PUBLIC AUDITING WITH PRIVACY PROTECTION FOR SECURE CLOUD STORAGE

[1]A.VENKATESWARLU, [2]MALLISETTY SRIDHAR

Asst. Professor, Dept of MCA, Audisankara Institute of Technology, Gudur, Tirupati(DT), Andhra Pradesh, India

PG Scholar, Dept of MCA, Audisankara Institute of Technology, Gudur, Tirupati (DT), Andhra Pradesh, India

ABSTRACT: With the help of cloud storage, users may remotely store their data and take advantage of high-quality apps and services that are available on demand from a pool of configurable computing resources, all without having to worry about maintaining and managing local data storage.However, since users no longer physically hold the outsourced data, protecting its integrity in cloud computing is a challenging issue, particularly for users with limited computing resources. Additionally, users should not need to worry about checking the integrity of the cloud storage; they should just be able to utilize it as if it were local. Hence, allowing for public auditability Cloud storage is crucial because customers can utilize a third party auditor (TPA) to verify the accuracy of data that has been outsourced while remaining at ease. The auditing procedure should not create any new online burdens for users or vulnerabilities affecting user data privacy in order to introduce a TPA securely and effectively. In this research, we provide a private public auditing mechanism for a secure cloud storage system. We further extend our finding such that the TPA may effectively and simultaneously conduct audits for a number of consumers. The proposed techniques are provably secure and extremely effective, according to a thorough investigation of security and performance.

I. INTRODUCTION

From the viewpoint of users, including both individuals and IT enterprises, storing data remotely to the cloud in a flexible on-demand manner offers enticing advantages: relief from the burden of storage management, global data access with no restrictions on physical locations, and avoidance of capital expenditure on hardware, software, and personnel maintenances, among others. These benefits are now more enticing than ever thanks to cloud computing, but it also poses new, difficult security risks to users' outsourced data. Considering that cloud service providers (CSP) are distinct Inreality, outsourcing data means that users lose final say over what happens to their information. As a result, the following reasons put the accuracy of the data in the cloud at risk. First off, despite being significantly more robust and reliable than personal computers, cloud infrastructures nevertheless face a wide range of internal and external risks to data integrity. It occasionally happens that well-known cloud services experience outages or security breaches. Second, cloud service providers may act dishonestly toward cloud users on the whereabouts of their outsourced data for a number of reasons. cloud service providers, for instance, could reclaimstorage for financial gain by deleting data that is rarely or never viewed, or even cover up data loss incidents

to protect a company's reputation. In conclusion, even while outsourcing data to the cloud is economically advantageous for long-term, extensive data storage, it does not provide any assurances about the availability and integrity of the data right away. If this issue is not adequately resolved, it could prevent the cloud architecture from being successfully deployed.

Traditional cryptographic primitives for the purpose of data security protection cannot be directly used since consumers no longer physically possess the storage of their data. In particular, due to the high I/O and network transmission costs, downloading the entire set of data is not a workable solution for verifying its integrity. Furthermore, it is frequently insufficient to just detect data corruption while viewing the data, as this does not provide users with the guarantee of correctness for those data that have not been viewed and may even be too late. to repair any damage or loss to the data. The duties of verifying the data accuracy in a cloud

owners and rely only on TPA to store their data securely, are responsible for maintaining the privacy of that data.Safety. Additionally, there are laws requiring that the outsourced data not be disclosed to third parties, such as the US Health Insurance Portability and Accountability Act (HIPAA). One method to address this privacy risk is to use data encryption before outsourcing, but this solution is only complimentary to the privacy-preserving public auditing scheme that will be put forth in this study. Encryption by itself cannot stop data from "flowing away" toward other parties during the auditing process in the absence of a properly developed auditing protocol. As a result, it only reduces the issue of maintaining data privacy to key management, rather than solving it entirely. Due to the potential for decryption keys to be exposed, unauthorized data leaking still poses a hazard.

Therefore, the issue we'll be trying to solve in this work is how to enable a third-party auditing protocol that protects privacy and is independent of data encryption. Our research, which focuses on data storage, is one of the first to support privacy-preserving public auditing in cloud computing. Additionally, an anticipated rise in the number of auditing assignments from various users may be assigned to third party auditor because to the prominence of cloud computing. The question of how to enable the third party auditor to effectively complete several auditing activities in a batch way, that is, concurrently, arises because the individual auditing of these expanding duties can be time-consuming and laborious.

