Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 125 (2018) 698–708

Procedia Computer Science

www.elsevier.com/locate/procedia

6th International Conference on Smart Computing and Communications, ICSCC 2017, 7-8 December 2017, Kurukshetra, India

Public integrity auditing for shared dynamic cloud data

Shubham Singh[a], Surmila Thokchom[b]

[a]*Dept. of Computer Science and Engineering, NIT Meghalaya, Shillong-793003, India*
[b]*Dept. of Computer Science and Engineering, NIT Meghalaya, Shillong-793003, India*

Abstract

Cloud computing is an important storage platform being researched nowadays. It provides various services to its users. Among them, one of the salient service offered is cloud storage which makes data outsourcing a rising trend. But the major concern associated is the integrity and seclusion of outsourced data. Users require their data to be secure from any modification or unauthorized access. Therefore someway to verify whether the data is intact or not, without retrieving, should exist. This boosts the need of secure remote data auditing. This paper proposes an auditing scheme based on vector commitment, identity based ring signature and group key agreement protocol, emanated on bilinear pairing. An experimental analysis of the proposed scheme, later in the end, shows that when compared with its pertinent schemes, the proposed scheme is also efficient.

Peer-review under responsibility of the scientific committee of the 6th International Conference on Smart Computing and Communications

Keywords: Cloud computing; public integrity auditing; vector commitment; ring signature; group key agreement protocol.

1. Introduction

Cloud computing can be defined as a network based computing which provides shared processing data & resources to its user when required. As per NIST, it is a model which enables pervasive, on-demand access to resources being shared that can be rapidly provisioned & released with minimum management effort [29]. It provides a user various capabilities for storing their data at the cloud server & processing it when required. This motivates organizations to outsource their data to an external storage server & improve the storage constraints of local devices, and enables them

[a] Shubham Singh. Tel.: +91-9027627808.
E-mail address: ShubhamSinghCMR@nitm.ac.in

[b] Surmila Thokchom. Tel.: +91-9458177016.
E-mail address: Surmila.Thokchom@nitm.ac.in

1877-0509
Peer-review under responsibility of the scientific committee of the 6th International Conference on Smart Computing and Communications
10.1016/j.procs.2017.12.090

to have more focus on their core competencies. Cloud user can access the data anytime, when required, from any part of the world but in some cases like hardware failure etc. the server may return an invalid result. Thus data integrity becomes the biggest concern for the cloud users as they no longer have a physical control over their outsourced data. Therefore to assure whether the data is secure or not, a way to verify its integrity and accessibility must exist for the users.

Some solutions have been put forward for assuring the integrity & availability of data stored at a faraway server. Authors in the references [2], [28], [10], [24], [25], [35], [37] and [32] proposes dynamic scheme which focuses on instances where only the owner can modify the stored data. Applications [19], [20] and [21], where cloud assistance is used as a cooperation platform, multiple group users shares the code and can access, revise and run it anytime anywhere. Such type of collaborative network makes remote auditing schemes impracticable where only the data owner can modify the data. It will cause a lot of computation and communication overhead to data owner and are inappropriate for him. If integrity verification can be done by person other than data owner, that is, by third party auditor, then the scheme is publicly verifiable. The scheme in [38] designs polynomial authentication tags and uses proxy tag update technique to support public auditing but the data confidentiality of group users is not considered. This means that the scheme supports data update & integrity checking for plaintext, not for cipher text. Yet no solution addresses the issue of public integrity checking with group user modification.

The dearth of these schemes prompts us to propose a reliable as well as an efficient way for remote data auditing. To the end, a construction is proposed which applies vector commitment over the database and supports group data encryption & decryption during its modification using group key agreement protocol [36]. Identity based ring signatures [39] are used to protect the anonymity of the signer.

1.1. Our Contribution

This paper put forwards a method for efficient public integrity auditing which is based on the scheme in the reference [27]. The issues of public integrity auditing were further studied and vector commitment, from the existing scheme [27], is incorporated with identity based ring signature to put forward an efficient public data integrity checking scheme. And in the end, a performance evaluation of the proposed scheme shows that it is more efficient than the existing scheme.

