Privacy Preserving Nested Sub Queries over Relational Databases

K.R.G.Krishna Murthy[1], D.HariKrishna[2], Dr. K. Rama Krishnaiah[3,] Govathoti Sudeepthi[4]

[1] Dept. of CSE, Nova College of Engineerng & Technology,Vijayawada,AP,India.

[2]Assistant Professor, Nova College of Engineerng & Technology,Vijayawada,AP,India.

[3]Professor & Principal, NVR College of Engineering and Technology, Tenali,AP,India.

[4]Anurag Group of Institutions, Ghatkesar, Hyderabad

Abstract

Data outsourcing is the main important task in present days but one of the key strategies is security for out sourcing data. Although the benefits 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 datasets. We propose a Trust DB for accessing efficient outsourcing results in common environment manner. The introduction of new cost models and insights that explain and quantify the advantages of deploying trusted hardware for data processing. The TrustedDB query parser does not yet support parsing of multi-level nested sub-queries and user defined views. We propose a new efficient approach, the nested relational approach, based on the nested relational algebra. Our approach directly unrests non-aggregate sub queries using hash joins, and treats all sub queries in a uniform manner, being able to deal with nested queries of any type and any level.

Key terms: Trusted DB, encryption and decryption, key management, nested queries.

2. INTRODUCTION

Although the benefits 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 datasets. 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, health care and government frameworks. Moreover, today's privacy guarantees for such services are at best declarative and subject customers to unreasonable fine print clauses. E.g., allowing the server operator to use customer behavior and content for commercial profiling or governmental surveillance purposes. As a result, we posit 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.

3. PREVIOUS WORK

Queries on Encrypted Data:

Hacigumus et al. [6] propose division of information into mystery parcels and revamping of extent questions over the first information as far as the ensuing allotment identifiers. This adjusts an exchange off in the middle of customer and server-side transforming, as a capacity of the information section size. In [7] the creators investigate ideal basin sizes for reach questions [1]. proposes utilizing tuple-level encryption and lists on the encoded tuples to help equity predicates. The fundamental commitment here is the investigation of trait presentation brought on by inquiry preparing prompting two bits of knowledge.

Request Preserving encryption for questioning scrambled xml databases. Moreover, a strategy alluded to as part and scaling is utilized to vary the recurrence dissemination of encoded information from that of the plain-message information. Vertical dividing of relations among different un-trusted servers is utilized in [3]. Here, the protection objective is to avoid access of a subset of qualities by any single server.

Ge et al. [4] propose an encryption conspire in a trusted-server model to guarantee security of information dwelling on circle. The FCE plan outlined here is proportionately secure as a square figure, be that as it may, with expanded proficiency. [10], in the same way as [4] just guarantee security of information dwelling on circle. Collection inquiries over social databases is given in [5] by making utilization of homo morphic encryption focused around Privacy Homomorphism [11]. The creators in [2] have proposed that this plan is helpless against a figure message just assault. Rather [2] proposes an option plan to perform collection questions focused around bucketization [6]. Here the information holder precomputes total values, for example, SUM and COUNT for segments and stores them scrambled at the server.

In catch up work, Ghostdb [9] proposes to implant a database inside a USB key outfitted with a CPU. Both [8] and [9] are liable to the stockpiling limits of trusted equipment which thusly restrains the measure of the database and the questions that can handled.

4. EXISTING SYSTEM

Data outsourcing is the principle critical errand in present days however one of the key technique is security for out sourcing information. In spite of the fact that the profits of outsourcing and mists are well known, noteworthy difficulties yet lie in the way of substantial scale reception since such administrations frequently require their clients to naturally believe the supplier with full get to the outsourced datasets. For above con templations in information out sourcing generally created a portion of the security that incorporates access protection and ventures on scrambled information. In the vast majority of these deliberations information is encoded before outsourcing. Once encoded notwithstanding, intrinsic limits in the sorts of primitive operations that can be performed on scrambled information lead to major expressiveness and common sense obligations. For getting to upgraded encryption approaches suitable methodology was presupposed amid this prerequisite.

PROBLEM STATEMENT

The presentation of new cost models and bits of knowledge that clarify and evaluate the focal points of sending trusted fittings for information preparing. The TrustedDB inquiry parser does not yet help parsing of multi-level settled sub-questions and client characterized perspectives. We propose another proficient methodology, the settled social methodology, in view of the settled social variable based math. Our methodology straightforwardly unrests non-total sub questions utilizing hash joins, and treats all sub inquiries in an uniform way, having the capacity to manage settled inquiries of any sort and any level.

