ijcrr
Vol 04 issue 10
Category: Research
Received on:16/04/12
Revised on:21/04/12
Accepted on:25/04/12

PRIVACY-PRESERVING UPDATES TO ANONYMOUS AND CONFIDENTIAL DATABASES USING CRYPTOGRAPHY WITH ARM

Vanitha.T[1], Judeth Deborah[1], G.S.Sooriya[1], Aruna.V[2]

[1]ME-Communication Systems, S.A.Engineering College, Chennai
[2]Dept of ECE, S.A.Engineering College, Chennai

E-mail of Corresponding Author: everfreshvani@gmail.com

ABSTRACT

Databases represent an important asset for many applications but their security is crucial. Today scenario there is an increased concern for privacy and confidentiality. Databases recording a variety of information about individuals which will be maintained by database owners and users respectively. If Alice owns a K-anonymous database and needs to determine whether her database, when inserted with a tuple owned by Bob, is still k-anonymous. Also, suppose that access to the database is strictly controlled, allowing Alice to directly read the contents of the database breaks the privacy of Bob. (e.g., a patient's medical record). Thus, the problem is to check whether the database inserted with the tuple is K-anonymous, without letting Alice and Bob know the contents of the tuple and the database respectively. To preserve and secure this database and its updates cryptography is used. In this paper, the existing cryptography is software based. The proposed system is hardware and software based using ARM processor. This system provides protection against brute force rewind attacks, offline parallel attacks, and other cryptanalysis attacks.

Keywords:-Privacy, anonymity, data management, secure computation.

INTRODUCTION

Today data confidentiality is particularly relevant because of the value, often not only monetary, it also preserve the updates .For example, medical data collected by following the history of patients over several years may represent an invaluable asset that needs to be adequately protected. Such a requirement has motivated a large variety of approaches aiming at better protecting data confidentiality and data ownership. Relevant approaches include query processing techniques for encrypted data and data watermarking techniques. Data confidentiality is not however the only requirement that needs to be addressed.

Although confidentiality and privacy are often used as synonyms, they are different concepts: data confidentiality is about the difficulty (or impossibility) by an unauthorized user to learn anything about data stored in the database. Usually, confidentiality is achieved by enforcing an access policy, or possibly by using some cryptographic tools. Privacy relates to what data can be safely disclosed without leaking sensitive information regarding the legitimate owner [3]. There are many ways to perform data anonymization. We only focus on the k-anonymization approach [4], [5].

To better understand the difference between confidentiality and anonymity, consider the case of a medical facility connected with a research institution. Suppose that all patients treated at the facility are asked before leaving the facility to donate their personal health care records and medical histories (under the condition that each patient’s privacy is protected) to the research institution, which collects the records in a research database. To guarantee the maximum privacy to each patient, the medical facility only sends to the research database an anonymized version of the patient record. Once this anonymized record is stored in the research database, the non-anonymized version of the record is removed from the system of the medical facility. Thus the research database used by the researchers is anonymous.

Addressing the problem of privacy via data anonymization, One well-known technique k-anonymization is used [4], [5]. Such technique protects privacy by modifying the data so that the probability of linking a given data value. The problem arises when data stored in a confidential, anonymity-preserving database need to be updated. The operation of updating such a database, e.g., by inserting a tuple containing information about a given individual, introduces two problems concerning both the anonymity and confidentiality of the data stored in the database and the privacy of the individual to whom the data to be inserted are related: (i) Is the updated database still privacy-preserving? and (ii) Does the database owner need to know the data to be inserted? The two problems will be overcome by two protocols used in this existing paper. .