2. Related Work

Availability, integrity & confidentiality are the key attributes of data stored at cloud server. Serious work has been done looking for a way to securely outsource local data to faraway storage server and its remote auditing. In the papers [2] & [28], homomorphic authentication scheme is used by the authors to decrease the communication and computation cost. Later, deviants of these schemes are drafted to refine their efficiency like allowing public data auditing [35], [34], [37] and data update [24], [25].In the ref. [4] a scheme is proposed that supports user revocation. But it is constructed on the conjecture that there exists no collusion, neither it occurs, between the revoked user and the cloud server. It assumes that an exclusive authenticated channel exist between entities. Yuan & Yu proposed a dynamic auditing scheme in [38] which uses proxy tag update method and is built on polynomial authentication tags but doesnt considers ciphertext store. Benabbas et al. [8] proposes a verifiable database scheme but the problem is that public verifiability is not supported in it.

Authors puts forward a new way in the ref. [16] to build a database that is verifiable using vector commitment that supports public verifiability. This scheme assumes outsourced database to be of fixed size with the client having the ability to obtain information about the outsourcing function beforehand. Backes et al. presents a scheme in [5] having properties which eliminates this assumption. Group signature was first introduced in the ref. [17]. This signing method allows each member of the group to have a secret key and sign a message. Only the group manager knows the identity of actual signer, who is a trusted entity, and thus, in this way, this form of signing confirms signers anonymity. This signature type has been studied in the references [18] [15] [14] [3] [13] [33]. The authors have proposed a group signature which supports verifier local revocation in the paper [9]. In the proposed scheme, revocation information is sent to the signature verifier in user revocation. However, an initialization procedure is needed to specify a group in this scheme which may not be feasible under some conditions.

The idea of ring signature was introduced in the ref. [31]. The proposed scheme in [31] is built upon RSA public key [30] & its security is analysed in ideal cipher model [6]. The paper [11] improves this scheme and presents a ring signature whose security is analysed in the random oracle model [7].

3. Problem formulation

This section details the architecture of proposed system & its design, while taking into consideration the threat model and security goals of the existing scheme in ref. [27].

3.1. System architecture

Proposed scheme consists of three entities: Group users, Cloud storage server and Third Party Auditor (TPA), as shown in the Fig. 1.

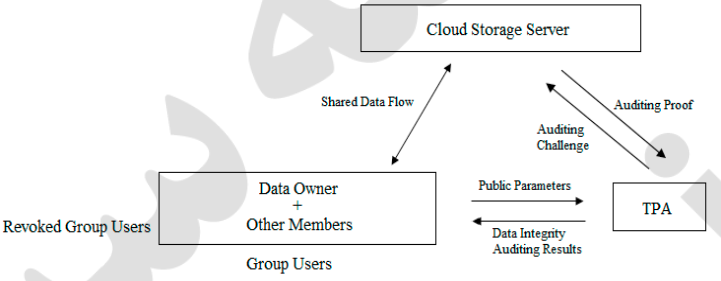

Fig. 1: System architecture

- Group users: They consist of data owners & other members of the group with whom the data owner can share his stored data along with privileges to access, modify etc. it.
- Cloud storage server: It is responsible for providing storage services to the cloud user. It is semi-trusted and hence, attacks are possible.
- Third Party Auditor: It is liable for verifying the integrity of stored data upon request. It audits the file and sends back the result.

3.2. System design

- Data owner: If a cloud user wants to share some data amongst a number of users then he will be acting as a data owner. Data owner may share his stored data among the members of an existing group or can create a new group which will include only those members with whom he wants to share the data. If he do not want to share some of his data with any member, he can easily do so. Moreover, he has the privilege to secure his data by encrypting it before uploading on the cloud. Fig. 2 showcases the jobs a dataowner can do.

