RESEARCH ARTICLE OPEN ACCESS

Attribute based Cloud Data Integrity Auditing for Secure Cloud Storage

Mr D Purushothaman MCA., M.E., [1], C RaghuRam [2]

[1] Asst. Professor, Department of Computer Applications

[2] Student, Department of Computer Applications

[1], [2] Chadalawada Ramanamma Engineering College (Autonomous)

ABSTRACT

Outsourced storage such as cloud storage can significantly reduce the burden of data management of data owners. Despite of a long list of merits of cloud storage, it triggers many security risks at the same time. Data integrity, one of the most burning challenges in secure cloudstorage, is a fundamental and pivotal element in outsourcing services. Outsourced data auditing protocols enable a verifier to efficiently check the integrity of the outsourced files without downloading the entire file from the cloud, which can dramatically reduce the communication overhead between the cloud server and the verifier. Existing protocols are mostly based on public key infrastructure or an exact identity, which lacks flexibility of key management. In this paper, we seek to address the complex key management challenge in cloud data integrity checking by introducing attribute-based cloud data auditing, where users can upload files to cloud through some customized attribute set and specify some designated auditor set to check the integrity of the outsourced data. We formalize the system model and the security model for this new primitive, and describe a concrete construction of attribute-based cloud data integrity auditing protocol. The new protocol offers desirable properties namely attribute privacy- preserving and collusion-resistance. We prove soundness of our protocol based on the computational Diffie-Hellman assumption and the discrete logarithm assumption. Finally, we develop a prototype of the protocol which demonstrates the practicality of the protocol.

Keywords-- Cloud Storage, Data Integrity, Attribute-Based Cryptography, Threshold Secret Sharing.

I. INTRODUCTION

CLOUD storage, one of the most basic services of IaaS [1], is a configurable data storage model that enables data owners to store their files in the cloud without retaining a local copy, which greatly reduces data owners' storage and management burden of local files. Moreover, itis quite convenient for users to retrieve their files via terminals which have cloud access, such as mobile phones and tablet PCs. Cloud storage services have a number of significant advantages compared with traditional storage approaches, such as anytime and anywhere access, location independent, on-demand services, flexible resources. Currently, an increasing number of individuals and enterprises are enjoying the convenience provided by cloud storage. Cloud storage provides convenient, fast and unlimited capacity IT services to its users. However, due to the separation between data ownership and data management, cloud storage introduces some new data security challenges since data are hosted by cloud servers rather than data owners themselves. The cloud servers are not fully trusted. Any accidental data deletion by the cloud server, or worse, a physical catastrophe such as a fire or earthquake, might lead to permanentloss of users' data. This is not exaggerating the dangers to frighten people. Symantec, a well known information security company, reported a survey and showed that 43% of respondents experienced cloud data loss accidents and had to recover the data from backups1. Thus, it is fair to claim that data integrity is the premise and basis of reliable cloud computing as well as bigdata analysis. If the integrity of cloud data is not ensured, the correctness of big data analysis and cloud computing cannot be guaranteed. As a consequence, data owners require a strong integrity guarantee of their outsourced data to make sure the cloud servers store their data correctly.

In order to address the issue mentioned above, the concept of cloud data integrity auditing was presented, which can be mainly divided into two categories, namely Proof of Retrieveability (PoR) and Provable Data Possession (PDP). PDP is a probabilistic detection protocol which employs randomly sampled data blocks rather than the entire file to perform cloud data integrity checking, which is more efficient than the deterministic auditing protocols [2], especially for large files. PoR protocols, similar to PDP, can not only detect the integrity of cloud data but also provide data retrieveability. By using error-correction coding techniques, PoR can improve the storage reliability. Both PDP protocols and PoR protocols are challenge-response protocols, where homomorphic verifiable authenticators are employed to reduce the communication and computation costs between cloud server and Third-Party Auditor (TPA) when conducting the cloud data auditing protocols.

