Privacy-Preserving Public Auditing with Fair Arbitration for Cloud Data

D. Parameswari
Assistant Professor
Department of Information Technology Kongunadu College Of Engineering and Technology Trichy, India

R. Aruna
Assistant Professor
Department of Information Technology Kongunadu College Of Engineering and Technology Trichy, India

S. Lavanya
Assistant Professor
Department of Information Technology Kongunadu College Of Engineering and Technology Trichy, India

T. K. Revathi
Assistant Professor
Department of Information Technology Kongunadu College Of Engineering and Technology Trichy, India

***Abstract*— Cloud storage is one of the service provided by cloud computing in which data is maintained, managed, backed up remotely and made available to users over a network. The user is concerned about the integrity of data stored in the cloud as the users data can be attacked or modified by outside attacker. Therefore a new concept called data auditing which introduced which check the integrity of data with the help of an third party auditor(TPA). In this project, we propose Proof of Retrieval, a privacy-preserving auditing scheme for shared data with large groups in the cloud. We utilize hash signatures to compute verification information on shared data, so that the TPA is able to audit the correctness of shared data, but cannot reveal the identity of the signer on each block. Hash signature and Keys are generated by Merkle Hash Tree. We can implement auditing scheme to perform efficient public auditing to protect both identity and data privacy in cloud environments. And also users can access the data from data owner through cloud provider in real time dynamic cloud environment.**

Keywords—Public Auditing, Arbitration, Cloud Storage Security

I. INTRODUCTION

Cloud computing is known as distributed computing, due to its long list of unprecedented advantages in the IT history: on-demand self-service, ubiquitous network access, location independent resource pooling, rapid resource elasticity, usage-based pricing and transference of risk. As a disruptive technology with profound implications, cloud computing is transforming the very nature of how businesses use information technology. One fundamental aspect of this paradigm shifting is that data are being centralized or outsourced to the cloud. From users' perspective, including both individuals and IT enterprises, storing data remotely to the cloud in a flexible on-demand manner brings appealing benefits: relief of the burden for storage management, While cloud computing makes these advantages more appealing than ever, it also brings new and challenging security threats toward users' outsourced data.

Since cloud service providers (CSP) are separate administrative entities, data outsourcing is actually relinquishing user's ultimate control over the fate of their data. As a result, the correctness of the data in the cloud is being put at risk due to the following reasons. First of all, although the infrastructures under the cloud are much more powerful and reliable than personal computing devices, they are still facing the broad range of both internal and external threats for data integrity. Second, there do exist various motivations for CSP to behave unfaithfully toward the cloud users regarding their outsourced data status.

To address these problems, our work utilizes the technique of public auditing, which enables TPA to perform the auditing without demanding the local copy of data and thus drastically reduces the communication and computation overhead as compared to the straightforward data auditing approaches. By integrating the Merkle Hash Tree with signature, our protocol guarantees that the TPA could not learn any knowledge about the data content stored in the cloud server (CS) during the efficient auditing process. The aggregation and algebraic properties of the authenticator further benefit our design for the batch auditing. The contribution of the project is:Motivate the public auditing system of data storage security in cloud computing and provide a privacy-preserving auditing protocol.

