International Journal of Research
ISSN: 2348-6848 Vol-3, Special Issue-1
National Conference on Advanced Computing Technologies
Held on 21st July 2015 at Hyderabad city organized by **Global Research Academy,** Hyderabad, Telangana, India.

Design & Development of a Trusted Database Prototype to Execute SQL Queries with Privacy and Compliance

P.Vandana[1]; K.Anusha[2]& Prof.Dr.G.Manoj Someswar[3]

[1]M.Tech.(CSE) from Sai Sudhir Institute of Engineering & Technology, Affiliated to JNTUH, Hyderabad, Telangana, India

[2]M.Tech. (CSE), Assistant Professor, Department of CSE, Sai Sudhir Institute of Engineering & Technology, Affiliated to JNTUH, Hyderabad, Telangana, India

[3]B.Tech., M.S.(USA), M.C.A., Ph.D., Principal & Professor, Department Of CSE, Anwar-ul-uloom College of Engineering & Technology, Affiliated to JNTUH, Vikarabad, Telangana, India

ABSTRACT:

Traditionally, as soon as confidentiality becomes a concern, data is encrypted before outsourcing to a service provider. Any software-based cryptographic constructs then deployed, for server-side query processing on the encrypted data, inherently limit query expressiveness. Here, we introduce Trusted DB, an outsourced database prototype that allows clients to execute SQL queries with privacy and under regulatory compliance constraints by leveraging server-hosted, tamper-proof trusted hardware in critical query processing stages, thereby removing any limitations on the type of supported queries. 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 DB is built and runs on actual hardware and its performance and costs are evaluated here.

KEY WORDS: Classification and Regression Trees (CART); Chi Square Automatic Interaction Detection (CHAID); Content Addressable Memory(CAM); Conscious Space Saving with Stream Summary (CSSSS); Genetic Algorithms; Artificial Neural Networks; Rule Induction

INTRODUCTION

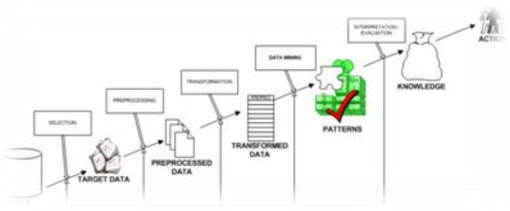

Figure 1 : Structure of Data Mining

Generally, data mining (sometimes called data or knowledge discovery) is the process of analyzing data from different perspectives and summarizing it into useful information - information that can be used to increase revenue, cuts costs, or both. Data mining software is one of a number of analytical tools for analyzing data. It allows users to analyze data from many different dimensions or angles, categorize it, and summarize the relationships identified. Technically, data mining is the process of finding correlations or patterns among dozens of fields in large relational databases.[1]

While large-scale information technology has been evolving separate transaction and analytical systems, data mining provides the link between the two. Data mining software analyzes relationships and patterns in stored transaction data based on open-ended user queries. Several types of analytical software are available: statistical, machine learning, and neural networks. **Generally, any of four types of relationships are sought:[2]**

- **Classes**: Stored data is used to locate data in predetermined groups. For example, a restaurant chain could mine customer purchase data to determine when customers visit and what they typically order. This information could be used to increase traffic by having daily specials.

- **Clusters**: Data items are grouped according to logical relationships or consumer preferences. For example, data can be mined to identify market segments or consumer affinities.
- **Associations**: Data can be mined to identify associations. The beer-diaper example is an example of associative mining.
- **Sequential patterns**: Data is mined to anticipate behavior patterns and trends. For example, an outdoor equipment retailer could predict the likelihood of a backpack being purchased based on a consumer's purchase of sleeping bags and hiking shoes.[3]

Data mining consists of five major elements:

1) Extract, transform, and load transaction data onto the data warehouse system.
2) Store and manage the data in a multidimensional database system.
3) Provide data access to business analysts and information technology professionals.
4) Analyze the data by application software.
5) Present the data in a useful format, such as a graph or table.[4]

