Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X

IJCSMC, Vol. 4, Issue. 5, May 2015, pg.653 – 661

RESEARCH ARTICLE

Privacy Preserving Public Auditing Mechanism Using HARS For Cloud Data Sharing In Oruta

Nagasruthi J[1], Dr. Shubhangi.D.C[2]

[1]Computer Science and Engineering, VTU, India

[2]Computer Science and Engineering, VTU, India

[1] nagasruthi.rymec@gmail.com; [2] shubhangidc@yahoo.co.in

Abstract— In Cloud Computing Public auditing on the integrity of the shared data is very big issue, because the public auditing scheme will disclose private information, and identity privacy to public verifiers. This review, propose a new scheme called a privacy preserving mechanism that supports public auditing for shared data stored in the cloud. It takes the use of ring signatures to compute the verification metadata. The ring signature scheme is essential to audit the correctness of shared data. With this mechanism, the identity of the signer on each block in shared data is kept secret from public verifiers. Public verifiers are one who is able to conveniently verify shared data integrity without retrieving the entire file stored in the cloud. In addition, this mechanism is able to perform multiple auditing tasks simultaneously rather than single auditing task. This survey shows the performance and competence of our mechanism when auditing the shared data integrity.

Keywords— public auditing, privacy preserving, shared data, cloud computing.

I. INTRODUCTION

With the cloud computing and storage, users are able to access and also to share the resources provided by cloud service providers at a very low cost. It is a regular for users to take the benefit of cloud storage services to share data with others in the group, where data sharing in cloud becomes standard feature in most cloud storage services like including Drop box, I Cloud and Google Drive.

The traditional approach for checking data correctness is to retrieve the entire data from cloud and then verify the data integrity by checking the rightness of signatures (e.g., RSA) or hash values (e.g., MD5) of the entire data, similarly this approach is able to successfully check the rightness of cloud data.

The main reason is that the size of the cloud data is large in general. Downloading the entire cloud data to verify data integrity will cost and waste users amount of computation and communication assets, particularly when data has been corrupted in the cloud. Recently, many methods have been proposed to allow not only a data owner but also a TPA to competently perform integrity checking without downloading the complete data from the cloud, which is referred to as public auditing. In this mechanism, data has been divided into many small blocks, where each block is individually signed by the owner, and a random combination of all the blocks instead of the complete data is retrieved during integrity checking. A public verifier could be data user who would like to exploit the owner's data through the cloud or a third-party auditor (TPA) who can provide professional integrity examination services.

Existing public auditing mechanisms can actually be extensive to verify shared data integrity and data the freshness. Similarly, a new important privacy issue introduced in the case of shared data with the use of existing mechanisms is the leakage of identity privacy to public verifiers. To protect the private information, it is necessary and critical to preserve identity privacy from public verifiers during the public auditing.

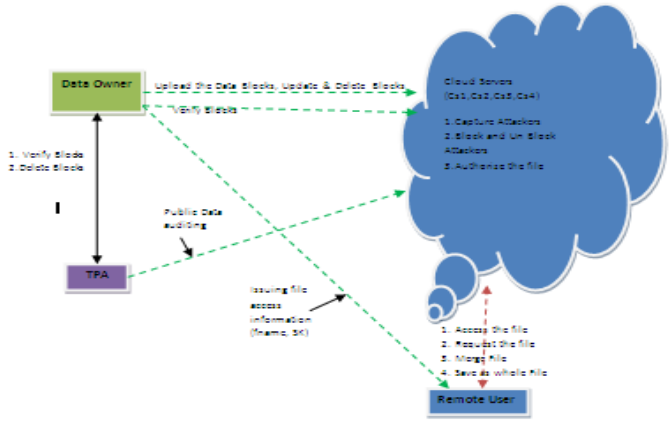
Failing to preserve identity privacy on the shared data during public auditing will disclose important and private information to public verifiers. To solve the privacy issue on the shared data, a novel privacy preserving public auditing mechanism has been proposed. Here we use ring signature to construct homomorphic authenticators in the oruta, so that the public verifier is able to verify the integrity of the shared data without retrieving the entire data and the identity of the signer on each block in shared data is kept private from the public verifier. We, extend this mechanism to support batch auditing which performs multiple auditing tasks simultaneously and improve the efficiency of verification for multiple auditing tasks. Meanwhile, Oruta is compatible with random masking; Oruta stands for One Ring to Rule Them All.

