IJARIIT INTERNATIONAL JOURNAL OF ADVANCE RESEARCH, IDEAS AND INNOVATIONS IN TECHNOLOGY

ISSN: 2454-132X
Impact factor: 4.295
(Volume3, Issue3)
Available online at www.ijariit.com

Secure and Constant Cost Public Cloud Storage Auditing with Deduplication

Shubham Srivastav
Mtech Student
RITM Lucknow, Uttar Pradesh

Harikesh Pandey
Assistant Professor
RITM Lucknow, Uttar Pradesh

***Abstract**: Information uprightness and capacity effectiveness are two essential necessities for distributed storage. Verification of Retrievability (POR) and Confirmation of Information Ownership (PDP) strategies guarantee information respectability for distributed storage. Evidence of Proprietorship (POW) enhances stockpiling proficiency by safely evacuating superfluously copied the information on the capacity server. Be that as it may, an insignificant blend of the two systems, with a specific end goal to accomplish both information trustworthiness and capacity proficiency, brings about non-minor duplication of metadata (i.e., validation labels), which repudiates the destinations of POW. Late endeavors to this issue present huge computational and correspondence costs and have likewise been demonstrated not secure. It requires another answer for bolster effective and secure information trustworthiness inspecting with capacity deduplication for distributed storage. In this paper, we take care of this open issue with a novel plan in view of strategies including polynomial-based validation labels and homomorphic straight authenticators. Our plan permits deduplication of both documents and their relating confirmation labels. Information respectability examining and capacity deduplication are accomplished all the while. Our proposed plan is likewise portrayed by consistent ongoing correspondence and computational cost on the client side. Open inspecting and group reviewing are both upheld. Henceforth, our proposed conspire beats existing POR and PDP plans while giving the extra usefulness of deduplication. We demonstrate the security of our proposed conspire in light of the Computational Diffie-Hellman issue, the Static Diffie-Hellman issue, and the t-Solid Diffie-Hellman issue. Numerical investigation and trial come about on Amazon AWS demonstrate that our plan is proficient and versatile.*

Keywords: Integrity, Retrievability, Deduplication, Authentication.

I. INTRODUCTION

Distributed storage has been progressively pervasive in light of its focal points [1]. As of now, business distributed storage administrations including Microsoft Skydrive, Amazon S3, and Google Distributed storage have pulled in a large number of clients. Distributed storage remains for the enormous figuring framework as well as the financial matters of scale. Under such a pattern, it ends up noticeably pressing to guarantee the nature of information stockpiling administrations which include two continuous worries from both cloud clients and cloud specialist organizations: information uprightness and capacity productivity. On one hand, with the numerous information misfortune and debasement occasions announced for those best-known cloud specialist co-ops [2], [3], information proprietors, who are additionally cloud clients, want to intermittently review the honesty of their outsourced information. Then again, for cloud specialist organizations it is important to enhance the productivity of distributed storage to exploit the financial matters of scale. As per a current review by EMC [4], 75% of today's computerized information are copied duplicates. To diminish the superfluously excess duplicates, the distributed storage servers would deduplicate by keeping just a single or few duplicates for each document and making a connection to the record for each client who makes a request to store the record. Cloud clients (i.e., information proprietors) should dependably have the capacity to check the uprightness of the record whenever. For capacity effectiveness, it is attractive to deduplicate both the record and the metadata (e.g., confirmation labels) required for information trustworthiness check. Taking noxious or trouble making clients or cloud servers into thought, the cloud server needs to check that the client really possesses the document before making a connection to this record for him/her; the client likewise needs to affirm that the cloud really has the record in its stockpiling and review the uprightness of the document all through its lifetime.