PROPOSED SYSTEM

Encryption systems that permit reckoning of self-assertive capacities without unscrambling the inputs. Sadly real occurrences of such components appear to be decades from being handy. We propose a Trust DB for getting to effective outsourcing brings about basic environment way. The presentation of new cost models and bits of knowledge that clarify and evaluate the favorable circumstances of conveying trusted fittings for information processing.a trusted equipment based social database with full information classified ness and no limits on inquiry expressiveness. This proposed system gives after preferences:

The Trusted DB inquiry parser does not yet help parsing of multi-level settled sub-queries and user defined views.

We propose another proficient methodology, the settled social methodology, in view of the settled social polynomial math. Our methodology specifically unrests non-total sub inquiries utilizing hash joins, and treats all sub questions in an uniform way, having the capacity to manage settled inquiries of any sort and any level. We cover test work that affirms that current methodologies experience issues managing non-total sub- questions, and that our methodology offers better execution. We likewise talk about a few conceivable outcomes for arithmetical enhancement and the issue of coordinating our methodology in a social database framework. Our settled social methodology is general and treats all settled inquiries with any kind of interfacing administrator and any level of settling in an uniform way. The vicinity or nonappearance of lists does not have any impact on our methodology.

ALGORITHM:

```
Algorithm Compute(node,relational-expression)
Require: : a nested query with non-aggregate subqueries
Ensure: : the result of a query
PROCEDURE compute(node, rel) {
    if (node is a leaf) then
        return;
    else
        for each n belongs to children(node) do
            Ti = name(n);
            Cij = linkc(node; n);
            Li = linkL(node; n);
            if (Cij not=empty); then
                rel = rel semi join Cij Ti or rel = rel joinCij Ti;
            else
                rel = rel * Ti;
            end if
            compute(n, rel);
            rel = V{ T1.*,…} {Ti.*} (rel);
            rel= sigma Li(rel) or sigma complement Li(rel);
        end for
    end if
}
```

The algorithm works equally for nested linear queries and nested tree queries. In the first case, there is only one child for each node; the net effect is that of going down the tree joining or outer joining.

Optimizations

Algorithm can evaluate nested queries containing non aggregate sub queries with any type of linking predicates and any level of nesting in a uniform manner.

Reduce nesting operations:

In the first approach, we process each one interfacing predicate by utilizing one settling operation took after by one joining choice. In any case, analyzing the parameters of the home administrator, it is clear that more elevated amounts settle by a prefix of the settling qualities utilized by lower levels, and utilize a piece of the postfix of those settling characteristics as the settled traits.

Pipelining: Pipelining is possible in the context of our algorithm. In particular, it seems clear that it should be possible to pipeline the linking selection with the nesting that is immediately adjacent to it.

Linear correlation: Calculation could be further upgraded for some uncommon inquiries to increase better execution. One such case is direct connection. A settled inquiry is straight connected if every inward question piece is just corresponded to its nearby external query block.

PERFORMANCE ANALYSIS

The bellow two diagrams show the existing and proposed system performance in time manner.

Fig 1: Existing system performance in seconds.

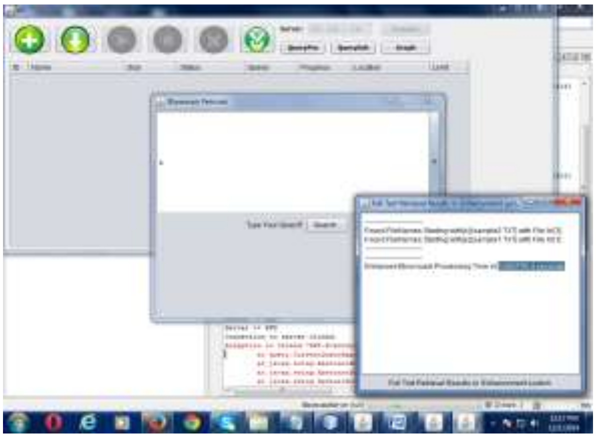


Fig 2: praposed system performance in seconds.

To verify the efficiency of the nested relational approach, three queries and their variations with four different sizes derived from the TPC-H benchmark were tested in our experiments.

