International Journal of Engineering Research in Computer Science and Engineering (IJERCSE) Vol 2, Issue 1, January 2015

DYNAMIC SECURE SYSTEM FOR DETECTING AND ELIMINATING FRAUDULENCE IN CLOUD STORAGE

Kalaivani A[1], Ranjith Kumar M[2], Sabarish M[3], Sai Kishore R[4]
Assistant Professor, Dept. of CSE, R.M.K College of Engineering and Technology, Puduvoyal, Chennai.[1,2.]
Third Year , Dept. of CSE, R.M.K College of Engineering and Technology, Puduvoyal, Chennai.[3.4]

Abstract: **When data is hosted by data owners and made available to clients, security challenges are faced. Thus, we need to audit the data to solve security challenges. Existing remote integrity checking methods can only serve for static archive data. Thus, cannot be applied to the auditing service since the data in the cloud can be dynamically updated. Hence, a dynamic System is desired to ensure that the data are correctly stored in the cloud. In this Paper, we create a monitoring tool that detects fraudulent using link analysis and checks for similar details among multiple databases will be created. As there is no interlinking between different banks database an individual can create many accounts with different identity proofs So that they can do malicious activities in the network. To avoid this we use link analysis for finding similarity link in the entire combined cloud stored database. It uses Joint Threshold Administrative Model (JTAM) for authenticating database storage and handles fault tolerance effectively using the monitoring tool**

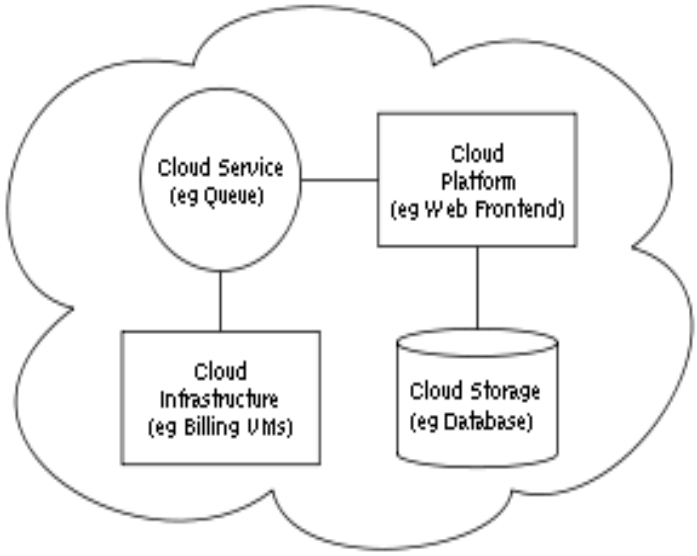
LIST OF TABLES
LITERATURE SURVEY

LITERATURE SURVEY	EXISTING SYSTEM	PROPOSED SYSTEM
Secure Untrusted Data Repository (SUNDR)	SUNDR is a network file system designed to store data securely on untrusted servers. SUNDR lets clients detect any attempts at unauthorized file modification by malicious server operators or users.	SUNDR's protocol achieves a property called *fork consistency*, which guarantees that clients can detect any integrity or consistency failures as long as they see each other's file modifications. An implementation is described that performs comparably with NFS (sometimes better and sometimes worse), while offering significantly stronger security.
A View of Cloud Computing	The interesting thing about cloud computing is that we've redefined cloud computing to include everything that we already do.I don't understand what we would do differently in the light of cloud computing other than change the wording of some of our ads.	This article is to reduce that confusion by clarifying terms, providing simple figures to quantify comparisons between of cloud and conventional computing, and identifying the top technical and non-technical obstacles and opportunities of cloud computing.
Efficient Byzantine-tolerant erasure-coded storage	Survivable storage systems spread data redundantly	A decentralized consistency protocol for survivable storage

