Public Auditabilityand Data Storage SecurityInCloud Computing By Using RSASS And ECCSS

PoonamPawar

Information Technology, MumbaiUniversity,

Mumbai, Maharashtra, India

Abstract--Cloud computing is Internet based technology where the users can subscribe high quality of services from data and software that resides in the remote servers. Cloud computing a brings improvement of resources utilization efficiency. One of the biggest problem in cloud computing is data storage security. The data stored at the untrusted server lacks data integrity. Thus client should need knowledge about the remote data by continuously monitoring the integrity of the stored data. Integrity problem is solved by using public auditability. This paper discusses two methods of security RSA based storage security (RSASS) method and Elliptic curve cryptosystem security scheme (ECCSS) method. Both methods are uses for public auditing of the remote data and public key cryptography techniques for providing strong security. Finally both methods compares on various parameters.

Keywords: CloudComputing,Integrity,ECCSS,RSASS.

I. INTRODUCTION

Cloud computing is extensively used technology nowadays where the resources are shared like applications, software, and business processes, infrastructure. It moves the application databases and software to the large data centers that are remotely stored in large number of computers as server. The client creates the data and stored in the remote server. The client can access or modify the data using Internet technology like web access and stored it. Cloud computing has various challenging design issues that possess security and performance concerns. One ofthe biggest concerns in security issues is data storage security. The remote data stored at the untrusted server so that lacksof data integrity.[1]

Mostly cloud storage network architecture defines three different network entities: Client (user), can be either individual clients or organizations. Cloud Storage Server (CSS), is the prover in the process of data integrity verification CSS is managed by the cloud service provider (CSP),.Third-party Auditor (TPA), independent and impartial role which has capabilities and expertise that clients do not have. In the verification process, TPA and client will be the verifier.[3]

Considering the role of the verifier in the model, all the schemes presented fall into two categories: pubic auditability and private auditability. Private auditability supports the user to verify the stored data availability and integrity. In this does not TPA induced. Public auditability has been proved effective in verification of availability and integrity. In public auditability, the client sends the generated public key to the third party auditor (TPA). The TPA monitors the stored file in the remote server and inform the client about the stored file security .[4]

The RSASS method is used for auditing the data that are stored in the remote server. This RSASS method is based on generating the large signature using RSA algorithm and posing several challenges to the server by the client for frequent integrity checking. RSASSmethod provides maximum probability detection with identification of misbehaving server. Using this method the server computation time is much reduced and we can also applied this method to variable size file blocks.[1]

The Ellipticcurve cryptosystem security scheme (ECCSS) is used for ensure the data integrity on the remote server. To make the data operations dynamicThis schemeimplements the concepts of the provable data possession (PDP). Without downloading data every time the client can audit the data on the cloud itself. By using the proof of retrievability (PoR) scheme the presence of data can be checked out by challenging the server. Homomorphic encryption is executed on the signatures for security. The client can do its operations on the cipher text itself without decrypting it with the help of this encryption. Client encrypts the data and provides the public key to the TPA. This method can be applied to file blocks of variable size and takes comparable less computation.[2]

II.RSA BASED STORAGE SECURITY SYSTEM

A) Architecture

In the RSA based Storage Security (RSASS), there are three entities for carrying out the overall process flow in the system . The client creates the data and sendsthe file data to the remote cloud server. The remote cloud service provider stores the file data in its local data store. By using the third party auditor (TPA),the client continuously monitors the stored file data in the server . The TPA is a monitoring tool which analysis the integrity of the stored file in the remote server using the RSA based signature generation algorithm and report it to the client about the status of the file data. If the file data is affected, any intrusion or attacks is notified to the client using proper message flow.

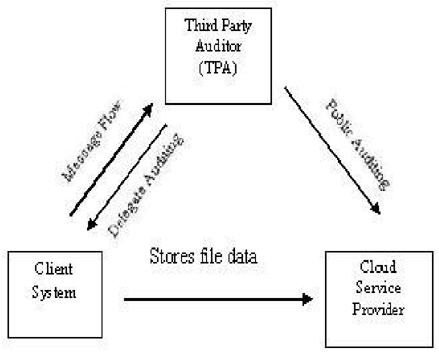

Figure 1: RSASS Data Flow Architecture [1]

B) Methodology

In the RSA based Storage Security system (RSASS),thestored file data is continuously monitored for maintaining security using RSA based signature algorithm. It is based on the concept of provable data possession (PDP) model. The PDP is a challenge and response protocol model. In this PDP model, the client using the monitoring tool, poses challenge to the cloud server and gets the proof for the challenge. The challenge is the subset of file blocks stored in the remote server and proof is the value generated for the selected subset of file blocks. The client verifies the proof that it received from the server and ensures the storage correctness of the file in the remote cloud server.[1]

C) Algorithm

RSASS scheme consists of the following four algorithms.[1]

keyGen

Input: None

Output: public key rpk, secrete key rsk, generator g.

