Volume 4, Issue 9, September 2014 ISSN: 2277 128X

International Journal of Advanced Research in Computer Science and Software Engineering

Research Paper

Available online at: www.ijarcsse.com

Privacy-Preserving Updates to Anonymous Databases with Security

Pallavi S. Patil
Department of Computer Sci.Engg,
DYCOET, Kolhapur, India

Prof. K. B. Manwade
Department of Computer Sci.Engg.,
AMGI, Vathar, India

Prof. G. A. Patil
Department of Computer Sci.Engg,
DYCOET, Kolhapur,India

Abstract— In today's world security becomes valuable and necessary asset for many applications, so security becomes critical. Bank information, medical research database all this information can be dangerous if it fail into wrong hands. Privacy of this invaluable asset needs to be protected. However, Privacy is not only the issue. Data confidentiality is also concern to all the problems regarding database. Although privacy and confidentiality both are different concept. Privacy relates to person whereas confidentiality relates to data. So problem arises at this point where database needs to be updated. When tuple is to be inserted in the database problem occurs relating to privacy and confidentiality that is database owner decide that whether database preserve privacy without knowing what new tuple to be inserted. To carry out task of privacy, confidentiality to anonymous database, two approaches can be used. One is Suppression and the other is Generalization.

Keywords- Anonymous, Confidentiality, Privacy, Third party.

I. INTRODUCTION

Propels in advanced estimation and designing innovations empower the catch of gigantic measures of information in fields, for example, stargazing, pharmaceutical, and seismology. The exertion for information accumulation and handling, and also its potential utility for exploration or business, makes esteem for the information manager. He wishes to store them and permit get to without anyone else's input, partners, and other (trusted) researchers or clients. This might be backed by outsourced servers that offer low stockpiling expenses for extensive databases. Case in point, outsourcing focused around distributed computing is getting to be progressively alluring, as it guarantees pay-as-yougo, low stockpiling expenses and also simple information access. Nonetheless, mind needs to be taken to shield information that is profitable or touchy against unapproved access. In this setting, we call anything in an information accumulation an article, people with approved access question clients, and the substance offering the stockpiling administration the administration supplier. We represent the affectability issues with a few situations. Initially, consider space projects, for example, the NASA Apollo program on the Earth's Moon1 or the ESA Mars Express2 that gather logically significant and uncommon information.

Anonymization means concealing. That is distinguishing data is expelled from the first information to ensure individual or private data. Information Anonymization empowers exchanging data between two associations, by changing over content information into non-intelligible structure utilizing encryption strategy [4]. The K-namelessness is accomplished by hindering all the unique qualities (Suppression) or by supplanting them with a less particular basic reliable quality (Generalization). Consequently K-obscurity is exchange off between information utility and information confidentiality. It is essential to note that the information misfortune happens for identifiers and quasi-identifiers just.

So problem arises at this point where database needs to be updated. When tuple is to be inserted in the database problem occurs relating to privacy and confidentiality that is database owner decide that whether database preserve privacy without knowing what new tuple to be inserted. To carry out task of privacy, confidentiality to anonymous database, two approaches can be used. One is Suppression and the other is Generalization.

II. SYSTEM ARCHITECTURE

- **Problem Definition**

In the existing system data are store in database directly. Anyone can easily retrieve information like username, password. Etc. The cryptography security is not maintained here. The classification of database is carried out from local system only. Any unauthorized person can easily access the database. Authorized person can view the other user's data too. Data confidentiality is particularly relevant because of the value, often not only monetary, that data possess. A requirement has motivated a large variety of approaches aiming at better protecting data confidentiality and data ownership. The availability of huge numbers of databases recording a large variety of information about individuals makes it possible to discover information about specific individuals by simply correlating all the available databases.

