Third Party Public Auditing For Shared Data In The Cloud Using ECC

Divya R Nair¹, Syam Gopi²

¹Research Scholar, Department of Computer Science & Engg., Amal Jyothi College of Engg., Kanjirappally, Kerala, India
²Assistant Professor, Department of Computer Science & Engg., Amal Jyothi College of Engg., Kanjirappally, Kerala, India

Corresponding Author: Divya R Nair

ABSTRACT

A Third Party Auditor audits the shared data in the cloud without any modification of the data content. To provide this security for cloud data a TPA can check the integrity of shared data. The third party auditor can be able to audit the integrity of data without accessing the entire data from the cloud. The files are divided into blocks and allow the TPA to audit the data with a specified block. If a new user wishes to join the group, TPA approves the user and added to the group. A public auditing mechanism that verifies the integrity of data stored in the cloud with the help of a digital signature, which can be aggregated to each block of data. The proposed scheme provides an efficient user revocation mechanism i.e. when a user revoked from the group, to resign the blocks that are signed by that revoked user. The proposed scheme also supports dynamic operations such as update, delete and insert operations. The TPA handles multiple auditing tasks at a time with batch auditing protocol.

Keywords: Public auditing, user revocation, cloud computing, digital signature, batch auditing.

I. INTRODUCTION

Cloud computing[1] is an on demand internet based computing that provides shared computer processing data[2] and resources to other devices or computers. Cloud computing estimates the need of administrators to manage computing resources. It allows users to pay only for used resources. There are services like private, public or hybrid cloud computing models. The private cloud provides services from a business data center to internet users. In public cloud, a third party delivers the services over the internet. Hybrid services are a combination of both private and public models.

There are so many mechanisms to provide a public auditing scheme[8] that promises the data integrity in the cloud by checking the correctness of shared data. A third party auditor[3], [6] is assigned to check the integrity of data in the cloud. The third party auditor can perform the audit task without retrieving the whole data[8]. The uploaded files are divided into a number of blocks and each block is signed by the user. If anyone can modify the block, the user needs to resign the modified block by using her private key. To promise the data confidentiality the auditor allows the auditing task by select a particular block.

One of the users in the group may misbehave or exit from the group then revokes the user from the group. The proposed system introduces a public auditing mechanism [9] for shared data in the cloud. In this scheme a third party auditor for shared data in the cloud, a digital signature is used to verify the integrity of shared data without retrieving the whole data. The system model also manages the revocation of users[12] in the group.

The proposed system overcome the disadvantages of some existing mechanisms such as to perform multiple auditing tasks at a time by using batch auditing scheme[7] and handle efficient user revocation. An elliptic curve digital signature algorithm (ECDSA)[4] is used to generate the signature of each block. The elliptic curve Diffie-Hellman algorithm is used to generate the secreted key and the data encryption standard supports the encryption and decryption process. This scheme supports dynamic operations[10] such as update, delete and insert operations. The proposed scheme is assure one, which supports an efficient user revocation[4] and handles auditing task efficiently. It also promises to reduce the auditing time of the data in the cloud.

The remainder of this work organized as follows. The system and threat model is described in section 2, and the cryptographic methods are...
explained in section 3. Section 4 describes the proposed methodology and the performance evaluation is explained in section 5. Section 6 provides the description of the related work in the field. Section 7 concludes this paper.

II. PROBLEM STATEMENT

2.1 System and Threat model

The system and threat models of the proposed system are described in this section. The proposed system model is shown in Fig. 1. To ensure the data integrity and save the cloud users from the online burden, here introduce a third party auditor to audit the data stored in the cloud when needed. The auditor can check the correctness of data on behalf of the user. The proposed scheme includes three entities such as the user, the cloud, and the third party auditor.

![Fig. 1. The Public Auditing System Model](image)

1) The User: The user is one of the main entities in the proposed system. The user first registers into the system. After the approval from the admin, log in with a valid user id and password. The user can be able to perform file operations such as upload and download the file. The user splits the files according to its size and divided into different blocks. The user signs the file blocks and sends an audit request to the auditor.

2) The Cloud Service: To provide computational resources and other services for users. The cloud sends the file detail to the auditor and shares the data between users. The cloud generates the metadata and provides services for users.

3) The TPA: A trusted service that checks the data integrity in the cloud through public auditing mechanism on behalf of the user. The TPA generates a valid signature and performs auditing tasks effectively. The audit result is sent back to the user.

The data integrity is threatened by several factors. Due to some human or system errors, the cloud provider may corrupt the data. A revoked user is another factor that affects data integrity. The revoked user may try to modify the data by an illegal way. To guarantee the data integrity, a signature is attached with each block of data. One of the users who modify a block, there need to sign the block with his/her private key. The misbehaving user must need to remove or revoke from the group and blocks signed by this revoked user are resigned by the existing user.

2.2 Design Goals

To handle the revoked user and provide data integrity in the cloud, the proposed mechanism must follow 1) Correctness: The auditor can periodically check the data integrity and provide audit results to the user. 2) Efficient user revocation: The misbehaving user must be revoked from the group and signature is recalculated. 3) Batch auditing: To audit one or more files at a time.