II. RELATED WORKS

Considering only integrity auditing for data outsourced to cloud servers, a number of POR schemes [5], [6], [7], [8] and PDP schemes [9], [10], [11], [12] have been proposed. Among that ref.[5] has the best performance which achieves public auditing at a constant communication cost. Similar to other POR or PDP schemes, users in ref.[5] still need to perform O(k) multiplication and addition operations over the underlying field, where k is the number of checking data blocks. Batch auditing for multiple requests scenarios is not supported in ref.[5]. For secure storage deduplication, Halevi et al. [13] introduced the first POW scheme based on the Merkle hash tree. Pietro et al. [14] enhanced ref.[13] and proposed a secure POW scheme which reduces the computational cost to a constant number of pseudorandom function operations. Nevertheless, these POW schemes do not consider data integrity auditing. To achieve both data integrity auditing and storage deduplication, one trivial solution is to directly combine an existing POR/PDP scheme with a POW scheme. This trivial solution, however, will result in an O(W) storage overhead for each file, where W is the number of owners of this file. This is because of the data owners, lacking mutual trust, need to separately store their own authentication tags in the cloud for file integrity auditing. Since these tags are created for auditing the same file, storing O(W) such copies represents a type of duplication which contradicts the objective of POW for saving storage cost. For efficient proof of storage with deduplication (POSD), Zheng et al. [15] proposed a scheme aiming at providing both public data integrity auditing and secure storage deduplication. In ref.[15] the communication cost and computational cost on the user side are linear to the number of elements in each data block as well as the number of checking blocks during the integrity auditing process. With an increasing population of mobile users, who access cloud through mobile apps (e.g., iAWS, iCloud, etc.) and have constrained computational resources and bandwidth (e.g., mobile phones with limited data plan), such a communication and computational complexity could represent a barrier to accessing the cloud storage service. Preferably, computational cost and communication cost on the user side shall be constant. Moreover, ref.[15] has been proven not secure [16]. Specifically, by setting the elements in secret keys to some special values, a data owner who outsources data to the cloud server is able to use the server as a malware distribution platform. Therefore, it still calls for a new solution to support efficient and secure data integrity auditing with storage deduplication for cloud storage.

Our Contribution

In this paper, we solve this open problem and propose the first Public and Constant cost storage integrity Auditing scheme with secure Deduplication (PCAD) based on techniques including polynomial-based authentication tags and homomorphic linear authenticators. The proposed PCAD scheme is characterized by following salient properties: 1) PCAD is able to securely “deduplicate" the authentication tags by aggregating the tags of the same file from different owners, and hence make the storage overhead independent of the number of owners of the file; 2) the communication cost in our PCAD scheme is made constant thanks to our novel design of polynomial-based authentication tags and secure data aggregation; 3) the computational cost on cloud users is also constant because most computational tasks can be securely offloaded to the cloud server; 4) PCAD supports public auditing, i.e., the data integrity auditing operation can be securely performed by any third party other than the owner(s); 5) PCAD allows batch auditing, i.e., multiple auditing requests can be securely aggregated, which substantially reduces the auditing cost for simultaneous requests; 6) in PCAD data integrity auditing and secure deduplication operations can be performed without the help of existing owners. Noticeably, our PCAD outperforms existing POR and PDP schemes while providing an additional functionality of data deduplication. The main idea of our scheme can be summarized as follows: The data owner outsources the erasure-coded file to the cloud server together with the corresponding authentication tags. To audit, the integrity of the outsourced file, a user (who may not be the owner) challenges the cloud with a challenging message. On receiving the message, the cloud generates the proof information based on the public key and sends it to the user. The user verifies the data integrity with the proof information, using our verification algorithm. In order to deduplicate data, when a user wants to upload a data file that already exists in the cloud, the cloud server executes a checking algorithm to see whether or not this user actually possesses the whole file. If the user passes the checking, he/she can directly use the file existed on the server without uploading it again. The security of our proposed scheme is proven under the Computational Diffie- Hellman (CDH) problem, the Static Diffie-Hellman problem, and the t-Strong Diffie-Hellman (SDH) problem. Thorough analysis and experimental results on Amazon EC2 Cloud show that our scheme is efficient and scalable. Our main contributions can be summarized as below.

- We propose the first public and constant cost storage integrity auditing scheme with secure deduplication, which can also efficiently handle multiple auditing requests with batch operations.
- We formally prove the security of PCAD. The advantages of PCAD are validated by both numerical analysis and real experiments on Amazon AWS Cloud.
- Our design of polynomial based authentication tag can be used as an independent solution for other related applications, such as verifiable SQL search, encrypted keyword search, etc.