Our work uses the public key based homomorphic linear authenticator (abbreviated HLA) technique to address these issues. This technique allows third party auditor to perform auditing without needing a local copy of the data, significantly reducing communication and computation overhead compared to more traditional data auditing approaches. Our protocol ensures that the third party auditor won't be able to discover any information about the data content stored on the cloud server throughout the effective auditing procedure by combining the HLA with random masking. Our design for the batch auditing benefits further from the authenticator's aggregation and algebraic

II. PROBLEM STATEMENT:

The System and Threat Model

We take into account a cloud data storage service with three distinct parties, as shown in Fig. 1: the third party auditor (TPA), who is entrusted to evaluate the cloud storage service reliability on behalf of the cloud user (U), who has a large volume of data files to be stored in the cloud; the cloud server (CS),

which is managed by the cloud service provider (CSP) to provide data storage services and has significant storage space and computation resources (we will not differentiate cloud server andcloud service provider hereafter); and the cloud user (U), who has a large volume of data files to be stored in the cloud. The cloud server is used by users to maintain and save cloud data. To access and change their information, they can also dynamically engage with the cloud server stored information for a variety of applications. Cloud users may turn to third party auditor for assuring the storage integrity of their outsourced data in order to save compute resources and online workload, while yet trying to keep their data secret from third party auditor.

we take into account the possibility of a semi-trusted cloud server. In other words, it behaves properly most of the time and follows the protocol execution instructions exactly. However, the cloud server might overlook to maintain or purposely erase infrequently used data files that belong to common cloud users in order to benefit themselves. Additionally, to protect its reputation, the cloud server can opt to cover up data corruptions brought on by server hacks or Byzantine errors. We presume that the third party auditor, who is in the auditing business, is trustworthy and impartial and, as a result, has no motivation to work with the cloud server or the users to obstruct the auditing process. The user would suffer harm, nevertheless, if the TPA discovered the outsourced data after the audit.

To authorize the cloud server to respond to the audit delegated to TPA's, the user can sign a certificate granting audit rights to the TPA's public key, and all audits from the third party auditor are authenticated against such a certificate. These authentication handshakes are omitted in the following presentation.

We take into account a cloud data storage service with three distinct parties, as shown in Fig. 1: the cloud service provider (CSP), which is in charge of the cloud server (CS), which has a lot of storage space and computing power to provide data storage services; the cloud user (U), who has a lot of data files to be stored in the cloud; the third party auditor (TPA), who has skills and abilities that cloud users lack and is trusted to evaluate the cloud storage service reliability on behalf of the cloud users.

Design Goals

Our protocol design ought to meet the following security and performance guarantees in order to enable privacy-preserving public auditing for cloud data storage under the aforementioned model.

1) Public auditability: To enable third party auditor to immediately confirm the accuracy of the cloud data without

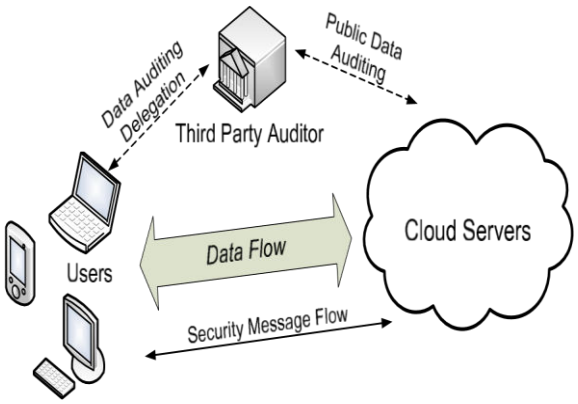


Fig. 1: The architecture of cloud data storage service

obtaining a complete copy of the data or placing an additional online strain on cloud users.

2) Storage accuracy: to make sure that no fraudulent cloud server exists that could pass the TPA audit without really keeping users' data intact.

3) Privacy-preserving: to make sure the third party auditor cannot determine the content of users' data from the data gathered during the auditing process.

4) Batch auditing: To provide third party auditor with secure and effective auditing capabilities to handle multiple auditing delegations from a variety of users, maybe a large number of users, at once.