II. RELATED WORK

G. Ateniese et al worked on Provable Data Possession at Untrusted Stores which allows a verifier to test the correctness of a client's data stored at an untrusted server by utilizing RSA-based homomorphic authenticators and the sampling strategies, the verifier is able to publicly audit the integrity of data without retrieving the entire data, which is reffered as public auditing, this mechanism is only suitable for auditing the integrity of personal data. Ensuring Data Storage Security in Cloud Computing by C. Wang has used leverage homomorphic tokens to ensure the correctness of erasure codes-based data distributed on multiple servers, the major contribution of this mechanism is able to support dynamic data, identify misbehaved servers.[3][4] Invented Remote Data Checking for Network Coding-Based Distributed Storage for auditing the correctness of data under multiple server scenarios, where these data are encoded by network coding instead of using erasure codes, this scheme minimizes communication overhead in the phase of data repair. Scalable and Efficient Provable Data Possession to Support Dynamic Data presents an efficient PDP mechanism based on symmetric keys, it supports update and delete operations on data, and it exploits symmetric keys to verify the integrity of data [6]. A. Juels et al worked on PORs: Proofs of Retrievability for Large Files it provides POR's scheme which is able to check the correctness of data in an untrusted server, the original file is added with a set of randomly-valued check blocks called the sentinels, the verifier challenges untrusted server by specifying the positions of a collection of sentinels and asking the untrusted server to return associated sentinel values, Sentinel based POR protocol is amenable to the real world application. [7] To solve the privacy issues on the shared data, we introduce a scheme called Oruta in a cloud computing scenario, Oruta is a privacy preserving public auditing mechanism, which uses the ring signatures used to construct the homomorphic authenticators in Oruta scheme, where a public verifier is able to verify the integrity correctness of shared data without retrieving the original data fully, in Oruta the identity of the signer on each block in shared data is kept private/secret from the public verifier, where it support the batch auditing to perform multiple auditing tasks simultaneously instead of doing single tasks, to improve the efficiency of the verification methods.

III.METHODOLOGY

A. Architecture



Architecture involves three parties, the cloud server, a group of users and a TPA. There are two types of users in a group, the original user and a number of groups of users. The original user annually creates shared data in the cloud, and shares it with group of users. Both the original user and group users are members of group. Every members of group is grant to access and modify the shared data. Shared data and its verification metadata are both stored in the cloud server. A TPA, such as a Third Part Auditor providing expert data auditing services or a data user outside the group attempt to utilize shared data, is able to publicly verify the integrity of shared data stored in the cloud server.

When TPA intended to check the integrity of shared data, it sends an auditing challenge to the cloud server. After receiving the auditing challenge, the cloud server responds to the TPA with an auditing proof of the tenancy of the shared data. Then, this TPA checks the exactness of the entire data by verifying the exactness of auditing proof. Essentially, the process of public auditing is a challenge and response protocol between a TPA and the cloud server. The identity of the signer on each block in the shared data is kept secret from the third part auditor.

B. Ring Signature Scheme

The design of the new homomorphic authenticable ring signature (HARS) scheme, which is extended from the standard ring signature scheme. The ring signature is developed by HARS mechanism are not only able to preserve identity privacy but also able to support block less verifiability.

HARS contains three algorithms: KeyGen, RingSign and Ring Verify.

- KeyGen: Each user in the group develops their private key and public key.
- RingSign: A user in a group is able to develop a signature on a block and its block identifier with his/her private key and all group members' public keys. A block identifier is a string that can characterize the comparable block from others.
- Ring Verify: A verifier is able to check whether a given block is signed by a group of member.

C. Public Auditing Mechanism

Using the HARS and its properties, we construct Oruta, a privacy preserving public auditing mechanism for shared data in the cloud. With oruta, the public verifier can verify integrity of shared data without retrieving the entire data. Meanwhile, the identity of the signer on each block in shared data is kept private from public verifier during the auditing.

We present the details of our public auditing mechanism. It includes five algorithms: KeyGen, SigGen, Modify, ProofGen, and ProofVerify. In KeyGen, users develop their own public/private key pairs. In SignGen, a user is able to compute ring signatures and blocks in shared data by using its own

private key and all group members' public keys. Each user in the group is able to perform an insert, delete or update operation on a block, and compute the new ring signature on this new block in Modify. ProofGen is operated by a public verifier and the cloud server together to indicatively generate a proof of possession of shared data. In ProofVerify, the public verifier audits the integrity of shared data by verifying the proof.