- **Present System**

The general taking an interest of our application is information supplier enter information onto the framework is as tuple then framework perform nameless verification and overhaul operation on tuple lastly put away that tuple in database The client can likewise get to the put away record by just utilizing inquiry. In any case, a general issue is that on the off chance that we consider that the put away information is classified, then how it conceivable to concede for overhaul and safeguard secrecy of database. In such a state of change methodology of including people data, two issues are acquaints agreeing with namelessness and classifiedness of the information in the database and the protection of information supplier. 1) Is the altered database even now safeguarding the security? What's more 2) would it say it is important to know the real embedded information to database manager?

- **Proposed System**

We propose two protocols tackling this issue on suppression-based and generalization-based K-anonymous and secret databases. The protocols depend on well-known cryptographic suppositions, and we give hypothetical dissects to demonstrate their soundness and trial results to show their proficiency. It is today well comprehended that databases speak to an imperative possession for some applications and accordingly their security is vital. As of late, procedures tending to the issue of protection by means of information anonymization have been created, hence making it harder to link delicate data to particular people. One well known system is k-anonymization.

Cryptography technique is using secure data storing in server. The protocols we propose to solve Problem 1 rely on the fact that the anonymity of database is not affected by inserting t if the information contained in t, properly anonymized, and is already contained in DB [1]. Then, Problem 1 is equivalent to privately checking whether there is a match between (a properly anonymized version of) t and (at least) one tuple contained in DB. The block diagram of proposed system shown in Figure 1.

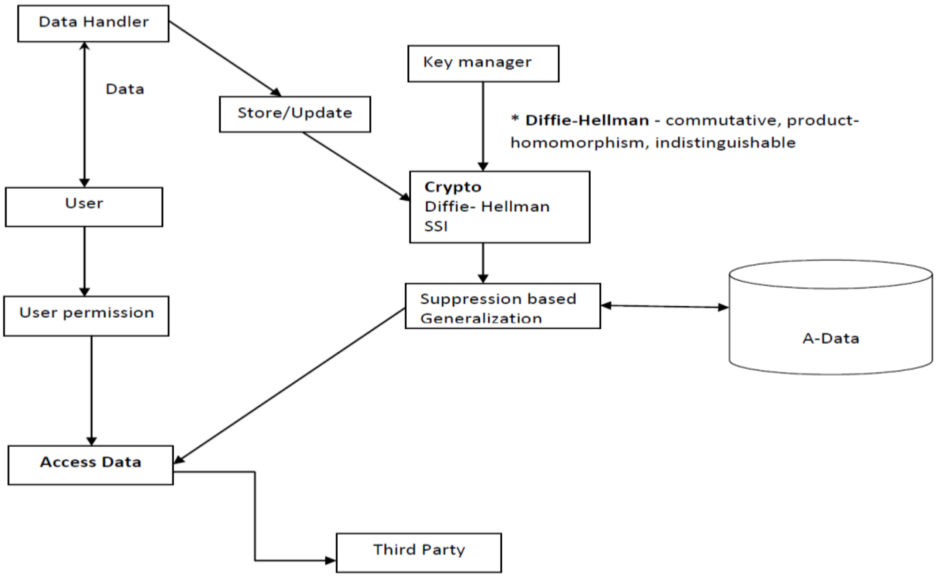


Figure 1: Block Diagram of Proposed System

III. A PRIVATE UPDATE PROTOCOL FOR SUPPRESSION BASED ANONYMOUS DATABASES

In this segment, we expect that the database is anonymized utilizing a suppression-based technique. Note that our protocols are not needed to further enhance the security of clients other than that gave by the way that the upgraded database is still kanonymous. Assume that Alice claims a k-anonymous table T over the QI qualities. Alice need to choose whether T when embedded with a tuple t, claimed by Bob, is still k-anonymous, without straightforwardly knowing the qualities in t (accepting t and T have the same construction). This issue adds up to choose whether t matches any tuple in T on the non-stifled Q I characteristics. The protocol works as follows:

Step 1, Alice sends Bob an encrypted version of Ji, containing only the s non-suppressed QI attributes.
Step 2, Bob encrypts the information received from Alice and sends it to her, along with encrypted version of each value in his tuple t.
Steps 3-4, Alice examines if the non-suppressed QI attributes of hi is equal to those of t.

