PRIVACY-PRESERVING PUBLIC AUDITING FOR DATA STORAGE SECURITY IN CLOUD COMPUTING

D. Pratiba, Dr. G. Shobha
Department of Computer Science and Engineering, R V College of Engineering,
Visvesvaraya technological university, Bangalore, Karnataka, India.

ABSTRACT

Cloud Computing is a model for enabling convenient on-demand network access to a shared pool of configurable computing resources that can be rapidly provisioned and released with minimal management effort or service provider interaction. By data outsourcing, users can be relieved from the burden of local data storage and maintenance. Thus, enabling public auditability for cloud data storage security is of critical importance so that users can resort to an external audit party to check the integrity of outsourced data when needed. Threshold-multisignature scheme and Third Party Auditor are used for this purpose. Third Party Auditor should be able to efficiently audit the cloud data storage without demanding the local copy of data, and introduce no additional on-line burden to the cloud user. This paper uniquely defines the fundamental properties of threshold-multisignature schemes and shows that the proposed scheme satisfies these properties and eliminates the latest attacks to which other similar schemes are subject.

General Terms-Encryption, Decryption, user, key.

Keywords-Third Party Auditor, threshold multi-signature, cloud computing.

I. INTRODUCTION

Cloud Computing has been envisioned as the next-generation architecture of IT enterprise, due to its long list of unprecedented advantages in the IT history: on-demand self-service, ubiquitous network access, location independent resource pooling, rapid resource elasticity, usage-based pricing and transference of risk [1].

441
Secure cloud computing schemes combine the properties of threshold group-oriented signature schemes and multisignature schemes[2]. Threshold multisignature schemes are also referred to as threshold signature schemes with traceability. The combined properties guarantee the signature verifier that at least ‘t’ members participated in the generation of the group-oriented signature and that the identities of the signers can be easily established.

Secure cloud computing schemes can be differentiated from threshold group signatures by the fact that by definition, the individual signers remain anonymous since it is computationally hard to derive the identities from the group signature, with the exception of the group managers. In contrast, by the above-defined traceability property of secure cloud computing schemes, the individual signers are publicly traceable and do not enjoy anonymity. Consequently, the traceability property of secure cloud computing schemes allows the individual signers to be held accountable in the public domain and renders the unlinkability property of threshold group signature schemes, as defined in, inapplicable.

II. BACKGROUND STUDIES

As a disruptive technology with profound implications, Cloud Computing is transforming the very nature of how businesses use information technology. From users’ perspective, including both individuals and IT enterprises, storing data remotely into the cloud in a flexible on-demand manner brings appealing benefits: relief of the burden for storage management, universal data access with independent geographical locations, and avoidance of capital expenditure on hardware, software, and personnel maintenances, etc [3].

As users no longer physically possess the storage of their data, traditional cryptographic primitives for the purpose of data security protection cannot be directly adopted. Thus, how to efficiently verify the correctness of outsourced cloud data without the local copy of data files becomes a big challenge for data storage security in Cloud Computing.

Considering the large size of the outsourced data and the user’s constrained resource capability, the ability to audit the correctness of the data in a cloud environment can be formidable and expensive for the cloud users [4,5].

Based on the audit result, TPA could release an audit report, which would not only help users to evaluate the risk of their subscribed cloud data services, but also be beneficial for the cloud service provider to improve their cloud based service platform. In a word, enabling public risk auditing protocols will play an important role for this nascent cloud economy to become fully established, where users will need ways to assess risk and gain trust in Cloud.

The aggregation and algebraic properties of the authenticator further benefit the design for the batch auditing.

Specifically, contribution in this work can be summarized as the following three aspects:

- Motivate the public auditing system of data storage security in Cloud Computing and provide a privacy-preserving auditing protocol, i.e., the scheme supports an external auditor to audit user’s outsourced data in the cloud without learning knowledge on the data content.
• Apparently, this scheme is the first to support scalable and efficient public auditing in the Cloud Computing. In particular, our scheme achieves batch auditing where multiple delegated auditing tasks from different users can be performed simultaneously by the TPA.
• To prove the security and justify the performance of proposed schemes through concrete experiments and comparisons with the state-of-the-art.