Now, we discuss security properties of HARS, including exactness, affectability, and block less verifiability, normal liability and identity privacy

Given any block m, its block identifier id, and its ring signature $\sigma = (\sigma_{1,\ldots\ldots}\sigma_2)$, *a* verifier is able to correctly check the integrity of this block under HARS.

Based on properties of bilinear maps, the correctness of this scheme can be proved as fallows.

Let G_1, G_2 and $G_T$be multiplicative cyclic groups of order p, $g_1\ and g_2$ be a generator of $G_1\ and\ G_2$ respectively.

A bilinear map e is a mape $= G_1 \times G_2 \rightarrow G_T$, it remain an efficiently computable algorithm for computing map e, for all$u \in G_1, V \in G_2\ and\ a, b \in Z_p$,$e(u^a, v^b) = e(u, v)^{ab}$, and φ: $G_{12} \times G_1$be a computable homomorphism with$\varphi(g_2) = g_1$

For a user u_i, he/she randomly picks $x_i \leftarrow z_p$ and computes$\omega_i = {g_2}^{x_i} \in G_2$. Then, user$u_i$ public key is $pk_i = \omega_i$ and his/her private key is $sk_i = x_i$.

$$\prod_{i=1}^{d} e(\sigma_i, \omega_i) = e(\sigma_s, \omega_s).\prod_{i\neq s} e(\sigma_i, \omega_i)$$

$$= e\left(\left(\frac{\beta}{\varphi(\prod_{i\neq s}{\omega_i}^{a_i})}\right)^{\frac{1}{x_s}}, {g_2}^{x_s}\right).\prod_{i\neq s} e({g_1}^{a_i}, {g_2}^{x_i})$$

$$= e\left(\frac{\beta}{\varphi\left(\prod_{i\neq s}{g_2}^{x_i a_i}\right)}, g_2\right).\prod_{i\neq s} e({g_1}^{a_i x_i}, g_2)$$

$$= e\left(\frac{\beta}{\prod_{i\neq s}{g_2}^{a_i x_i}}.g_2\right).e(\prod_{i\neq s}{g_1}^{a_i x_i}, g_2)$$

$$= e\left(\frac{\beta}{\prod_{i\neq s}{g_1}^{a_i x_i}}.\prod_{i\neq s}{g_1}^{a_i x_i}, g_2\right)$$

$$= e(\beta, g_2)$$

Where β is computed as $\beta = H_1(id){g_1}^m$ (1)

And sets,

$$\sigma_s = \left(\frac{\beta}{\varphi(\prod_{i\neq s}{\omega_i}^{a_i})}\right)^{\frac{1}{x_s}} \in G_1 \quad (2)$$

The ring signature of block m is $\sigma = (\sigma_1, \ldots\ldots, \sigma_d) \epsilon {G_1}^d$.

Given all the d users public keys$(pk_1, \ldots., pk_d) = (w_1, \ldots\ldots, w_d)$, a block m, an identifier id and a ring signature $\sigma = (\sigma_{1,\ldots\ldots,}\sigma_d)$, a verifier first computes $\beta = H_1(id){g_1}^m \in G_1$and then checks

$$e(\beta, g_2) = \prod_{i=1}^{d} e(\sigma_i, \omega_i) \quad (3)$$

If the above equation holds, then the given block m is signed by one of these d users in the group.

Now, we examine the security properties of Oruta, including its correctness, unforgeability, identity privacy and data privacy. A public verifier is able to correctly audit the integrity of shared data under oruta.

According to the characterization of ProofVerify, a public verifier will conclude the integrity of shared data is correct if equation 6 holds. So, the correctness of our scheme can be proved by verifying the correctness of equation. Based on the properties of bilinear maps, the right hand side of RHS of equation 6 can be expanded as fallows.