- **Protocol Algorithm**

1. Alice codes her tuple Ji into c((vi,. . . , v;)), denoted as ~ (6 ~T)h.e n, she encrypts c(Ji) with her private key and sends EA(c(Ji)) to Bob.
2. Bob individually codes each attribute value in to get the tuple of coded values (c(vl), . . . , vu)), encrypts each coding and EA(C(Oi)) with his key B and sends
   (i) (EB(C(VI)), ... , EB(c(vu))),
   (ii) EB (EA(c(bi))) to Alice.

3. Since E is a commutative encryption scheme, EB(EA(c(bi)))= EA(EB(c(bi))), Alice decrypts
EB (~ (b i)))t o c(6i)).
4. Since the encrypted values sent by Bob are ordered in a tuple according to the schema of T, Alice knows which is, among the encrypted values sent by Bob, the one corresponding to the suppressed and non-suppressed QI attributes. Thus Alice computes

$$\text{(iii)} \quad E_B(c(v_1)) \times \ldots \times E_B(c(v_s))$$

Where vl, . . . , v, are the values of non-suppressed attributes contained in tuple. As already mentioned, E is a product-homomorphic encryption scheme. Based also on the definition of function c(.), this implies that Expression 3 is equal to

$$\text{(iv)} \quad E_B(c(\langle v_1, \ldots, v_s \rangle))$$

5. Alice checks whether EB(c((v1, . . . , v,))) = EB(c((vi, . . . , vi))). If true, t (properly anonymized) can be inserted to table Otherwise, when inserted to breaks k-anonymity.

IV. A PRIVATE UPDATE PROTOCOL FOR GENERALIZATION BASED ANONYMOUS DATABASES

In this section, we assume that the table T is anonymized using a generalization-based method; let rl, . . . , I?, be u disjoint value generalization hierarchies (VGHs) corresponding to A1,. . . , A, E Atnon known to Alice. Let S E T, and let GetSpec(SIAl, . . . ,A,], rl, . . . , I',) (GetSpec(6) for short) denote a function which returns a set y of specific values (values at the bottom of a VGH) related to each attribute Ai E Atnon such that every value in y can be generalized to S[Ai] for some i according to ri.

Let t be Bob's private tuple, and assume that Bob knows the set AFnon. Bob can generate a set T containing the corresponding values t[A1], . . . , t[A,]; the size of T is always u. We denote by SSI(y, T) as a secure protocol that computes the cardinality of y n ~. Upon receiving an initial request from Bob, Alice starts the protocol by randomly choosing a tuple S from the witness set Tw of T. After Alice computes y = GetSpec(S), she and Bob privately compute SSI(y, 7). Note that Bob does not need to know any I'i. We claim that if SSI(y, T) = u (the size of A:non), t[A1,. . . ,A,] can be generalized to S, and hence this insertion into T can be safely performed without breaking the k-anonymity property.

The protocol works as follows:
Step1. Alice randomly chooses a δ Tw.
Step2. Alice computes γ Γ GetSpec(δ)γ
Step3. Alice and Bob collaboratively compute s= SSI(γ,Γ)
Step4. If s =u then t's generalized form can be safely inserted to T.
Step5. Otherwise, Alice computes Tw ←Tw –{δ}and repeat the above procedures until either s =u or Tw =Ø

Where GetSpec (δ) denote a function which returns a set of specific values each attribute.
Where X is an server user & Y is an client user.

