PERPECTUAL AUTHENTICATION AND INTEGRITY SCHEMES FOR REDISTRIBUTED DATABASES

[1]Karpagam T
Karpagamdv83@gmail.com
Assistant Professor

[2]Maheswari B
mahebala1296@gmail.com

[3]Saranya J
saranmj55@gmail.com

[4]Vaishalie M
vsdec4@gmail.com

[2][3][4]UG students
Department of Computer Science and Engineering
T.J.S. Engineering College

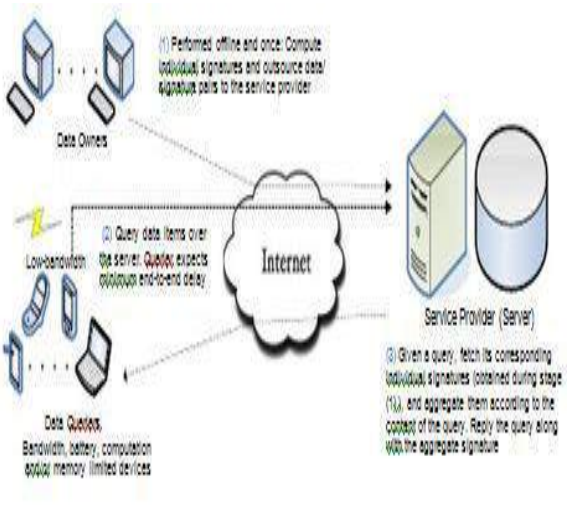
Abstract

Database outsourcing is a prominent trend that enables organizations to deliver their data management overhead (e.g., query handling) to the external service provider. To provide authentication and integrity for redistributed databases using the immutable signature tool, this property is called immutability. In existing system, the algorithm has less security due to low key size. In order to overcome those problems and to improve their security proposed technique RSA is used to provide high security with no attacks.

Index terms: *Redistributed database, immutable digital signature, RSA algorithm.*

1. Introduction

It is a growing trend that the data is outsourced and being managed on remote servers, which are maintained by third party outsourcing vendors. One such data outsourcing approach is "database as a service" (DAS) in which clients outsource their data to a database service provider that offers a reliable maintenance/access for the hosted data.Data outsourcing can significantly reduce the cost of data management (e.g., via continuous service, expertise, maintenance) and therefore it is highly beneficial for entities with limited management capabilities such as small to medium businesses . However, despite its merits, data outsourcing brings various security challenges, since the sensitive data is hosted in a (semi)untrusted environment. These security challenges include but not limited to the confidentiality , access privacy, authentication and integrity . Another challenge is to provide the security efficiently such that the data outsourcing still remains practicaland cost efficient.The focus of this paper is to provide authentication and integrity of outsourced data via aggregate signatures while also guaranteeing a vital security property called signature immutability in a practical manner. Differences between this article and its preliminary version In this article, we develop a new construction and also give a more comprehensive security and performance analysis over the preliminary version.We introduce a new scheme called PISB -RP that offers the lowest end-to-end delay among existing alternatives.


(1) Performed offline and once: Compute individual signatures and outsource data/ signature pairs to the service provider
Data Owners
Low-bandwidth
(2) Query data items over the server. Querier expects minimum end-to-end delay
Internet
Service Provider (Server)
(3) Given a query, fetch its corresponding individual signatures (obtained during stage (1)), and aggregate them according to the context of the query. Reply the query along with the aggregate signature
Data Queriers,
Bandwidth, battery, computation and/or memory limited devices

Mykletun et al.'s Outsourced Database Model (ODB)

2. System and Data Model

We follow Mykletun et al.'s Outsourced Database Model (ODB) ,as a variant of "database as a service".

System Model: There are three types of entities in the system; data owners, server (database service provider) and data queriers (clients). These entities behave as follows.

• **Data Owners**: A data owner can be a single or a logical entity such as an organization. Each data owner in the system signs her database elements (e.g., each tuple separately) and then outsources them along with their signatures to the server. This protects the integrity and authentication of outsourced data against both the server and outside adversaries (e.g., in the case of the server is compromised).

The data owner computes the individual signature of each database element (e.g., each tuple) with an aggregate signature scheme, which allows the combination of these signatures according to the content of a query. This enables the server to reply any query on the outsourced data with a compact constant size signature (instead of sending a signature for each element in the query, which entails a linear communication overhead). This outsourcing step is performed offline, and therefore its cost is not the main concern.

• **Server (Service Provider):** The server maintains the data and handles the queries of data queriers. The server is trusted with these services, but it is not trusted with the integrity and authentication of the data. Hence, each data owner digitally signs her data before outsourcing it as described previously.Once a data querier (i.e., clients who perform data queries) queries the server, the server computes a constant size signature by aggregating the corresponding individual signatures of database elements associated with this query. Recall that the server knows these individual signatures, since the data owner provided all individual signatures to the server at the offline phase. The server then performs necessary cryptographic operations to ensure the immutability of this aggregate signature. Observe that the server faithfully follows the immutability operations, since the immutability prevents external parties to offer similar services free of charge. The query handling phase is performed online. The server is expected to handle larger number of queries simultaneously with a minimum end-to-end delay. Therefore, the cost of signature immutability operations is highly critical.