II. RELATED WORKS

Deswarte et al. [2] put forward the con-cept of remote data integrity checking for the first time and presented a scheme based on RSA. Filho et al. [3] put forward a new protocol, which can greatly improve the date integrity auditing efficiency, that is, it costs 20 seconds for 1MB file. Yamamoto et al. [4] proposed an efficient scheme by offering batch processing [5] based on the homomorphic hash function. The similar technique was employed in Sebe [6], in which they proposed a Diffie-Hellman protocol based on group Zp but the length of each data block is limited and the storage overhead of the client is O(n). Juels et al. [7] came up with the concept of PoR and described a concrete protocol by inserting some special blocks, named sentinels, intothe original file. The cloud server is challenged by verifying some sentinels. Ateniese et al. [8]
[9] proposed a PDP protocol based on homomorphic verifiable tag (HVT). HVT can aggregate responses of n challenged blocks into a single value, which can significantly reduce the communication cost of cloud server and TPA. Erway et al. [10] gave a framework supporting dynamic PDP by extending the protocol in [8], and proposed an efficient construction. Shacham and Waters [11] presented two PoR schemes using homomorphic message authentication code and BLS short signature [12]. The previous one supports private verification, while the latter one supports public verification. Recently, a variety of cloud data integrity auditing protocols with various eyecatching properties have been proposed such as supporting dynamic operations auditing [13], privacy-preserving auditing [14], [15], [16], public auditing [17], [18], and multiple copies auditing [19]. The aforementioned protocols are based on public key infrastructure (PKI), which consists of a set of roles, policies and procedures that needed to issue, manage, distribute, store and revoke digital certificates. The most commonly adopted digital certificate in our daily life is X.509 certificates, an ITU-T standard for a PKI and privilege management infrastructure. However, there are three weaknesses when involving PKI based protocols. Firstly, the generation, management and revocation of digital certificates requires a highly complicated structure. Secondly, a PKI system is a tree structure and the authentication to the current CA relies on its parent CA. Thus, the root CA is a trusted center and self-signed, which is vulnerable since compromising root CA means all the related certificates should be reissued. Thirdly, the certificates issued by a CA may not secure enough to ensure the security of one's secret key. For example, Dell's self root certificate was reported to expose users' encrypteddata to spy in 2015. 2. In order to reduce the complexity of certificate management in PKI, identity based (ID-based) cryptology [20] was proposed by Shamir, in which the secret key binds with the user's identity. Therefore, users can communicate without exchanging digital certifications.

Due to the flexibility in key management, ID-based cryptology has been widely adopted in a variety of primitives, including in cloud data integrity auditing protocols. A number of ID- based cloud data auditing protocols have been proposed such as [22] [23] [24]. The most commonly used identity information in existing ID-based cloud data auditing protocols is an arbitrary bit string chosen by a user, such as names, IP and E-mail, which can be viewed as a text-based recognition related to the combinations of characters and numbers. With this identity information, one can register for a private key binding to his/her identity from the private key generation center. There are three weaknesses when making use of ID-based protocols. Firstly, identity might not be unique if identity information is not chosen properly. For example, the name "Nancy Helen" is probably not unique. Secondly, a user needs to "prove" to the privatekey generator centre that the claimed identities are indeed belong to him, which is typicallyverified by providing some additional documents such as one's passport or identity card. However, these supplementary documents themselves are subject to forgery. Thirdly, one has to keep in mind his/her identity information even sometimes an identity is too long to remember. We seek to address the issue mentioned above by proposing an alternative named attribute-basedcloud data integrity auditing. Different from the previous work that attribute-based cryptography is used to realize data sharing [25], [26] or access control [27] in a cloud environment. Thenotion of an attribute-based cloud data auditing protocol is a generalization of fuzzy identity-based cloud data auditing protocol [28]. In this primitive, it allows cloud users to define some attribute sets such as name, age and select a subset of those attributes to generate private keys to generate the metadata of the files which need outsourcing rather than some inherent attribute[28]. When it comes to auditing phase, the cloud users can designate a certain group of peoplewith a set of similar attributes to execute the cloud data integrity checking.