1. Client runs random number generation in java for yielding primes p and q. N=pq

2. RSA prime numbers e and d are generated such thated mod N = 1 mod N.

Let rpk=e and rsk=d.

3. A random number g is generated from the QR_N.

sigGen

Input: File Blocks F, secret key rsk, generator g.Output: Signature set Φ.

1. Client sign each file block using secret key rsk andgenerator g as

$Ti= (H (mi).g^{mi})^{rsk}$for i Є n.

2. The completed set of signatures is grouped togetheras Φ

proofGen

Input: Subset of file blocks mi, coefficient ai

Output: Proof P

1. The prover (server) generates the tag block T, data block M, and Auxiliary Authenticate Information (AAI) for the client to generate the MHT and confirms the root R.

2. $P=\{T, M, \{H(mi), \Omega i\}_{s1 \leq i \leq sc}, sig_{rsk}(H(R))\}$

verProof

Input: Proof P

Output: Boolean value {TRUE, FALSE}

1. The verifier (client or TPA) validates the proof by generating tree using AAI.

2. If step 1 is TRUE, the verifier further validate the file blocks else emitFALSE..

III. ECC BASED STORAGE SECURITY SYSTEM

A) Architecture

The ECC based on storage security system. Mainly there are three entities the cloud service provider (CSP), the TPA and the client as shown in Fig.2. The system presents the overall process flow. The client sends the data to store to the CSP, and then CSPstores them in remote server. The TPA continuously monitors the stored file data in the server. The TPA does this by using the ECC based algorithm and informs the client about the status of the filedata. If the file data is corrupted by any means, then it is notified to the client using message flow.[5]

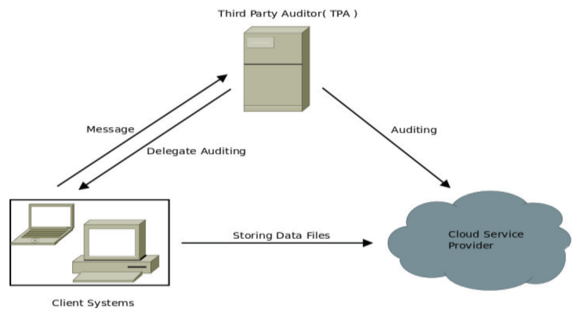

Figure 2: Data Flow Architecture[2]

B) Methodology

In ECC based storage security system the server is always under continuous security using the. In this PORapproach is implemented. The POR is a challenge and response protocol model. In this POR model, the client using the monitoring tool, poses challenge to the cloud server and gets the proof for the challenge. The challenge is the subset of file blocks stored in the remote server and proof is the value generated for the selected subset of file blocks. The client verifies the proof that it received from the server and ensures the storage correctness of the file in the remote cloud server.[2]

C)Algorithms

The ECCSS scheme consists of the following four algorithms. Key-generation (KeyGen) algorithm is used to generate the public and private key for the signature. The Signature-generation (SigGen) algorithm generates the signature for each file block. The Proof- generation(ProofGen) algorithm generates the proof for the challenge send. The last one Verify-proof (VerProof) is carried out to verify the proof generated.[2]

KeyGen

Input: None

Output: public key p, private key x and point Q.

1) The client will run the algorithm which will generate the public key p and the private keyx.

2) A point Q is chosen on the elliptic curve E (K), where K isa finite field.

3) It selects a pseudo random number x such that

$1 \leq x \leq (n_A - 1)$.

4) Point p =xQ.

5) ECC key pair is (p, x), where p is the public key, x is theprivate key.

SigGen

Input: File block F, secret key x, chosen point Q.

Output: Signature set φ.

1) A random number k is generated such that

$1 \leq k \leq (n - 1)$.

2) The hash value of blocks are calculated

$ei = H(m_i)$.

3) Computes point kQ = (x1, y1).

4) Computes $r = x_1 (\text{mod } n)$. If r = 0, go to step 1.

5) Next computes $s = k^{-1} (ei + x\ r) (\text{mod } n)$, if s = 0, go to step1.

6) The signature for a particular block s = (r, s).