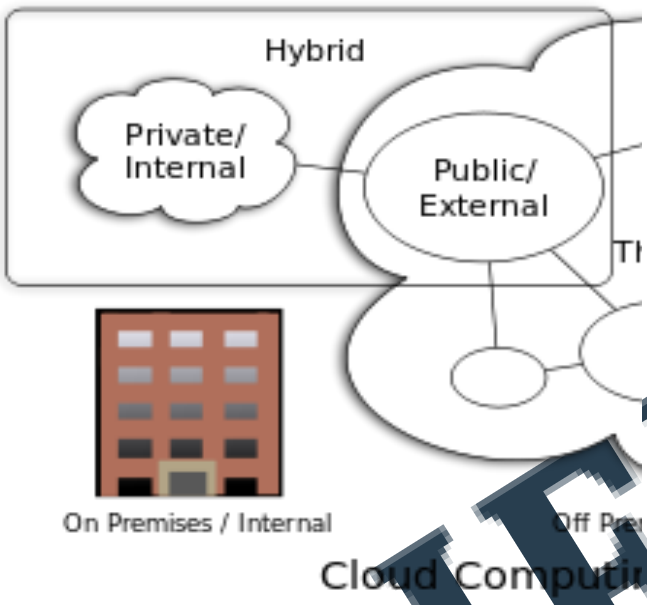
	across a set of distributed storage-nodes in an effort to ensure its availability despite the failure or compromise of storage-nodes. Such systems require some protocol to maintain data consistency in the presence of failures and concurrency.	that exploits local data versioning within each storage-node is described. Such versioning enables the protocol to efficiently provide linearizability and wait-freedom of read and write operations to erasure-coded data in asynchronous environments with Byzantine failures of clients and servers.
The Complexity of Online Memory Checking	We consider the problem of storing a large file on a remote and unreliable server. To verify that the file has not been corrupted, a user could store a small private (randomized) finger-print on his own computer. This is the setting for the well-studied authentication problem in cryptography, and the required fingerprint size is well understood.	It was previously shown that when the adversary is computationally bounded, under the assumption that one-way functions exist, it is possible to construct much better online memory checkers. The same is also true for sub-linear authentication schemes.
Efficient Remote Data Possession Checking in Critical Information Infrastructures	Checking data possession in networked information systems such as those related to critical infrastructures (power facilities, airports, data vaults, defense systems, and so forth) is a matter of crucial importance	In this paper, we present a new remote data possession checking protocol such that 1) it allows an unlimited number of file integrity verifications and 2) its maximum running time can be chosen at set-up time and traded off against storage at the verifier.
Securing Distributed Storage: Challenges, Techniques, and Systems	The rapid increase of sensitive data and the growing number of government regulations that require longterm data retention and protection have forced enterprises to pay serious attention to storage security.	we discuss important security issues related to storage and present a comprehensive survey of the security services provided by the existing storage systems. We cover a broad range of the storage security literature, present a critical review of the existing solutions, compare them, and highlight potential research issues

LIST OF FIGURES

Cloud Architecture



CLOUD COMPUTING TYPES



SYSTEM REQUIREMENT SPECIFICATION
Analysis
SYSTEM ANALYSIS

EXISTING SYSTEM

The data owners host their data on cloud servers and users (data consumers) can access the data from cloud servers. Due to the data outsourcing, however, this new paradigm of data hosting service also introduces new security challenges, which requires an independent auditing service to check the data integrity in the cloud. Some existing remote integrity checking methods can only serve for static archive data and, thus, cannot be applied to the auditing service since the data in the cloud can be dynamically updated. Thus, an efficient and secure dynamic auditing protocol is desired to convince data owners that the data are correctly stored in the cloud. Here we first design an auditing framework for cloud storage systems and propose an efficient and privacy-preserving auditing protocol. Then, we extend our auditing protocol to support the data dynamic operations, which is efficient and provably secure in the random oracle model.

DRAWBACKS IN EXISTING SYSTEM:

- In dynamic auditing protocol cloud server have threat for security for data storage.
- When any system is suspected that will be under tracked only under complaint.
- There is no link between all the databases where an individual have different accounts.

PROPOSED SYSTEM

A monitoring tool that detects fraudulent using link analysis and checks for similar details among multiple databases will be created. As there is no interlinking between different banks database an individual can create many accounts with different identity proofs So that they can do malicious activities in the network. To avoid this we use link analysis for finding similarity link in the entire combined cloud stored database. It uses Joint Threshold Administrative Model (JTAM) for authenticating database storage and handles fault tolerance effectively using the monitoring tool.