Compared with traditional cloud data integrity checking, the advantages of attribute- based data integrity auditing protocols are as follows. Firstly, an attribute-based cloud data auditing protocol enables the data owners to specify the scope of the auditors, which avoids the situation of single-point failure in traditional protocols which has a single TPA. Secondly, an attribute based cloud data auditing scheme allows users to select their attribute sets when uploading files. Generally speaking, one with n atomic attributes can enjoy 2n combined attributes to manipulate the file. This can be implemented by an attribute-based data auditing scheme with the key size O(n), rather than O(2n) if employing traditional data auditing schemes. Thus, attribute-based cloud data integrity protocols are more flexible and practical compared with the traditional proposals in many real-world scenarios.

Contributions. In this paper, we attempt to simplify the key management issue of traditional cloud data integrity auditing protocols by incorporating attribute-based cryptography. Our contributions are three-fold. 1) We propose the notion of attribute-based cloud data integrity auditing, where users can choose some arbitrary attributes to generate private keys and upload files to cloud server. Moreover, the data owners can specify the set of auditors who are able to check the integrity of the outsourced data. 2) We formalize the system model as well as the security model of this new primitive to ensure the security named soundness of cloud data integrity auditing. 3) We describe a concrete construction of attribute based cloud data integrity auditing protocol. We then prove the security of the protocol under Shacham-Waters game- based proof framework [11].

III. PROPOSED METHOD

An attribute-based signature (ABS) [33] involves two entities, key generation center (KGC) and a user. KGC is responsible for generating the corresponding secret key for a user with the claimed attribute set. Upon receiving secret key from KGC, a user can generate an attribute based signature. This primitive consists of the following four algorithms.

Setup(k): This is a probabilistic algorithm, which takes a security parameter k as input and outputs the master key MK as well as the public parameter PK.

Extract(MK,A): This is a probabilistic algorithm which takes a master key MK and an attribute set A as input. It generates secret key SKA for the user.

Sign(PK; SKA;_;M): This is a probabilistic algorithm which takes the public parameter PK, a secret key SKA, a predicate _ and a message M as input. It outputs a signature.

Verify (PK, B, M): This is a deterministic algorithm which takes the public parameter PK, an attribute set B, a predicate, the message M and its alleged signature as input. It returns 1 or 0 toindicate the signature is valid or not.

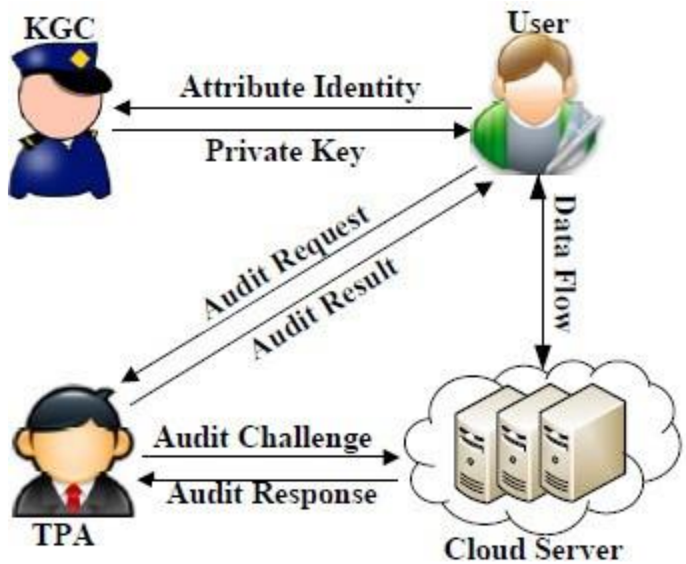

Fig. 1. The system model of attribute-based data integrity auditing protocol

An attribute-based cloud data integrity auditing protocol should satisfy the following properties [11] 1) **Correctness.** Correctness states that for a valid proof, which is generated by the Response algorithm, the Verify algorithm can accept it with an overwhelming probability. 2) **Soundness.** Soundness requires that, any cheating prover, who can generate a valid proof that can pass the Verify algorithm is actually storing the challenged file. In other words, there is no adversary, who does not store the file, can generate a valid proof of the challenge. 3) **Collusion resistance.** Collusion resistance indicates that a group of users can complete cloud data auditing if at least one individual has the permission to do so. In other words, if a group of users cannot generate a valid response individually, the advantage to output a valid response will not increase even all the users collude. Note that in the security model of **Soundness**, the adversary can make Extract queries to inquire the private key of selected attributes, where the overlap of the selected attributes and the set of challenge attributes must be less than d. This is resemble the collusion resistance scenario. Therefore, in the security model of **Soundness**, the adversary has the ability to perform collusion attack. Thus, the property of collusion resistance holds naturally if the property of **Soundness** holds. 4) **Attribute privacy-preserving.**