Different levels of analysis are available:

- **Artificial neural networks**: Non-linear predictive models that learn through training and resemble biological neural networks in structure.
- **Genetic algorithms**: Optimization techniques that use process such as genetic combination, mutation, and natural selection in a design based on the concepts of natural evolution.
- **Decision trees**: Tree-shaped structures that represent sets of decisions. These decisions generate rules for the classification of a dataset. Specific decision tree methods include Classification and Regression Trees (CART) and Chi Square Automatic Interaction Detection (CHAID). CART and CHAID are decision tree techniques used for classification of a dataset. They provide a set of rules that you can apply to a new (unclassified) dataset to predict which records will have a given outcome. CART segments a dataset by creating 2-way splits while CHAID segments using chi square tests to create multi-way splits. CART typically requires less data preparation than CHAID.[5]
- **Nearest neighbor method**: A technique that classifies each record in a dataset based on a combination of the classes of the *k* record(s) most similar to it in a historical dataset (where *k*=1). Sometimes called the *k*-nearest neighbor technique.
- **Rule induction**: The extraction of useful if-then rules from data based on statistical significance.
- **Data visualization**: The visual interpretation of complex relationships in multidimensional data. Graphics tools are used to illustrate data relationships.

Characteristics of Data Mining:

- **Large quantities of data**: The volume of data so great it has to be analyzed by automated techniques e.g. satellite information, credit card transactions etc.
- **Noisy, incomplete data**: Imprecise data is the characteristic of all data collection.
- **Complex data structure**: conventional statistical analysis not possible
- **Heterogeneous data stored in legacy systems[6]**

Benefits of Data Mining:

1) It's one of the most effective services that are available today. With the help of data mining, one can discover precious information about the customers and their behavior for a specific

set of products and evaluate and analyze, store, mine and load data related to them

2) An analytical CRM model and strategic business related decisions can be made with the help of data mining as it helps in providing a complete synopsis of customers
3) An endless number of organizations have installed data mining projects and it has helped them see their own companies make an unprecedented improvement in their marketing strategies (Campaigns)
4) Data mining is generally used by organizations with a solid customer focus. For its flexible nature as far as applicability is concerned is being used vehemently in applications to foresee crucial data including industry analysis and consumer buying behaviors
5) Fast paced and prompt access to data along with economic processing techniques have made data mining one of the most suitable services that a company seek[7].

Advantages of Data Mining:

1. **Marketing / Retail:**

Data mining helps marketing companies build models based on historical data to predict who will respond to the new marketing campaigns such as direct mail, online marketing campaign…etc. Through the results, marketers will have appropriate approach to sell profitable products to targeted customers.

Data mining brings a lot of benefits to retail companies in the same way as marketing. Through market basket analysis, a store can have an appropriate production arrangement in a way that customers can buy frequent buying products together with pleasant. In addition, it also helps the retail companies offer certain discounts for particular products that will attract more customers.

2. **Finance / Banking**

Data mining gives financial institutions information about loan information and credit reporting. By building a model from historical customer's data, the bank and financial institution can determine good and bad loans. In addition, data mining helps banks detect fraudulent credit card transactions to protect credit card's owner.[8]

3. **Manufacturing**

By applying data mining in operational engineering data, manufacturers can detect faulty equipments and determine optimal control parameters. For example semi-conductor manufacturers has a challenge that even the conditions of manufacturing environments at different wafer production plants are similar, the quality of wafer are lot the same and some for unknown reasons even has defects. Data mining has been applying to determine the ranges of control parameters that lead to the production of golden wafer. Then those optimal control parameters are used to manufacture wafers with desired quality.

4. **Governments**

Data mining helps government agency by digging and analyzing records of financial transaction to build patterns that can detect money laundering or criminal activities.

5. **Law enforcement:**

Data mining can aid law enforcers in identifying criminal suspects as well as apprehending these criminals by examining trends in location, crime type, habit, and other patterns of behaviors.