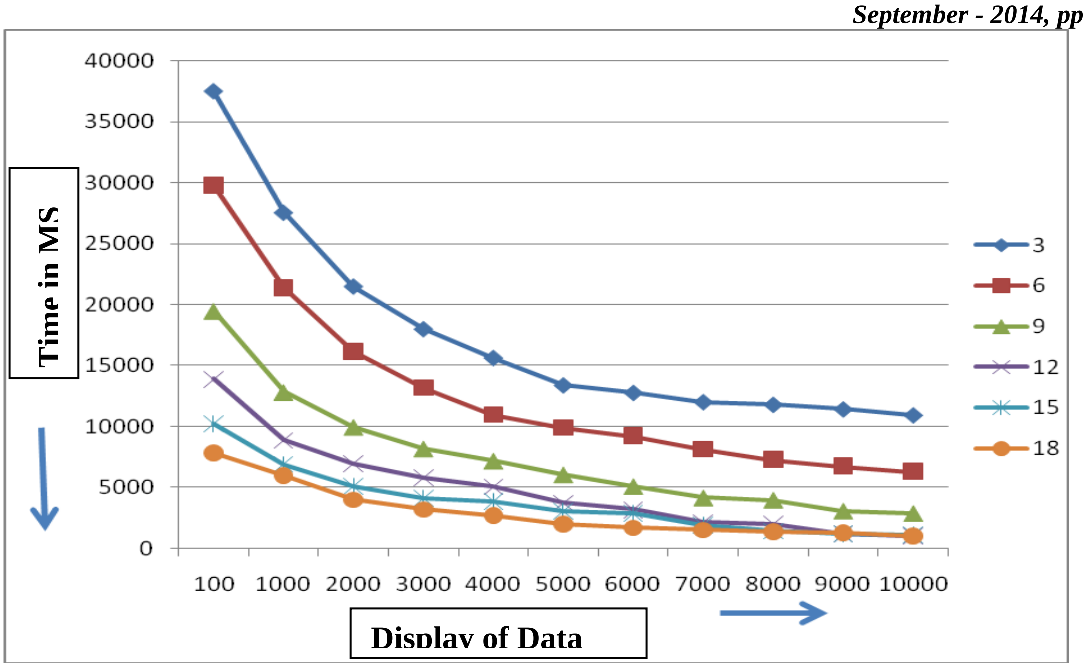
V. EXPERIMENTAL SETUP & IMPLEMENTATION

- **Implementation**
- To implement above system some pre work to do.
- To form Formation 1 protocol prepare module from proposed work:
  - Crypto Module
  - Checker Module.
  - Loader Module.

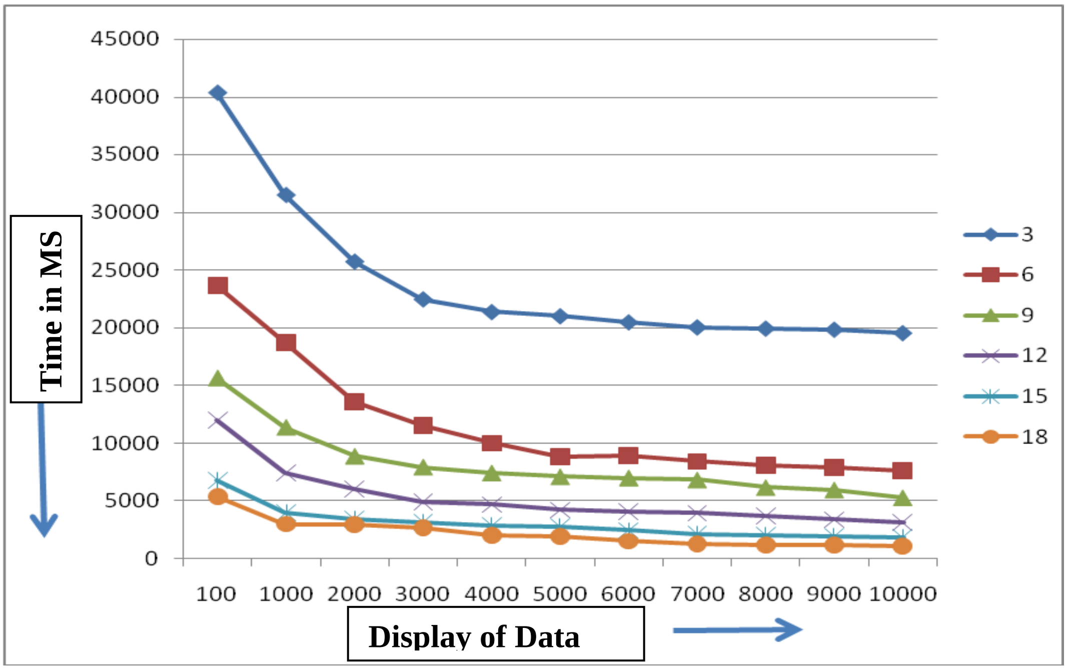
- A crypto module that is in charge of encrypting all the tuples exchanged between an user and the Private Updater, using the formation1 protocol.
- A checker module that performs all the controls, as prescribed by Protocols.
- For implementation of crypto & checker module DSA algorithm is used,
- Loader module that reads chunks of anonymized tuples from the k-anonymous DB.
- The chunk size is fixed in order to minimize the network overload.