• **Data Queriers (Clients):** Queriers are heterogeneous entities, which may be resource-constrained in terms bandwidth, battery and/or computation (e.g., a PDA). A querier can make a query on the database elements belonging to a single or multiple data owners. The former is called single signer queries while the latter is called multiple signer queries. The data querier verifies the aggregate signature of her query, along with cryptographic tokens transmitted for the immutability.

Data Model: We assume that the data is managed with a traditional relational database management system and the queries are formulated with SQL. Our work handles only SQL queries involving SELECT clauses, which return the selection of a set of records or fields matching a given predicate. The granularity of data integrity and authentication may vary according to the application (e.g., attribute level). For example, one possible choice is to provide them at the tuple level (i.e., sign each tuple individually), which offers a balance between the storage, transmission and computation overheads introduced by the cryptographic scheme.

3. Literature Review

Enable a frequent querier to eventually amass enough aggregated signatures to answer other (un-posed) queries using immutable. Secure data storage and retrieval of the relevant information from the traditional as well as in cloud computing environment. Enables query processing while providing privacy protection of genomic databases**.** Authentication mechanisms are considered the typical method to secure financial websites. Context authentication and multifactor authentication. Automated Turing Tests continue to be an effective, easy-to-deploy approach to identify automated malicious login attempts with reasonable costs of inconvenience to users.

4. Existing System

Data outsourcing can significantly reduce the cost of data management (e.g., via continuous service, expertise, maintenance) and therefore it is highly beneficial for entities with limited management capabilities such as small to medium businesses.

However, despite its merits, data outsourcing brings various security challenges, since the sensitive data is hosted in a (semi) untrusted environment. These security challenges include but not limited to the confidentiality, access privacy, authentication and integrity. Another challenge is to provide the security efficiently such that the data outsourcing still remains practical and cost efficient.

5. Proposed System

The focus of this paper is to provide authentication and integrity of outsourced data via aggregate signatures, while also guaranteeing a vital security property called signature immutability in a practical manner. Differences between this article and its preliminary version. In this article, we develop a new construction and also give a more comprehensive security and performance analysis over the preliminary version.

We introduce a new scheme called PISB-RP that offers the lowest end-to-end delay among existing alternatives. We investigate the relationship between signature immutability and aggregate signature extraction, which has been omitted in previous outsourced database authentication schemes. We proved that the signature extraction is a necessary condition for some immutable signature constructions such as PISB-RP. We discuss pros and cons of various PISB instantiations by highlighting their performance characteristics

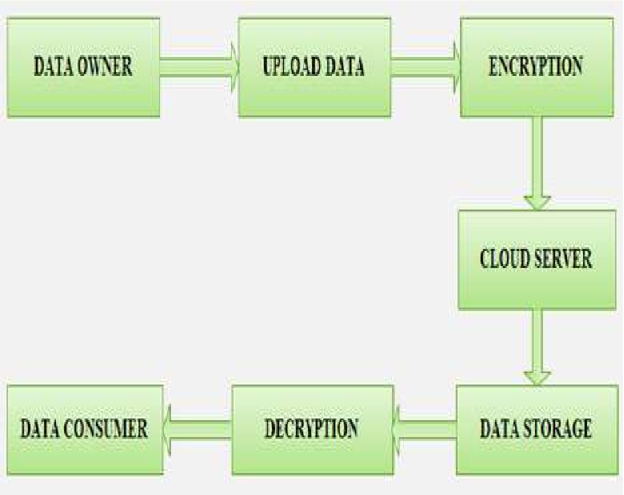
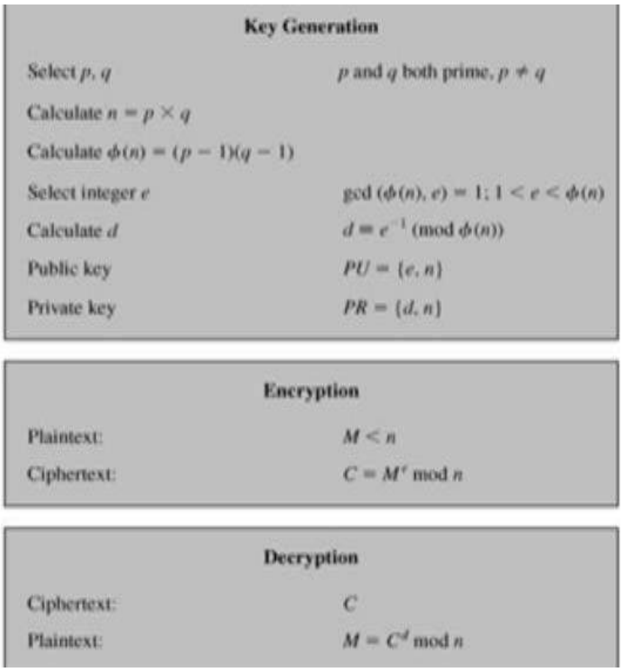


Fig. System Architecture

RSA Algorithm:



