International Journal of Computer Applications Technology and Research
Volume 3– Issue 11, 661 - 663, 2014, ISSN: 2319–8656

Protecting Global Records Sharing with Identity Based Access Control List

Vidhya . P
Computer Science and Engineering
V.S.B Engineering College
Tamilnadu, India

A.P.V Raghavendra
Computer Science and Engineering
V.S.B Engineering College
Tamilnadu, India

Abstract: Generally, the information is stored in the database. Protecting sensitive information are encrypted before outsourcing to a service provider. We send the request to service provider through SQL queries. The query expressiveness is limited by means of any software-based cryptographical constructs then deployed, for server-side query working on the encrypted data.Data sharing in the service provider is emerging as a promising technique for allowing users to access data. The growing number of customers who stores their data in service provider is increasingly challenging users' privacy and the security of data. The TrustedDB an outsourced database prototype that allows clients to execute SQL queries with privacy and under regulatory compliance constraints by leveraging server-hosted. Tamper-proof believed hardware in crucial query processing levels, thereby removing any limits on the type of supported queries. It focuses on providing a dependable and secure data sharing service that allows users dynamic access to their information. TrustedDB is constructed and runs on hardware, and its performance and costs are evaluated here.

Keywords: Database architectures, security, privacy, special-purpose hardware**.**

1. INTRODUCTION

Although the benefits of outsourcing and clouds are well known, imperative 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 data sets. Numerous instances of illicit insider behavior or data leaks have left clients backward to place known data under the control of a private, unknown provider, without practical assurances of privacy and confidentiality, especially in occupation, healthcare, and public sectors. Moreover, today's privacy assurance for such services are at best informative and subject customers to unreasonable fine-print clauses. For example, allowing the server operator to use customer behavior and content for commercial profiling or governmental control purposes.

Existing analysis addresses several such safety aspects, including entry's privacy and finds on enciphered data. In most of these efforts, data are encrypted before outsourcing. Once enciphered however, essential limitations in the types of basic operations that can be performed on encrypted data lead to fundamental expressiveness and practicality constraints. However, recent insights into the cost performance tradeoff seem to suggest that things stand somewhat differently. Specifically, at scale, in outsourced contexts,computation inside secure processors is orders of degree lower than any equivalent cryptographic operations performed on the provider's unsecured server hardware, despite the overall higher achievement cost of secure hardware.

We conclude that a complete privacy enabling secure database leveraging server-side trusted hardware can be built and run at a fraction of the cost of any (existing or future) cryptography-enabled private data working on common server hardware. We verify this by signing and building TrustedDB, an a SQL database processing engine that makes use of tamper-proof cryptographic coprocessors such as the IBM 4764 [1] in close adjacency to the outsourced data.

Change opposes designs, however, is imperatively constrained in both computational ability and memory capacity which makes implementing fully featured database solutions using secure coprocessors (SCPUs) very demand. Trusted complete this by applying common unsecured server resources to the maximum available duration. For example, Trusted permits the SCPU to clearly access external storage while protecting data confidentiality in the family of encryption. This removes the limitations on the size of databases that can be supported. Moreover, client queries are preprocessed to identify knowing components to be run inside the SCPU. Non knowing operations are offloaded to the interested server owner. This highly increases work and decrease the cost of the agreement.

Overall, despite the overheads and performance limitations of trusted hardware, the costs of running TrustedDB are orders of degree lower than any (existing or) potential future cryptography-only mechanisms. Moreover, it does not limit query expressiveness. This paper's improvement is triples:1)The opening of current cost setups and judgments that explain and specify the advantages of utilizing trusted hardware for data working, 2) the design and advancement of TrustedDB, a reliable hardware based relational database with full data confidentiality and no limitations on query expressiveness, and 3) defined query optimization approach in a trusted hardware-based query execution model.

2. RELATED WORK

Queries on encrypted data. To propose a division of data into secret partitions and rewriting of range queries over the original data in terms of the resulting separation qualifiers. This balances a judge between client and server-side working, as a function of the data subdivision size. Some authors explore optimal bucket sizes for range queries.