To execute auditing with the least amount of communication and calculation overhead, the TPA must be lightweight**.**

III. THE PROPOSED SCHEMES

This section outlines our public auditing programme, which offers a total outsourcing solution for data, including both the data's integrity verification and the data itself. We begin with a summary of our public auditing system before talking about two simple methods and their shortcomings. We next go through our primary scheme and demonstrate how to expand it to provide batch auditing for the TPA when several users delegate to it. Finally, we go over how to apply our public auditing approach that protects privacy and supports data dynamics in general.

Definitions and Framework

We adopt the foundation for our privacy-preserving public auditing system and adhere to a definition of previously presented methods in the context of remote data integrity checking.

(KeyGen, SigGen, GenProof, VerifyProof) algorithms. The user runs a key generation algorithm called KeyGen to set up the scheme. The user uses SigGen to create verification metadata, including message authentication code, signatures, and other relevant data that will be utilised for auditing. The cloud server executes GenProof to produce a proof of the accuracy of data storage, and the TPA executes VerifyProof to verify the proof from the cloud server.

There are two stages to running a public auditing system: setup and audit.

• **Setup**: The user runs KeyGen to initialise the system's public and secret parameters. They next use SigGen to pre-process the data file F and create the verification metadata. The user then deletes its local copy and stores the data file F and the verification information at the cloud server. The user can expand or add more metadata to the data file F during pre-processing so that it can be saved on the server.

• **Audit**: To ensure that the cloud server has correctly retained the data file F at the time of the audit, the TPA sends the cloud server an audit message or challenge. GenProof will be used by the cloud server .

Notation and Preliminaries

• F - the data file that needs to be outsourced, represented as a series of n blocks m1,..., mn Zp for a large prime p.

• The message authentication code (MAC) function has the following definition: K 0, 1 0, 1 l, where K stands for the key space.

• Cryptographic hash algorithms H(), h()

Now that the cryptographic foundation is established, we may discuss our suggested system.

Map with bilines. The multiplicative cyclic groups G1, G2, and GT should be of prime order p. Let g1 and g2 be the respective generators of G1 and G2. A bilinear map is one where e: G1 G2 GT such that e(u a, vb) = e(u, v) ab for any u G1, v G2 and a, b Zp. This linearity implies that e(u1 u2, v) = e(u1, v) e for all u1, u2 G1, v G2 (u2, v). Since e is nondegenerate and there is an efficient procedure for computing it, the map should also be non-trivial: e(g1, g2) 6= 1.

The Basic Schemes

As a warm-up, we look at two groups of schemes before presenting our major finding. The first option is a MAC-based solution that has unfavorable systematic flaws, including constrained consumption and stateful verification, which could increase users' online workloads in a

scenario with open auditing. This implies that despite the introduction of a TPA, the auditing problem is still difficult to overcome. The second method, called a homomorphic linear authenticator (HLA) system, is applicable to several current proof-of-storage schemes. We shall identify the cause of all current HLA-based systems' lack of privacy protection. Our primary finding, which eliminates all of these shortcomings, comes from the examination of these fundamental designs. Our primary plan to be provided is founded on a certainHLA system.

Solution based on message authentication code. Using message authentication code to authenticate the data can be done in one of two ways. A simple method is to send the relevant secret key sk to the TPA and upload the data blocks with their message

TPA
Cloud Server

1. Retrieve file tag t, verify its signature, and quit if fail;
{(i,vi)}i∈I

2. Generate a random challenge
chal = {(i, νi)}i∈I ;

3. Compute μ ′ = P i∈I νimi , and also σ = Q i∈I σ νi i ;

4. Randomly pick r ← Zp, and compute R = e(u, v) r and γ = h(R);

5. Compute μ = r + γμ′ mod p ;

{μ,σ,R}← storage correctness proof

6. Compute γ = h(R), and then verify {μ, σ, R} via Equation 1.

authentication code to the server. Later, the TPA can use their message authentication code to randomly retrieve blocks and use sk to verify that they are accurate. Aside from the significant communication and processing complexity (linear in the sampled data amount), the TPA demands an understanding of the data blocks for verification.