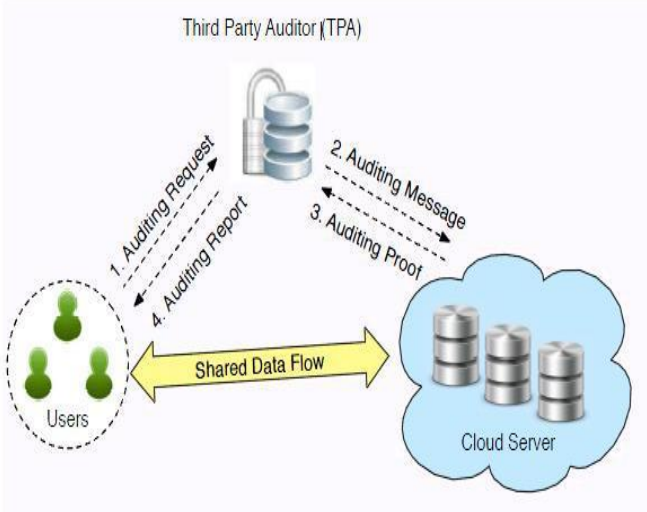

Fig.1 Cloud Storage Public auditing

II. LITERATURE SURVEY

A. PRIVACY-PRESERVING PUBLIC AUDITING FOR SECURE CLOUD STORAGE

In privacy-preserving public auditing in Cloud Computing, with a focus on data storage[1]. Besides, with the prevalence of Cloud Computing, a foreseeable increase of auditing tasks from different users may be delegated to TPA. As the individual auditing of these growing tasks can be tedious and cumbersome, a natural demand is then how to enable the TPA to efficiently perform multiple auditing tasks in a batch manner, i.e., simultaneously. To address these problems, this proposed work utilizes the technique of public key based homomorphism linear authenticator (or HLA for short), which enables TPA to perform the auditing without demanding the local copy of data and thus drastically reduces the communication and computation overhead as compared to the straightforward data auditing approaches. By integrating the HLA with random masking, protocol guarantees that the TPA could not learn any knowledge about the data content stored in the cloud server during the efficient auditing process. The aggregation and algebraic properties of the authenticator further benefit our design for the batch auditing. To authorize the CS 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 TPA are authenticated against such a certificate.

B.FAIR AND DYNAMIC PROOFS OF RETRIEVABILITY

In the concept of fair and dynamic proof of retrivability (FDPOR), a useful extension of static POR in practice[2]. Then, present a formal definition of FDPOR and its security properties called soundness and fairness. We explain why the extension of static POR to FDPOR is non-trivial and discuss in detail: (i) the state of- the-art static POR scheme is insecure when directly used in the setting of dynamic POR; (ii) Need a new authenticated data structure to ensure soundness in the setting of dynamic POR; (iii) We need a special kind of technique to ensure fairness in the setting of dynamic POR; (iv) Need to use error-correcting code in a fashion different from its counterpart in the setting of static POR. These observations guided us in designing an efficient FDPOR scheme, which simultaneously offers both retrivability and fairness in the setting of dynamic data. Specifically, scheme is built on top of two new building-blocks, which might be of independent value. On the other hand, if it is appropriate to assume that the number of challenges issued by data owner is bounded from above and the dynamic data operations are only append-like, then a very efficient PDP scheme can be found.Using the Template

C. ENABLING PUBLIC AUDITABITITY AND DATA DYNAMIC FOR STORAGE SECURITY IN CLOUD COMPUTING

Cloud Computing has been envisioned as the next-generation architecture of IT Enterprise[3]. It moves the application software and databases to the centralized large data centers, where the management of the data and services may not be fully trustworthy. This unique paradigm brings about many new security challenges, which have not been well understood. This work studies the problem of ensuring the integrity of data storage in Cloud Computing. In particular, Here consider the task of allowing a third party auditor (TPA), on behalf of the cloud client, to verify the integrity of the dynamic data stored in the cloud. The introduction of TPA eliminates the involvement of the client through the auditing of whether his data stored in the cloud are indeed intact, which can be important in achieving economies of scale for Cloud Computing. The support for data dynamics via the most general forms of data operation, such as block modification, insertion, and deletion, is also a significant step toward practicality, since services in Cloud Computing are not limited to archive or backup data only. While prior works on ensuring remote data integrity often lacks the support of either public auditability or dynamic data operations, this project eves both. Here first identify the difficulties and potential security problems of direct extensions with fully dynamic data updates from prior works and then show how to construct an elegant verification scheme for the seamless integration of these two salient features in our protocol design. In particular, to achieve efficient data dynamics, Here improve the existing proof of storage models by manipulating the classic Merkle Hash Tree construction for block tag authentication.

III.ALOGRITHM AND IMPLEMENTATIONS

A.SYSTEM ARCHITECTURE

The system model in this project involves three parties: the cloud server, a group of users and a public verifier. There are two types of users in a group: the original user and a number of group users. The original user initially creates shared data in the cloud, and shares it with group users. Both the original user and group users are members of the group. Every member of the group is allowed to access and modify shared data. Shared data and its verification metadata (i.e. signatures) are both stored in the cloud server. A public verifier, such as a third-party auditor (TPA) providing expert data auditing services or a data user outside the group intending to utilize shared data, is able to publicly verify the integrity of shared data stored in the cloud server. When a public verifier wishes to check the integrity of shared data, it first sends an auditing challenge to the cloud server. After receiving the auditing challenge, the cloud server responds to the public verifier with an auditing proof of the possession of shared data. Then, this public verifier checks the correctness of the entire data by verifying the correctness of the auditing proof. Essentially, the process of public auditing is a challenge and-response protocol between a public verifier and the cloud server.