$$\text{RHS} = \left(\prod_{i=1}^{d} e\left(\prod_{j\in J} \sigma_{j,i}{}^{\omega_i}, \omega_i\right)\right).e\left(\prod_{l=1}^{k} \lambda_l{}^{h(\lambda_l)}, g_2\right)$$

$$= \left(\prod_{j\in J}\left(\prod_{i=1}^{d} e(\sigma_{j,i}, \omega)^{y_j}\right)\right).e\left(\prod_{l=1}^{k} \eta_l{}^{t_l{}^{h(\lambda_l)}}, g_2\right)$$

$$= \left(\prod_{j\in J} e(\beta_j, g_2)^{y_j}\right).e\left(\prod_{l=1}^{k} \eta_l{}^{t_l{}^{h(\lambda_l)}}.g_2\right)$$

$$= e\left(\prod_{j\in J}\left(H_1(id_j)\prod_{l=1}^{k} \eta_l{}^{m_{j,l}}\right)^{y_j}, g_2\right).e\left(\prod_{l=1}^{k} \eta_l{}^{t_l{}^{h(\lambda_l)}}, g_2\right)$$

$$= e\left(\prod_{j\in J} H_1(id_j)^{y_j}.\prod_{l=1}^{k} \eta_l{}^{m_{j,l}y_j}.\prod_{l=1}^{k} \eta_l{}^{t_l{}^{h(\lambda_l)}}, g_2\right)$$

$$= e\left(\prod_{j\in J} H_1(id_j)^{y_j}.\prod_{l=1}^{k} \eta_l{}^{\mu_l}.g_2\right)$$

Given all the d group members public keys$(pK_1, \ldots, pK_d) = (\omega_1, \ldots, \omega_d)$, a block $m_j = (m_{j,1}, \ldots, m_{j,k})$, its identifiers$id_j$, a private key $sk_s$for some s, user u_s computes a ring signature of this block as fallows.

Aggregate block m_j with the public aggregate key pak, and computes

$$\beta_{j=H_1(id_j)}\prod_{l=1}^{k} \eta_l{}^{m_{j,l}} \in G_1 \quad (4)$$

Random choose $a_{j,i} \in Z_p$ and sets $\sigma_{j,i} = g_1{}^{a_{j,i}}$ for all $i \neq s$. then, calculates

$$\sigma_{j,s} = \left(\frac{\beta_j}{\varphi\left(\prod_{i\neq s} \omega_i{}^{a_{j,i}}\right)}\right)^{1/x_s} \in G_1 \quad (5)$$

The ring signature of block m_j is $\sigma_j = (\sigma_{j,1}, \ldots\ldots, \sigma_{j,d})$

The public verifier analyze the definiteness of this data by analyzing the following equation

$$e\left(\prod_{j\in J} H_1(id_j)^{y_j}\prod_{l=1}^{k} \eta_l{}^{m_l}, g_2\right) \stackrel{?}{=} \left(\prod_{i=1}^{d} e(\emptyset_i, \omega_i)\right).e\left(\prod_{l=1}^{k} \lambda_l{}^{h(\lambda_l)}, g_2\right) \quad (6)$$

If the above query holds, then the public verifier terminate that the block in shared data are all correct. Otherwise, the integrity of shared data is incorrect.

D. Batch Auditing

Frequently, a public verifier may need to verify the exactness of multiple auditing tasks in a very short time. Precisely verifying these multiple auditing tasks independently would be disorganized.

By leveraging the properties of bilinear maps, we can further extend Oruta to support batch auditing, which can verify the definiteness of multiple auditing tasks concurrently and enhance the efficiency of public auditing.

Subsequently receiving all the B auditing proofs, the public verifier analyze the inaccuracy of these B proofs subsequently by analyzing the fallowing query with all

$$e\left(\prod_{b=1}^{B}\left(\prod_{j\in J}H(id_{b,j})^{y_j}.\prod_{l=1}^{k}\eta_{b,l}{}^{\mu_{b,l}}\right),g_2\right)$$

$$\stackrel{?}{=}\left(\prod_{b=1}^{B}\prod_{l=1}^{d_b}e(\emptyset_{b,i},\omega_{b,i})\right).e\left(\prod_{b=1}^{B}\prod_{l=1}^{K}\lambda_{b,l}{}^{h(\lambda_{b,l})},g_2\right) \quad (7)$$

Where $pK_{b,i} = w_{b,i} = g^{x_{b,i}}$ and $sK_{b,i} = x_{b,i.}$

If the above verification equation holds, then the public verifier believes that the integrity of all the B shared data is correct. Otherwise, there is at least one shared data is corrupted.