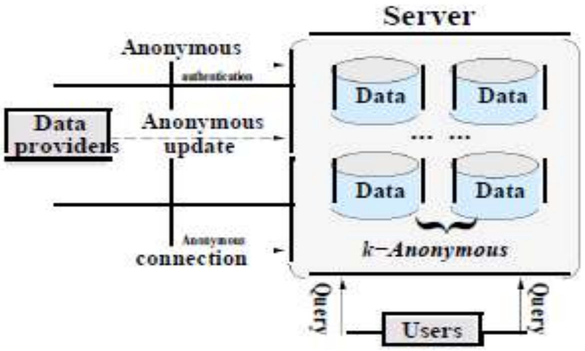

Figure 1: Anonymous Database System

Problem Statement

Figure1 captures the main participating parties in our application domain. We assume that the information concerning a single patient (or data provider)is stored in a single tuple,and DB(Database) is kept confidentially at the server. The users in Figure 1 can be treated as medical researchers who have the access to DB. Since DB is anonymous, the data provider’s privacy is protected from these researchers. (Note that to follow the traditional convention, in Section 4 and later sections; we use Bob and Alice to represent the data provider and the server respectively.

The modification of the anonymous database DB can be performed as follows: In this paper, software based cryptography is existed. The basic concept used here is embedded cryptography. The objective is to design hardware and software based cryptography. The two parties Alice and Bob were considered as two ARM processors to provide the database and transfer the information. Another ARM also considered as intruder (Eve).

Note that to assure a higher level of anonymity to the party inserting the data, we require that the communication between this party and the database occurs through an anonymous connection, as provided by protocols like Crowds or Onion routing [6].

Table1: Anonymous Database System Requirements

Requirement	Objective	Protocol
Anonymous connection	Protect IP address and sensitive info	Crowds [27], Onion Routing [26]
Anonymous authentication	Protect sensitive authentication info	Policy-hiding access control [20]
Anonymous update	Protect non-anonymous data u	**Proposed in this paper**

Figure1 summarizes the various phases of a comprehensive approach to the problem of anonymous updates to confidential databases, while Table 1 summarizes the required techniques and identifies the role of our techniques in such approach.

Proposed Solutions

To create a hardware and software based algorithm that is designed to encrypt computer data in such a way that it cannot be recovered without access to the key. Software encryption is a fundamental part of all aspects of modern computer communication and files protection and may include features like file shredding.

The purpose of encryption is to prevent third parties from recovering the original information. This is particularly important for sensitive data like credit card numbers.

Hardware Cryptography

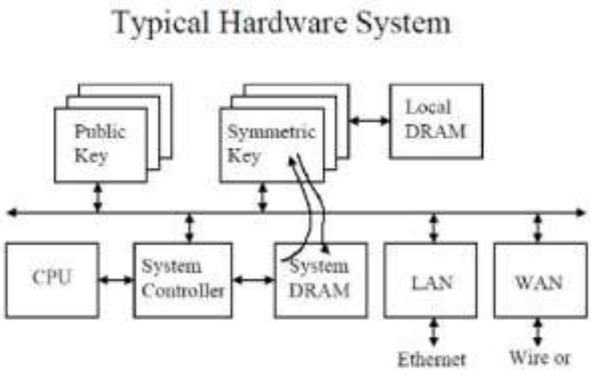

Figure 2: Typical cryptography hardware system

Figure 2 shows the basic hardware cryptography system. This consists of two keys public key and private key (symmetric key). The keys are assigned to Alice and Bob respectively. Embedded cryptography is nothing but the cryptography is in built within the embedded system. It provides security for the embedded devices. [1], [2]

Commonly there are lots of problems attacks on the embedded system such as brute force rewind attacks, offline parallel attacks, or other cryptanalysis attacks. This system handles two cryptography. And also design hardware based cryptography. Encryption software executes an types of databases as suppression and generalization based databases.

These databases are analyzed using the software as keil microvision with embedded C. It also preserves the updates of the database. In this,

we have proposed the hardware to protect the updates of the databases and also preserve the privacy of the individual users of the database management.