- **Public Auditing** - A public verifier is able to publicly verify the integrity of shared data without retrieving the entire data from the cloud.
- **Correctness** - A public verifier is able to correctly verify shared data integrity.
- **Unforgetability** - Only a user in the group can generate valid verification metadata (i.e., signatures) on shared data.
- **Identity Privacy** - A public verifier cannot distinguish the identity of the signer on each block in shared data during the process of auditing.

With cloud computing and storage, users are able to access and to share resources offered by cloud service providers at a lower marginal cost. It is routine for users to leverage cloud storage services to share data with others in a group, as data sharing becomes standard feature in most cloud storage offerings, including Dropbox, iCloud and Google Drive. The integrity of data in cloud storage, however, is subject to

skepticism and scrutiny, as data stored in the cloud can easily be lost or corrupted due to the inevitable hardware/software failures and human errors. The traditional approach for checking data correctness is to retrieve the entire data from the cloud, and then verify data integrity by checking the correctness of signatures or hash values of the entire data. Certainly, this conventional approach able to successfully check the correctness of cloud data. However, the efficiency of using this traditional approach on cloud data is in doubt. The main reason is that the size of cloud data is large in general. Downloading the entire cloud data to verify data integrity will cost or even waste user's amounts of computation and communication resources, especially when data have been corrupted in the cloud. Recently, many mechanisms have been proposed to allow not only a data owner itself but also a public verifier to efficiently perform integrity checking without downloading the entire data from the cloud, which is referred to as public auditing. In these mechanisms, data is divided into many small blocks, where each block is independently signed by the owner; and a random combination of all the blocks instead of the whole data is retrieved during integrity checking. A public verifier could be a data user (e.g. researcher) who would like to utilize the owner's data via the cloud or a third-party auditor (TPA) who can provide expert integrity checking service and it can be described in Fig.2.

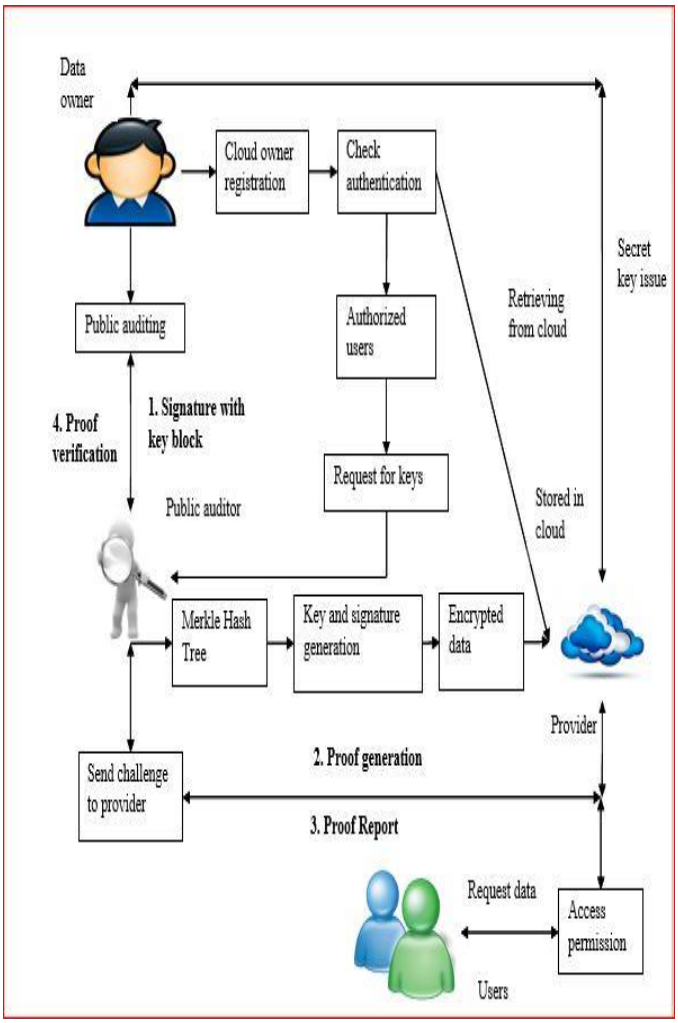

Fig.2 System Architecture

B.MERKLE HASH TREE

To achieve privacy-preserving public auditing[1], propose to uniquely integrate the linear authenticator with binary tree technique. In our protocol, the linear combination of sampled blocks in the server's response is masked with randomness generated by the server. With random masking, the TPA no longer has all the necessary information to build up a correct group of linear equations and therefore cannot derive the user's data content, no matter how many linear combinations of the same set of file blocks can be collected. On the other hand, the correctness validation of the block-authenticator pairs can still be carried out in a new way which will be shown shortly, even with the presence of the randomness. Our design makes use of a public key-based MHT, to equip the auditing protocol with public audit ability. A MHT Encryption scheme is comprised of a tuple of algorithms (Gen, E,D, Eval), and is defined with respect to a circuit C with t inputs. Though a MHT scheme can be either a public-key or symmetric-key system, we will define it as a public-key system here. The key generation algorithm Gen takes the security parameter 1k as input, and outputs the public key and private key for the system