Therefore, to fully ensure the data security and save the cloud users’ computation resources, it is of critical importance to enable public auditability for cloud data storage so that the users may resort to a third party auditor (TPA), who has expertise and capabilities that the users do not, to audit the outsourced data when needed.

III. DESIGN

A. Design Goals

To enable privacy-preserving public auditing for cloud data storage under the aforementioned model, protocol design should achieve the following security and performance guarantee[5]:

• **Public auditability**: to allow TPA to verify the correctness of the cloud data on demand without retrieving a copy of the whole data or introducing additional on-line burden to the users.
• **Storage correctness**: to ensure that there exists no cheating cloud server that can pass the audit from TPA without indeed storing users’ data intact.
• **Privacy-preserving**: to ensure that there exists no way for TPA to derive users’ data content from the information collected during the auditing process.
• **Batch auditing**: to enable TPA with secure and efficient auditing capability to cope with multiple auditing delegations from possibly large number of different users simultaneously.
• **Lightweight**: to allow TPA to perform auditing with minimum communication and computation overhead.

B. Modules

**Sender:**
In this module there is a group with ‘N’ number of users. Each user can send a message to a user of another group or a group can send a message to another group. When sending a message from group to group, each user signs the message.

**Third Party Auditor:**
The third party is a trusted party which generates public and private keys for the users. Each time a user requests for key, a new key will be generated. Encryption and decryption of messages is also done by Third Party Auditor.

**Receiver:**
Here the receiver is either a user of a group or a complete group. The user or a group receives the decrypted message from the Third Party Auditor.
Fig 1- Architecture of cloud storage device

Fig 2- The existing system

Fig 2 shows the existing system of the implementation of effective third party auditor for secure cloud computing.

A cloud data storage service involving three different entities, as illustrated in Fig 2: the cloud user (U), who has large amount of data files to be stored in the cloud; the cloud server (CS), which is managed by cloud service provider (CSP) to provide data storage service and has significant storage space and computation resources the third party auditor (TPA), who has expertise and capabilities that cloud users do not have and is trusted to assess the cloud storage service security on behalf of the user upon request.
Users rely on the CS for cloud data storage and maintenance. They may also dynamically interact with the CS to access and update the stored data for various application purposes. The users may resort to TPA for ensuring the storage security of their outsourced data, while hoping to keep their data private from TPA. Considering the existence of a semi-trusted CS in the sense that in most of time it behaves properly and does not deviate from the prescribed protocol execution.

It's assumed that the TPA, who is in the business of auditing, is reliable and independent, and thus has no incentive to collude with either the CS or the users during the auditing process. TPA should be able to efficiently audit the cloud data storage without local copy of data and without bringing in additional on-line burden to cloud users. However, any possible leakage of user’s outsourced data towards TPA through the auditing protocol should be prohibited.

C. The Proposed Scheme

In Multisignature Scheme, two pairs of prime numbers are generated. Different pairs of keys are generated for different stages as opposed to RSA where the same key is used for encryption in various stages[6][7]. Decryption requires the sets of pair of keys to be used in the reverse order increasing security. This makes group validation also possible unlike RSA[8].

- A structured multisignature scheme is an order-sensitive multisignature
- Participating signers need to sign messages with a specified signing order
- Assume \( t \) signers \( U_1, U_2, \ldots, U_t \) in a group
- The specified signing order is \( U_1, U_2, \ldots, U_t \)
  - Follow RSA scheme
  - Select two large secret primes \( p_i \) and \( q_i \)
  - Publish product \( n_i \)
  - Determine the public key \( e_i \) and private key \( d_i \)
  - Publicly known products \( n_1, n_2, \ldots, n_t \) need to satisfy \( n_1 < n_2 < \ldots < n_t \)
  - Sign the message
    - A message \( m \)
    - \( U_1 \) computes \( S_1 = m^{e_1} \pmod{n_1} \)
    - Send \( S_1 \) to \( U_2 \)
    - \( U_2 \) computes \( S_2 = S_1^{e_2} \pmod{n_2} \)
    - Send \( S_2 \) to \( U_3 \)
    - \( U_3, U_4, \ldots, U_t \) do the similar computation
    - \( S_t \) is the order-sensitive multisignature
  - Verify the Multisignature
    - The verifier needs to reverse the signing order
    - Check \( h(m) = (((S_t \mod n_t)^{e_{t-1} \pmod{n-1}}) \ldots)^{e_1 \pmod{n_1}} \)
    - i.e., \( S_3 = C^{d_2 \pmod{n}} \)
    - \( S_2 = S_3^{d_1 \pmod{n}} \)
    - If the multisignature is signed by signers without following the specified order, the multisignature cannot be verified successfully[9][10].
To overcome the problem of Path Identification, the data and keys generated are sent to a Trusted Third Party Auditor who sends the ciphertext and keys in the required order to the receiver. Hence the data is safe and prevents hacking.

