International Journal of Emerging Technology in Computer Science & Electronics (IJETCSE)
ISSN: 0976-1353 Volume 25 Issue 4 – APRIL 2018.

ENHANCED PRIVACY PRESERVING MODEL FOR TRUSTED BASED DATABASE SYSTEMS

R. Mohana Priya[#1], G. Indhumathi [#2] and K. Bhuvaneshwari[*3]

[# 1, 2] *M.Sc computer science (Student), Department of computer science, Kamban College of arts and science for women, India.*

[*] *Assistant Professor, Department of computer science, Kamban College of arts and science for women, India.*

***Abstract*- Data confidentiality is one of the major security concerns of the data outsourcing systems. Techniques to protect data against curious attackers are particularly important for users of public cloud services. 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. In this paper, we propose an enhanced privacy and confidentiality mechanism for TrustedDB. The objective of this study is to provide an enhanced privacy preserving model for trusted hardware based database systems. It's a prototype developed using SQL queries. The client executes their data by secured SQL queries under regulatory compliance that leverages the server based components. By doing so, the cost overhead and performance of trusted components are effectively achieved. Experimental results have shown the efficiency of the proposed systems.**

Keywords: Data confidentiality, Data outsourcing, SQL operation, Queries, Privacy and cost overhead.

I. INTRODUCTION

The Overview of outsourcing and clouds are well known, 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 data sets. Numerous instances of illicit insider behavior or data leaks 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 fine print clauses. It's allowing the server operator to use customer behavior and content for commercial profiling or governmental surveillance purposes [1]. Tamper resistant designs, however, are significantly constrained in both computational ability and memory capacity which makes implementing fully featured database solutions using secure coprocessors (SCPUs) very challenging.

Despite the cost overhead and performance limitations of trusted hardware, we show that the costs per query are orders of magnitude lower than any (existing or) potential future software-only mechanisms. Trusted is built and runs on actual hardware and its performance and costs are evaluated here. In most of these efforts, data are encrypted before outsourcing. Once encrypted however, inherent limitations in the types of primitive operations that can be performed on encrypted data lead to fundamental expressiveness and practicality constraints. As soon as confidentiality becomes a concern, data needs to be encrypted before outsourcing [2]. Once encrypted, solutions can be envisioned that: (A) straightforwardly transfer data. Back to the client where it can be decrypted and queried, (B) deploy cryptographic constructs server-side to process encrypted data, and (C) process encrypted data server-side inside tamper-proof enclosures of trusted hardware.

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 [3]. This is due to cryptography expenses that allow processing on encrypted data even for simple operations extremely high.

The rest of the paper is organized as follows: Section II presents the related work; Section III presents the proposed work; Section IV represents experimental analysis and finally concludes in Section V.

II. RELATED WORK

This section presents the prior technique suggested of the proposed study. Clients of outsourced databases need Query Authentication (QA) guaranteeing the integrity (correctness and completeness), and authenticity of the query results returned by potentially compromised providers. Existing results provide QA assurances for a limited class of queries by deploying several software cryptographic constructs. Here, we show that, to achieve QA, however, it is significantly cheaper and more practical to deploy server-hosted, tamper-proof co-processors, despite their higher acquisition costs [4]. Further, this provides the ability to handle arbitrary queries. To reach this insight, we extensively survey existing QA work and identify interdependencies and efficiency relationships. We then introduce CorrectDB, a new DBMS with full QA assurances, leveraging server-hosted, tamper-proof, trusted hardware in close proximity to the outsourced data.

Data is placed by the data owner with a remote, untrusted service provider. For authenticity and integrity of query results the data owner computes and places additional authentication data with the provider. After the initial upload, the server-side, trusted hardware manages the authentication data on behalf of the data owner. The data owner also issues search and update queries to the provider [5]. Client's authorized by the data owner query the outsourced datasets through an interface exposed by the provider. A client query can either perform a search or request a data-only-update operation. Client's have no access to the server-side authentication data. A secure and privacy-assured service outsourcing is used in cloud computing which uses Linear programming [6] and compressed sensing techniques to transform images, which aims to take security, complexity, and efficiency into consideration from the very beginning of the service flow.