Query 1:

select o_orderkey, o_order priority
from orders
where o_orderdate>=x1 and o_orderdate<x2 and
o_totalprice > all
(select l_extendedprice
from lineitem
where l_orderkey=o_orderkey and
l_commitdate<l_receiptdate and
l_shipdate<l_commitdate)

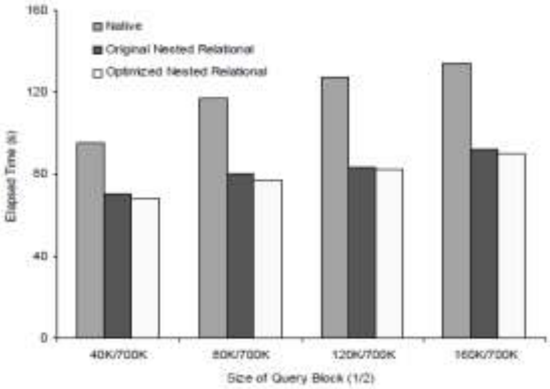


Fig 3: Query 1 performance.

The conditions o_orderdate>=x1 and o_orderdate<x2 and l_commitdate<l_receiptdate and l_shipdate<l_commitdate are used to regulate the size of each query block.

Query 2:

select p_partkey, p_name
from part
where p_size>=x1 and p_size<=x2 and
p_retailprice < [any|all]
(select ps_supplycost
from partsupp
where ps_partkey=p_partkey and

ps_availqty<y and

not exists

*(select **

from lineitem

where ps_partkey=l_partkey and

ps_suppkey=l_suppkey and

l_quantity=z))

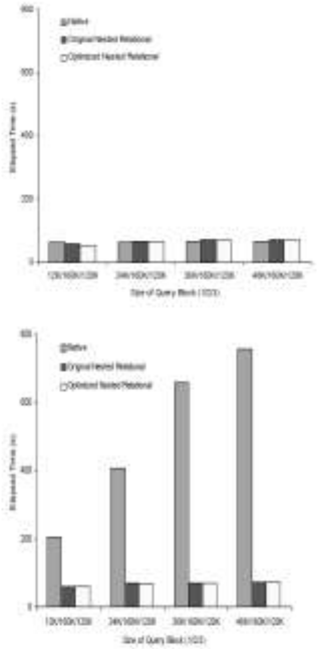


Fig 4: Query 2a(mixed: ANY/NOT EXISTS)

Query 2b(negative: ALL/NOT EXISTS)

Our First variation of Query 2 is Query 2a with the mixed ANY and NOT EXISTS operators. The native approach evaluates Query 2a from the bottom up, that is, first per- forms an anti-join of part supp and line item to form a view for the NOT EXISTS linking predicate.

Query 3:

select p_partkey, p_name

from part

where p_size>=x1 and p_size<=x2 and

p_retailprice < [all|any]

(select ps_supplycost

from partsupp

where ps_partkey=p_partkey and

ps_availqty<y and

[exists|not exists]

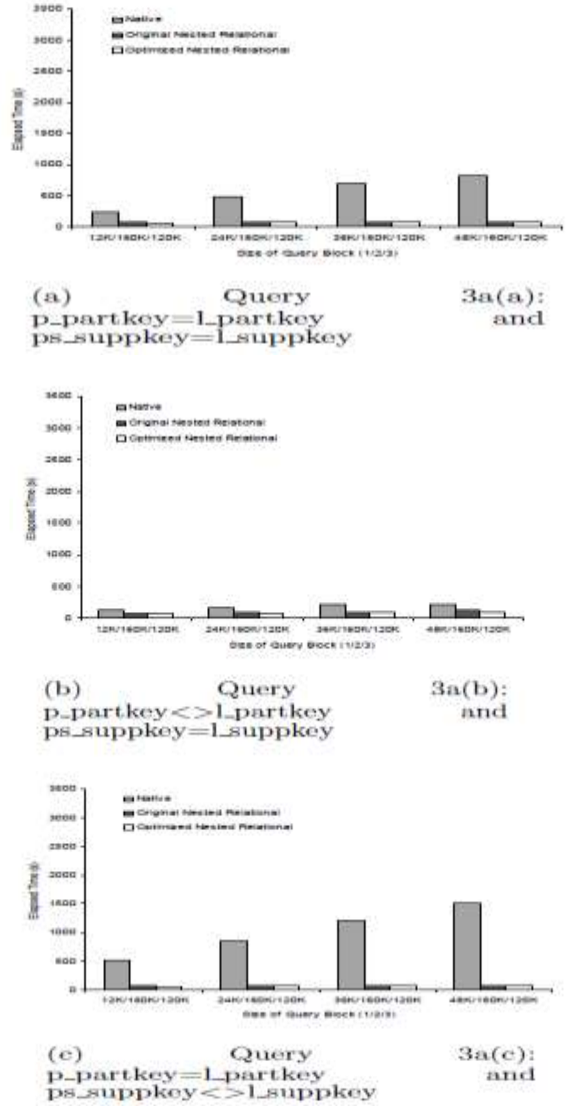
*(select **

from lineitem

where p_partkey[=|<>]l_partkey and

ps_suppkey[=|<>]l_suppkey and

l_quantity=z))