VI. RESULT

We watch and perform really insertion of record into database around then our framework ascertains the normal execution times means what number of time used in insertion or updation of database (computed in milliseconds) of both suppression and generalization-based methodologies. Framework to produce the diagram of ascertained normal execution times in Graph 1 and Graph 2 shows execution times in suppression and generalization based methods separately. In the figure it is noted that time used by both methods in testing whether the tuple might be securely embedded in the anonymized database diminishes as the estimation of k increments.



Graph 1: Execution times of protocol as the parameter k increases (Suppression Protocol)



Graph 2: Execution times of protocol as the parameter k increases (Generalization Protocol)

In a general view, the insertion of record having two different background procedure of database k-secrecy checking and the genuine redesign into two separate stages, in the first stage, when a tuple is embedded into database then it check embedded tuple with existing information and whether the upgraded database is stays k-anonymous, then updation will happened. In this checking methodology checker don't have a clue about the substance of client's tuple. In second stage, framework really redesigns the database relies on upon the consequence of the secrecy checker. Sometimes the insertion or updation fizzled in k-anonymous database then it holds up until k-1 quality gets to be sure and different tuples fail the insertion.

Assuming the worst-case scenario, this has to be executed w times, where w is wittiness set (attribute set). Thus the number of messages is 6 X w. The complexity of protocol relies on the size of tuple and the complexity of the SSI protocol. The number of calls to the SSI protocol is bounded by each tuple.

VII. CONCLUSION

In this, we focus on a privacy preserving of k-anonymous database. We have presented two secure protocols suppression based and generalization-based for database anonymization techniques for protecting individual's privacy. In a anonymous database when new tuple is inserted then database owner privately check that whether anonymous database remains anonymity. Since the proposed protocols work perfect on updated database will definitely k-anonymous. Thus by using proposed protocol the database is updated properly. System every time check when new tuple inserted into database and if it satisfies k-anonymity then tuple is accepted for insertions otherwise it will prohibited. Suppressed the value of attribute by replacing '*' and Generalized the value with related possible general value to maintain the k-anonymity in database. Thus by making such k-anonymity in table it becomes complicated for third party to identify the record.

VIII. FUTURE WORK

- In case of unauthorized user, non-colluding third party, implementing a real-world anonymous database system.
- How to increase the efficiency of implementation and quality of the released output data in such a way to get the various requirements.

REFERENCES

[1] G. Aggarwal, T. Feder, K. Kenthapadi, R. Motwani, R. Panigrahy,D. Thomas, and A. Zhu, "Anonymizing Tables," Proc. Int'l Conf. Database Theory (ICDT), 2005.

[2] L. Sweeney, "k-Anonymity: A Model for Protecting Privacy," Int'l J. Uncertainty, Fuzziness and Knowledge-Based Systems, vol. 10, no. 5, pp. 557-570, 2002.

[3] D. Bonch The decision Diffic-Hellman problem in Proc. Of Int Algorithmic Number theory Symposium.

[4] Privacy-Preserving Updates to Anonymous and Confidential Databases, Alberto Trombetta, Wei Jiang, Elisa Bertino and Lorenzo Bossi, Department of Computer Science and Communication, University of Insubria, Italy.

[5] A. Trombetta and E. Bertino, "Private Updates to Anonymous Databases," Proc. Int'l Conf. Data Eng. (ICDE), 2006.