6. **Researchers:**

Data mining can assist researchers by speeding up their data analyzing process; thus, allowing those more time to work on other projects. [9]

LITERATURE SURVEY

Recent trends towards database outsourcing, as well as concerns and laws governing data privacy, have led to great interest in enabling secure database services. Previous approaches to enabling such a service have been based on data encryption, causing a large overhead in query processing. We propose a new, distributed architecture that allows an organization to outsource its data management to two un trusted servers while preserving data privacy. We show how the presence of two servers enables efficient partitioning of data so that the contents at any one server are guaranteed not to breach data privacy. We show how to optimize and execute queries in this architecture, and discuss new challenges that emerge in designing the database schema.

HMAC was proved by Bellare, Canetti and Krawczyk (1996) to be a PRF assuming that (1) the underlying compression function is a PRF, and (2) the iterated hash function is weakly collision-resistant.[10] However, recent attacks show that assumption (2) is false for MD5 and SHA-1, removing the proof-based support for HMAC in these cases. This paper proves that HMAC is a PRF under the sole assumption that the compression function is a PRF. This recovers a proof based guarantee since no known attacks compromise the pseudo randomness of the compression function, and it also helps explain the resistance-to-attack that HMAC has shown even when implemented with hash functions whose (weak) collision resistance is compromised. We also show that an even weaker-than-PRF condition on the compression function, namely that it is a privacy-preserving MAC, suffices to establish HMAC is a secure MAC as long as the hash function meets the very weak requirement of being computationally almost universal, where again the value lies in the fact that known attacks do not invalidate the assumptions made.[11]

Frequent elements and top-k queries constitute an important class of queries for data stream analysis applications. Certain applications require answers for both frequent elements and top-k queries on the same stream. In addition, the ever increasing data rates call for providing[12] fast answers to the queries, and researchers have been looking towards exploiting specialized hardware for this purpose. Content Addressable Memory(CAM) provides an efficient way of looking up elements and hence are well suited for the class of algorithms that involve lookups. In this paper, we present a fast and efficient CAM conscious integrated solution for answering both frequent elements and top-k queries on the same stream. We call our scheme CAM conscious Space Saving with Stream Summary (CSSwSS), and it can efficiently answer continuous queries. We provide an implementation of the proposed scheme using commodity CAM chips, and the experimental evaluation demonstrates that not only does the proposed scheme outperforms existing CAM conscious techniques by an order of magnitude at query loads of about 10%, but the proposed scheme can also efficiently answer continuous queries.[13]

The impact of privacy requirements in the development of modern applications is increasing very quickly. Many commercial and legal regulations are driving the need to develop reliable solutions for protecting sensitive information whenever it is stored, processed, or communicated to external parties. To this purpose, encryption techniques are currently used in many scenarios where data protection is required since they provide a layer of protection against the disclosure of personal information, which safeguards companies from the costs that may arise from exposing their data to privacy breaches. However, dealing with encrypted data may make query processing more expensive. In this paper, we address these issues by proposing a solution to enforce privacy of data collections that combines data fragmentation with encryption. We model privacy requirements as confidentiality constraints expressing the sensitivity of attributes and their associations. We then use encryption as an underlying (conveniently available) measure for making data unintelligible, while exploiting fragmentation as a way to break sensitive associations among attributes. We formalize the problem of minimizing the impact of fragmentation in terms of number of fragments and their affinity and present two heuristic algorithms for solving such problems.[14]

We survey the recent wave of extensions to the popular map-reduce systems, including those that have begun to address the implementation of recursive queries using the same computing environment as map-reduce. A central problem is that recursive tasks cannot deliver their output only at the end, which makes recovery from failures much more complicated than in map-reduce and its non recursive extensions. We propose several algorithmic ideas for efficient implementation of recursions in the map-reduce environment and discuss several alternatives for supporting recovery from failures without restarting the entire job.[15]

SYSTEM STUDY

FEASIBILITY STUDY

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

Three key considerations involved in the feasibility analysis are

- ECONOMICAL FEASIBILITY
- TECHNICAL FEASIBILITY
- SOCIAL FEASIBILITY

ECONOMICAL FEASIBILITY

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

TECHNICAL FEASIBILITY

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

SOCIAL FEASIBILITY

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

SYSTEM DESIGN

SYSTEM ARCHITECTURE:

