PRIVACY PRESERVING FOR DATA PUBLISHING WITH SEARCH AND COMPUTE ON ENCRYPTED DATA

Megala.R[1], Revathy.S[2] , Mrs.Pushpa.S[3]
[1,2]Under Graduate Students, [3]Asst.Professor
[1,2,3]Department of CSE, New Prince Shri Bhavani College Of Engineering & Technology, Chennai-73.

***Abstract* – Privacy preserving for data publishing that indicates large number of data is formed due to number of providers are more when the data to be published and time elapsed for data searching. This task is non-trivial, because the utility measuring usually requires the aggregated raw data, which is not revealed to the data users due to privacy concerns. Furthermore, the data publishers may even cheat in the raw data, since no one including the individual providers, knows the full data set. The relevant encrypted data sets in sequence. We are interested in efficiently processing queries that require both operations to be performed on fully encrypted databases. One immediate solution is to use several special-purpose encryption schemes simultaneously; however, this approach is associated with a high computational cost for maintaining multiple encryption contexts. Another solution is to use a privacy scheme. However, no secure solutions have been developed that satisfy the efficiency requirements. We constructs a unified framework to efficiently and privately process queries. In this paper, we propose secure data publishing without having any duplication and privately process queries with search and compute operations on a database.**

***Index Terms* –** *Data publishing, Duplication, Indexing, Searching*

I.INTRODUCTION

Information security refers to the process and methodologies which are designed and implemented to protect print, electronic or any other of confidential, private and sensitive information or data from unauthorized access, use, misuse, disclosure, destruction, modification or disruption. SERVICE providers have the ability to collect large amount of user data. Data should be published securely, that can use duplication for unique publication. PRIVACY homomorphism is an important concept for encrypting clear data while allowing one to perform operation on encrypted data without decryption. After publishing the data, the data should be searched in a privacy manner with less time using indexing variables as local variables and global variables. The user data read as string and preprocessed. The string is encrypted. The random key is used cipher text which is processed to next module for duplication. The key is stored for further needs.

II. RELATED WORK

We have presented the first two-party differentially private data release algorithm for vertically partitioned data. We have shown that the proposed algorithm is differentially private and secure under the security definition of the semi honest adversary model. Moreover, we have experimentally evaluated the data utility for classification analysis. It provides similar data utility compared to the recently proposed single-party algorithm and better data utility than the distributed K-anonymity algorithm. [1]

Secure anonymous database search (SAD) system that provides exact keyword match capability. Using a new reroutable encryption and the ideas of bloom filters and deterministic encryption, SAD lets multiple parties efficiently execute exact-match queries over distributed encrypted databases in a controlled manner. The article further considers a more general search setting allowing similarity searches, going beyond existing work that considers similarity in terms of error tolerance and Hamming distance. [2]

Database outsourcing is an emerging data management paradigm which has the potential to transform the IT operations. In this paper, address privacy threats in database outsourcing scenarios where trust in the service provider is limited. Specifically privacy (e.g., estimating the value of a data element within a small error margin) and identify statistical measures of data privacy in the context of these attacks. We also investigate precise privacy guarantees of data partitioning which form the basic building blocks of our index. We then develop a model for the fundamental privacy-utility tradeoff and design a novel algorithm for achieving the desired balance between privacy and utility (accuracy of range query evaluation) of the index [3]

Plan to extend the theory to model and reason about higher level systems, such as computer system of hospitals and other distributed systems that allow interactions of the system with data providers and with data analysis , while protecting the privacy

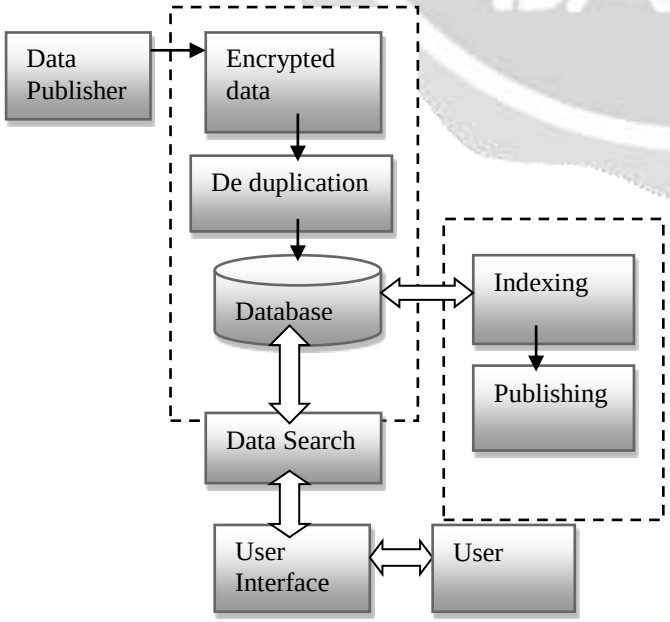
of the data stored and manipulated by the systems. For example, Allows computations over data distributed in a cloud, and combines mandatory access control with differential privacy where differential privacy is used to facilitate declassification governed by the privacy error bound set by a data provider. Our techniques can currently apply to the verification of differential privacy property of the system using a whole-system model. [4].