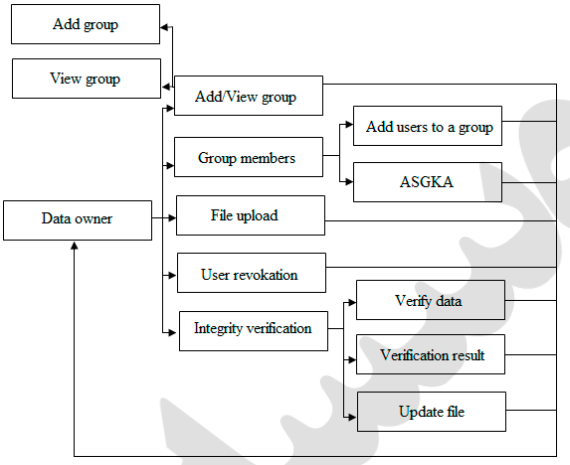

Fig. 2: Data owner

- Cloud server: Cloud server is the remote storage server for the users data. All the data uploaded by the data owner is stored at this storage server. Since it is semi trusted, it is possible that the cloud may try to gain access to the stored data as shown in the Fig. 3. Such an attempt by the cloud server is unacceptable and this type of unauthorized access to stored data with the purpose to view or modify it, leaves data owners data vulnerable. Thus to protect the revelation of data if such an attempt is made by the cloud server, data should be encrypted before being stored at a remote location which adds a level of security to the stored data.

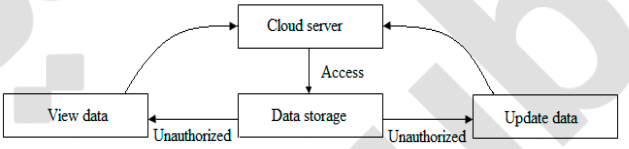

Fig. 3: Cloud server

- Third Party Auditor: It is liable for verifying data integrity when requested by a member of the group as shown in Fig. 4. Whenever a request to verify integrity comes, it generates a challenge & sends it to cloud server. TPA upon getting a response from the cloud performs the auditing task & sends the result back to the user.

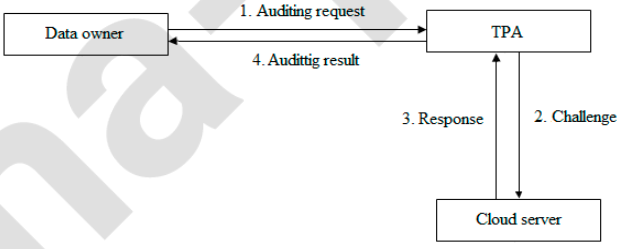

Fig. 4: Third Party Auditor

4. Preliminaries

This section reviews all the primitives used in the proposed scheme.

4.1. Bilinear groups

Let consider two multiplicative groups G_1 & G_2 having prime order p with g_1 & g_2 being the generators. Let ψ be an isomorphism from G_2 to G_1 with $\psi(g_2) = g_1$ and $e : G_1 X G_2 \longrightarrow G_T$ be a bilinear map with properties:

- Countability: An efficient algorithm to compute e exists.
- Bilinearity: For every $u \in G_1, v \in G_2$ and $a, b \in Z_p$: $e(u^a, v^b) = e(u, v)^{ab}$
- Non-degeneracy: $e(g_1, g_2) \neq 1$

4.2. Complexity assumptions

Security of the proposed method depends on the complications of problems like Decision Linear problem, Strong Diffie-Hellman problem and Computational Diffie-Hellman problem, described below:

- Decision Linear problem: Let G_1 be a cyclic group having g_1 as generator and prime order p. Given $u, v, h, u^a, u^b, u^c \in G_1$ as input; if $a + b = c$, output yes else no.
- Strong Diffie-Hellman problem: Let consider two groups G_1 and G_2 of prime order p ($G_1 = G_2$)(possibly) with g_1 being the generator of group G_1 & g_2 of G_2. Given a $(q + 2)$ tuple $(g_1, g_2, g_2^Y, g_2^{Y^2}, ...)$ as input, produce a pair $(g_1^{\frac{1}{Y+x}}, x)$ as output where $x \in Z_p^*$.
- Computational Diffie-Hellman problem: Let G_1 be a cyclic group having g_1 as generator. Let the prime order of the group is p. Given (g, g^x) for $x \in Z_p$, produce g^{x^2} as output.