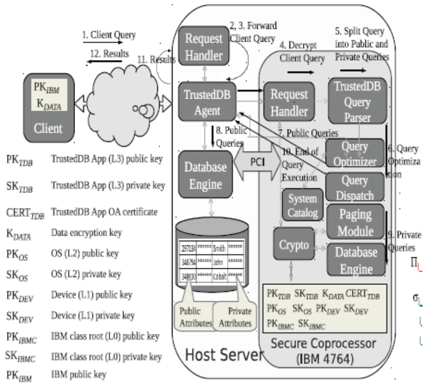
1. Client Query
12. Results
11. Results
2, 3. Forward Client Query
4. Decrypt Client Query
5. Split Query into Public and Private Queries
PK_IBM
K_DATA
Client
Request Handler
TrustedDB Agent
Request Handler
TrustedDB Query Parser
8. Public Queries
7. Public Queries
Query Optimizer
6. Query Optimiza tion
10. End of Query Execution
Database Engine
PCI
Query Dispatch
System Catalog
Paging Module
Crypto
Database Engine
9. Private Queries
Public Attributes
Private Attributes
PK_TDB SK_TDB K_DATA CERT_TDB PK_OS SK_OS PK_DEV SK_DEV PK_IBMC SK_IBMC
Host Server
Secure Coprocessor (IBM 4764)
PK_TDB TrustedDB App (L3) public key
SK_TDB TrustedDB App (L3) private key
CERT_TDB TrustedDB App OA certificate
K_DATA Data encryption key
PK_OS OS (L2) public key
SK_OS OS (L2) private key
PK_DEV Device (L1) public key
SK_DEV Device (L1) private key
PK_IBMC IBM class root (L0) public key
SK_IBMC IBM class root (L0) private key
PK_IBM IBM public key

Figure 2 : System Architecture

DATA FLOW DIAGRAM:

1. The DFD is also called as bubble chart. It is a simple graphical formalism that can be used to represent a system in terms of input data to the system, various processing carried out on this data, and the output data is generated by this system.
2. The data flow diagram (DFD) is one of the most important modeling tools. It is used to model the system components. These components are the system process, the data used by the process, an external entity that interacts with the system and the information flows in the system.
3. DFD shows how the information moves through the system and how it is modified by a series of transformations. It is a graphical technique that depicts information flow and the

transformations that are applied as data moves from input to output.

4. DFD is also known as bubble chart. A DFD may be used to represent a system at any level of abstraction. DFD may be partitioned into levels that represent increasing information flow and functional detail.

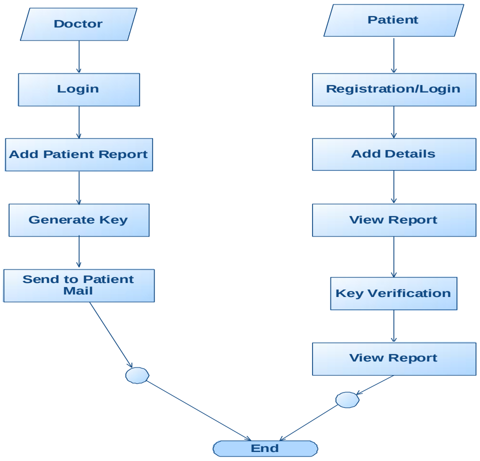

Figure 3 : Data Flow Diagram

UML DIAGRAMS

UML stands for Unified Modeling Language. UML is a standardized general-purpose modeling language in the field of object-oriented software engineering. The standard is managed, and was created by, the Object Management Group.

The goal is for UML to become a common language for creating models of object oriented computer software. In its current form UML is comprised of two major components: a Meta-model and a notation. In the future, some form of method or process may also be added to; or associated with, UML.