An attribute-based cloud data integrity auditing protocol should satisfy the following properties [11] 1) **Correctness.** Correctness states that for a valid proof, which is generated by the Response algorithm, the Verify algorithm can accept it with an overwhelming probability. 2) **Soundness.** Soundness requires that, any cheating prover, who can generate a valid proof that can pass the Verify algorithm is actually storing the challenged file. In other words, there is no adversary, who does not store the file, can generate a valid proof of the challenge. 3) **Collusion resistance.** Collusion resistance indicates that a group of users can complete cloud data auditing if at least one individual has the permission to do so. In other words, if a group of users cannot generate a valid response individually, the advantage to output a valid response will not increase even all the users collude. Note that in the security model of **Soundness**, the adversary can make Extract queries to inquire the private key of selected attributes, where the overlap of the selected attributes and the set of challenge attributes must be less than d. This is resemble the collusion resistance scenario. Therefore, in the security model of **Soundness**, the adversary has the ability to perform collusion attack. Thus, the property of collusion resistance holds naturally if the property of **Soundness** holds. 4) **Attribute privacy-preserving.** Attribute privacy preserving property denotes that, during cloud data auditing phase, TPA can not deduce the set of attributes used by users to upload the file except the d common attributes selected by cloud server. Therefore, we require that if TPA can guess the user's attribute from the response, it can also complete the deduction when only given the intersection with d attributes. This property ensures that only the intersection attributes selected by the cloud server are possibly revealed to TPA when executing the challenge-response protocol. The proposed attribute-based cloud data integrity auditing protocol consists of three procedures, namely Enroll, Store and Audit. Enroll phase involves the cloud user and a KGC following Setup and Extract algorithm. The user chooses some attribute set and submits it to KGC. KGC checks the validity and generates the corresponding private key for the cloud user with the master secret key with Extract algorithm. Store phase involves the cloud user and the cloud server with MetadataGen algorithm. The user preprocesses the File F to be uploaded into F. Then generates the file tag and block authenticators using the private key using MetadataGen algorithm. After that, the cloud user uploads the metadata to the cloud server and deletes the local copy. The Audit phase involves an auditor(or the cloud user), cloud server and a TPA. The auditor sends his own attribute set to the TPA as an audit request and TPA runs the Challenge- Response protocol with cloud server to check the integrity of the file stored on the sever. TPA firstly generates a challenge and forwards audit request as well as the challenge set to cloud server. Upon receiving the challenge fromTPA, the cloud server checks the overlap attribute set between the cloud user's and the auditor's. If the number of intersection is less than the auditing precision d, which is set by the cloud userin Setup phase, cloud server emits failure and returns signature. Otherwise, cloud server generates a response with the challenged file F together with the corresponding block authenticators. To achieve user privacy-preserving, the cloud server first chooses an intersection of A and B with d elements and converts the response accordingly to prevent TPA learning the signer's attributes outside A U B, and forwards the converted response to TPA. Finally, TPA verifies the response and returns the auditing result to the user.

IV. RESULTS AND DISCUSSION

In this section, we report the performance of the proposed protocol. In ourimplementation, all the algorithms are conducted on a Win 8 64-bit laptop with Intel Core (TM)i5-4300 @ 2.49GHz CPU and an 8 GB SSD. The projects are written in C++ language underVisual Studio 2010 compiler and we call the Miracl library [34] API to construct elliptic curves.In the first part, we present the time consumption of both **Setup** and **Extract** algorithms. As canbe seen from Fig. 2, the time cost of the **Setup** algorithm exhibits a strictly linear growth with the maximum number of attributes m in the system. This is due to the fact that the function T needs to perform m multiplications. Thus, with the increasing of m, the time cost of Setup will increasemultiply as well. Fig. 3 shows that the time consumption of **Extract** algorithm grows linearlywith the number of attributes required for a user. The results are consistent with our empiricalanalysis, since the user's private key is calculated for each attribute in a user's attribute set, sothe more attributes an identity includes, the longer it takes for the key extraction algorithm. In thesecond part, we test the time consumption of generating the metadata for a file. We choose a file with a fixed size of 1MB and select the maximum number of attributes in a set to be 10, three ofwhich to describe a user's attribute information. The block size varies from 1KB to 100KB withthe increment of 10KB.We divide the Metadata- Gen algorithm into two parts, say, online andoffline phase, where the offline phase refers to the portion that can be calculated before theuploaded file is selected and the online phase is the portion that must be determined after obtaining the file. Since the off-line part changes rapidly in the range of 1-10KB, four points areadded in this interval to observe the trend of the curve.