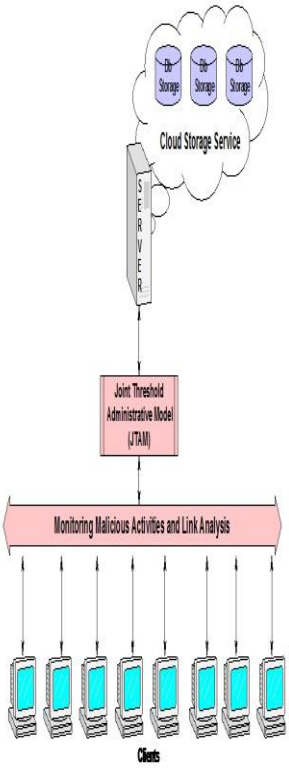
ADVANTAGES IN PROPOSED SYSTEM

- Creating monitoring tool that tracks fraud account creation using link analysis.
- Monitors and handles all the suspected system involved in malicious activities.
- Joint Threshold Administrative Tool (JTAM) is used for permitting privileges for data storage.
- Handles fault tolerance effectively.

Function

- It uses link analysis algorithm for similarity details.
- Joint Threshold Administrative Tool (JTAM) is used for permitting privileges for data storage.
- A monitoring tool is created for monitoring malicious activities on database.