Because data explosion [7] is the fast-growing trend to outsource the image management systems to cloud and leverage its economic yet abundant computing resources to efficiently and effectively acquire, store, and share images from data owners to a large number of data users. Although outsourcing the image services is quite promising, in order to become truly successful, it still faces a number of fundamental and critical challenges, among which security is the top concern. To initiate the investigation for these challenges and propose a novel outsourced image recovery service (OIRS) architecture [8] with privacy assurance [9]. For the simplicity of data acquisition at data owner side, OIRS is specifically designed under the compressed sensing framework. The acquired image samples from data owners are later sent to cloud, which can be considered as a central data hub and is responsible for image sample storage and provides on-demand image reconstruction service for data users. But it makes more complexity because the data is sent in its raw form to one cloud [10]. The cryptography schemes are computationally more complex.

III. PROPOSED WORK

This section presents the proposed privacy preserving models. The proposed model composes of four phases in which each represents its own security measures.

A) QUERY parsing and execution

In this stage, the client's responsibility on their data is been discussed. The client stores their data using database schema. In general, the sensitive attributes are labeled as Sensitive keywords and stored onto the database. The sample query shows the encrypting process of the corresponding attributes.

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

By this standard SQL query, the client transfers their query to host server. At the client side, the query is encrypted with the secret key. In similar way, the host server can't decrypt the query. The request handler component decrypts the query and then forward to the query browser. The query is parsed generating a set of plans. Each plan is constructed by rewriting the original client query into a set of sub-queries, and, according to their target data set classification, each sub-query in the plan is identified as being either public or private. The Query Optimizer then estimates the execution costs of each of the plans and selects the best plan (one with least cost) for execution forwarding it to the dispatcher. The Query Dispatcher forwards the public queries to the host server and the private queries to the SCPU database engine while handling dependencies. The net result is that the maximum possible work is run on the host server's cheap cycles. The final query result is assembled, encrypted, digitally signed by the SCPU Query Dispatcher, and sent to the client.

B) Query optimization process:

Query optimization process works as follows:

(i) The Query Plan Generator constructs possibly multiple plans for the client query.

(ii) For each constructed plan the Query Cost Estimator computes an estimate of the execution cost of that plan.

(iii) The best plan i.e., one with the least cost, is then selected and passed on to the Query Plan Interpreter for execution.

C) Catalog of the system:

Any query plan is composed of multiple individual execution steps. To estimate the cost of the entire plan it is essential to estimate the cost of individual steps and aggregate them. In order to estimate these costs the Query Cost Estimator needs access to some key information. These sets of information are collected and stored in the System Catalog. Most available DBMS today have some form of periodically updated System Catalog.

D) Basic query operations:

The cost of a plan is the aggregate of the cost of the steps that comprise it. A data scan is employed to collect statistics about the actual data. This scan is configured to run periodically. The statistics on public attributes are collected server-side.

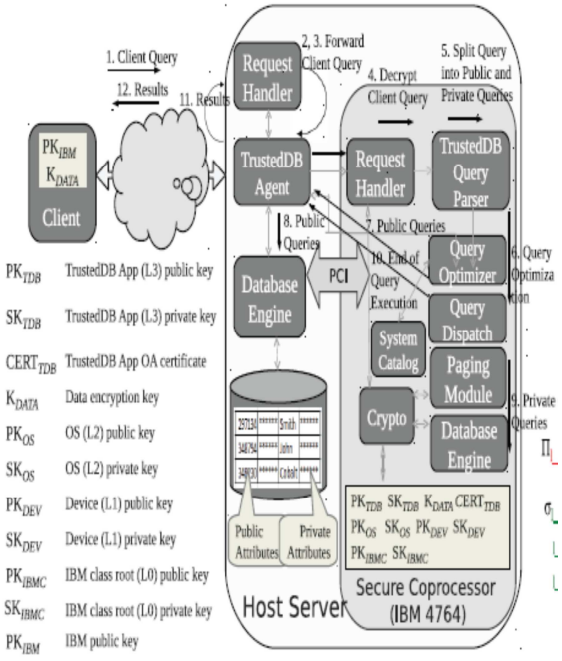

Fig.3.1 Proposed architecture

IV. EXPERIMENTAL RESULTS AND ANALYSIS

This section presents the experimental analysis of our proposed work via .NET framework. The proposed model has been tested on patient data of Healthcare systems of the data.