One can limit the verification to only consist of equality testing to get around the requirement of the data in TPA verification. This is the concept. The cloud user selects s random message authentication code keys sk 1s, pre-computes s (deterministic) message authentication code, (F)1s for the entire data file F, and publishes these verification metadata (the keys and the message authentication code) to TPA before to data outsourcing. The TPA can request a new keyed message authentication code for comparison with each audit and reveal a secret key sk to the cloud server. As long as it is not possible to fully recover F given MACsk (F) and sk, this preserves privacy. However, it has the following significant flaws:

1) The amount of secret keys that need to be fixed a priori determines how many times a specific data file can be audited. The user must then fully extract the data in order to re-compute and re-publish new message authentication code to TPA
once all secret key possibilities have been exhausted;

2) The TPA must also keep track of the exposed message authentication code keys and update the situation between audits. Maintaining such states for TPA can be challenging and error-prone given the possible huge number of audit delegations from multiple users;

3. It can only support static data and is completely incapable of handling dynamic data. Supporting data dynamics, however, is equally crucial for
cloud storage systems. Our basic protocol will be explained based on static data in order to keep it brType equation here.if and understandable. We'll go over how to modify our protocol for dynamic data.

HLA-based Approach. The HLAapproach can be used to successfully offer public

used. HLAs are some unforgeable verification metadata that authenticate a data block's integrity, just like MACs. The distinction is that HLAs can be combined. An aggregated HLA that validates a linear combination of the individual data blocks can be calculated. An HLA-based proof of storage system functions roughly as follows. Each element of F = (m1, , mn) is still authenticated by the user using a set of HLAs. The F, is kept on the cloud server. The TPA sends a random set of challenge "vi" to validate the cloud storage. In more specific terms, as F,, and I are all vectors, I is an ordered set,or I I should be sent. In order to authenticate, the cloud server then returns = P I mi and an aggregated authenticator (both are computed from F, I and I

Although these HLA-based solutions allow for effective data audits and only use constant bandwidth, they are still insufficient for our needs. This is due to the fact that the linear combination of blocks, = P I vi mi, could possibly divulge user data information to TPA and violates the assurance of privacy preservation. More specifically, the TPA can easily determine the user's data content by solving a system of linear equations provided enough linear combinations of the same blocks are gathered.

Privacy-Preserving Public Auditing Scheme

Overview: We suggest using a special integration of the homomorphic linear authenticator and random masking technique to accomplish privacy-preserving public auditing. In our approach, randomness generated by the server is used to conceal the linear combination of sampled blocks in auditability without having to access the data blocks directl

the server's response. No matter how many linear combinations of the same set of file blocks may be gathered, random masking prevents the TPA from having all the information needed to construct a valid group of linear equations and as a result, it is impossible to determine the user's data content. However, the block authenticator pairs' correctness checking can still be done in a new with the randomness present, as will be demonstrated shortly. To give the auditing protocol public auditability, our design uses a public key based HLA. In particular, we employ the HLA suggested in which is based on the BLS signature, a short signature scheme provided by Boney, Lynn, and Shechem.

Scheme Details: If G1, G2, and GT are multiplicative cyclic groups of order p and e, respectively: As stated in the introduction, G1 G2 GT is a bilinear map.

Let g be a generator for G2. H() is a safe map-to-point hash function that uniformly maps text to G1 at the points 0 and 1. Another hash function, h(), translates the group element of GT evenly to Zp. The planned plan is as follows:

verification equation mentioned above is accurate:

Specifications of Our Protocol. Our protocol successfully achieves public auditability, which is clear. The auditing protocol does not put any possible online strain on users, and the TPA is not required to store or maintain any secret keying material or states in between audits. This method uses a random masking r to conceal a linear combination of the data blocks, ensuring the privacy of user data content during the auditing process.

IV. EVALUATION:

Security Analysis

We assess the proposed scheme's security by looking at how well it upholds the security guarantee outlined, specifically the storage correctness and privacy-preserving property. We begin with the single user case, from which our primary finding emerged. The security assurance of batch auditing for the TPA in a multi-user situation is then demonstrated.

Storage Correctness Guarantee

In order to successfully respond to the TPA, we must demonstrate that the cloud server must faithfully store the data that was taken. The cloud server is now regarded as an enemy. The extractor manages the random oracle h () and responds to the cloud server's hash query. The cloud server out puts, R for a challenge = h(R) returned by the extractor so that the equation below holds.

The extractor then receives, ′ = () / () as a legitimate answer of the underlying storage system evidenceRecall that I = (H(Wi) u mi) x to observe.