The Unified Modeling Language is a standard language for specifying, Visualization, Constructing and documenting the artifacts of software system, as well as for business modeling and other non-software systems.

The UML represents a collection of best engineering practices that have proven successful in the modeling of large and complex systems.

The UML is a very important part of developing objects oriented software and the software development process. The UML uses mostly graphical notations to express the design of software projects.

GOALS:

The Primary goals in the design of the UML are as follows:

1. Provide users a ready-to-use, expressive visual modeling Language so that they can develop and exchange meaningful models.
2. Provide extendibility and specialization mechanisms to extend the core concepts.
3. Be independent of particular programming languages and development process.
4. Provide a formal basis for understanding the modeling language.
5. Encourage the growth of OO tools market.
6. Support higher level development concepts such as collaborations, frameworks, patterns and components.
7. Integrate best practices.

USE CASE DIAGRAM:

A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.

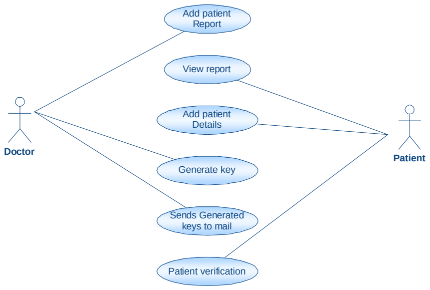

Figure 4 : Use Case Diagram

CLASS DIAGRAM:

In software engineering, a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes. It explains which class contains information.

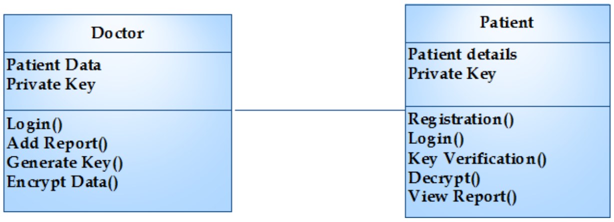

Figure 5 : class Diagram

SEQUENCE DIAGRAM:

A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.

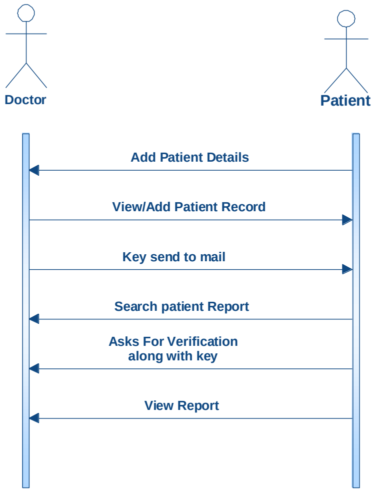

Figure 6 : Sequence Diagram

ACTIVITY DIAGRAM:

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.

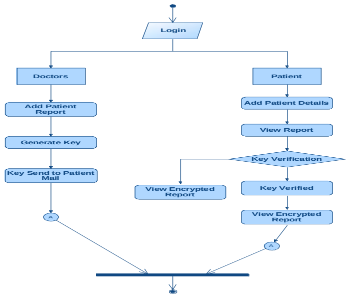

Figure 7 : Activity Diagram

INPUT DESIGN

The input design is the link between the information system and the user. It comprises the developing specification and procedures for data preparation and those steps are necessary to put transaction data in to a usable form for processing can be achieved by inspecting the computer to read data from a written or printed document or it can occur by having people keying the data directly into the system. The design of input focuses on controlling the amount of input required, controlling the errors, avoiding delay, avoiding extra steps and keeping the process simple. The input is designed in such a way so that it provides security and ease of use with retaining the privacy. Input Design considered the following things:

- ➢ What data should be given as input?
- ➢ How the data should be arranged or coded?
- ➢ The dialog to guide the operating personnel in providing input.
- ➢ Methods for preparing input validations and steps to follow when error occur.

OBJECTIVES

1.Input Design is the process of converting a user-oriented description of the input into a computer-based system. This design is important to avoid errors in the data input process and show the correct direction to the management for getting correct information from the computerized system.