7) The block tag generated for a block is $T_i=(e,k^{m}{}_{i})x$ for i Єn.

8) The signatures grouped together as φ.

ProofGen

Input: Subset of file blocks mi, co-efficient ai.

Output: Proof P.

1) The server generates T, M and AAI for the client togenerate MHT.

2) The proof P contains

$\{T, M, \{H(m_i), \Omega_i\}_{s1 \leq i \leq sc}, sig_x(H(R))\}$.

Verproof

Input: Proof P.

Output: Boolean value TRUE, FALSE.

1) The verifier validates the proof by generating the MHTusing AAI.

2) If step 1 is TRUE, then the further verification is done,else it yields FALSE.

a) Initially verification of (r, s) is done over the interval[1, n-1].

b) Next it computes $e = H(m_i)$.

c) Then computes $w = s^{-1} (\text{mod } n)$.

d) Next $u_1 = ew(\text{mod } n)$ and $u2 = rw(\text{mod } n)$ are computed.

e) Then computes $X = u_1Q + u_2P$.

f) If X = 0, then S is rejected.

Otherwise $v = x_1(\text{mod } n)$ iscomputed.

g) The proof is accepted only if v = r.

IV .PHASE

The RSASS and ECCSS contain two phases, namely the setup phase and the integrity phase.

A)Setup Phase

In the setup phase, the client generates a file F, which is a finite ordered collection of n blocksi.e.F= $\{m_1,m_2,\ldots m_n\}$. It is generates public key 'rpk' and secrete key 'rsk' using the key generation of algorithms. The overall process flow is given in fig4.1. There are five steps involved in this phase. Firstly, the client selects a random number k, such that $1 \leq k \leq (n-1)$ generates the signature (tag) for each file block using the secret key rsk and hash algorithm as $T_i= (H\ (m_i).g^{mi)\ rsk}$. Secondly, acollective signature set $\Phi= \{_{Ti}\}$ is generated which is the collection of signature of file blocks. In the third step, aMerkle Hash Tree (MHT) is built for each file block using hash algorithm. In the fourth step, the root R of the MHTconstructed is signed using the secret key as $sig_{rsk}(H(R))=H(R^{)rsk}$. Finally, the generated client advertise $\{F, \Phi, sig_{rsk}\ (H(R))\}$ to the server and then deletes Φ and sig_{rsk} (H(R)) from its local storage. The client delivers the public key to the third party auditor (TPA) for monitoring the remote files.[1][2]

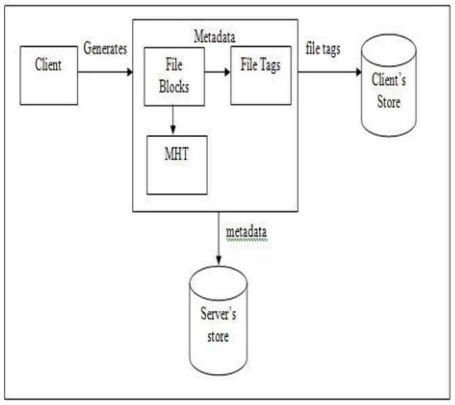

Figure 3: Preprocessing of file blocks [1][2]

B)Integrity Phase

In the integrity phase, the client (or TPA)will generates the challenge by selecting a subset of file blocks as $I = \{s_1, s_2\ldots s_c\}$ of set [1, n] such that $s1 \leq s2 \leq\ldots\leq sc$ and random coefficient $a_i=f_k\ (s_i)$ for i Є Iselected and k is the security parameter. The client sends the challenge $\{i,a_i\}$ to the server. The server generates the proof based on the challenge it receives. Theproof P contains $\{T, M, \{H(m_i), \Omega_i\ \}_{s1\ \leq\ i\leq\ sc}$, $sig_{rsk}(H(R))\}$ where $T=\Pi\ T_i^{ai}$ mod N (for $i=s_1$ to s_c) is the tag blocks, $M=\Sigma a_i m_i$ mod N (for $i=s_1$ to s_c) is the data blocks, $\{\Omega_i\}_{s1\ \leq\ i\leq\ sc}$is the auxiliary authenticate information (the node siblings from the leaves $\{h(H(m_i))\}_{s1\ \leq\ i\leq\ sc}$to the root R of the MHT). The server then sends the generatedProof P to the client. The client validates the proof by generating the root R using $\{\Omega_i, H(m_i)\}_{s1\ \leq\ i\leq\ sc}$ and authenticates by checking its secret key rsk. This is the first level of authentication. If the authentication fails, then the proof is rejected. The client verifies$T^{rpk} = g^M$.If this is true, then the signature verification will be done. The integrity verification process flow is depicted in fig 4.[1][2]