III. ELLIPTIC CURVE CRYPTOGRAPHY

The Elliptic Curve Cryptography [4] is an alternative mechanism for implementing public key cryptography. The equation of an elliptic curve is

\[ y^2 = x^3 + ax + b \]  

(1)

E is the elliptic curve, P is the point on the curve, and n is the maximum limit. The Elliptic Curve Cryptography is used to generate a private and public key pair.

3.1. Key generation with ECC

The ECC is used to generate both public key and private key. ECC will work in a cyclic subgroup of an elliptic curve over a finite field. The main parameters are:

1. The size of the finite field p.
2. The co-efficient a and b.
3. G is the base point of the subgroup.
4. n is the order of the subgroup.
5. h is the co-factor of the subgroup.

Thus the parameters are (p, a, b, G, n, h)

1. The private key generation: The private key is selected as a random integer (d) and chosen from (1, n-1). Thus d is the private key and n is the order of the subgroup.
2. The public key generation: The public key is denoted as H. This is calculated as:

\[ H = dG \]

Thus the generated key pair is denoted as (d, H).

3.2. The encryption with ECDH

ECDH is the Elliptic Curve Diffie-Hellman algorithm [4]. It is a variant of the Diffie-Hellman algorithm. It describes the encryption of data with generated key pair and works as:

1. Consider A and B wants to share a secret. A have the private key d_A and public key H_A. Also B have the private key d_B and public key H_B.
2. A calculate \( S = d_A H_B \) and B calculates \( S = d_B H_A \). Thus S is the same key for both A and B. If a and b have obtained the secret S, they can exchange the data with DES algorithm.
IV. PROPOSED METHOD

4.1. Public Auditing

The public auditing module includes key generation, signature generation and signature verification. In key generation phase, users generate their own private key and public key by using elliptic curve cryptography. The Elliptic Curve Cryptography is used to generate a private and public key pair. The ECC is used to generate both public key and private key. The generated key pair is denoted as \( (d, H) \).

4.1.1. ECDSA

The signature generation and verification are performed by using elliptic curve digital signature algorithm (ECDSA). The ECDSA [4] contains two phases such as signature generation process and verification process. A digital signature is used to audit the data integrity and by using this signature a user signs a block of the message. The ECDSA is the Elliptic Curve Digital Signature Algorithm and it consists mainly of two algorithms such as:

1) Signature Generation Algorithm
2) Signature Verification Algorithm

1) Signature Generation Algorithm

- Take a random integer \( k \) from \((1...n-1)\).
- The point \((x_1, y_1)\) is calculated as \((x_1, y_1) = kG\).
- Calculate \( r = x_1 \mod n \)
- If \( r = 0 \), then choose another \( k \) and repeat the steps.
- Calculate \( s = k^{-1}(z + rA) \mod n \), where \( k^{-1} \) is the multiplicative inverse of \( k \) modulo \( n \).
- If \( s = 0 \), then choose another \( k \) and repeat the steps.

Thus the generated pair \((r, s)\) is considered as signature pair.

2) Signature Verification Algorithm

To verify the signature, we need \( H_A, z \) and \((r, s)\).

The algorithm works as:

- Calculate the integer \( u_1 = s^{-1}z \mod n \) and \( u_2 = s^{-1}r \mod n \).
- Calculate the point \((x_2, y_2) = u_1G + u_2H_A\).
- Calculate \( v = x_2 \mod n \).
- If \( v = r \), then the proof is accepted.

The TPA collects the metadata and generates signatures for each block. Finally, the auditor verifies the signatures and sends the result to the user. The result contains details of both authorized and suspicious data blocks. If one of the users is revoked from the group, the blocks signed by that revoked user are considered as suspicious blocks. The auditor checks the suspicious block and included in the audit request.

4.2. User Revocation

The proposed system is efficient and secure during user revocation. When a user is revoked from the group, the block signed by the revoked user is resigned by a resigning key. Recomputed the signature on those blocks and attach the signature to each block. The revocation of the user is secure because the existing users are able to sign the data blocks. During the auditing, the blocks signed by the revoked user are considered as suspicious blocks.

4.3. Batch Auditing

The TPA must handle multiple auditing tasks [5] at a time. It reduces the auditing time of TPA. The TPA can perform auditing of different file blocks at a time [13] and reduces the computation cost of the auditor. TPA views the list of files uploaded by the user, selects all the files and performs auditing.

4.4. Data Dynamic Operation

The dynamic module supports dynamic operations [11] on file blocks. To enable each user to modify the data stored in the cloud, the dynamic operations include an insertion, deletion and update operations. After the modification of a file block, the new signature is calculated and attached to the modified block.

V. PERFORMANCE EVALUATION

5.1. Performance of Public Auditing

To perform the auditing, we need to log in to the TPA console. The user selects a file or multiple files and submits for auditing. The user submits the files such as file1.java, file2.java, and file3.java for auditing task. The blocks signed by the revoked user are considered as suspicious blocks. The blocks signed by the revoked user are considered as suspicious blocks. The result shows that the total percentage of both suspicious and verified blocks is shown in Fig. 2, Fig. 3 and Fig. 4.

