalog Along with this Database parameters such as CPU and host server configurations and Data statistics about the database table are feed as inputs to the Cost estimator. This information is periodically updated by running batch process overnight or weekly.

4. KEY ASPECTS

4.1 Security

Data encryption is an aspect which plays vital role in providing security and ensure confidentiality is maintained. There are other aspects where privacy of client data is ensured by having tamper-proof SCPU which is done by having tamper-resistant physical requirements. Processing of confidential data inside the SCPU by deploying public-private key cryptography in key messaging stages. Both client and the SCPU contains a Public-private key and messages sent between them are encrypted.

4.2 Key Management

Data stored on host server disk can be encrypted/decrypted only by use of single master encryption key known only to the SCPU. Since all operations are performed by the SCPU, the master key is stored within SCPU and is never communicated outside. All decryptions required as part of query processing uses the master key. The SCPU encrypts the only when a sensitive attribute value is to be communicated to the client. This way, only the authorized client can access the data.

5. CONCLUSION

The above thesis scales out an outsourced contexts, computation inside secure processors is orders of magnitude cheaper than cryptography operations performed and provides low cost with optimized techniques leading to high secured database.

6. ACKNOWLEDGMENTS

This paper has benefited from conversations with many different people – far more than can be acknowledged completely here.

7. REFERENCES

[1] H.Hacigumus, B.Iyer, C.Li, and S.Mehrotra, “Executing SQL over Encrypted Data in the Database-Service-Provider Model,”Proc. ACM SIGMOD Int’l Conf. Management of Data (SIGMOD ’02), pp. 216-227, 2002.

[2] H. Wang and L.V.S.Lakshmanan, “Efficient Secure Query Evaluation over Encrypted XML Databases,” Proc. 32nd Int’l Conf. Very Large Data Bases (VLDB), 2006.

[3] T.Ge and S.Zdonik, “Fast Secure Encryption for Indexing in a Column-Oriented DBMS,” Proc. IEEE 23rd Int’l Conf. Data Eng.(ICDE), 2007.

[4] L.Bouganim and P.Pucheral, “Chip-Secured Data Access: Confidential Data on Untrusted Server,” Proc. 28th Int’l Conf.Very Large Data Bases (VLDB ’02), pp. 131-141, 2002.

[5] B.I.H.Hacigumus and S.Mehrotra, “Efficient Execution of Aggregation Queries over Encrypted Relational Databases,” Proc.Ninth Int’l Conf. Database Systems for Advanced Applications, vol. 2973, pp. 633-650, 2004.

[6] Marten Van Dijk, Craig Gentry, Shai Halevi, and Vinod Vaikuntanathan.”Fully homomorphic encryption over the integers”. Volume 6110 of Lecture Notes in Computer Science, pages 24–43. Springer, 2010.

[7] 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.

[8] O.Goldreich. Foundations of Cryptography I. Cambridge University Press, 2001.

[9] M.O.Rabin “Digitalized signatures and public-key functions as intractable as factorization”. Technical Report TR-212, Cambridge, MA, USA, 1979.

[10] Sean W Smith. Outbound authentication for programmable secure coprocessors. Darmouth College, Technical Report TR2001-401. Online at http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.58.4066, 2001..

[11] Sumeet Bajaj and Radu Sion. TrustedDB: A Trusted Hardware based Database with Privacy and Data Confidentiality. In Proceedings of the ACM SIGMOD Conference, pages 205–216, 2011.