III. MODELS AND GOALS

A. System Model

In this work, we consider a system consisting of four major entities: Trust Authority (TA), Data Owner, Cloud Server and User. The TA in our design is a party only responsible for generating part of the public keys for the system and will go off-line after the key generation. The TA will not participate in any other operations during integrity auditing and deduplication processes. The data owner has a number of data files and stores them on the cloud server together with the authentication tags. Each owner in our design will also generate its own secret keys and public keys for authentication tag generation and data integrity verification. A user to whom the owner shares the data files can access and check the integrity of data files using the public key. A user can also be a Third Party Authority (TPA), who has capabilities/expertise and can periodically audit the integrity of data files being stored on the behalf of data owners. When a user wants to upload data files which are already stored in the cloud, the cloud server just creates a link to this file, instead of storing another copy, for this user if the user has been proven a true owner of the file with our scheme. During the integrity auditing and deduplication processes, the user and the cloud server only use the public key and do not need any help from the data owner. While cloud servers are always equipped with abundant computing resources, data owners and users may have constrained computational power or bandwidth (e.g., mobile phones with limited data plan).

B. Security Model

In our PCAD scheme we consider the following factors that may impact integrity of data stored on cloud servers: 1) attackers corrupting data stored on cloud servers; 2) attackers claiming the ownership of file stored in the cloud even if they do not possess the whole file; 3) hardware/software failures of cloud servers and operational errors of system administrator. The cloud server in our scheme is considered as selfish, which may potentially misbehave in order to save resources (e.g., deleting data stored on it). This assumption is consistent with the previous POSD scheme [15]. We also allow attackers, cloud servers, and some data owners to collude with each other in order to pass the integrity verification of valid users.

C. Design Goals

To securely and efficiently verify the integrity of the shared data on a cloud with deduplication, our PCAD scheme should achieve the following properties at the same time:

- Efficiency: the communication cost and computational cost for users to verify the integrity of data stored in the cloud should be constant.
- Functionality: Public data integrity verification and deduplication should be supported at the same time without introducing functionally duplicated authentication tags.
- Correctness: The proposed scheme should accept all valid secret keys and public keys, all valid authentication tags, all valid proof information generated based on valid public keys and all valid data blocks.
- Soundness: Any polynomial-time adversary cannot forge-proof information based on modified data and pass the verification algorithm in our scheme; any polynomial-time adversary without the whole data file cannot pass the ownership checking process; any polynomial-time adversary who colludes with other data owners and cloud servers cannot forge-proof information and pass the integrity verification process.

IV. PERFORMANCE EVALUATION

A. Numerical Analysis

In this section, we numerically analyze our PCAD scheme and compare it with ref.[15], [5], [14]. For simplicity, in the rest of this paper, we use MUL and EXP1 to denote the complexity of one multiplication operation and one exponentiation operation on Group G respectively. Pairing is a bilinear pairing operation.

1) Communication: In our PCAD scheme, the communication cost of the auditing process is caused by the challenging message CM = {K , r} and the proof information Prf = { ῳ̇ , σ , y}. The CM consists of a set K with k block ids and a random number r. The user can randomly challenge k = 600 data blocks to assure at least 99:999% error detection probability. If an error detection probability a fixed parameter, the size of K can be considered as constant and the complexity of challenging message CM is O(1). The proof information is composed a polynomial y and two group elements ῳ̇ and σ. Therefore, the total communication complexity of auditing process in our PCAD scheme is also O(1). In the Deduplication process of our scheme, the user only needs to send d encoded data blocks to the cloud server to prove that it actually owns the whole file. As we discussed in Section III-E, the cloud server only needs challenging 300 blocks or 460 blocks to achieve 95% or 99% confidence whether the user actually owns the whole data file. Therefore, the size of D can be bounded and the total communication complexity of the Deduplication process in our scheme is O(1). Now, we compare our PCAD scheme with existing schemes [15], [5], [14]. In ref.[15], the Auditing process requires the cloud server to send k authentication tags of the challenging blocks and s aggregated data blocks to the user, where s is the number of elements in an encoded block. Thus, its communication complexity during the Auditing process is O(s + k). To perform the Deduplication process, the user needs to sends 2s aggregated data blocks to the cloud server and thus introduces the communication complexity as O(s). Differently, the aggregation of communication information in our design enables our scheme to achieve O(1) communication complexity for both Auditing and Deduplication processes. The POR schemes proposed by Yuan et al. [5] achieve constant communication complexity for the Auditing process same as our PCAD scheme. However, their scheme does not support the Deduplication process and batch auditing and introduces much higher computational cost on the user side (Discuss later in Section IV-A2). Considering the deduplication process only, ref.[14] also requires O(1) communication cost, but their scheme cannot support the data integrity auditing.