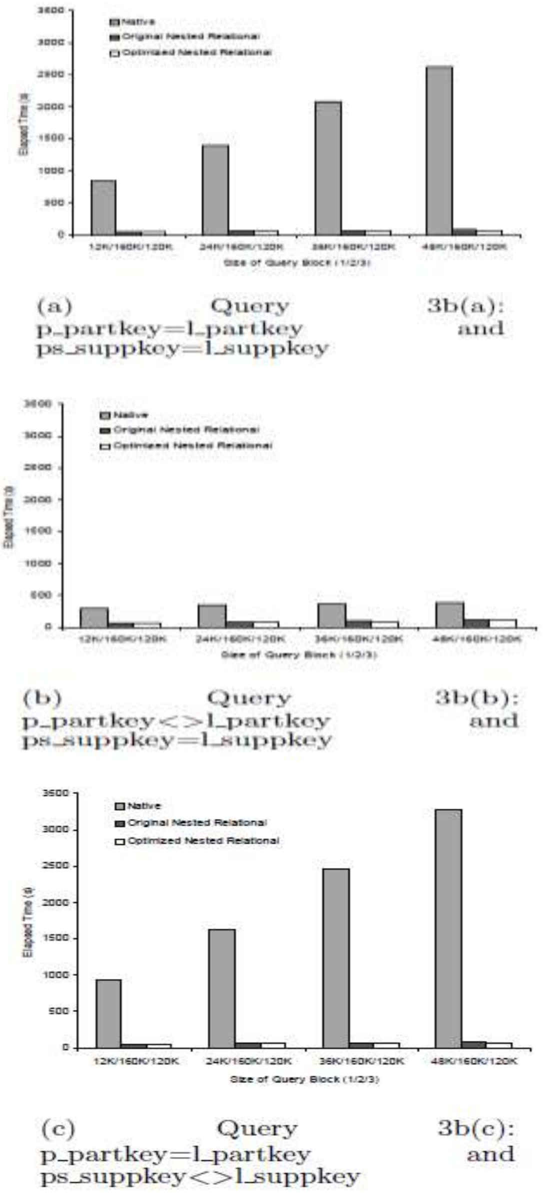
(a) Query 3a(a): p_partkey=l_partkey and ps_suppkey=l_suppkey

(b) Query 3a(b): p_partkey<>l_partkey and ps_suppkey=l_suppkey

(c) Query 3a(c): p_partkey=l_partkey and ps_suppkey<>l_suppkey

Figure 5: Query 3a(mixed: ALL/EXISTS)

The varieties of Query 3 are utilized to test blended connecting administrators, positive connecting administrators, and negative connecting administrators, with equivalent and non-equivalent connected predicates. The ¯rst variety is Query 3a with the blended connecting operators ALL and EXISTS, the second variety is Query 3b with two negative connecting administrators ALL and NOT EXISTS, furthermore the third variety is Query 3c with two positive connecting administrators ANY and EXISTS. By and large, streamlining agent creates question arrangement relying upon the connecting administrators as well as the related predicates.