4.3. Vector commitment

One of the most important things in cryptography is a commitment scheme. Vector commitment [16] is one among them. It allows a user to bind to a chosen value, keeping it hidden from others in a way that committed value can be revealed later. This commitment scheme satisfies two properties, namely, hiding and binding. Thus the primitives built upon vector commitment are useful in solving the problem of verifiable data outsourcing. This commitment scheme consists of following algorithms:

- *VC.KeyGen*: Taking security parameter & the size of committed vector as input, this algorithm generate parameters which are made public.
- *VC.Com*: Given an input message m & the public parameters, it computes a commitment string C for the message.
- *VC.Open*: This algorithm generates a proof that m is the $i - th$ committed message.
- *VC.Ver*: This verification algorithm accepts only if the proof that C was created for the $i - th$ message is valid.
- *VC.Update*: This algorithm of vector commitment scheme is executed by the committer who generated the commitment string & wants to update it by changing the $i - th$ message m to m'. It takes old message, new message & the position i as input and produces new commitment as output.

4.4. Identity based ring signature

Identity (ID) based ring signature [39] is an amalgamation of general ring signature and an ID based one. More precisely, this scheme is basically the ring signature one in which user's public key is signers identity. This scheme consists of three entities: signer, user & the Trusted Authority (TA); and is a collection of four algorithms, as described below:

- *Setup*: Executed by *TA*, it generates some parameters *param* and a master key on receiving a security parameter as input.
- *Extract*: This algorithm generates private key for a user when executed by *TA* on an input of *param*, master key & an arbitrary *ID*. *ID* is the signers identity and used a signers public key.
- *Sign*: On receiving *param*, signer's private key, list of identities which includes the identity of signer and a message, it outputs a signature for the message.
- *Verify*: This algorithm verifies whether a signature is valid or not for an input message.

5. Proposed scheme

5.1. Overview

The proposed scheme is based upon an existing scheme in the ref. [27] which uses group signature. This signature form allows any group member to sign messages on behalf of whole group while providing anonymity and privacy. There exists a group manager who adds members to the group and can reveal the signer in case of any dispute. In the proposed scheme, identity based ring signature is used in place of group signature. Ring signature allows any number of users to comeup together and form a group to share data among them without requiring group manager or some additional setup. It is an adhoc collection of users in which anyone can verify whether the signature is generated by a member belonging to the collection. This type of signature is very useful in anonymity protection as there is no way to identify who the signer is. Fig. 5 showcases the algorithms used in the proposed scheme:

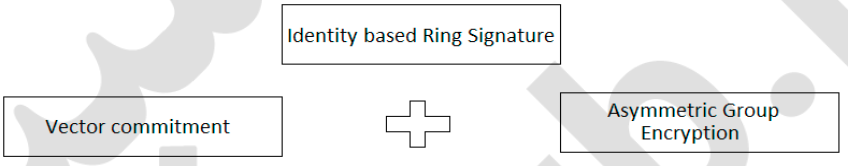

Fig. 5: Proposed scheme

Although the work done using group signature scheme is varied, there are certain issues which needs to be addressed like complex key management, group manager administration, computation overhead etc. Identity based ring signature scheme deals successfully with all these issues. The technique of using a publicly known string about the user like his email, phone no. etc. as his public key simplifies key management to an extent. Ring signature scheme involves lesser number of computations than group signature scheme, in computing the signature on a message. Moreover absence of a group manager act as an advantage for the users as any number of users can create a group anytime without the involvement of a third party.

5.2. Concrete scheme

The proposed scheme contains five algorithms, namely setup, verify, query, update & proofupdate. The major difference from the existing scheme is in the signature generation and verification part. This section presents the concrete definition of proposed scheme based on vector commitment and identity based ring signature.

Consider a database having tuples (i, m_i) where i is the index & m_i is the associated value. n users of a group, with only one data owner, shares this database. Let $M = Z_p$ be the message space and k, security parameter.