The figure 3 shows the various possible security boundaries for the embedded devices. The most possible security aspects are IPSec, firewall etc.,

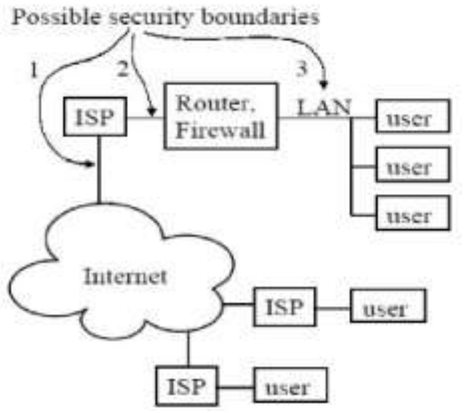

Figure 3: Possible security boundaries

The unbreakable hardware cryptography is implemented in our paper. Almost all of the practical cryptosystems are theoretically breakable given the time and computational resources. However, there is one system which is even theoretically unbreakable. One-time pad requires exchanging key that is as long as the plaintext. However impractical, it is still being used in certain applications which necessitate very high-level security. Security of one-time pad systems relies on the condition that keys are generated using truly random sources.

Experimental Setup for Hardware Cryptography

The main basis of attacking a crypto server is based on development of a Database. The database is acquired through a separate experimental setup built around an embedded system with encryption algorithm. Fig 4 shows the experimental setup used for the acquisition of data base. The ARM crypto server is fed with known Input data, Encryption Algorithm and the key value from HOST using RS232 communication. The crypto server performs the encryption operation. During the encryption, HOST sends a command signal for verify that the route is free of unauthorized access. Logic Analyzer senses command from the HOST and sends the ACK to the HOST. Crypto Server also makes available the Encrypted output to the HOST. A database is created by the HOST combining the inputs and outputs received from crypto server and Logic Analyzer.

The database is created for different samples of data sent from the HOST and the results generated by the crypto server and the Logic Analyzer is padded with the input data to form an entry into the database.

The database created out of execution of several samples of data submitted through the HOST and the output obtained from Crypto Sever and the Logic Analyzer.

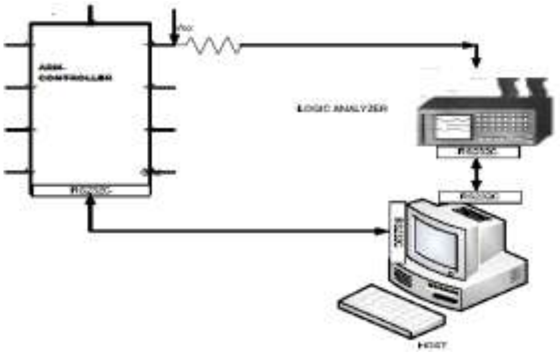

Figure 4: Experimental setup

Advantages

Hardware-based encryption, when implemented in a secure manner, is demonstrably superior to software-based encryption. That being said, hardware-based encryption products can also vary in the level of protection they provide against brute force rewind attacks, offline parallel attacks, or other cryptanalysis attacks.

Hardware-based encryption has other benefits for users. Software based encryption typically runs much more slowly than hardware-based encryption. Iron Key devices are specially optimized for high-speed data transfer, performing at the top of their class by reading data at up to 29 megabytes per second and writing data at 18 megabytes per second.

Architecture and Experimental Results

Our prototype of a Private Checker (that is, Alice) is com-posed by the following modules: a crypto module that is in charge of encrypting all the tuples exchanged between a user (that is, Bob) and the Private Updater.

Modules are represented along with labeled arrows denoting what information are exchanged among them. Note that the functionality provided by the Private Checker prototype regards the check on whether the tuple insertion into the k-anonymous DB is possible. We do not address the issue of actually inserting a properly anonymized version of the tuple.

The simulation results are shown in the graph.