SYSTEM DESIGN SPECIFICATION
Architecture

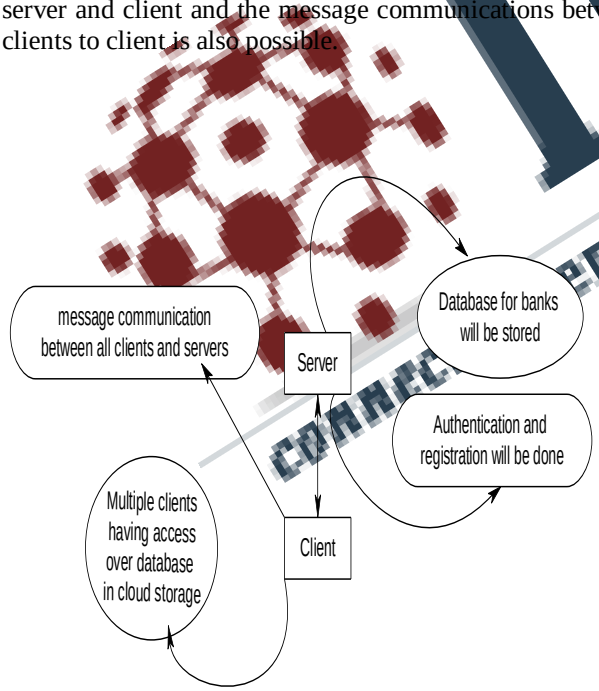


Modules

- SERVER/CLIENT ESTABLISHMENT
- CLOUD STORAGE
- MONITORING TOOL CREATION
- JTAM IMPLEMENTATION

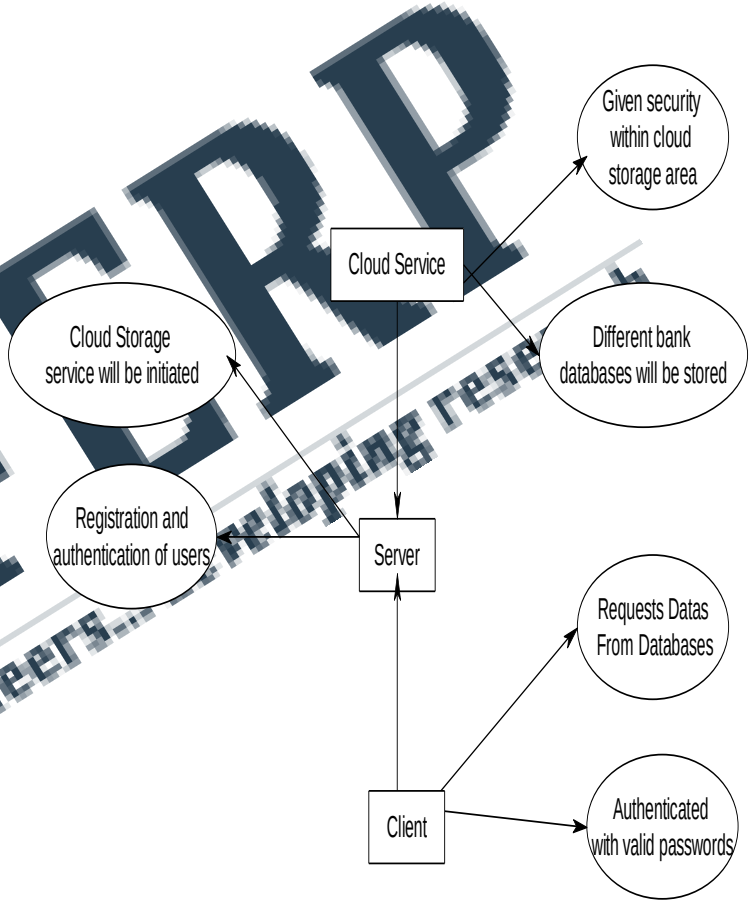
SERVER/CLIENT ESTABLISHMENT

In this module we create server & client systems. We create different databases denoting different bank data. These all will be given strong security thus no one can hack any message communication in terms of money. We thus create many client systems with some system having secured database. . Here we create a single server and multiple client systems which have intermediate communications between each other. Thus they can be easily pass messages between server and client and the message communications between clients to client is also possible.



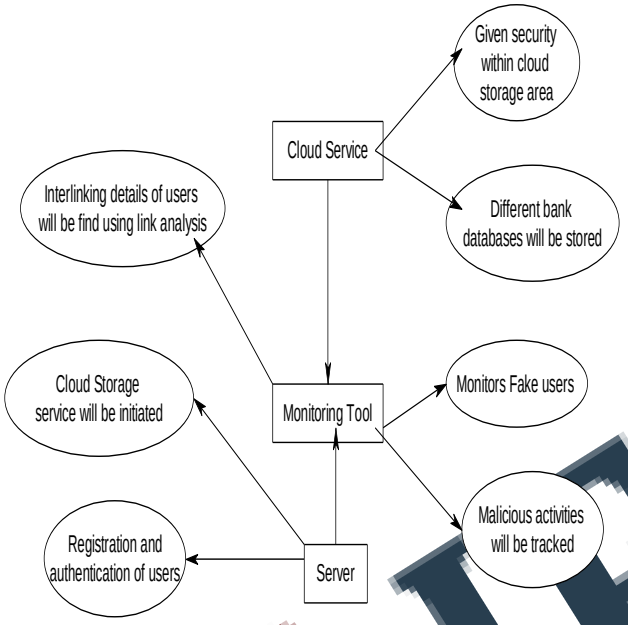
CLOUD STORAGE

Data from the different databases combined and implemented in cloud storage for linking all the data for further data storage and retrieval. Cloud storage will be online storage system that provides secured space for storing data for different banks. Thus every bank will have individual memory space for storing their authenticated secured user information and their money transactions between bank and their customers. Cloud computing concept of storage as a service will be implemented in this module for storing databases for different banks.