- Setup

1. Let consider two bilinear groups G and G_T having prime order p. Consider bilinear map $e : G \longrightarrow G_T$ and g, a generator of G. To start, select randomly $z_1, ...z_q \longleftarrow Z_p$. Now $\forall i = 1, ..., q$, compute $h_i = g^{z_i}$ and $\forall i, j = 1, ..., q, i \neq j$, compute $h_{i,j} = g^{z_i z_j}$. Now to obtain the public parameters PP, $VC.KeyGen$ is executed by the data owner. The public parameters are $PP = (p, q, G, G_T, H, g, (h_i)_{i\in[q]}, \{h_{i,j}\}_{i,j\in[q],i\neq j})$.

2. Let consider a group G having generator P. Choose $s \in Z_q^*$, randomly, & set $P_{pub} = sP$. Let s be the master key of TA. On receiving an identity ID as input, output $S_{id} = sH_1(ID)$ as the private key associated with ID, public key being $Q_{id} = H_1(ID)$. In this way, generate secret keys for all the users.
3. Determine the commitment string and other auxiliary information using the computing algorithm of vector commitment scheme as follows:

$$C = h_1^{m_1}, h_2^{m_2}, ..., h_q^{m_q}$$
$$aux = (m_1, m_2, ..., m_q)$$

4. Next, to determine the signature for an input message m, run the signing algorithm of identity based ring signature scheme over the commitment. Select an element $A \in G$, determine $c_{k+1} = H(L \parallel m \parallel e(A, P))$. Now a forward ring sequence is to be generated. For $i = k+1, ..., n-1, 0, 1, ..., k-1$, choose $T_i \in G$ & find:

$$c_{i+1} = H(L \parallel m \parallel e(T_i, P)e(c_i H_1(ID_i), P_{pub}))$$

Determine:

$$T_k = A - c_k S_{ID_k}$$

The signature for the message is $(n+1)$ tuple: $(c_0, T_0, T_1, ..., T_{n-1})$.

- Query
Group user executes $VC.Open$ to compute a proof:

$$\Lambda_i^t = \Pi_{j-1, j\neq i}^q \cdot h_{i,j}^{m_j^t} = (\Pi_{j=1, j\neq i}^q \cdot h_j^{m_j^t})^{z_i}$$

of the $i-th$ committed message.
- Verify

1. On an input of a message and its corresponding signature, the third party auditor first verifies the signature validity. For it, the auditor runs the verification algorithm of ring signature scheme. This algorithm computes:

$$c_{i+1} = H(L \parallel m \parallel e(T_i, P)e(c_i H_1(ID_i), P_{pub})) \text{ for } i = 0, 1, ..., n-1$$

on receiving a signature $(c_0, T_0, T_1, ..., T_{n-1})$ as input & accepts if:

$$(c_n = c_0)\ .$$

2. If the signature is valid, auditor executes $VC.Ver$ to verify that whether the equation:

$$e(C^t / h_i^{m_i^t}, h_i) = e(\Lambda_i^t, g)$$

holds or not. If it outputs one, the equation holds..

- Update

1. Group user verifies the current database to ensure its validity.
2. To update a message m_i to m_i', the user executes $VC.Update$ and provides an updated commitment $C' = C \cdot h_i^{m'-m}$ & updated information $U = (m, m', i)$ as output.

- ProofUpdate

1. TPA executes $VC.ProofUpdate$ to compute an updated proof Λ_j for the message at position j such that this new proof is valid with respect to new commitment C' which contains the updated message m' as the new message at position j.
2. The auditor uses the update information to generate a proof of update. If $i \neq j$, find the updated commitment $C' = C \cdot h_i^{m'-m}$ and the updated proof $\Lambda'_j = \Lambda_j \cdot (h_i^{m'-m})^{z_j} = \Lambda_j \cdot h_{j,i}^{m'-m}$. If $i = j$, find $C' = C \cdot h_i^{m'-m}$ with the proof remaining same.

6. Performance evaluation

The proposed scheme solves the efficiency problems of public integrity auditing schemes where the data is signed by a user belonging to the group in which data is shared and any member can conduct verifiable data update. This section provides the performance analysis of the proposed scheme & conducts a comparison with existing scheme. It is assumed that the underlying building blocks of the scheme are secure.

The experiments are simulated on a windows 10 machine with Intel Core¢ i7-4770 CPU running at 3.40GHz and 8G memory. A group having fewer members is taken into consideration, having all the privileges granted by the data owner. First of all, one time computational effort is needed in the setup phase. It is efficient as the resources invested by the client do not depend on database size and the most expensive computations are outsourced to the cloud server.