Assume that messages $M \in \{0, 1\}\, l\,(k)$ (1)

The encryption algorithm E takes a public key and a message as input, and outputs a ciphertext C, (Notation: $C \leftarrow E(pk,M)$ for $M \in \{0, 1\}l(k)$) (2)

The decryption algorithm D takes a secret key and a ciphertext, and returns a message, (Notation: $M \leftarrow D(sk,C)$ and $M \in \{0, 1\}l$).Finally, the evaluation algorithm Eval takes as input a public key, a description of a t-input circuit C, and t ciphertexts C1, . . . ,Ct such that $Ci \leftarrow E(pk,Mi)$, and produces as output C*, (Notation: $C^* \leftarrow Eval(pk,C, C1, \ldots ,Ct)$).

We add a new correctness property to the standard correctness requirement for an encryption scheme as follows. We say that an encryption scheme is homomorphic with respect to a t-input circuit C if $\forall k, \forall M1, \ldots ,Mt$, $Pr[(pk, sk) \leftarrow Gen(1k); C1, \ldots Ct \leftarrow E(pk,M1), \ldots , E(pk,Mt); C^* \leftarrow Eval(pk,C, C1, \ldots ,Ct) : D(sk,C^*) = C(M1, \ldots ,Mt)] = 1$.

Similarly, a scheme with respect to a family of circuits {Ci} if the correctness property holds for any circuit $C \in \{Ci\}$. Note that so far, our definition makes no requirement that the output C* of Eval should look like a standard ciphertext. Indeed, without some additional restriction on C*, every standard encryption scheme (Gen, E,D) can be trivially modified to yield a homomorphic encryption scheme (Gen', E',D', Eval') with respect to all circuits as follows.

Gen' runs as Gen.

E' runs as E.

The Eval' is constructed to take a public key, a circuit description, and up to t ciphertexts, and then output the circuit description concatenated with each of the ciphertexts, as $C^* \leftarrow Eval''(pk,C, C1, \ldots ,Ct) = C|C1| \ldots |Ct$, with | used to denote concatenation.

On special cipher texts C* containing a circuit description, D' parses its input into C, C1, . . . ,Ct, runs the original decryption algorithm D on the ciphertexts to obtain messages $Mi \leftarrow D(sk,Ci)$, and runs the circuit C on these messages, to obtain $D'(sk,C^*) = C(M1, \ldots ,Mt)$, satisfying the homomorphic correctness property. On ciphertexts without circuit descriptions, $D'(sk,C)$ simply returns $D(sk,C)$.

C.OPOR (Public Auditing)

With the establishment of privacy-preserving public auditing, the TPA may concurrently handle multiple auditing upon different users' delegation. The individual auditing of these tasks for the TPA can be tedious and very inefficient. Given K auditing delegations on K distinct data files from K different users, it is more advantageous for the TPA to batch these multiple tasks together and audit at one time. Keeping this natural demand in mind, we slightly modify the protocol in a single user case, and achieve the aggregation of K verification equations (for K auditing tasks) into a single one. As a result, a secure batch auditing protocol for simultaneous auditing of multiple tasks is obtained.

Verify file tag tk for each user k, and quit if fail

For each user k (1)

Generate a random challenge

Compute as single user case;

Chal = {(I, Vi)} i

Compute R=R1, R2,….Rk

L = vk1||vk2||…..||vkk

Compute µk :

Compute for each user k and do batch auditing

D.CLOUD FRAMEWORK:

Clouds are the hottest issue in the field of IT from a year now. Introduction of cloud computing has made a revolutionary change in the field of IT. Cloud computing is a most recent area which offers a different model for IT. Cloud computing is emerging technology which consists of existing techniques combined with new technology paradigms. In this technology, we shared different resources like software's, hardware's and information is provided to its users and other peoples on internet whenever demanded. Today's world relies on cloud computing to store different data such as their public as well as some personal information which is needed by the user itself or some other persons. Cloud service is any service offered to its users by cloud. As cloud computing comes in service there are some drawbacks such as privacy of user's data, security of user data is very important aspects. Cloud computing is demand on shared computing resources. With the continuous development of cloud computing technology, its appliance is more and more widely.