Key Generation	
Select p, q	p and q both prime, $p \neq q$
Calculate $n = p \times q$	
Calculate $\phi(n) = (p-1)(q-1)$	
Select integer e	$\gcd(\phi(n), e) = 1; 1 < e < \phi(n)$
Calculate d	$d \equiv e^{-1} \pmod{\phi(n)}$
Public key	$PU = \{e, n\}$
Private key	$PR = \{d, n\}$

Encryption	
Plaintext:	$M < n$
Ciphertext:	$C = M^e \bmod n$

Decryption	
Ciphertext:	C
Plaintext:	$M = C^d \bmod n$

SOFTWARE FEASIBILITY

FEASIBILITY STUDY:

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure plan for the project and some cost estimates. 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 TESTING

After finishing the development of any computer based system the next complicated time consuming process is system testing. During the time of testing only the development company can know that, how far the user requirements have been met out, and so on.Following are the some of the testing methods applied to this effective project:

SOURCE CODE TESTING:

This examines the logic of the system. If we are getting the output that is required by the user, then we can say that the logic is perfect.

SPECIFICATION TESTING:

We can set with, what program should do and how it should perform under various condition. This testing is a comparative study of evolution of system performance and system requirements.

MODULE LEVEL TESTING:

In this the error will be found at each individual module, it encourages the programmer to find and rectify the errors without affecting the other modules.

UNIT TESTING:

Unit testing focuses on verifying the effort on the smallest unit of software-module. The local data structure is examined to ensure that the date stored temporarily maintains its integrity during all steps in the algorithm's execution. Boundary conditions are tested to ensure that the module operates properly at boundaries established to limit or restrict processing.

INTEGRATION TESTING:

Data can be tested across an interface. One module can have an inadvertent, adverse effect on the other. Integration testing is a systematic technique for constructing a program structure while conducting tests to uncover errors associated with interring.

VALIDATION TESTING:

It begins after the integration testing is successfully assembled. Validation succeeds when the software functions in a manner that can be reasonably accepted by the client. In this the majority of the validation is done during the data entry operation where there is a maximum possibility of entering wrong data. Other validation will be performed in all process where correct details and data should be entered to get the required results.

RECOVERY TESTING:

Recovery Testing is a system that forces the software to fail in variety of ways and verifies that the recovery is properly performed. If recovery is automatic, re-initialization, and data recovery are each evaluated for correctness.

SECURITY TESTING:

Security testing attempts to verify that protection mechanism built into system will in fact protect it from improper penetration. The tester may attempt to acquire password through external clerical means, may attack the system with custom software design to break down any defenses to others, and may purposely cause errors.

PERFORMANCE TESTING:

Performance Testing is used to test runtime performance of software within the context of an integrated system. Performance test are often coupled with stress testing and require both software instrumentation.

BLACKBOX TESTING:

Black- box testing focuses on functional requirement of software. It enables to derive ets of input conditions that will fully exercise all functional requirements for a program.

Black box testing attempts to find error in the following category:

Incorrect or missing function

Interface errors

Errors in data structures or external database access and performance errors.

OUTPUT TESTING:

After performing the validation testing, the next step is output testing of the proposed system since no system would be termed as useful until it does produce the required output in the specified format. Output format is considered in two ways, the screen format and the printer format.

USER ACCEPTANCE TESTING:

User Acceptance Testing is the key factor for the success of any system. The system under consideration is tested for user acceptance by constantly keeping in touch with prospective system users at the time of developing and making changes whenever required.

CONCLUSION:

In this paper, we developed new cryptographic schemes called PISB , which provide practical immutable signatures for out-sourced databases. We also gave the first formal security assessment of immutable signatures for outsourced databases, highlighted the relationship between aggregate extraction problem and signature immutability, and schemes are much more efficient than previous immutable signatures: PISB -Generic describes a simple yet efficient way to obtain immutable then provided formal proofs for PISB schemes. We also demonstrated that PISB constructions via standard signatures that is more outsourced database systems.

REFERENCES:

[1] H. Hacigumus, B. Iyer, and S. Mehrotra, “Providing database as a service,” in Proceedings of the 18th International Conference on Data Engineering, ser. ICDE ’02, Washington, DC, USA, 2002, pp. 29–38.

[2] R. S., “Secure data outsourcing,” in Proceedings of the 33rd international conference on Very large data bases (VLDB), 2007, pp. 1431–1432.

[3] E. Mykletun, M. Narasimha, and G. Tsudik, “Authentication and integrity in outsourced databases,” Transaction on Storage (TOS), vol. 2, no. 2, pp. 107–138, 2006.

[4] A. Patel, S. J. Nirmala, and S. M. Bhanu, “Security and availability of data in the cloud,” in Advances in Computing and Information Technology, ser. Advances in Intelligent Systems and Computing. Springer Berlin Heidelberg, 2012, vol. 176, pp. 255–261.

[5] H. Wang and L. V. S. Lakshmanan, “Efficient secure query evaluation over encrypted xml databases,” in Proceedings of the 32nd international conference on Very large data bases, ser. VLDB ’06, 2006, pp. 127–138.