Vertical partitioning of relations amongst multiple untrusted servers is employed in [2]. Here, the privacy goal is to a prevent access of a subset of attributes by any single server. For example, {Name, Address} can be a privacy sensitive access-pair and query processing needs to ensure that they are not jointly visible to any single server. The client query is split into multiple queries wherein each subquery fetches the relevant data from a server and the client combines results from multiple servers. TrustedDB is equivalent to when the size of the privacy subset is one and hence a single server answers. In this case, each attributes column demand encryption to ensure privacy. Hence, TrustedDB to optimize for querying encrypted columns since otherwise they rely on client-side decryption and processing.

To introduce the concept of logical fragments to achieve the same partitioning effect on a single server. A fragment here is simply a relation wherein attributes not desired to be visible in that fragment are encrypted. TrustedDB (and other solutions) are in effect concrete mechanisms to efficiently query any individual fragment from [10]. The work on the other hand, can be used to determine the set of attributes that should be encrypted in TrustedDB.

The goal is to minimize the lifetime of sensitive data and keys in server memory after decryption. In TrustedDB, there is no such disclosure risk since decryptions are performed only within the SCPU. In TrustedDB, all decryptions are performed within the secure confinements of the SCPU, thereby processing is done on the plaintext data. This removes any limitation on the nature of predicates that can now be employed on encrypted attributes including arbitrary user decided functions. We note that certain solutions destined for a very specific set of predicates can be more efficient albeit at the loss of functionality.

Trusted hardware. In [4], SCPUs are used to retrieve X509 certificates from a database. However, this only supports key based lookup. Each record has a single key and a user can query for a record by specifying the key. Multiple SCPUs are used to provide key based search. The entire database is scanned by the SCPUs to return matching records. Chip-Secured Data Access [5] uses a smart card for query processing and for enforcing access rights. The client query is split such that the server performs the 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. In follow-up work, GhostDB [6] proposes to embed a database inside a USB key equipped with a CPU. It allows linking of private data carried in the USB Key and public data available on a server. Ghost DB ensures that the only information revealed to a potential spy is the query issued and the public data accessed.

Both [5] and [6] are subject to the storage limitations of trusted hardware which in turn limits the size of the database and the queries that can processed. In contrast, TrustedDB uses external storage to store the entire database and reads information into the trusted hardware as needed which enables it to be used with large databases. Moreover, database pages can be swapped out of the trusted hardware to external storage during query processing.

3. MODULES DESCRIPTION

A. Query Parsing and Execution

In the first stage a client defines a database schema and partially occupy it. Knowing attributes are noted using the SENSITIVE keyword which the client layer transparent processes by encrypting the corresponding attributes:

```
CREATE TABLE customer (ID integer
    primary key,
Name char (72) SENSITIVE, Address
    char (120)   SENSITIVE).
```

(1) Later, a client sends a query request to the host server through a standard SQL interface. The query is clearly enciphered at the client site using the public key of the SCPU. The server owner thus cannot decipher the query.
(2) The server owner serves the enciphered query to the Request Handler inside the SCPU.
(3) The Request Handler deciphers the query and serves it to the Query Parser. The query is parsed achieving a set of ideas. Each idea is worked by editing the original client query into a set of sub-queries, and, presenting to their end data set allocation, each sub-query in the idea is qualified as being either public or private.
(4)The Query Optimizer then evaluates the execution costs of each of ideas and selects the best idea (one with least cost) for execution serving it to the dispatcher.
(5) The Query Dispatcher serves the public queries to the server owner and the private queries to the SCPU database engine while handling responsibilities. The net result is that the maximum possible work is run on the server owner reduced cycles.
(6) The final query result is assembled, enciphered, digitally signed by the SCPU Query Dispatcher, and dispatched to the client.

B. Query optimization process

At a high level query optimization in a database system works as follows. (i) The Query Plan Achiever works possibly multiple ideas for the client query. (ii) For each worked idea the Query Cost Evaluator computes an evaluate of the execution cost of that idea. (iii) The best idea i.e., one with the lowest cost, is then selected and passed on to the Query Idea Interpretor for execution. The query development process in TrustedDB works similarly with key differences in the Query Cost Evaluator due to the logical separating of data mentioned above.

C. System Catalog

Any query idea is composed of multiple individual execution steps. To evaluate the cost of the entire idea it is essential to evaluate the cost of individual steps and combine them. In order to evaluate these costs the Query Cost Evaluator needs access to some key information. E.g., the possibility of an index or the knowledge of possible distinct values of an attribute. These sets of instruction are collected and stored in the System Catalog. Most available databases today have some form of the continuously updated System Catalog.