Now a days, cloud computing is often used with different synonymous like grid computing, cluster computing, distributed computing, autonomic computing. Privacy is an important issue in cloud computing, whenever user wants to make use of data that involve individual sensitive information. With the rapid development of internet technology, privacy preserving data publication has become one of the most important research topics and become a serious concern in publication of personal data in recent years. However, for data owners who are becoming increasingly concerned about their privacy of the data which contains some personal information about individuals.

In this module, cloud data storage service three different entities such as the cloud user, who has large amount of data files to be stored in the cloud; the cloud server, which is managed by the cloud service provider to provide data storage service and has significant storage space and computation resources; the third-party auditor, who has expertise and capabilities that cloud users do not have and is trusted to assess the cloud storage service reliability on behalf of the user upon request.

E.KEY MANAGEMENT:

Merkle hash tree contains three algorithms: KeyGen, Sign and Verify. In KeyGen, each user in the group generates his/her public key and private key. In Sign, a user in the group is able to generate a signature on a block and its block identifier with his/her private key and all the group members' public keys. A block identifier is a string that can distinguish the corresponding block from others. A verifier is able to check whether a given block is signed by a group member in Ring Verify.

MHT Encryption systems are used to perform operations on encrypted data without knowing the private key (without decryption), the client is the only holder of the secret key. When we decrypt the result of any operation, it is the same as if we had carried out the calculation on the raw data. A privacy preserving remote data integrity checking protocol with data dynamics and public verifiability make use of a Remote Data Integrity Checking Protocol. The protocol provides public verifiability without the help of a third party auditor. It doesn't leak any privacy information to third party, which provides good performance without the support of the trusted third party and provides a method for independent arbitration of data retention contracts. But it gives unnecessary computation and communication cost. The public auditing protocol: To achieve privacy-preserving public auditing, we propose to uniquely integrate the MHT with signature technique. In our protocol, the linear combination of sampled blocks in the server's response is masked with randomness generated by the server.

F.DATA INTEGRITY ANALYSIS

In this module, TPA checks the correctness of data storage to ensure that there exists no cheating cloud server that can pass the TPA's audit without indeed storing users' data intact to ensure that the TPA cannot derive users' data content from the information collected during the auditing process. And implement the batch auditing scheme to enable TPA with secure and efficient auditing capability to cope with multiple auditing delegations from possibly large number of different users simultaneously. Auditor monitors the manipulations between the data owner and cloud service provider, receives the meta information of the data component, tag generation key and random challenge from the data owner, now by making a request to the cloud server auditor gets the meta information of the data component, before processing the request checks for authentication the and checks with the meta information which is received from the data owner.

Data owner hosts the data over cloud servers. Here the data which is fragmented and encrypted by the data owner, data owner can access the information when ever required from the cloud server. Auditor access the information for auditing purpose if he is authenticated, Submits the access process to the data owner when ever required.

G.DYNAMIC AUDITING & SECURE DATA SHARING

To fully ensure the data integrity and save the cloud users' computation resources as well as online burden, it is of critical importance to enable public auditing service for cloud data storage, so that users may resort to an independent third-party auditor (TPA) to audit the outsourced data when needed.

In this module, allow TPA to verify the correctness of the cloud data on demand without retrieving a copy of the whole data or introducing additional online burden to the cloud users. Through the organization of privacy-preserving public auditing in Cloud Computing, TPA may concurrently handle multiple auditing delegations upon different user requests. The individual auditing of these tasks for TPA can be and very difficult and inefficient. Batch auditing not only allows TPA to perform the multiple auditing tasks at the same time, but also greatly reduces the computation cost on the TPA side.

This is because of aggregating K verification equations into helps to reduce the number of quite expensive paring operation from 2k, as required in individual auditing ,to K+1, by which saves a considerable amount of auditing time. Data dynamic support is achieved by replace information index in computation of block authenticator and by using one of the best data structure and supporting data dynamics for privacy-preserving public risk auditing is also of supreme importance. Now we show how our main scheme can be adapted to build upon the obtainable work to support data dynamics, including block level operations of modification, deletion and insertion. We can accept this technique in our design to achieve privacy-preserving public risk auditing with support of data dynamics

Each user is assigned to data owner from the Provider. Each user can freely get the cipher texts from the server. To decrypt a cipher text, each user may submit their secret keys issued by data owner together with its global public key to the server and ask it to generate decryption token for some cipher text. Upon receiving the decryption token, the user can decrypt the cipher text by using its global secret key.