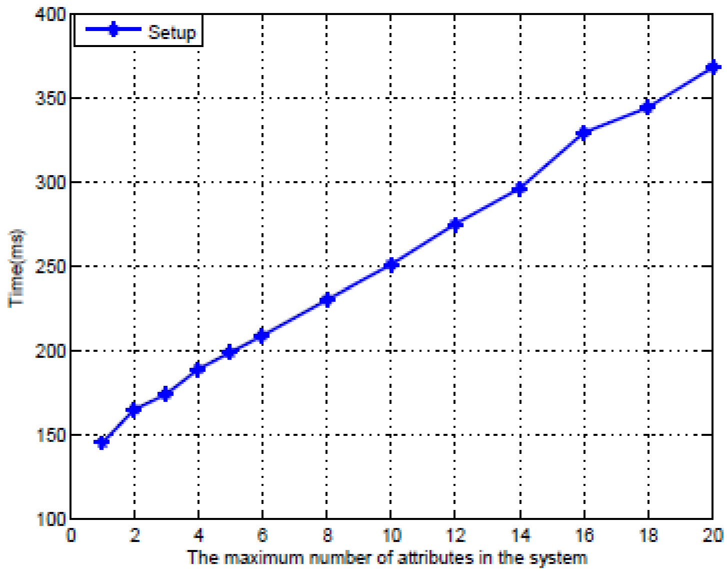

Fig. 2: Time consumption for **Setup** algorithm

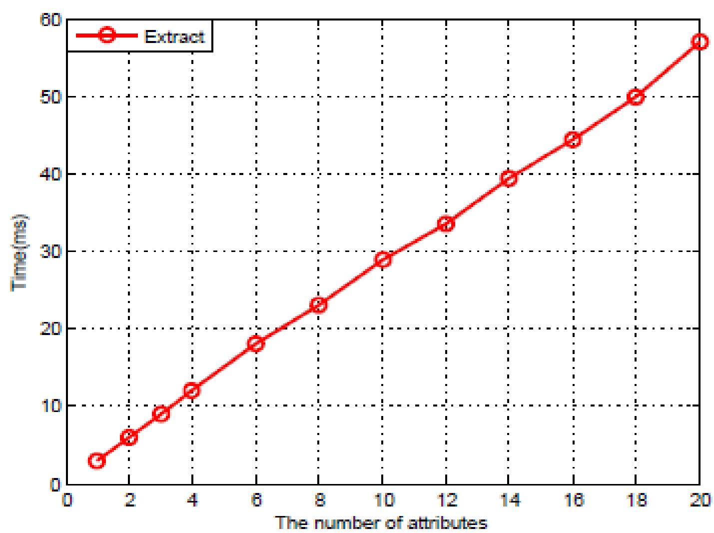

Fig. 3: Time consumption for **Extract** algorithm

V. FUTURE SCOPE AND CONCLUSION

In the past few years, cloud data integrity has drawn much attention from both academia and industry. In this paper, we propose an attribute-based cloud data integrity auditing protocol, for the first time, to simplify the key management issue in traditional cloud data auditing schemes. We formalize the system model and security model for this new primitive. Subsequently, a concrete construction is presented by involving the idea of attribute-based cryptography. The proposed protocol can achieve the property of soundness, attribute privacy- preserving and collusion resistance. We prove the soundness of the protocol under Shacham- Waters game-based proof framework. The implementation illustrates the practicality and efficiency of the new proposal. Future Work. The construction in Section 4 provides a privacy- preserving guarantee that reveals nothing but the d common attributes chosen by cloud server when executing the auditing protocols. The authors are investigating a strong privacy-preserving mechanism that can ensure zeroknowledge in the auditing phase. Future work includes proposinga concrete construction that are both practical and with high efficiency.