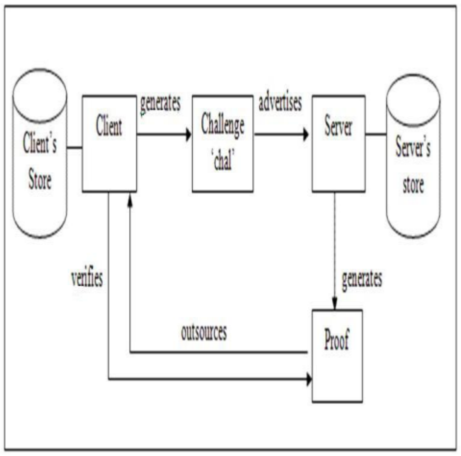

Figure 4: Integrity checking process flow [1][2]

V.COMPARISON OF ECCSS AND RSASS

Difference between ECCSSAnd RSASS are as follows.[8][9][10]

SR NO	Attribute	RSASS	ECCSS
1	Length of key	Longer key	Shorter keys
2	CPU consumption	More	Low
3	use of memory	More	Low
4	energy saving	Less	More
5	bandwidth saving	Less	More
6	Speed	Less faster	More faster
7	Key Generation	RSA grows exponentially	ECC grows linearly
8	signature verification	RSA uses very negligible time	ECC uses significant time
9	Implementation	RSA is easy	Ecc more complicate
10	Public parameters	is(n, e)	(E, P, Q)
11	Private key is	d (value s.t.ed = 1 (mod φ(n))	(value s.t aP = Q)
12	Main strength parameter is	$\|n\|$, the number of bits in n	$\|E\|$, the number of bits in the order of E

VI. CONCLUSION

The Proposed RSASS system generates the signature using RSA algorithm which supports large and different size of files and provides much security in storing the file data. This system ensures the possession of the file stored in the remote server using frequent integrity checking. This system is applicable to large public databases such as digital libraries, medical archives, etc.

The proposed ECCSS system generates its signature using the ECC algorithm and provides efficient security using key of small size. The confidentiality and the integrity of thedata are preserved by it. It ensures the possession of data on the cloud by frequent checking the remote storage servers. The security can be increase from attacks like theExceptional procedure attack, Fault analysis attack .This system is applicable to various fields.

Future Work

In future, this RSASS and ECCSS system will be incorporated in a dynamic real time applicationto have much more effective data storage security in cloud computing.

REFERENCE

[1]M.Venkatesh,M.R.Sumalatha,Mr.C.SelvaKumar," Improving Public Auditability, Data Possession in Data Storage Security forCloud Computing" Department of Information Technology, MIT, Anna University,2012 IEEE

[2] TamalKantiChakraborty, Anil Dhami, PrakharBansal and Tripti Singh," Enhanced Public Auditability & Secure Data Storage in Cloud Computing" Department of Computer Science & Engineering Motilal Nehru National Institute of Technology AllahabadAllahabad, India,2012 IEEE

[3] Long Chen, Hongbo Chen," Ensuring Dynamic Data Integrity with Public Auditability for Cloud Storage" Institute of Computer Forensics Chongqing University of Posts and Telecommunications Chongqing, China 2012 IEEE

[4] Qian Wang, Cong Wang, KuiRen, Wenjing Lou, "Enabling Public Auditability and Data Dynamics for Storage Security in Cloud Computing", IEEE Transactions on Parallel andDistributed Systems, May 2011.

[5]C.Wang, Q.Wang, K.Ren, W.Lou, "Ensuring Data Storage Security in Cloud Computing", Proc. 17th Int'l Workshop Quality of Service (IWQos '09), 2009.

[6]Anthony T. Velte,TobyJ.Velte,RobertElsenpeter, "Cloud Computing A Practical approach".

[7]Ronal L. Krutz,Russell Dean Vines, "cloud security",August 09,2010.

[8]Nicholas Jansma,BrandonArrendondo ,"Performance Comparison of Elliptic Curve and RSA Digital Signatures",April 28, 2004.

[9] Arunkumar, Dr. S.S. Tyagi, ManishaRana, NehaAggarwal, PawanBhadana,"A Comparative Study of Public Key Cryptosystem based on ECC and RSA".

[10]Kamlesh Gupta, Sanjay Silakari,"ECC over RSA for Asymmetric Encryption: A Review",www.IJCSI.org,May 2011.