Based on the correctness of equation (6), the correctness of batch auditing in equation (7) can be presented as

$$\left(\prod_{b=1}^{B}\prod_{l=1}^{d_b}e(\emptyset_{b,i},\omega_{b,i})\right).e\left(\prod_{b=1}^{B}\prod_{l=1}^{K}\lambda_{b,l}{}^{h(\lambda_{b,l})},g_2\right)$$

.

$$=\prod_{b=1}^{B}\left(\left(\prod_{l=1}^{d_b}e(\emptyset_{b,i},\omega_{b,i})\right).e\left(\prod_{l=1}^{K}\lambda_{b,l}{}^{h(\lambda_{b,l})},g_2\right)\right)$$

$$=\prod_{b=1}^{B}e\left(\prod_{j\in J}H((id_{b,j})^{y_j}).\prod_{l=1}^{k}\eta_{b,l}{}^{\mu_{b,l}},g_2\right)$$

$$=e\left(\prod_{b=1}^{B}\left(\prod_{j\in J}H(id_{b,j})^{y_j}.\prod_{l=1}^{k}\eta_{b,l}{}^{\mu_{b,l}}\right),g_2\right)$$

If all the B shared data are from the same group, the public verifier can further improve the efficiency of batch auditing by verifying

$$e\left(\prod_{b=1}^{B}\left(\prod_{j\in J}H(id_{b,j})^{y_j}.\prod_{l=1}^{k}\eta_{b,l}{}^{\mu_{b,l}}\right),g_2\right)$$

$$\stackrel{?}{=} \left(\prod_{b=1}^{B} \prod_{l=1}^{d_b} e(\emptyset_{b,i}, \omega_{b,i}) \right) . e\left(\prod_{b=1}^{B} \prod_{l=1}^{K} {\lambda_{b,l}}^{h(\lambda_{b,l})}, g_2 \right) \quad (8)$$

This can save the public verifier about (d-1) B pairing operations in total compared to equation (7).

IV. RESULT AND DISCUSSION

In this system, we propose Oruta, a privacy preserving public auditing mechanism for shared data in the cloud. We use ring signatures to construct homomorphic authenticators, so that a TPA is able to audit shared data integrity without retrieving entire data, where it cannot differentiate who is the signer on each block.

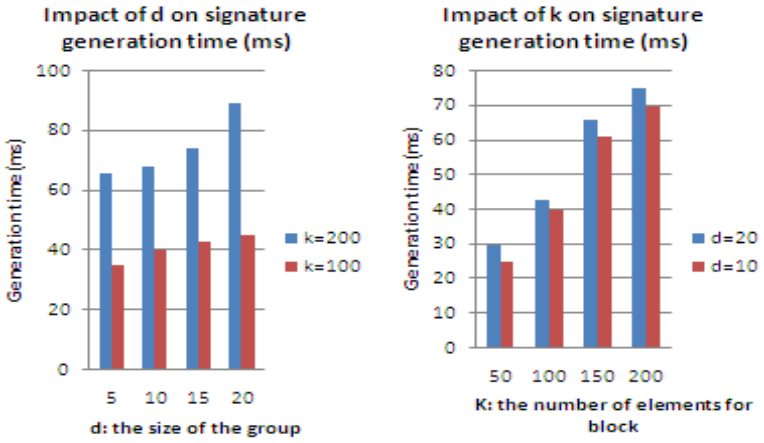


Fig.1. Performance of signature generation

According to the generation time of ring signature on a block is resolved by the number of users in the group and the number of elements in each block. From the above figure, when k is fixed, the generation time of ring signature is generally increasing with the size of group, when d is fixed; the generation time of ring signature is linearly increasing with the number of elements in each block. Especially, when d=10 and k=100 a user in the group requires about 37 milliseconds to compute a ring signature on a block in shared data.

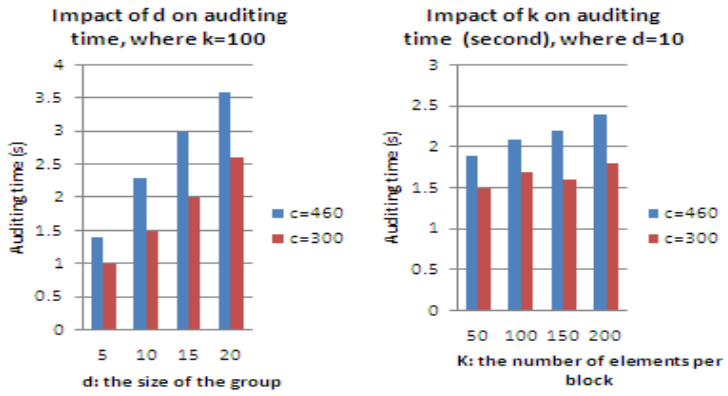


Fig.2. Performance of auditing time

Based on our preceding analysis, the auditing performance of oruta under different detection probabilities is illustrated in the figure 2, 3 the auditing time is linearly increasing with the size of the group. When c=300, if two users sharing the data in the cloud, the auditing time is only about 0.5 seconds; when the number of group members increases to 20, it takes 2.5 seconds to finish the same auditing task.