REFERENCES

[1] M. Hogan, F. Liu, A. Sokol and J. Tong. "NIST Cloud Computing Standards Roadmap". NIST Cloud Computing Standards Roadmap Working

Group, SP 500-291-v1.0, NIST, Jul, 2011.

[2] Y. Deswarte, J. J. Quisquater and A. Saidane. “Remote integrity checking”. Integrity and Internal Control in Information Systems VI. Springer US, pp.1-11, 2004.

[3] G. Filho D L, Barreto P S L M. “Demonstrating data possession and uncheatable data transfer”. IACR Cryptology ePrint Archive, 2006, 150.

[4] G. Yamamoto, S. Oda, K. Aoki. “Fast integrity for large data”. Proc. ECRYPT Workshop Software Performance Enhancement for Encryption and Decryption. Amsterdam, Netherlands 2007, 21-32.

[5] K. Chida, G. Yamamoto. “Batch processing of interactive proofs”. Cryptographers Track at the RSA Conference. Springer Berlin Heidelberg, San Francisco, USA, 2007, 196-207.

[6] F. Sebe, A. Martinez-Balleste, Y. Deswarte, et al. “Time-bounded remote file integrity checking”. Technical Report 04429, LAAS, 2004.

[7] A. Juels, B. S. Kaliski Jr. “PORs: Proofs of retrievability for large files”. Proceedings of the 14th ACM conference on Computer and communications security. Acm, Alexandria, 2007, 584-597.

[8] G. Ateniese, R. C. Burns, R. Curtmola, J. Herring, L. Kissner, Z. N. J. Peterson and D. X. Song. “Provable data possession at untrusted stores”. in Proc. of ACM Conference on Computer and Communications Security, pp.598-609, 2007.

[9] G. Ateniese, S. Kamara and J. Katz. “Proofs of storage from homomorphic identification protocols”. Proc. of ASIACRYPT, pp.319-333, 2009.

[10] Erway C C, Kupcu A, Papamanthou C, et al. “Dynamic provable data possession”. ACM Transactions on Information and System Security (TISSEC), 2015, 17(4): 15.

[11] H. Shacham and B. Waters. “Compact proofs of retrievability”. Proc. of Cryptology- ASIACRYPT, 5350, pp.90-107, 2008.

[12] D. Boneh , B. Lynn, and H. Shacham. “Short signatures from the weil pairing”. In Proc. of Asiacrypt 2001, pp.514-532, 2001.

[13] Y. Yu, J.B. Ni, M. H. Au, H.Y. Liu, H.Wang and C.X. Xu. “Improved security of a dynamic remote data possession checking protocol for cloud storage”. Expert Syst. Appl. 41(17), pp.7789-7796, 2014.

[14] Y. Yu, M.H. Au, Y. Mu, S.H. Tang, J. Ren, W. Susilo and L.J. Dong. “Enhanced privacy of a remote data integrity-checking protocol for secure cloud storage”. International Journal of Information Sececurity. 14(4), pp.307-318, 2015.

[15] Jiangtao Li, Lei Zhang, Joseph K. Liu, Haifeng Qian, Zheming Dong, Privacy-Preserving Public Auditing Protocol for Low Performance End Devices in Cloud, IEEE Transactions on Information Forensics and Security 11(11): 2572-2583 (2016).

[16] Wang C, Zhang B, Ren K, et al. Privacy-assured outsourcing of image reconstruction service in cloud. IEEE Transactions on Emerging Topics in Computing, 2013, 1(1): 166-177.

[17] C. Wang, K. Ren, W. Lou, and J. Li. “Toward publicly auditable secure cloud data storage services”. IEEE Network, 24, pp.19-24, 2010.