The collection of digital information by governments, corporations and individuals has created tremendous opportunities for knowledge and information based decision making. Driven by mutual benefits or by regulations that require certain data to be published, there is a demand for the exchange and publication of data among various parties. Data in its original form, however typically contains sensitive information about individuals and publishing such data will violate individual privacy.[5]

III. PROBLEM DESCRIPTION

This proposal can measure the data utility based upon the encrypted frequencies of the aggregated raw data instead of plain values, which thus prevents privacy breach. Moreover, it is enabled to privately check the correctness of the encrypted frequencies provided by the publisher. Construct a unified framework to efficiently and privately process queries with search and compute operations. For this purpose, the first part of our work involves devising several underlying circuits as primitives for queries on encrypted data. Second, we apply two optimization technique to improve the efficiency of these circuit primitives. The other technique is to use a larger integer as a message space rather than a binary field. Even for an integer of k-bits with addition can be performed using degree circuits with a carry operation. Finally, we present various experiments performed by varying the considered parameters, such as the query type and the number of tuples , solution is to use a privacy homomorphic scheme. However, no secure solution have been developed that satisfy the efficiency requirement. we construct a unified framework to efficiently and privately process queries with search and compute operations. privacy-preserving utility verification mechanism based upon cryptographic technique for DiffiePart a differentially private scheme designed for set-valued data. This proposal can measure the data utility based upon the encrypted frequencies of the aggregated raw data instead of the plain values, which thus prevents privacy breach. Moreover, it is enabled to privately check the correctness of the encrypted frequencies provided by the publisher, which helps detect dishonest publishers. Our optimization of circuit primitives are developed such that they minimize the circuit depth and the number of homomorphic operations. We also extend this mechanism differentially private publishing scheme designed for relational data. Our theoretical and experimental evaluations demonstrate the security and efficiency of the proposed mechanism. A lot of privacy models and corresponding anonymisation mechanism have been proposed.

IV. ARCHITECTURE DIAGRAM



V. PROPOSED SYSTEM

Privacy Preserving for data publishing that indicates large number of data is formed due to number of providers are more when the data to be published and time elapsed for data searching. In this paper, we propose secure data publishing without having any duplication and private process queries with search and compute operations which collect large amount of user data. Data should be published securely, that can use duplication for unique publication. PRIVACY homomorphism is an important concept for encrypting clear data while allowing one to perform operations on encrypted data without decryption. After publishing the data, the data should be searched in a privacy manner with the less time using indexing variables as local and global variables.

A. Data Encryption

The publisher can publish the data in a secure manner using the encryption algorithm (AES). For publishing the data, every publisher can register his/her details, when he/she log into the page. After log in of the page, publisher can post their data's. But the data's are accepted, when it is verifying the duplication. AES algorithm is symmetric encryption algorithm which supports a block of 128 bits.

B. Data Duplication Checking

The encrypted data will be stored into the database. Using this verification, publisher can publish the unique data's. Before publishing, the data should be verified using duplication checking. A publisher one who publish the data that can be verified using the primary keys. After verifying the data's, it can be published if without having any duplication.

C. Optimized Search

The data set contains a large number of parts. It takes a more time to search for users, if users need a single part of the data.. So, we use indexing technique for time reducing. User's may require a part of a data, it can compare a data sets with user's need and it can display to the users.

D. Indexing Technique

Indexing technique can be divided in to two catagories. Local variables and Global variables. Local variables are defined as unusual words which cannot be defined by the admin. Global variables are defined as usual words which is normally used by the users or an admin. Global variables are defined by admin and it is used for comparing the users need.