Privacy Preserving Guarantee

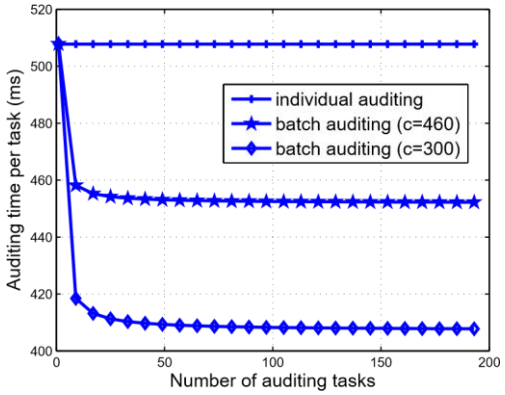
We want to ensure that the TPA cannot determine the substance of users' data from the data gathered during the auditing process.

Theorem 2: TPA cannot recover from the server's response of R.

Proof: In the random oracle model, we demonstrate the existence of a simulator that can nevertheless generate an accurate result even in the absence of knowledge about ′. The TPA is now viewed as an enemy. Given a valid obtained from the cloud server, first choose at random, from Zp, and then set R e ((Qsc i=s1 H(Wi) vi) u, v)/e (, g). Since the simulator is in charge of the random oracle h (), backpatch = h(R). We point out that the random oracle model uses this backpatching technique in the verification of the underlying scheme.

Cost of Privacy-Preserving Protocol

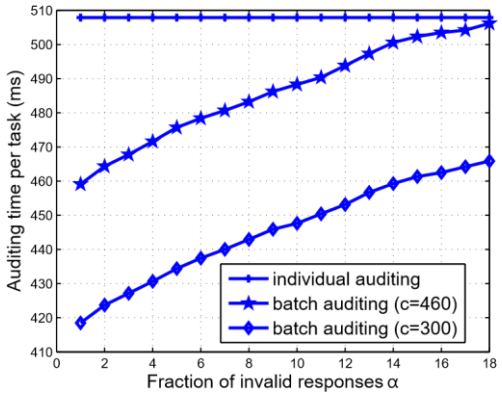
To start, we calculate the price in terms of fundamental cryptographic operations, as shown in Table 1. Let's say the challenge specifies c random blocks.



a time comparison between batch and individual audits. The total auditing time divided by the number of tasks is known as the per task auditing time. We exclude the straight curve for individual auditing when c=300 for the sake of clarity.

during the Audit phase, message chal. In this situation, we calculate the cost of the privacy-preserving auditing in terms of server, auditor, and communication overhead.

The resulting response from the server includes a blindfolded linear combination of sampled blocks = P iI imi + r Zp, where = h(R) Zp, an aggregated authenticator =Q iI vi I G1 and a random factor R = e(u, v)r GT. Hash1 Zp + Addc Zp + Multc+1 Zp, Exp1 GT, and c-MultExp1 G1 (|i|), respectively, are the associated computation costs. The additional cost for preserving user privacy resulting from the random mask R is only a constant: Exp1 GT (|p|) + Mult1 Zp + Hash1 Zp + Add1 Zp, and it has nothing to do with the quantity of sampled blocks c. This is in contrast to the current HLA-based solution for ensuring remote data integrity.



Comparison of batch and individual auditing times when less than 256 responses are legitimate. The total auditing time divided by the number of tasks is known as the per task auditing time.

shown that, even while our approach includes privacy-preserving assurance while does not, the performance of our scheme is nearly identical to that of. The server's response, R, comprises an additional random element R, which is a group element of GT and has the size close to 960 bits, in order to make up for the greater communication cost of our approach compared to.

V. CONCLUSION

In this study, we provide a public auditing method for data storage security in cloud computing that protects privacy. We use the homomorphic linear authenticator and random masking to ensure that the TPA won't discover any information about the data content stored on the cloud server during the effective auditing process, which not only frees up cloud users from the onerous and potentially expensive auditing task but also allays their concerns about the leakage of their outsourced data. We further extend our privacy-preserving public auditing protocol into a multi-user setting, where the TPA may do numerous auditing tasks in a batch manner for greater speed. TPA may concurrently manage several audit sessions from various users for their outsourced data files. A thorough investigation reveals that

REFERENCES

[1] P. Mell and T. Grance, "Draft NIST working definition of cloud computing," Referenced on June. 3rd, 2009 Online at http://csrc.nist.gov/groups/SNS/cloud-computing/index. html, 2009.

[2] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. H. Katz, A. Konwinski, G. Lee, D. A. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, "Above the clouds: A berkeley view of cloud computing," University of California, Berkeley, Tech. Rep. UCB-EECS-2009-28, Feb 2009.