The communication cost of an auditing task under different parameters is presented in fig 3. Compared to the size of entire shared data, the communication cost that a public verifier consumes in an auditing task is very small. It is clear in the table that when maintaining a higher detection probability, a public verifier needs to absorb more computation and communication overhead to finish the auditing task. Specifically when c=300, it takes a public verifier 1.32 seconds to audit the correctness of shared data, when the size of the shared data is 2GB; when c=460, a public verifier needs 1.94 seconds to verify the integrity of the same shared data.

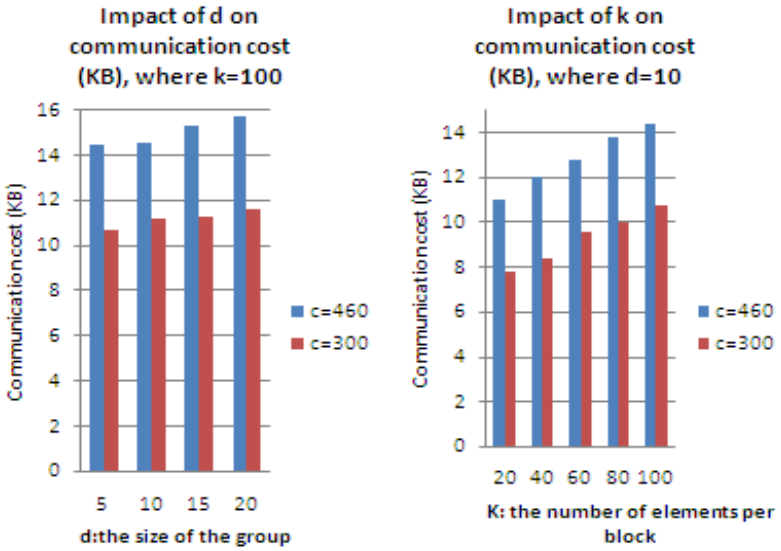


Fig.3. Performance of communication cost

To solve the privacy issue on shared data, we propose Oruta, a novel privacy preserving public auditing mechanism, we utilize ring signature to construct homomorphic authenticators in Oruta, so that a public verifier is able to verify the integrity of the shared data without retreiving the entire data,while the identity of the signer on each block in the shared data is kept private from the public verifier. Where, oruta is compatable with random masking which has been utilized in WWRL and can preserve data privacy and public verifiers.

V. CONCLUSION

We propose Oruta, a privacy preserving public auditing mechanism for shared data in the cloud by using ring signatures to construct homomorphic authenticators, so that a TPA is able to audit shared data integrity without retrieving entire data. Compared to the previous work this mechanism avoids high decoding computation cost for the data users and save computation resources for online data owners during data repair. We extend our mechanism to support batch auditing to efficiently audit the integrity of shared data with dynamic groups while still preserving the identity of the signer on each block from the third party auditor.

ACKNOWLEDGEMENT

I would like to thank my guide Dr.Shubhangi.D.C for assisting me in this paper work.

REFERENCES

[1] G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Peterson, and D. Song, "Provable Data Possession at Untrusted Stores", proc. 14th ACM Conf.Computer and Comm.Secutity (CCS'07), pp. 598-610, 2007.

[2] 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), pp. 1-9, 2009.

[3] B. Chen, R. Curtmola, G. Ateniese, and R. Burns, " Remote Data Checking for Network Coding based Distributed Storage System", proc. ACM Workshop Cloud Computing Security Workshop (CCSW'10), PP.31-42, 2010.

[4] B. Wang, B. Li, H. Li, "Certificates Public Auditing for Data Integrity in the Cloud", proc. IEEE Conf. Comm. And Network Security (CNS'13), pp. 276-284, 2013.

[5] A. Juels and B.S. Kaliski, "PORs: Proofs of Retrievability for Large Files", proc. 14th ACM Conf. Computer and Comm. Security (CCS'07), pp. 584-597, 2007.