VI. DATA INPUT AND ENCRYPTION WITH INDEXING

The user data read as string and preprocessed. The string is encrypted. The random key is used. Cipher text is processed to next module for duplication. The key is stored for further needs that the cipher text in the database are stored in an array. Admin can publish the data, if it does not have any duplication and send the key to the publisher. The data can be decrypted before publishing. Now, we are going to create a own tags for the data for search process. The data will be spllitted in to two terms that are local variables and global variables. The global variables are predefined in the library. The global variable in the data will be removed to form tags. The user query will be spllitted in to sub strings. The substrings are analyzed with the tags produced. A matched the content will be added to result set. The array will be created to store the result set. Then these values will be forwarded to front end. The result set which is obtained by the previous module will be taken as input. The random key will be selected. Each value will be decrypted using the key and the original text will be replaced in the cipher text location.

A. Generality for Other Data Types

We next discuss the applicability and generalization of the framework to other problems, rather than DiffPart which are interactive mechanisms but relying on taxonomy trees. According to our analysis, in order to apply our framework, the target problem should satisfy two requirements. First, the utility verifier has the ability to obtain the correct encrypted version of the raw data. Second, based on the published sanitized data and the encrypted raw data, the verifier could leverage homomorphic encryption techniques to compute the utility measurement within the cipher text space. According to our survey, there are many problems satisfying these two requirements. We first consider the problem of trajectory data anonymization. Propose the first differentially private publishing algorithm for trajectory data based on a noisy prefix tree. They introduce count queries to evaluate the utility of the sanitized database. Here, the result of a count query refers to the number of trajectories passing a specific location or area.

B. Test Datasets and Queries

Our solution supports basic conjunctive and disjunctive retrieval queries with aggregate functions. Similarly, we can implement an SQL query with join conditions. Currently, because all computations are performed on cipher texts, It is difficult to support SQL queries with the order by efficient manner.

C. Experimental support

We perform comprehensive experiments to evaluate the performance of various queries expressed using our techniques from both a theoretical and a practical perspective. Compared with the preliminary version of this paper this journal version now includes the following new results.

1) Additional circuit compositions to support more query
functionalities. Consequently, we can show how to handle an SQL query with multiple conditions, such a sequentially or/and greater-than conditions.

2) Improvements in the experimental studies achieved through the use of carefully selected parameters.

Privacy preserving utility verification for DiffPart perturbs the frequencies of the records based on a context-free taxonomy tree and no items in the original data are generalized Our proposal solves the challenge to verify the utility of the published data based on the encrypted frequencies of the original data records instead of their plain values. As a result it can protect the original data from the verifying parties(i.e., the data users) because they cannot learn whether or how many times a specific record appears in the raw data set without knowing its real frequency. In addition, since the encrypted frequencies are provided by the publisher, we also present a scheme for the verifying parties to incrementally verify its correctness. Our theoretical analysis demonstrates the correctness and the security of the proposed mechanism.

VI.ALGORITHM

Advanced Encryption Standard is well known block and group by clauses in an efficient manner. Cipher algorithm for portability and reasonable security. The encryption lends very well to the hardware capabilities. AES that module takes in 128 bit blocks of data. The algorithm for generating the 10 rounds of the round key. The generated round key module perform the algorithm that generates a single round key. It's input is multiplexed between the user inputted key and the last round key. The output is stored in a register to be used as input during the next iteration of the algorithm.

A. Data Read

The SBRO(Support Single Block Read Operation Only) mode upon reception of a valid read command, the card will be respond with a response taken followed by the data taken in the length defined by a previous SET_BLOCK_LENGTH command.

Expands the key material so that each round uses a unique round.

Round 0: 54 68 61 74 73 20 6D 79 20 4B 75 6E 67 20 46 75
• Round 1: E2 32 FC F1 91 12 91 88 B1 59 E4 E6 D6 79 A2 93
• Round 2: 56 08 20 07 C7 1A B1 8F 76 43 55 69 A0 3A F7 FA
• Round 3: D2 60 0D E7 15 7A BC 68 63 39 E9 01 C3 03 1E FB
• Round 4: A1 12 02 C9 B4 68 BE A1 D7 51 57 A0 14 52 49 5B
• Round 5: B1 29 3B 33 05 41 85 92 D2 10 D2 32 C6 42 9B 69
• Round 6: BD 3D C2 B7 B8 7C 47 15 6A 6C 95 27 AC 2E 0E 4E
• Round 7: CC 96 ED 16 74 EA AA 03 1E 86 3F 24 B2 A8 31 6A
• Round 8: 8E 51 EF 21 FA BB 45 22 E4 3D 7A 06 56 95 4B 6C
• Round 9: BF E2 BF 90 45 59 FA B2 A1 64 80 B4 F7 F1 CB D8
Round 10: 28 FD DE F8 6D A4 24 4A CC C0 A4 FE 3B 31 6F 2 6 3.