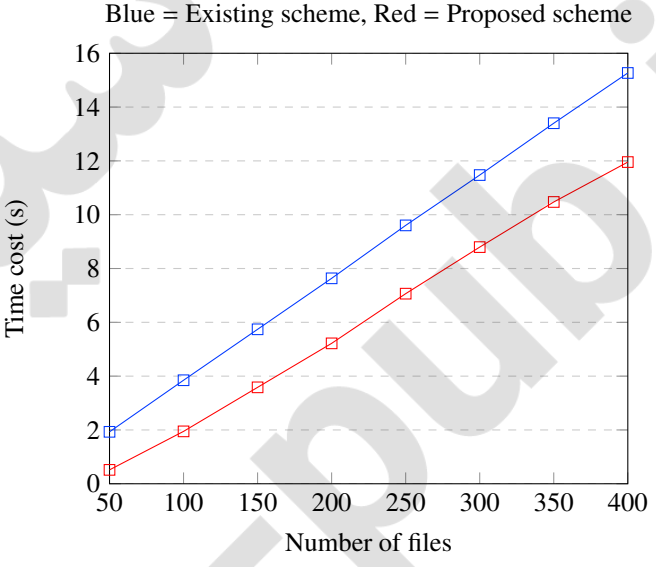

Fig. 6: Signature generation

The graph in Fig. 6 shows the time taken by the signature generation algorithms. The existing scheme uses group signature to compute a signature on a message whereas the proposed scheme uses ring signature to serve the purpose. As we are interested in small group of users, ring signature scheme takes lesser time as comparted to group signature one. Group signature scheme requires same number of computations irrespective of the number of users but the other scheme depends upon the number of users. It can be clearly seen from the graph that for smaller number of users, ring signature scheme has lesser computation time as compared to the group signature scheme.

VC.Open algorithm is responsible for generating a proof of committed message and the time taken by it is exponential to the number of files for both the existing as well as proposed scheme. As the number of files increases, the computation overhead increases.

The time taken by the third party auditor to verify a file is shown in the graph in Fig. 8. It can be clearly seen from

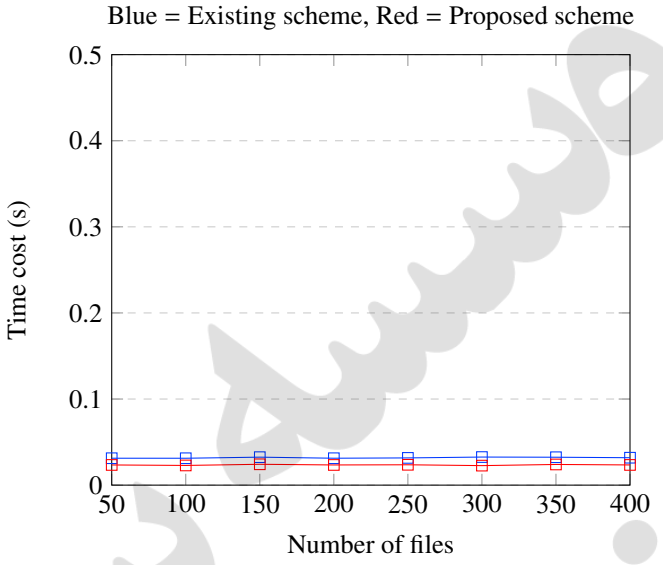

Fig. 7: TPA verification

the graph that the auditor takes lesser time in verifying a file whose signature is computed using ring signature scheme than the one whose signature is generated using group signature scheme. But still the time taken by the third party auditor is almost same in verifying the signature generated by the used schemes. This little difference in verification time is due to the number of computations involved in the verification process. The second scheme reduces this overhead to some extent as seen in the graph.