IV.CONCLUSION

Cloud computing securities are discussed and analyzed in previous study. In this project, some of the privacy threats are addressed and the techniques to overcome them are surveyed. While some approaches utilized traditional cryptographic methods to achieve privacy, some other approaches kept them away and focused on alternate methodologies in achieving privacy. Also, approaches to preserve privacy at the time of public auditing are also dis-cussed. Thus, to conclude it is necessary that every cloud user must be guaranteed that his data is stored, processed, accessed and audited in a secured manner at any time. Data freshness is essential to protect against misconfiguration errors or rollbacks caused intentionally and can develop an authenticated file system that supports the migration of an enterprise-class distributed file system into the cloud efficiently, transparently and in a scalable manner. It's authenticated in the sense that enables an enterprise tenant to verify the freshness of retrieved data while performing the file system operations. The user must be given complete access control over the published data. Also, powerful security mechanisms must always supplement every cloud application. Attaining all these would end up in achieving the long dreamt vision of secured Cloud Computing in the nearest future.

REFERENCES

[1] C.Wang, Q.Wang, K.Ren, and W.Lou,"Privacy-preserving public auditing for data storage in cloud computing," in INFOCOM, 2010,pp.525 - 533

[2] Q.Zheng and S.Xu,"Fair and dynamic proofs of retrievability," in CODASPY, 2011, pp, 237 – 248.

[3] Y.Zhu,H.Hu,G.J Ahn, and M.Yu,"Cooperative provable data possession for integrity verification in multicloud storage," IEEE Trans. Parallel Distrib. Syst., vol.23,no.12, pp. 2231 – 2244, 2012.

[4] R. Curtmola, O. Khan, R. Burns, and G. Ateniese, "Mr-pdp: Multiple-replica provable data possession," in *Proc. 28th Int'l Conf. Distributed Computing Systems (ICDCS '08)*, 2008, pp. 411–420.

[5] B. Wang, B. Li, and H. Li, "Oruta: Privacy-preserving public auditing for shared data in the cloud," in *Proc. 5th Int'l Conf. Cloud Computing*, 2012, pp. 295–302.

[6] C. Wang, Q. Wang, K. Ren, and W. Lou, "Privacy-preserving public auditing for data storage security in cloud computing," in *Proc. IEEE INFOCOM*, 2010, pp. 1–9.

[7] C. Wang, S. S. Chow, Q. Wang, K. Ren, and W. Lou, "Privacy-preserving public auditing for secure cloud storage," *IEEE Trans. Computers*, vol. 62, no. 2, pp. 362–375, 2013.

[8] B. Wang, B. Li, and H. Li, "Oruta: Privacy-preserving public au- diting for shared data in the cloud," *IEEE Trans. Cloud Computing*, vol. 2, no. 1, pp. 43–56, 2014.

[9] D. Boneh, C. Gentry, B. Lynn, and H. Shacham, "Aggregate and verifiably encrypted signatures from bilinear maps," in *Proc. 22nd Intl Conf. Theory and Applications of Cryptographic Techniques: Ad-vances in Cryptology (EUROCRYPT03)*, 2003, pp. 416–432.

[10] P. A. Bernstein and N. Goodman, "An algorithm for concurrency control and recovery in replicated distributed databases," *ACM Trans. Database Systems*, vol. 9, no. 4, pp. 596–615, 1984.

[11] J. Hendricks, G. R. Ganger, and M. K. Reiter, "Low-overhead byzantine fault-tolerant storage," in *ACM SIGOPS Operating Sys- tems Review*, vol. 41, no. 6, 2007, pp. 73–86.

[12] J. Gray, P. Helland, P. O'Neil, and D. Shasha, "The dangers of replication and a solution," in *ACM SIGMOD Record*, vol. 25, no. 2, 1996, pp. 173–182.

[13] M. Blum, W. Evans, P. Gemmell, S. Kannan, and M. Naor, "Check- ing the correctness of memories," *Algorithmica*, vol. 12, no. 2-3, pp. 225–244, 1994.

[14] R. C. Merkle, "Protocols for public key cryptosystems," in *Proc. IEEE Symp. Security and Privacy*, 1980, pp. 122–133.