![Fig. 2. Audit result for file1.java](image)

![Fig. 3. Audit result for file2.java](image)
The file1.java contains two modified blocks signed by the revoked user. The file2.java contains one modified block signed by the revoked user and the file3.java contains no modified blocks. The total number of verified and suspicious bytes is mentioned in Table I.

### Table I: Audit Result

<table>
<thead>
<tr>
<th>Filename</th>
<th>File1.java</th>
<th>File2.java</th>
<th>File3.java</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total blocks</td>
<td>4</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>Modified blocks</td>
<td>(1,2)</td>
<td>0</td>
<td>Nil</td>
</tr>
<tr>
<td>Total bytes</td>
<td>3696</td>
<td>1896</td>
<td>2904</td>
</tr>
<tr>
<td>Verified bytes</td>
<td>1640</td>
<td>848</td>
<td>1048</td>
</tr>
<tr>
<td>Suspicious bytes</td>
<td>2904</td>
<td>2904</td>
<td>0</td>
</tr>
</tbody>
</table>

5.2. Performance of Batch Auditing

The TPA can perform both individual auditing and batch auditing as per the user’s wish. The independent auditing, the auditor needs more time and communication overhead to perform auditing task. In order to reduce the time and communication overhead, TPA performs batch auditing. The experimental results show that the batch auditing helps to perform auditing task effectively.

5.3. Efficiency Analysis

Here compare the efficiency of the proposed system with the RSA-based scheme. The proposed system uses ECC for encryption and decryption. We compare the encryption and decryption process with the RSA algorithm. The comparison is described in Fig. 5. To compare the time taken for initialization, encryption and decryption process of both ECC and RSA algorithm.

The RSA takes 357 ms, 94 ms and 1888 ms for initialization, encryption, and decryption respectively. But ECC takes 16 ms, 219 ms and 172 ms for initialization, encryption, and decryption respectively. Thus the analysis shows that ECC takes less time than RSA for initialization and encryption process.

VI. RELATED WORK

A secure data sharing in clouds, SeDaSc mechanism that provides data confidentiality and integrity. In this scheme [2] encrypts a file with a single encryption key. In ensuring data storage security in Cloud Computing [11], that introduces the new security threats in the cloud and some techniques that provides data security in the cloud. The data owner submits the data, the list of the users, and the parameters to the cryptographic server. In this scheme the cryptographic server is the third party and it is responsible for key management, encryption and decryption. This scheme also handles the user inclusion and revocation in the cloud. This scheme ensures the data confidentiality but it needs secure channel for secret key exchange. The third party public auditing scheme for cloud storage [3], proposed a public auditing mechanism for data in the cloud. This scheme verifies the correctness of the cloud data with the help of a TPA, without retrieving the entire data. The proposed system consists of three entities such as the owner, the cloud server and the TPA. This ensures that no data content is leaked to TPA during the public auditing. This scheme reduces the overhead of the client. The user selects a file and splits into blocks. The TPA performs auditing of a block when a user sends an audit request. The TPA verifies the signature generated by itself with the user-generated signature and sends the result to the user. It maintains the storage correctness of data but does not support data dynamic operations.

A new scheme, An Efficient Public verifiability and Data Integrity Using Multiple TPAs in Cloud Data Storage [4], are remote data storage correctness scheme based on an elliptic curve digital signature algorithm (ECDSA) that supports the public auditing. This scheme identifies the misbehaving servers. Here proposed a main TPA and a secondary TPA for checking the data integrity. ECDSA used to generate the signature to verify the data integrity. It consists of KeyGen, SignGen, ProofGen and VerifyProof algorithms. In this scheme, if main TPA will fail to work, the secondary TPA must be able to handle the audit process. The Privacy-Preserving Public Auditing for Secure Cloud Storage [7], ensures the data integrity as well as the online burden of the user in the cloud. The TPA can periodically check the integrity of the data stored in the cloud. A privacy preserving auditing mechanism, Oruta [8] that uses a homomorphic ring signature method for signature generation and
verification. This scheme supports batch auditing task. Another scheme, Public auditing for shared data with efficient user revocation in the cloud [9] that handles the user revocation efficiently. InPrivacy-preserving public auditing for shared cloud data supporting group dynamics [10], proposed an auditing scheme that provides data privacy and supports dynamic groups in the cloud. Another auditing scheme, Efficient integrity auditing for shared data in the cloud with Secure User Revocation [12] that supports revocation of the user from the group. An Efficient Public Batch Auditing Protocol for Data Security in Multi-cloud Storage [13], to perform multiple audits task simultaneously and supports multi cloud data.

ACKNOWLEDGEMENTS

The authors would like to thank Boyang Wang, Student Member, IEEE, Baocun Li, Senior Member, IEEE, and Hui Li, Member, IEEE.

REFERENCES