7. Conclusion

The proposed scheme is successfully implemented to maintain the privacy and integrity of data stored at remote cloud servers. The different schemes used realizes secure data integrity auditing. Identity based ring signature scheme takes lesser time to compute signature for small group users for a message when compared with group signature scheme used in the existing method. For a small group, the number of computations carried out in ring signature scheme is lesser than those in group signature scheme during the generation as well as verification phase of a signature which reduces the computation overhead. Moreover, there is no entity like group manager in case of ring signature which allows any number of users to come up and create a group for sharing files among them, unlike group signature where group manager is responsible for performing the task. Hence, the combination of these primitives enables the proposed method to outsource encrypted database to remote cloud server and provides data confidentiality for small group. of members Also, performance analysis shows that the proposed scheme is more efficient in the small group sceneraio as compared to the existing scheme.

8. Future work

The proposed scheme, at the moment, does not support group user revocation as well as user accountability mechanism. This acts as an disadvantage for the scheme in case when the data owner no longer wants to share his data with some member of the group or when it is necessary to identify who the signer of a message is. Enhancements can be made to the proposed scheme by incorporating these features.

References

[1] Abe, M., Ohkubo, M., Suzuki, K., 2002. 1-out-of-n signatures from a variety of keys, in: International Conference on the Theory and Application of Cryptology and Information Security, Springer. pp. 415–432.

[2] Ateniese, G., Burns, R., Curtmola, R., Herring, J., Kissner, L., Peterson, Z., Song, D., 2007. Provable data possession at untrusted stores, in: Proceedings of the 14th ACM conference on Computer and communications security, Acm. pp. 598–609.

[3] Ateniese, G., Camenisch, J., Joye, M., Tsudik, G., 2000. A practical and provably secure coalition-resistant group signature scheme, in: Annual International Cryptology Conference, Springer. pp. 255–270.

[4] B. Wang, L.B., Hui, L., 2013. Public auditing for shared data with efficient user revocation in the cloud, in: IEEE INFOCOM 2013, Turin, Italy, IEEE. pp. 2904–2912.

[5] Backes, M., Fiore, D., Reischuk, R.M., 2013. Verifiable delegation of computation on outsourced data, in: Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security, ACM. pp. 863–874.

[6] Bellare, M., Pointcheval, D., Rogaway, P., 2000. Authenticated key exchange secure against dictionary attacks, in: Advances in CryptologyEUROCRYPT 2000, Springer. pp. 139–155.

[7] Bellare, M., Rogaway, P., 1993. Random oracles are practical: A paradigm for designing efficient protocols, in: Proceedings of the 1st ACM conference on Computer and communications security, ACM. pp. 62–73.

[8] Benabbas, S., Gennaro, R., Vahlis, Y., 2011. Verifiable delegation of computation over large datasets, in: Annual Cryptology Conference, Springer. pp. 111–131.

[9] Boneh, D., Shacham, H., 2004. Group signatures with verifier-local revocation, in: Proceedings of the 11th ACM conference on Computer and communications security, ACM. pp. 168–177.

[10] Bowers, K.D., Juels, A., Oprea, A., 2009. Proofs of retrievability: Theory and implementation, in: Proceedings of the 2009 ACM workshop on Cloud computing security, ACM. pp. 43–54.

[11] Bresson, E., Stern, J., Szydlo, M., 2002. Threshold ring signatures and applications to ad-hoc groups, in: Annual International Cryptology Conference, Springer. pp. 465–480.

[12] Camenisch, J., 1997. Efficient and generalized group signatures, in: International Conference on the Theory and Applications of Cryptographic Techniques, Springer. pp. 465–479.

[13] Camenisch, J., Lysyanskaya, A., 2002. Dynamic accumulators and application to efficient revocation of anonymous credentials, in: Annual International Cryptology Conference, Springer. pp. 61–76.

[14] Camenisch, J., Michels, M., 1998. A group signature scheme based on an rsa-variant. BRICS Report Series 5.

[15] Camenisch, J., Stadler, M., 1997. Efficient group signature schemes for large groups. Advances in CryptologyCRYPTO'97 , 410–424.

[16] Catalano, D., Fiore, D., 2013. Vector commitments and their applications, in: Public-Key Cryptography–PKC 2013. Springer, pp. 55–72.

[17] Chaum, D., Van Heyst, E., 1991. Group signatures, in: Workshop on the Theory and Application of of Cryptographic Techniques, Springer. pp. 257–265.