D. Analysis of Basic Query Operations

The cost of an idea is the separate of the cost of the steps that comprise it. In this section we present how execution times for a certain set of basic query idea steps are evaluated.

4. SYSTEM FLOW DIAGRAM

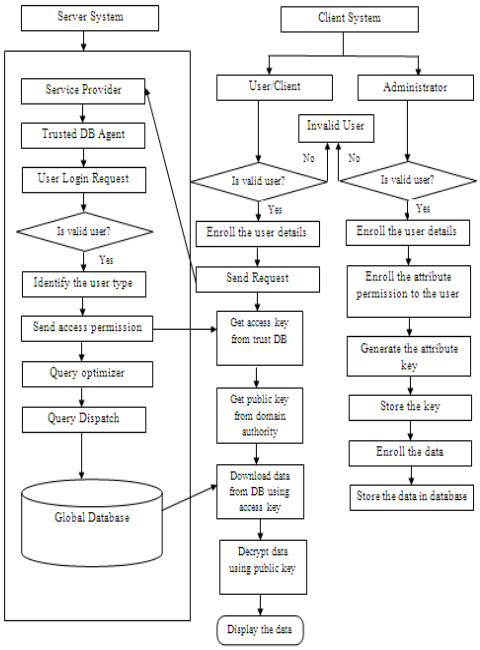

Fig: 1 System Flow Diagram

A client sends a query request to the host server through a standard SQL interface. The query is clearly enciphered at the client site using the public key of the SCPU. The server owner thus cannot decipher the query. The server owner serves the enciphered query to the Request Handler inside the SCPU. The Request Handler deciphers the query and serves it to the Query Parser. The query is parsed achieving a set of ideas. Each idea is worked by editing the original client query into a set of sub-queries, and, presenting to their end data set allocation, each sub-query in the idea is qualified as being either public or private. The Query Optimizer then evaluates the execution costs of each of ideas and selects the best idea (one with least cost) for execution serving it to the dispatcher. The Query Dispatcher serves the public queries to the server owner and the private queries to the SCPU database engine while handling responsibilities. The net result is that the maximum possible work is run on the server owner reduced cycles. The final query result is assembled, enciphered, digitally signed by the SCPU Query Dispatcher, and dispatched to the client.

4. CONCLUSION

This paper's improvement are triples:

1)The opening of current cost setups and judgments that explain and specify the advantages of utilizing trusted hardware for data working, 2) the design and advancement of TrustedDB, a reliable hardware based relational database with full data confidentiality and no limitations on query expressiveness, and 3) defined query optimization approach in a trusted hardware-based query execution model.

This work's inherent proposal is that, at order, in expand contexts, computation inside secure hardware processors is orders of degree lower than equivalent cryptography performed on suppliers' unsecured server hardware, even though the overall higher achievement cost of protected hardware. We thus propose to make reliable hardware a first-class member in the secure data management field. Moreover, we promise that cost-centric insights and architectural standards will fundamentally change the way systems and algorithms are signed.

5. REFERENCES

[1] IBM 4764 PCI-X Cryptographic Coprocessor, http://www03.ibm.com/security/cryptocards/pcixcc/overview.shtml, 2007.

[2] V. Ganapathy, D. Thomas, T. Feder, H. Garcia-Molina, and R.Motwani, "Distributing Data for Secure Database Services," Proc.Fourth Int'l Workshop Privacy and Anonymity in the Information Soc.(PAIS '11), pp. 8:1-8:10, 2011.

[3] V. Ciriani, S.D.C. di Vimercati, S. Foresti, S. Jajodia, S. Paraboschi,and P. Samarati, "Combining Fragmentation and Encryption to Protect Privacy in Data Storage," ACM Trans. Information and System Security, vol. 13, no. 3, pp. 22:1-22:33, July 2010.

[4] A. Iliev and S.W. Smith, "Protecting Client Privacy with Trusted Computing at the Server," IEEE Security and Privacy, vol. 3, no. 2, pp. 20-28, Mar./Apr. 2005.

[5] 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.

[6] N. Anciaux, M. Benzine, L. Bouganim, P. Pucheral, and D. Shasha,"GhostDB: Querying Visible and Hidden Data Without Leaks,"Proc. 26th Int'l ACM Conf. Managem ent of Data (SIGMOD), 2007.