Fig.4.1 Patient's registering with TrustedDB systems

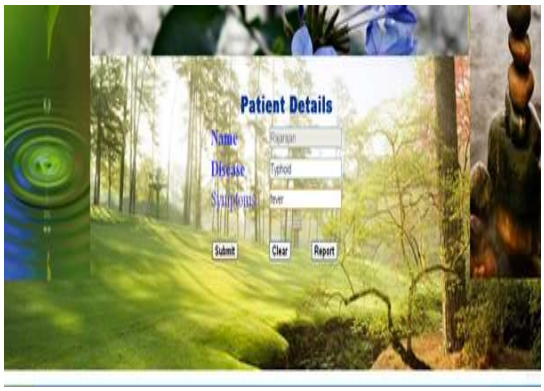

Fig.4.2 Receiving the patient details for creating database schema

Fig.4.3 Doctor's viewing the patient details from their login

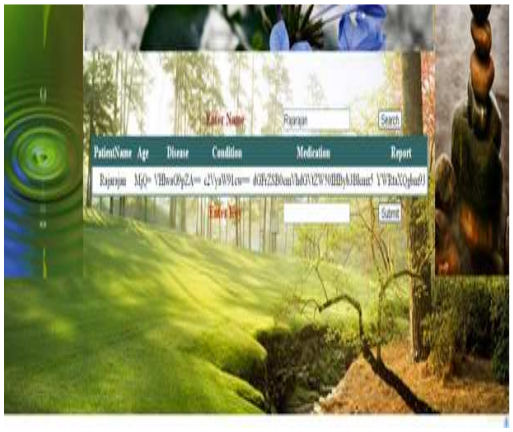

Fig.4.4 Performing searching operations

Fig. 4.5 Secured data modeling before outsourcing

V. CONCLUSION

The goal of the TrustedDB system is to help developers to protect their confidential data against curious attackers (e.g., database administrators) and achieve high performance at the same time. Just as in any other database system, the physical database design is crucial to achieve high performance. This paper discussed the most critical physical design decisions that are specific to a system like TrustedDB. Concretely, this paper discussed the performance implications of static data security (choice of encryption scheme), runtime data security (kind and amount of work that needs to be carried out by trusted hardware), and other tuning considerations such as clustering rows and columns. We are currently prototyping TrustedDB and in the process of evaluating and quantifying the trade-offs in using the different physical design options outlined in this paper. We believe that TrustedDB is particularly suited to be used as the infrastructure for a secure database-as-a-service where the goal is to achieve the confidentiality that is needed at the lowest possible cost.

REFERENCES

[1] Sumeet Bajaj and Radu Sion, “TrustedDB: A Trusted Hardware-Based Database with Privacy and Data Confidentiality”, IEEE Transactions on Knowledge and Data Engineering, 26(3), 2014.

[2] A. Arasu et al. Orthogonal security with cipherbase. In CIDR, 2013.

[3] S. Bajaj and R. Sion. TrustedDB: a trusted hardware based database with privacy and data confidentiality. In SIGMOD, 2011.

[4] A. Boldyreva, N. Chenette, Y. Lee, and A. O’Neill. Order-preserving symmetric encryption. In EUROCRYPT ’09, 2009. [5] K. Eguro and R. Venkatesan. FPGAs for trusted cloud computing. In FPL, 2012.

[6] C. Gentry. Computing arbitrary functions of encrypted data. Commun. ACM, 53(3), 2010.

[7] H. Hacigumus, S. Mehrotra, and B. R. Iyer. Providing database as a service. In ICDE, pages 29–38, 2002.

[8] S. Hildenbrand, D. Kossmann, T. Sanamrad, C. Binning, F. Faerber, and J. Woehler. Query processing on encrypted data in the cloud. In Technical Report No. 735, ETH Zurich, 2011.

[9] Microsoft Corporation. SQL Azure. http://www.windowsazure.com/en-us/home/features/sql-azure/.

10] R. A. Popa, C. M. S. Redfield, N. Zeldovich, et al. Cryptdb: protecting confidentiality with encrypted query processing. In SOSP, pages 85–100, 2011.