[3] M. Arrington, "Gmail disaster: Reports of mass email deletions," Online at http://www.techcrunch.com/2006/ 12/28/gmail-disasterreports-of-mass-email-deletions/, December 2006.

[4] J. Kincaid, "MediaMax/TheLinkup Closes Its Doors," Online at http://www.techcrunch.com/2008/07/10/ mediamaxthelinkup-closes-its-doors/, July 2008.

[5] Amazon.com, "Amazon s3 availability event: July 20, 2008," Online at http://status.aws.amazon.com/s3-20080720.html, 2008.

[6] S. Wilson, "Appengine outage," Online at http://www. cio-weblog.com/50226711/appengine outage.php, June 2008.

[7] B. Krebs, "Payment Processor Breach May Be Largest Ever," Online at http://voices.washingtonpost.com/securityfix/ 2009/01/payment processor breach may b.html, Jan. 2009.

[8] G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Peterson, and D. Song, "Provable data possession at untrusted stores," in Proc. of CCS'07, Alexandria, VA, October 2007, pp. 598–609.

[9] M. A. Shah, R. Swaminathan, and M. Baker, "Privacypreserving audit and extraction of digital contents,"

Cryptology ePrint Archive, Report 2008/186, 2008.

[10] Q. Wang, C. Wang, J. Li, K. Ren, and W. Lou, "Enabling public verifiability and data dynamics for storage security in cloud computing," in Proc. of ESORICS'09, volume 5789 of LNCS. Springer-Verlag, Sep. 2009, pp. 355–370.

[11] A. Juels and J. Burton S. Kaliski, "Pors: Proofs of retrievability for large files," in Proc. of CCS'07, Alexandria, VA, October 2007, pp. 584–597.

[12] Cloud Security Alliance, "Security guidance for critical areas of focus in cloud computing," 2009, http://www.cloudsecurityalliance.org.

[13] H. Shacham and B. Waters, "Compact proofs of retrievability," in Proc. of Asiacrypt 2008, vol. 5350, Dec 2008, pp. 90–107.

[14] M. A. Shah, M. Baker, J. C. Mogul, and R. Swaminathan, "Auditing to keep online storage services honest," in Proc. of HotOS'07. Berkeley, CA, USA: USENIX Association, 2007, pp. 1–6.

[15] 104th United States Congress, "Health Insurance Portability and Accountability Act of 1996 (HIPPA)," Online at http://aspe.hhs.gov/admnsimp/pl104191.htm, 1996.

[16] S. Yu, C. Wang, K. Ren, and W. Lou, "Achieving secure, scalable, and fine-grained access control in cloud computing," in Proc. of IEEE INFOCOM'10, San Diego, CA, USA, March 2010. [17] D. Boneh, B. Lynn, and H. Shacham, "Short signatures from the Weil pairing," J. Cryptology, vol. 17, no. 4, pp. 297–319, 2004.

[18] A. L. Ferrara, M. Greeny, S. Hohenberger, and M. Pedersen, "Practical short signature batch verification," in Proceedings of CT-RSA, volume 5473 of LNCS. Springer-Verlag, 2009, pp. 309– 324.

[19] G. Ateniese, R. D. Pietro, L. V. Mancini, and G. Tsudik, "Scalable and efficient provable data possession," in Proc. of SecureComm'08, 2008, pp. 1–10. [20] C. Wang, Q. Wang, K. Ren, and W. Lou, "Ensuring data storage security in cloud computing," in Proc. of IWQoS'09, July 2009, pp. 1–9.

[21] C. Erway, A. Kupcu, C. Papamanthou, and R. Tamassia, "Dynamic provable data possession," in Proc. of CCS'09, 2009, pp. 213–222.

[22] R. C. Merkle, "Protocols for public key cryptosystems," in Proc. of IEEE Symposium on Security and Privacy, Los Alamitos, CA, USA, 1980.

[23] G. Ateniese, S. Kamara, and J. Katz, "Proofs of storage from homomorphic identification protocols," in ASIACRYPT, 2009, pp. 319–333.

[24] M. Bellare and G. Neven, "Multi-signatures in the plain public key model and a general forking lemma," in ACM Conference on Computer and Communications Security, 2006, pp. 390–399.

[25] Y. Dodis, S. P. Vadhan, and D. Wicks, "Proofs of retrievability via hardness amplification," in TCC, 2009, pp. 109–127.