2) Computation: KeyGen, Setup, Challenge, Prove, Verify and Deduplication. Among these algorithms, KeyGen and Setup are prepossessing procedures, which are performed by the data owner offline. To produce authentication tags for an encoded file with n blocks, each of which has s elements, the data owner needs (s+2)n EXP and sn MUL operations. Note that the cost in the prepossessing of our scheme is a one-time cost for the data owner. After these prepossessing procedures, the data owner can go off-line. During the data integrity auditing process, the user performs Challenge algorithm to generate the challenging message CM by choosing a constant number of random numbers with negligible cost. On receiving the CM, the cloud server needs (k +s-1) MUL and (s+k) EXP operations to produce the proof information. To verify the integrity of the auditing file, the user performs 3 EXP, 3 MUL, and 4 Pairing operations. Therefore, the computational cost for the user to audit the data integrity of a single file is

O(1)MUL+O(1)EXP+O(1)Pairing. To perform the Deduplication algorithm in our scheme, no computation cost is required for the user. The cloud server needs to perform O(s + d) MUL+O(s) EXP+O(1)Pairing. We now compare our PCAD scheme with existing schemes [15], [5], [14] and summarize, the data integrity auditing process costs a user O(ks)MUL+O(k)EXP operations, and the deduplication processes introduces O(sk)MUL computational complexity to the user, where k is the number of challenging blocks and s is the number of element in a data block. Differently, by outsourcing most computational tasks of both auditing and deduplication processes to the cloud server, our PCAD scheme achieves constant computational cost on users and thus significantly outperforms ref.[15]. Compared with ref.[5] that only supports data integrity auditing, our PCAD scheme reduces the computational complexity on the user from O(k)MUL+O(k)EXP+O(1)Pairing_toO(1)MUL+O(1)EXP+O(1)Pairing. Considering only the deduplication process, ref.[14] requires O(1)PRF operation on the user side that is comparable to our PCAD scheme, where PRF is one pseudorandom function operation.

3) Auditing after Deduplication: In this section, we discuss the storage overhead saved by aggregation of authentication tags in our proposed scheme. Suppose W owners; 1 ≤ w ≤ W pass the deduplication checking of the file F0 existed on the cloud server. As these owners have no mutual trust with each other, each owner needs to store n authentication tags on the cloud server separately for future public integrity auditing of F0, where n is the number of encoded data blocks in F0. If the cloud server directly stores these authentication tags, an O(Wn) storage overhead complexity is introduced to it. Differently, by aggregating the tags for the same data block, the cloud server in our scheme can reduce the storage overhead complexity to O(n). With regard to the computational complexity and communication complexity on an auditing user, it remains the same as the constant level of auditing before deduplication, i.e., O(1)MUL+O(1)EXP+O(1)Pairing computational complexity and O(1) communication complexity.

4) Batch Auditing: In this section, we discuss the communication cost and computational cost saved by our batch auditing design for multiple requests scenarios. Suppose a TPA is hired by T data owners to help them audit the integrity of L outsourced files on the cloud server periodically. If the TPA processes these L auditing requests one by one, it needs 3L EXP, 3L MUL and 4L Pairing operations for computation, and 2L group elements, L random numbers and L polynomials for communication. With our batch auditing design, the cloud server can aggregate L _i into one group element and use one random number, one polynomial instead of L ones. Thus, compared with processing requests sequentially, our batch auditing design can help the TPA and the cloud server to save about 50% communication cost. From the perspective of computational cost, our batch auditing design enables the TPA to reduce the number of pairing operations from 4L to 3L, which is much more expensive compared with MUL and EXP operations. Therefore, about 25% computational tasks are saved for the TPA with our batch auditing design. Assume c% files are from same data owners, our batch auditing design can save additional c% communication cost and c% computational cost.