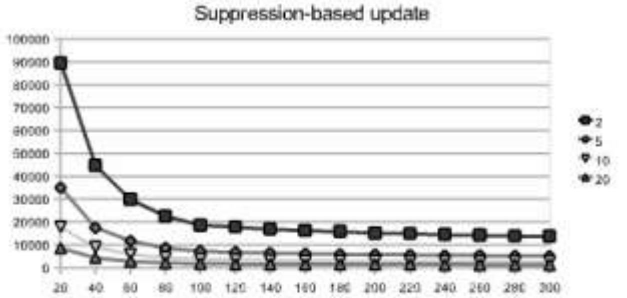

Fig 4.1: Execution time of suppression based updates

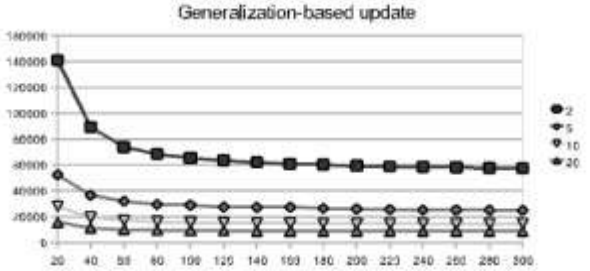

Fig 2: execution time of generalization based updates

Applications

The hardware based cryptography was designed to preserve the privacy of databases. It also maintains and protects the privacy of updates to databases. The hardware based encryption is implemented using ARM processor (LPC2148). The encryption consists of two keys such as public key and symmetric key. It is more securable than software based cryptography. The assumption of password and keys are impossible. It is applicable in maintaining databases in large organizations such as hospitals, large scale industries.

CONCLUSION

In this paper we have presented hardware based cryptography to preserve the privacy of individuals and database owners to maintain the two different updates suppression and generalization based approach. The implementation was done with the help of ARM processors. It provides security and confidentiality to the embedded devices.

In hardware based cryptography the encryption will be done with the help of standard encryption techniques advanced encryption standard (AES). By using this encryption standard the data are updated and preserved in the database system.

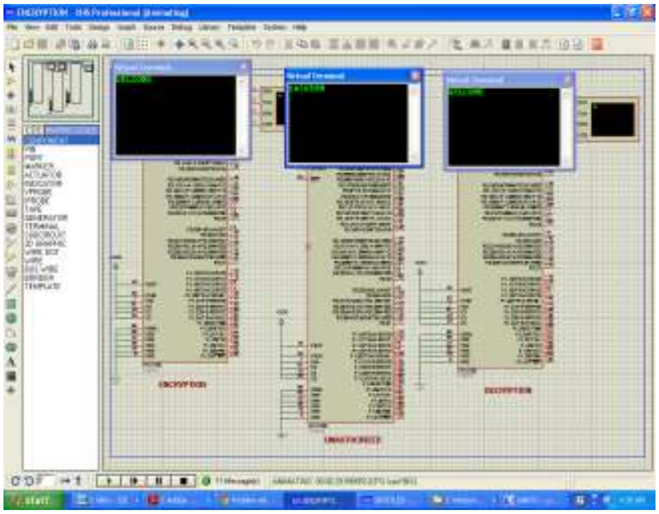

Figure 5: Simulation of Hardware Cryptography

REFERENCES

1. Cryptography using Arm processor, security and privacy, IEEE, issue date jan-feb 2006 volume: 5 issue 1.
2. The Hardware-based PKCS#11 Standard using the RSA Algorithm Latin America Transactions, IEEE (Revista IEEE America Latina) Issue Date: June 2009 Volume: 7 Issue: 2
3. E. Bertino, R. Sandhu. Database security - Concepts, approaches and challenges. IEEE Transactions on Dependable and Secure Com- puting, 2(1), 2005; 2– 19.
4. P.samarati.Protecting respondent’s privacy in microdata release.IEEE Transactions on knowledge and data engineering, vol.13, no.6, Pp.1010-1027 Nov/Dec 2001, UK, 2005.
5. L.Sweeney.K-anonymity: A model for protecting privacy international journal on uncertainty, Fuzziness and knowledge-based systems, 10(5), 557-570, 2002
6. M.Reed, P.Syverson, D. Goldschlag. Anonymous Connections and onion routing. IEEE journal of selected areas in communications, 16(4), 1998; 482-494