The features of AES are as follows

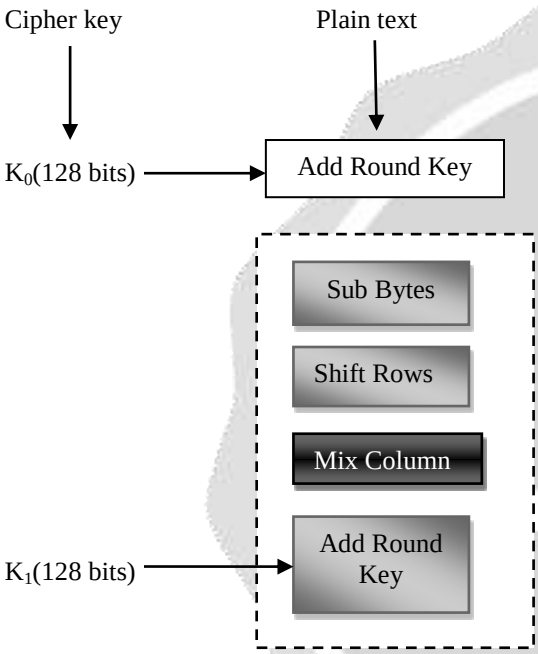
- Symmetric key symmetric block cipher.
- 128 bit data, 128/192/256 bit keys.
- Stronger and faster than Triple DES.
- Provide full specification and design details.
- Software implementable in C and JAVA.

B.Operation of AES

AES is an iterative rather than fiestel cipher. It is based on 'substitution-permutation network'. It comprises of a series of linked operations, some of which involve replacing inputs by specific outputs (substitutions) and others involve shuffling bit around (permutation). AES performs all its computations on bytes rather than bits. Hence, AES treats the 128 bits of plain text block as 16 bytes. These 16 bytes are arranged in four columns and four rows for processing as a matrix. DES, the number of rounds in AES is variable and depends on the length of the key. AES uses 10rounds for 128-bit keys, 12 rounds for 192-bit keys and 14 rounds for 256-bit keys. Each of these rounds uses a different 128-bit round key, which is calculated from the original AES key.

B. Encryption Process

Here, we restrict to description of a typical round of AES encryption. Each round comprise of four sub-processes. The first round process is depicted below



C. *Mix Column*

Each column of four bytes is now transformed using a special mathematical function. This function takes as input the four bytes of one column and outputs four completely new bytes, which replace the original column. The result is another new matrix consisting of 16 new bytes. This step cannot be performed in the last round.

D. Add Round Key

The 16 bytes of the matrix are now considered as 128 bits and XOR-ed to the 128 bits of the round key. If this is the last round then the output is the cipher text. Otherwise, the resulting 128 bits are interpreted as 16 bytes and we begin another similar round.

E. *Decryption Process*

The process of decryption of an AES cipher text is similar to the encryption process in the reverse order. Each round consists of the four processes conducted in the reverse order.

- Add Round Key
- Mix Column
- Shift Rows

- Byte Substitution

Since, sub-processes in each round are in reverse manner, unlike for a fiestel cipher, the encryption and decryption algorithm needs to be separately implemented, although they are very close related.

VII. CONCLUSION

In this paper, consider the problem of verifying the utility of data released by non-interactive differentially private methods. Similar mechanism are proposed to achieve the goal for secure publishing and search with the minimum time. The proposed solution require the publisher to provide auxiliary datasets in cipher text along with the publishing data. A published data can be stored in database with encrypted value for a securing purpose. The data will be published and the indexed variables are stored in an array. User's entries will be compared to the array, First, the matched part will be display to the users and the remaining part will be displayed below that part. It reduces a time when users can search the particular data's in a data set.

VIII. REFERENCES

[1] M. Raykova, SM.bellovin and T.Malkin "Secure anonymous data base search in proc.ACM workshop CCSWS,2009.
[2] P.Wang and CV.Ravishankar" secure and efficient range queries on outsourced data base using Rtrees in PROC.IEEE 29th ICDE.
[3] C.Dwork,F.Mcsherry,K.Nissim and A.Smith"calibratingnoise to sensitivity in data analysis" In proc theory cryptography 2006.
[4] B. C. M. Fung, K. Wang, R. Chen, and P. S. Yu, "Privacy preserving data publishing: A survey of recendevelopments,"ACM Comput. Surv.vol. 42, no. 4, 2010, Art. no. 14.
[5] N.Mohammed, D.Alhadidi, B.C.M fung, and M.Debbai, "Secure two-party differentially private data release for vertically partitioned data "IEEE trans dependable secure compute.