B. **Experimental Result**

To show that our proposed PCAD scheme is efficient and scalable, we conducted experiments on Amazon EC2 Cloud Platform using JAVA with Java Pairing-Based Cryptography library (jPBC) [24]. The machine we used for the TPA is a laptop running Mint Linux 13 with 2.50GHz Intel i5-2520M CPU and 8GB memory. For the cloud server, we utilize nodes that run Red Hat Enterprise Linux 6.3 with 8 Cores CPU and 16GB memory. We set the security parameter S= 160, which achieves 1024-bits RAS equivalent security on Group. All experimental results represent the mean of 10 trials. To verify our PCAD scheme's constant communication cost and computational cost on the user side, we vary the number of data blocks stored on the cloud server from 1000 to 10000. As shown in Fig.1 (a), the computational cost of users for performing an integrity auditing task almost keeps around 420 ms when the number of data blocks in the auditing files increases. With regard to the communication cost, it also remains stable as about 622 Bytes when the number of data blocks in the auditing files increases, Note that, although we do not perform experiment on more large files, it is easy to obtain that both computational cost and communication cost of our scheme are constant from the analysis.

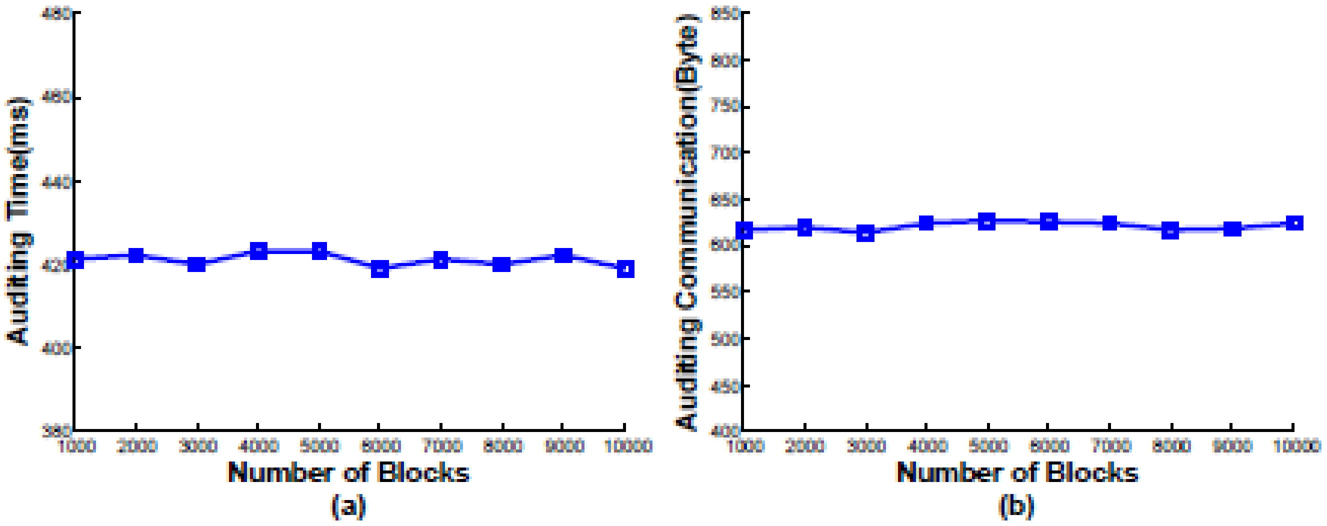

Fig. 1. (a) Auditing Time on Users (b) Auditing Communication Cost on Users

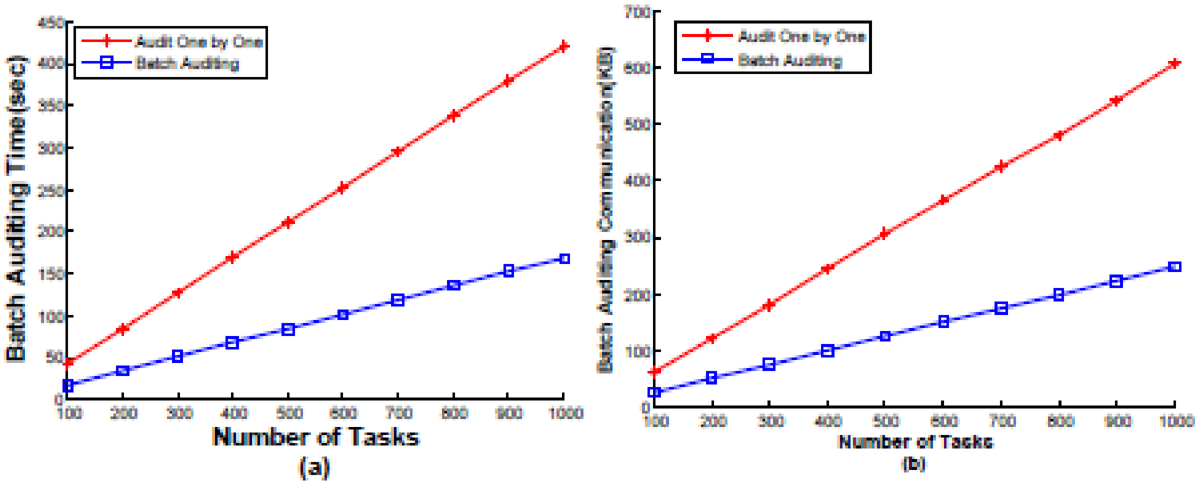

Fig. 2. (a) Auditing Time on TPA (b) Auditing Communication Cost on TPA

To show the benefits of our batch auditing design for multiple auditing tasks scenarios, we change the number of tasks a TPA needs to perform from 100 to 1000. Among these files, 20% files are from same data owners. As we demonstrated above, the number of data blocks in each file does not influence the performance of our scheme, we set the number of data blocks to 5000 