2. It is achieved by creating user-friendly screens for the data entry to handle large volume of data. The goal

of designing input is to make data entry easier and to be free from errors. The data entry screen is designed in such a way that all the data manipulates can be performed. It also provides record viewing facilities.

3.When the data is entered it will check for its validity. Data can be entered with the help of screens. Appropriate messages are provided as when needed so that the user will not be in maize of instant. Thus the objective of input design is to create an input layout that is easy to follow

OUTPUT DESIGN

A quality output is one, which meets the requirements of the end user and presents the information clearly. In any system results of processing are communicated to the users and to other system through outputs. In output design it is determined how the information is to be displaced for immediate need and also the hard copy output. It is the most important and direct source information to the user. Efficient and intelligent output design improves the system's relationship to help user decision-making.

1. Designing computer output should proceed in an organized, well thought out manner; the right output must be developed while ensuring that each output element is designed so that people will find the system can use easily and effectively. When analysis design computer output, they should Identify the specific output that is needed to meet the requirements.

2.Select methods for presenting information.

3.Create document, report, or other formats that contain information produced by the system.

The output form of an information system should accomplish one or more of the following objectives.

- ❖ Convey information about past activities, current status or projections of the
- ❖ Future.
- ❖ Signal important events, opportunities, problems, or warnings.
- ❖ Trigger an action.
- ❖ Confirm an action.

SYSTEM ANALYSIS

EXISTING SYSTEM:

Existing research addresses several such security aspects, including access privacy and searches on encrypted data. In most of these efforts data is 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. Recent theoretical cryptography results provide hope by proving the existence of universal homeomorphisms, i.e., encryption mechanisms that allow computation of arbitrary functions without decrypting the inputs. Unfortunately actual instances of such mechanisms seem to be decades away from being practical

DISADVANTAGES OF EXISTING SYSTEM:

Trusted hardware is generally impractical due to its performance limitations and higher acquisition costs. As a result, with very few exceptions, these efforts have stopped short of proposing or building full - fledged database processing engines.

Computation inside secure processors is orders of magnitude cheaper than any equivalent cryptographic operation performed on the provider's unsecured server hardware, despite the overall greater acquisition cost of secure hardware.

PROPOSED SYSTEM:

We propose that a full-fledged, 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 processing on common server hardware. We validate this by designing and building Trusted DB, a SQL database processing engine that makes use of tamperproof cryptographic coprocessors such as the IBM 4764 in close proximity to the outsourced data. 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. Trusted DB achieves this by utilizing common unsecured server resources to the maximum extent possible. E.g., Trusted DB enables the SCPU to transparently access external storage while preserving data confidentiality with on-the-fly encryption. This eliminates the limitations on the size of databases that can be supported. Moreover, client queries are pre-processed to identify sensitive components to be run inside the SCPU. Non-sensitive operations are off-loaded to the un trusted host server. This greatly improves performance and reduces the cost of transactions.

ADVANTAGES OF PROPOSED SYSTEM:

(i) The introduction of new cost models and insights that explain and quantify the advantages of deploying trusted hardware for data processing,

(ii) the design, development, and evaluation of Trusted DB, a trusted hardware based relational database with full data confidentiality, and

(iii) Detailed query optimization techniques in a trusted hardware-based query execution model.

SYSTEM TESTING

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the

Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

TYPES OF TESTS

Unit testing

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

Integration testing

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

Functional test

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted.

Invalid Input : identified classes of invalid input must be rejected.

Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised.

Systems/Procedures: interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

System Test

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

White Box Testing

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

Black Box Testing

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box .you cannot "see" into it. The test provides inputs and responds to outputs without considering how the software works.

Unit Testing:

Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases.

Test strategy and approach

Field testing will be performed manually and functional tests will be written in detail.

Test objectives

- All field entries must work properly.
- Pages must be activated from the identified link.
- The entry screen, messages and responses must not be delayed.

Features to be tested

- Verify that the entries are of the correct format
- No duplicate entries should be allowed
- All links should take the user to the correct page.

Integration Testing

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.