[18] Y. Yu, J.B. Ni, M. H. Au, Y. Mu, B.Y. Wang and H. Li. “Comments on a Public Auditing Mechanism for Shared Cloud Data Service”. IEEE Transantions on Services Computing, 8(6),pp.998-999, 2015.

[19] Y. Zhang, J. Ni, X., Y. Wang, Y. Yu. “Provable multiple replication data possession with full dynamics for secure cloud storage”. Concurrency and Computation: Practice and Experience, 28(4), pp.1161- 1173, 2016.

[20] A. Shamir. “Identity-based cryptosystems and signature schemes”. Advances in cryptology. pp.47-53, 1985.

[21] J. N. Zhao, C. X. Xu, F. G. Li, and W. Z. Zhang. “Identity-Based Public Verification with Privacy-Preserving for Data Storage Security in Cloud Computing”. IEICE Transactions, 96- A(12), pp.2709- 2716, 2013.

[22] Y. Yu, M. H. Au, G. Ateniese, X. Huang, W. Susilo, Y. Dai, G. Min. “Identity-Based Remote Data Integrity Checking With Perfect Data Privacy Preserving for Cloud Storage”. IEEE Trans. Information Forensics and Security 12(4), pp.767-778, 2017.

[23] Y. Yu, Y. F. Zhang, Y. Mu, W. Susilo and H. Y. Liu. “Provably Secure Identity Based Provable Data Possession”. Provable Security, pp.310-325, 2015.

[24] H. Q.Wang. “Identity-Based Distributed Provable Data Possession in Multicloud Storage”. IEEE Transactions on Services Computing, 8(2), pp.328-340, 2015.

[25] ShulanWang, Kaitai Liang, Joseph K. Liu, Jianyong Chen, Jianping Yu, Weixin Xie, Attribute-Based Data Sharing Scheme Revisited in Cloud Computing, IEEE Transactions on Information Forensics and Security 11(8): 1661-1673 (2016).

[26] Kaitai Liang, Man Ho Au, Joseph K. Liu, Willy Susilo, Duncan S. Wong, Guomin Yang, Yong Yu, Anjia Yang, A Secure and Expressive Ciphertext-Policy Attribute-Based Proxy Re-Encryption for Cloud Data Sharing, Future Generation Computer

Systems 52:95-108 (2015).

[27] Joseph K. Liu, Man Ho Au, Xinyi Huang, Rongxing Lu, Jin Li, Fine-grained Two-factor Access Control for Web-based Cloud Computing Services, IEEE Transactions on Information Forensics and Security 11(3): 484-497 (2016).

[28] Y. Li, Y. Yu, G. Min, W. Susilo, J. Ni, K-K. R. Choo. “Fuzzy Identity-Based Data Integrity Auditing for Reliable Cloud Storage Systems”. IEEE Transactions on Dependable and Secure Computing. http://dx.doi.org/10.1109/TDSC.2017.2662216.

[29] S. F. Shahandashti, R. Safavi-Naini. “Threshold attribute-based signatures and their application to anonymous credential systems”. In AFRICACRYPT’09, Gammarth, Tunisia, 2009, 198-216.

[30] D. Boneh and M. Franklin. “Identity-based encryption from the weil pairing”, Proc. of CRYPTO 2001, LNCS 2139, pp.213-229, 2001.

[31] J. Katz. “Digital Signatures”. Springer Science and Business Media, 2010.

[32] A. Shamir. “How to share a secret”. Communications of the ACM, 22(11), pp.612-613,1979.

[33] H. Maji, M. Prabhakaran, and M. Rosulek. “Attribute based signatures: Achieving attribute privacy and collusion-resistance”. 2008. Available at http://eprint.iacr.org/2008/328.

[34] https://certivox.org/display/EXT/MIRACL.

[35] D. Freeman, M. Scott and E. Teske, “A taxonomy of pairingfriendly elliptic curves”. J. Cryptol, 23(2), pp.224-280 (2010).

[36] Satish, Karuturi S R V, and M Swamy Das. "Quantum Leap in Cluster Efficiency by Analyzing Cost-Benefits in Cloud Computing." In Computer Science and Engineering by Auroras Scientific Technological & Research Academy Hyderabad, vol. 17, no. 2, pp. 58-71. Accessed 2018.