in each auditing task. Compared with performing these auditing tasks one by one, Fig.2 (a) shows that the TPA can save about 55% auditing time with batch auditing. From the perspective of communication cost, Fig.2 (b) shows our batch auditing saves about 60% bandwidth for the TPA. Considering the average cost per task, which is computed by dividing total auditing time and total auditing bandwidth cost by the number of tasks respectively.

CONCLUSION

To securely fulfill the two important requirements of cloud storage: data integrity and storage efficiency, a number of schemes have been proposed based on the concepts of POR, PDP, POW, and POSD. However, most existing schemes only focus on one aspect, because the trivial combination of existing POR/PDP schemes with POW schemes can contradict the objects of POW. The only one that simultaneously emphasized both aspects based on the concept of POSD suffers from tremendous computation and computational costs and has been proven not secure. In this work, we filled the gap between POR and POW and proposed a constant cost scheme that achieves secure public data integrity auditing and storage deduplication at the same time. Our proposed scheme enables the deduplication of both files and their corresponding authentication tags. In addition, we extend our design to support batch integrity auditing, and thus substantially save computational cost and communication cost for multiple requests scenarios. The security of our PCAD scheme is proved based on the CDH problem, the Static Diffie-Hellman problem and the t- SDH problem. We validate the efficiency and scalability of our scheme through numerical analysis and experimental results on Amazon EC2 Cloud. Our proposed polynomial based authentication tag can also be used as an independent solution for other related applications, such as verifiable SQL search, encrypted keyword search, etc.