(a) Query 3b(a): p_partkey=l_partkey and ps_suppkey=l_suppkey

(b) Query 3b(b): p_partkey<>l_partkey and ps_suppkey=l_suppkey

(c) Query 3b(c): p_partkey=l_partkey and ps_suppkey<>l_suppkey

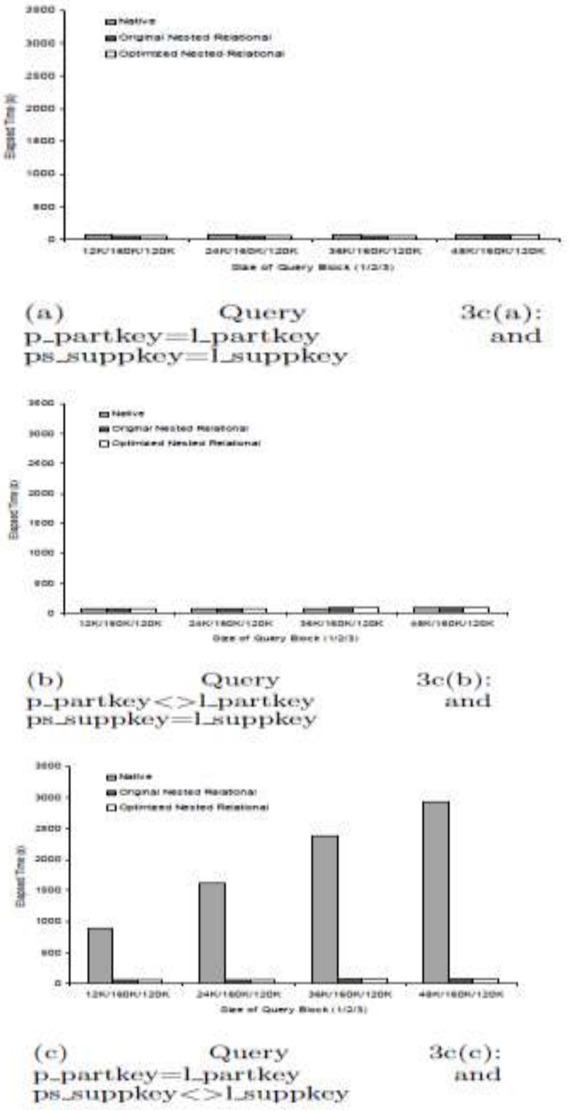
Figure 6: Query 3b(negative: ALL/NOT EXISTS)

(a) p_partkey=l_partkey and ps_suppkey=l_suppkey,

(b) p_partkey<>l_partkey and ps_suppkey=l_suppkey,

(c) p_partkey=l_partkey and ps_suppkey<>l_suppkey.

Figure 6 shows that Query 3b(a) and Query 3b(c) exhibit similar performance, but both of them perform worse than Query 3b(b). The reason is the same as that for Query 3a.



(a) Query 3c(a): p_partkey=l_partkey and ps_suppkey=l_suppkey

(b) Query 3c(b): p_partkey<>l_partkey and ps_suppkey=l_suppkey

(c) Query 3c(c): p_partkey=l_partkey and ps_suppkey<>l_suppkey

Figure 7: Query 3c(positive: ANY/EXISTS)

As needs be, the handling time of home and connecting determination is practically the same for Query 3a(a), Query 3b(a), Query 3a(c) and Query 3b(c), and somewhat more for Query 3a(b) and Query

3b(b). For Query 3c, the size of the intermediate result is much smaller due to the join operation: 1.4K, 2.9K, 4.2K and 5.6K for Query 3c(a), 113K, 228K, 341K and 452K for Query 3c(b), 4.2K, 8.8K, 12.9K and 17.3K for Query 3c(c).

As we can see from Fgure 5, Fgure 6 and Fgure 7, the nested relational approach performs almost equally among similar queries, while the native approach of System A varies significantly.

CONCLUSION:

This work's inherent thesis is that, at scale, in outsourced contexts, computation inside secure hardware processors is orders of magnitude cheaper than any equivalent cryptography performed on a provider's unsecured common server hardware, despite the overall greater acquisition cost of secure hardware. We propose a Trust DB for accessing efficient outsourcing results in common environment manner. The introduction of new cost models and insights that explain and quantify the advantages of deploying trusted hardware for data processing. The TrustedDB query parser does not yet support parsing of multi-level nested sub-queries and user defined views. We propose a new efficient approach, the nested relational approach, based on the nested relational algebra. Our approach directly unrests non-aggregate sub queries using hash joins, and treats all sub queries in a uniform manner, being able to deal with nested queries of any type and any level. We also discuss some possibilities for algebraic optimization and the issue of integrating our approach in a relational database system.

REFFERENCES:

[1] 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.

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

[3] 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.

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

[5] Bala Iyer Hakan Hacigumus and Sharad Mehrotra. Efficient execution of aggregation queries over encrypted relational databases. In *Database Systems for Advanced Applications*, volume 2973, pages 633–650, 2004.

[6] 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.

[7] B. Hore, S. Mehrotra, and G. Tsudik. A privacy-preserving index for range queries. In *Proceedings of ACM SIGMOD*, 2004.

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

[9] Nicolas Anciaux, Mehdi Benzine, Luc Bouganim, Philippe Pucheral and Dennis Shasha. GhostDB: Querying Visible and Hidden Data Without Leaks. In *Proceedings of SIGMOD*, 2007.

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

[11] Ronald Rivest, Len Adleman and Michael Dertouzos. On data banks and privacy homomorphisms. *Foundations of Secure Computation*, 1978.