[18] Chen, L., Pedersen, T., 1995. New group signature schemes, in: Advances in CryptologyEUROCRYPT'94, Springer. pp. 171–181.

[19] Cloud9, 2011. Your development environment, in the cloud. URL: `https://c9.io/`.

[20] Codeanywhere, 2011. Online code editor. codeanywhere. URL: `https://codeanywhere.com/`.

[21] Codenvy, 2002. Online code editor. cloud ide. URL: `https://codenvy.com/`.

[22] Cramer, R., Damgård, I., Schoenmakers, B., 1994. Proofs of partial knowledge and simplified design of witness hiding protocols, in: Annual International Cryptology Conference, Springer. pp. 174–187.

[23] De Santis, A., Di Crescenzo, G., Persiano, G., Yung, M., 1994. On monotone formula closure of szk, in: Foundations of Computer Science, 1994 Proceedings., 35th Annual Symposium on, IEEE. pp. 454–465.

[24] Dodis, Y., Vadhan, S., Wichs, D., 2009. Proofs of retrievability via hardness amplification, in: Theory of Cryptography Conference, Springer. pp. 109–127.

[25] Erway, C.C., Küpçü, A., Papamanthou, C., Tamassia, R., 2015. Dynamic provable data possession. ACM Transactions on Information and System Security (TISSEC) 17, 15.

[26] Fiat, A., Shamir, A., 1986. How to prove yourself: Practical solutions to identification and signature problems, in: Conference on the Theory and Application of Cryptographic Techniques, Springer. pp. 186–194.

[27] Jiang, T., Chen, X., Ma, J., 2016. Public integrity auditing for shared dynamic cloud data with group user revocation. IEEE Transactions on Computers 65, 2363–2373.

[28] Juels, A., Kaliski Jr, B.S., 2007. Pors: Proofs of retrievability for large files, in: Proceedings of the 14th ACM conference on Computer and communications security, Acm. pp. 584–597.

[29] Mell, P., Grance, T., et al., 2011. The nist definition of cloud computing .

[30] Rivest, R.L., Shamir, A., Adleman, L., 1978. A method for obtaining digital signatures and public-key cryptosystems. Communications of the ACM 21, 120–126.

[31] Rivest, R.L., Shamir, A., Tauman, Y., 2001. How to leak a secret, in: International Conference on the Theory and Application of Cryptology and Information Security, Springer. pp. 552–565.

[32] Shi, E., Stefanov, E., Papamanthou, C., 2013. Practical dynamic proofs of retrievability, in: Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security, ACM. pp. 325–336.

[33] Tsudik, G., Xu, S., 2003. Accumulating composites and improved group signing, in: International Conference on the Theory and Application of Cryptology and Information Security, Springer. pp. 269–286.

[34] Wang, B., Li, B., Li, H., 2012. Oruta: Privacy-preserving public auditing for shared data in the cloud, in: Cloud Computing (CLOUD), 2012

IEEE 5th International Conference on, IEEE. pp. 295–302.
[35] Wang, C., Wang, Q., Ren, K., Lou, W., 2010. Privacy-preserving public auditing for data storage security in cloud computing, in: Infocom, 2010 proceedings ieee, Ieee. pp. 1–9.
[36] Wu, Q., Mu, Y., Susilo, W., Qin, B., Domingo-Ferrer, J., 2009. Asymmetric group key agreement, in: Annual International Conference on the Theory and Applications of Cryptographic Techniques, Springer. pp. 153–170.
[37] Yuan, J., Yu, S., 2013. Proofs of retrievability with public verifiability and constant communication cost in cloud, in: Proceedings of the 2013 international workshop on Security in cloud computing, ACM. pp. 19–26.
[38] Yuan, J., Yu, S., 2014. Efficient public integrity checking for cloud data sharing with multi-user modification, in: INFOCOM, 2014 Proceedings IEEE, IEEE. pp. 2121–2129.
[39] Zhang, F., Kim, K., 2002. Id-based blind signature and ring signature from pairings, in: International Conference on the Theory and Application of Cryptology and Information Security, Springer. pp. 533–547.