REFERENCE

[1] G. Timothy and M. M. Peter, “The nist definition of cloud computing,” vol. NIST SP - 800-145, September 2011.
[2] “Amazon forum. major outage for Amazon s3 and ec2,” https://forums.aws.amazon.com/thread.jspa?threadID =19714&start=15&tstart=0.
[3] “Business Insider. amazon’s cloud crash disaster permanently destroyed many customers’ data,” http://www.businessinsider.com/amazon-lost-data-2011-4.
[4] J. Gantz and D. Reinsel, “The digital universe decade - are you ready?” http://www.emc.com/collateral/analyst-reports/idc-digitaluniverse- are-you-ready.pdf, May 2010.
[5] J. Yuan and S. Yu, “Proofs of Retrievability with public verifiability and constant communication cost in the cloud,” Proceedings of the ACM ASIACCS-SCC’13, 2013.
[6] H. Shacham and B. Waters, “Compact Proofs of Retrievability,” in Proceedings of the 14th International Conference on the Theory and Application of Cryptology and Information Security, ser. ASIACRYPT ’08, Berlin, Heidelberg, May 2008, pp. 90–107.
[7] A. Juels and B. S. Kaliski, Jr., “Pors: Proofs of retrievability for large files,” in Proceedings of the 14th ACM conference on Computer and communications security, ser. CCS ’07. New York, NY, USA: ACM, 2007, pp. 584–597.
[8] Y. Dodis, S. Vadhan, and D. Wichs, “Proofs of Retrievability via hardness amplification,” in Proceedings of the 6th Theory of Cryptography Conference on Theory of Cryptography, ser. TCC ’09, Berlin, Heidelberg, 2009, pp. 109–127.
[9] G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Peterson, and D. Song, “Provable data possession at untrusted stores,” in Proceedings of the 14th ACM conference on Computer and communications security, ser. CCS ’07. New York, NY, USA: ACM, 2007, pp. 598–609.

[10] G. Ateniese, R. Di Pietro, L. V. Mancini, and G. Tsudik, "Scalable and efficient provable data possession," in Proceedings of the 4th international conference on Security and privacy in communication netowrks, ser. SecureComm '08. New York, NY, USA: ACM, 2008.
[11] C. Erway, A. Küpçü, C. Papamanthou, and R. Tamassia, "Dynamic provable data possession," in Proceedings of the 16th ACM conference on Computer and communications security, ser. CCS '09. New York, NY, USA: ACM, 2009, pp. 213–222.
[12] Q. Wang, C. Wang, K. Ren, W. Lou, and J. Li, "Enabling public auditability and data dynamics for storage security in cloud computing," IEEE Trans. Parallel Distrib. Syst., vol. 22, no. 5, pp. 847–859, 2011.
[13] S. Halevi, D. Harnik, B. Pinkas, and A. Shulman-Peleg, "Proofs of ownership in remote storage systems," in Proceedings of the 18th ACM conference on Computer and communications security, ser. CCS '11. New York, NY, USA: ACM, 2011, pp. 491–500.
[14] R. Di Pietro and A. Sorniotti, "Boosting efficiency and security in proof of ownership for deduplication," in Proceedings of the 7th ACM Symposium on Information, Computer and Communications Security, ser. ASIACCS '12. New York, NY, USA: ACM, 2012, pp. 81–82.
[15] Q. Zheng and S. Xu, "Secure and efficient proof of storage with deduplication," in Proceedings of the second ACM conference on Data and Application Security and Privacy, ser. CODASPY '12. New York, NY, USA: ACM, 2012, pp. 1–12.
[16] K. K. Youngjoo Shin, Junbeom Hur, "Security weakness in the proof of storage with deduplication," Cryptology ePrint Archive, Report 2012/554, 2012, http://eprint.iacr.org/.
[17] D. Boneh and X. Boyen, "Short signatures without random oracles," in EUROCRYPT, 2004, pp. 56–73.
[18] I. S. Reed and G. Solomon, "Polynomial Codes Over Certain Finite Fields," Journal of the Society for Industrial and Applied Mathematics, vol. 8, no. 2, pp. 300–304, 1960.
[19] X. Jia and C. Ee-Chien, "Towards efficient provable data possession," in Proceedings of the 7th ACM Symposium on Information, Computer and Communications Security, ser. ASIACCS '12, Seoul, Korea, 2012.
[20] W. Diffie and M. Hellman, "New directions in cryptography," IEEE Trans. Inf. Theor., vol. 22, no. 6, pp. 644–654, Sep. 1976.
[21] D. R. L. Brown and R. P. Gallant, "The static diffie-hellman problem," Cryptology ePrint Archive, Report 2004/306, 2004, http://eprint.iacr.org/.
[22] A. Kate, G. M. Zaverucha, and I. Goldberg, "Constant-size commitments to polynomials and their applications," in ASIACRYPT, 2010, pp. 177–194.
[23] V. Shoup, A computational introduction to number theory and algebra. New York, NY, USA: Cambridge University Press, 2005.
[24] jPBC, "http://gas.dia.unisa.it/projects/jpbc/."