✓ **Client/Server**

A *server* is anything that has some resource that can be shared.

A *client* is simply any other entity that wants to gain access to a particular server.

✓ **Steps for Implementing a Server**
- Create a Server Socket object
  
  ```java
  ServerSocket listenSocket = new ServerSocket(portNumber);
  ```

- Create a Socket object from ServerSocket
  
  ```java
  while(someCondition) {
    Socket server = listenSocket.accept();
    doSomethingWith(server);
  }
  ```

- Create an input stream to read client input
  
  ```java
  BufferedReader in = new BufferedReader(new InputStreamReader(server.getInputStream()));
  ```

- Create an output stream that can be used to send info back to the client.
  
  ```java
  PrintWriter out = new PrintWriter(server.getOutputStream(), true);
  ```

- Do I/O with input and output Streams
- Close the socket when done
  
  ```java
  server.close().
  ```

✓ **Steps for Implementing a Client**
- Create a Socket object
  
  ```java
  Socket client = new Socket("hostname", portNumber);
  ```

- Create an output stream that can be used to send info to the Socket
  
  ```java
  PrintWriter out = new PrintWriter(client.getOutputStream(), true);
  ```

- Create an input stream to read the response from the server
  
  ```java
  BufferedReader in = new BufferedReader(new InputStreamReader(client.getInputStream()));
  ```

- Do I/O with the input and output Streams
- Close the socket when done
  
  ```java
  client.close()
  ```

**D. Selection of a Programming Language**

Java is an object oriented programming language and it was intended to serve as a new way to manage software complexity. Java refers to a number of computer software products and specifications from Sun Microsystems that together provide a system for developing application software and deploying it in a cross-platform environment. Java is used in a variety of computing platforms from embedded devices and mobile phones on the low end, to enterprise servers and supercomputers on the high end.
IV. CONCLUSIONS

This paper proposes a privacy-preserving public auditing system for data storage security in cloud computing, where the Third Party Auditor (TPA) can perform the storage auditing without demanding the local copy of data. Utilizing the homomorphic authenticator and random mask technique to guarantee that TPA would not learn any knowledge about the data content stored on the cloud server during the efficient auditing process, which not only eliminates the burden of cloud server from the tedious and possibly expensive auditing task, but also alleviates the users' fear of their outsourced data leakage. Considering TPA may concurrently handle multiple audit sessions from different users for their outsourced data files, further extending the privacy-preserving public auditing protocol into a multi-user setting, where TPA can perform the multiple auditing tasks in a batch manner, i.e., simultaneously. Extensive security and performance analysis shows that the proposed schemes are provably secure and highly efficient. It is believed that all these advantages of the proposed schemes will shed light on economies of scale for Cloud Computing.

Advantages of using clouds are unarguable, due to the opaqueness of the Cloud—as separate administrative entities, the internal operation details of cloud service providers (CSP) may not be known by cloud users—data outsourcing is also relinquishing user’s ultimate control over the fate of their data. As a result, the correctness of the data in the cloud is being put at risk due to the following reasons. First of all, although the infrastructures under the cloud are much more powerful and reliable than personal computing devices, they are still facing the broad range of both internal and external threats for data integrity. Examples of outages and security breaches of noteworthy cloud services appear from time to time. Secondly, for the benefits of their own, there do exist various motivations for cloud service providers to behave unfaithfully towards the cloud users regarding the status of their outsourced data. The message size can be increased beyond 500 characters. One can further enhance this by making it closer to the real time system. This can be implemented by allowing all the users of the group to login separately. The number of users on the sender and the receiver side can be increased.

REFERENCES