[6] G. Ateniese, R.D. Pietro, L.V. Mancini, and G. Tsudik, "Scalable and Efficientoicat Provable Data Possession", proc. Fourth Int'l Conf.Security and privacy in Comm.Networks (SecureComm'08), 2008.

[7] B. Wang, B. Li, and H. Li, "Oruta: Privacy Preserving Public Auditing for Shared Data in the Cloud", proc. IEEE Fifth Int'l Conf.Cloud Computing, pp. 295-302, 2012.

[8] M. Armburst, A. Fox, R. Grifith, A.Joseph, R.H. Katz, A. Konwinski, G. Lee, D.A. Patterson, A. Rabkhin, I. Stoica, and M. Zaharia, " A View of Cloud ComputiM, vol.53, no.4, pp. 50-58, ang", Comm. ACM, vol. 53, no. 4, pp. 50-58, Apr. 2010.

[9] K. Ren, C.wang, and Q. Wang, "Security Challenges for the Public Cloud", IEEE Internet Computing, vol.16, no.1, pp.69-73,2012.

[10] D.Song, E. Shi, I. Fischer, and U. Shankar, "Cloud Data Protection for the Masses", Computer vol. 45, no. 1, pp. 39-45, 2012.

[11] C. Wang, Q. Wang, K. Ren, and W. Lou, "Privacy-Preserving Public Auditing for Shared Data in the cloud", proc. IEEE INFOCOM, pp.525-533, 2010.

[12] B.Wang, M. Li, S.S. Chow, and H. Li, "Computing Encrypted Cloud Data Efficiency under Multiple Keys", Proc. IEEE Conf.Comm and Network Security (CNS'13), pp. 90-99,2013

[13] R. Rivest, A. Shamir, and L.Adleman, "A Method for Obtaining Digital Signatures and Public Key Cryptosystems", Comm. ACM, vol. 21, no.2, pp. 120-126, 1978.

[14] C. Erway, A. Kupsu, C. Papamanthou, and R. Tamassia,"Dynamic Provbale Data Possession", proc. 16th ACM Conf. Computer and Comm.Security (CCS'09), pp.213-222, 2009.

[15] B. Wang, S.S. Chow, M. Li, and H. Li, "Storing Shared Data on the Cloud via Security-Mediator," Proc. IEEE 33rd Int'l Conf. Distributed Computing Systems (ICDCS'13), pp.124-133, 2013.

[16] D. Bonesh, X. Boyen, and H. Shacham, "Short Group Signatures", Proc. 24th Ann. Int'l Cryptology Conf. (CRYPTO'04), pp.41-55, 2004.

[17] B. Wang, B. Li, and H. Li, "Knox: Privacy-Preserving Auditing for Shared Data With Large Groups in the Cloud", Proc. 10th Int'l Conf. Applied Cryptography and Network Security (ACNS'12), pp. 507-525, June 2012.

[18] E. Brickell, J. Camenisch, and L. Chen, "Direct Anonymous Attestation", proc. 11th ACM Conf. Computer and Comm. Security (CCS'04), pp. 132-145, 2004.

[19] D. Bonesh, B. Lynn, and H. Shacham, "Short Signatures from the Weil Pairing", Proc seventh Int'l Conf. Theory and Applications of Cryptology and Information Security: Advances in Cryptology (ASIACRYPT'01), pp. 552-565, 2001.

[20] R.L. Rivest, A. Shamir, and Y.Tauman, " How to Leak a Secret", Proc. Seventh Int'l Conf. Theory and Applications Of Cryptology and Information Security: Advances in Cryptology (ASIACRYPT'01), pp. 552-565,2001.

[21] Q. Wang, C. Wang, J. Li, K. Ren, and W. Lou, "Enabling Public Verifiability and Data Dynamic for Storage Security in Cloud Computing, "proc. 14th European Conf. Research in Computer Security (ESORICS'09), pp.355-370, 2009.

[22] B. Wang, S.S. Chow, M. Li, and H. Li, "Storing Shared Data on the Cloud via Security-Mediator," Proc. IEEE 33rd Int'l Conf. Distributed Computing Systems (ICDCS'13), pp.124-133, 2013.

[23] D. Bonesh, X. Boyen, and H. Shacham, "Short Group Signatures", Proc. 24th Ann. Int'l Cryptology Conf. (CRYPTO'04), pp.41-55, 2004.