The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error.

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

Acceptance Testing

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

IMPLEMENTATION

MODULES:

1. Query Parsing and Execution
2. Query optimization process
3. System Catalog
4. Analysis of Basic Query Operations

MODULES DESCRIPTION:

Query Parsing and Execution

In the first stage a client defines a database schema and partially populates it. Sensitive attributes are marked using the SENSITIVE keyword which the client layer transparently 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 transparently encrypted at the client site using the public key of the SCPU. The host server thus cannot decrypt the query. (2) The host server forwards the encrypted query to the Request Handler inside the SCPU. (3) The Request Handler decrypts the query and forwards it to the Query Parser. 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. (4)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.(5) 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. (6) The final query result is assembled, encrypted, digitally signed by the SCPU Query Dispatcher, and sent to the client.

Query optimization process:

At a high level query optimization in a database system 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 Interpretor for execution.

The query optimization process in Trusted DB works similarly with key differences in the Query Cost Estimator due to the logical partitioning of data mentioned above.

System Catalog:

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. E.g., the availability of an index or the knowledge of possible distinct values of an attribute. These sets of information are collected and stored in the System Catalog. Most available DBMS today have some form of periodically updated System Catalog.

Analysis of Basic Query Operations:

The cost of a plan is the aggregate of the cost of the steps that comprise it. In this section we present how execution times for a certain set of basic query plan steps are estimated.

RESULTS AND CONCLUSION

This paper's contributions are threefold: (i) the introduction of new cost models and insights that explain and quantify the advantages of deploying trusted hardware for data processing, (ii) the design and development of Trusted DB, a trusted hardware based relational database with full data confidentiality and no limitations on query expressiveness, and (iii) detailed query optimization techniques in a trusted hardware-based query execution model.

This work's inherent thesis is that, at scale, in outsourced contexts, computation inside secure hardware processors is orders of magnitude cheaper than equivalent cryptography performed on provider's unsecured server hardware, despite the overall greater acquisition cost of secure hardware. We thus propose to make trusted hardware a first-class citizen in the secure data management arena. Moreover, we hope that cost-centric insights and architectural paradigms will fundamentally change the way systems and algorithms are designed.

REFERENCES

[1] FIPS PUB 140-2, Security Requirements for Cryptographic Modules. Online at http://csrc.nist.gov/groups/STM/cmvp/standards.html#02.

[2] TPC-H Benchmark. Online at http://www.tpc.org/tpch/.

[3] IBM 4764 PCI-X Cryptographic Coprocessor. Online at http://www-03.ibm.com/security/cryptocards/pcixcc/overview.shtml,2007.

[4] 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.

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

[6] Mihir Bellare. New proofs for nmac and hmac: Security without collision-resistance. pages 602–619. Springer-Verlag, 2006.

[7] Bishwaranjan Bhattacharjee, Naoki Abe, Kenneth Goldman, Bianca Zadrozny, Chid Apte, Vamsavardhana R. Chillakuru and Marysabel del Carpio. Using secure coprocessors for privacy preserving collaborative data mining and analysis. In *Proceedings of DaMoN*, 2006.

[8] 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.

[9] Yao Chen and Radu Sion. To cloud or not to cloud?: musings on costs and viability. In *Proceedings of SOCC*, pages 29:1–29:7. ACM, 2011.

[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] Tom Denis. *Cryptography for Developers*. Syngress.

[12] Damiani E., Vimercati C., Jajodia S., Paraboschi S., and Samarati P. Balancing confidentiality and efficiency in untrusted relational dbmss. In *Proceedings of ACM CCS*, 2003.

[13] Einar Mykletun and Gene Tsudik. Aggregation Queries in the Database-As-a-Service Model. *Data and Applications Security*, 4127, 2006.

[14] Foto N. Afrati and Vinayak Borkar and Michael Carey and Neoklis Polyzotis and Jeffrey D. Ullman. Map-reduce extensions and recursive queries. In *Proceedings of EDBT*, pages 1–8. ACM, 2011.

[15] 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.