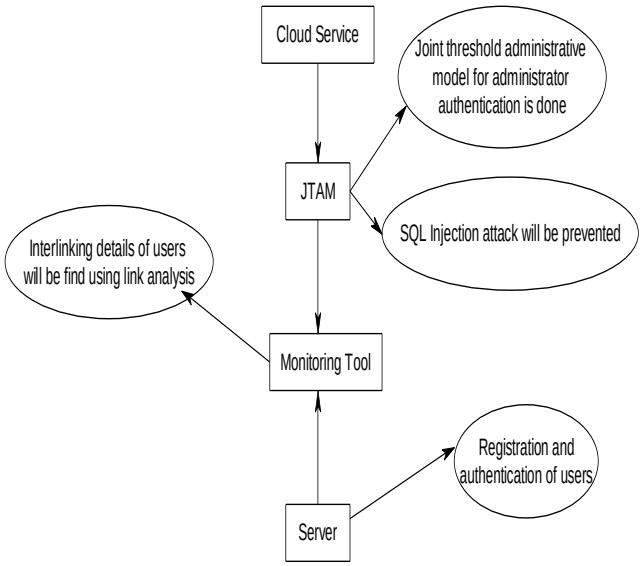
MONITORING TOOL CREATION

In this module we create a monitoring tool in the cloud database which monitors the stored data for their similarity interlinking details. It helps to monitor the details of users who is trying to create fake database or even the users having multiple accounts. The similarity details of a particular user will be kept track so that the user's malicious activity on the network using their different account databases. Such users will be caught and informed about their malicious
database activity. The interlinking activities will be handled using link analysis mapping for all the possible similarity details.



JTAM IMPLEMENTATION

Joint Threshold Administrative Model (JTAM) tool is created with different administrators implied in different databases. Admin gives authentication for storage of data and subsequently analyse similarity details in cloud storage database. This JTAM model will ask for authentication to each and every administrator all the time. Only when all the administrators given authentication to that particular task then only that task will be proceeds otherwise it will wait until every admin given authentication. This handles SQL injection attack at any database from the secured database.



System Requirements

Software Requirement:

- Front End : Jdk 1.7
- Back End (Server Edition) : Oracle 10g
- IDE :NetBeans
- Operating System :Windows Xp,Windows 7

Hardware Requirement:

- Processor above :Pentium 4 or
- Ram above :512mb or
- Hard disk :80gb

SYSTEM IMPLEMENTATION

At first the node topology is created using server socket program .the client module uses the socket and the server module uses the server socket. Java swing is used to design the front end. Quaqua look and feel is used to enhance the design part of the project.

The encryption algorithm of aes is implemented using java code .the system port number and the dsn details in the network are gathered so that the message is transferred through the specified path. The back end is sql server where we are storing data and retrieving from there. The node connected in the network topology and the dsn of all nodes and the port number of all nodes is retrieved from this sql server database 2000. we are creating table in sql server for storing this details . The front end and the backend is connected with the help of jdbcodbc driver.

Modules implement procedure

Link Analysis

TRADITIONAL statistical, machine learning, pattern recognition, and data mining approaches usually assume a random sample of independent objects from a single relation. Many of these techniques have gone through the extraction of knowledge from data (typically extracted from relational databases), almost always leading, in the end, to the classical double-entry tabular format, containing features for a sample of the population. These features are therefore used in order to learn from the sample, provided that it is representative of the population as a whole. However, real-world data coming from many fields (such as World Wide Web, marketing, social networks, or biology; see) are often multirelational and interrelated. The work recently performed in statistical relational learning, aiming at working with such data sets, incorporates research topics, such as link analysis, web mining social network analysis or graph mining. All these research fields intend to find and exploit links between objects (in addition to features—as is also the case in the field of spatial statistics), which could be of various types and involved in different kinds of relationships.

The focus of the techniques has moved over from the analysis of the features describing each instance belonging to the population of interest (attribute value analysis) to the analysis of the links existing between these instances (relational analysis), in addition to the features. This paper precisely proposes a link-analysis-based technique allowing to discover relationships existing between elements of a relational database or, more generally, a graph. More specifically, this work is based on a random walk through the database defining a Markov chain having as many states as elements in the database. Suppose, for instance, we are interested in analyzing the relationships between elements contained in two different tables of a relational database. To this end, a two-step procedure is developed. First, a much smaller, reduced, Markov chain, only containing the elements of interest typically the elements contained in the two tables and preserving the main characteristics of the initial chain, is extracted by stochastic complementation. An efficient algorithm for extracting the reduced Markov chain from the large, sparse, Markov chain representing the database is proposed. Then, the reduced chain is analyzed by, for instance, projecting the states in the subspace spanned by the right eigenvectors of the transition matrix or by computing a kernel principal component analysis, on a diffusion map kernel computed from the reduced graph and visualizing the results. Indeed, a valid graph kernel based on the diffusion map distance, extending the basic diffusion map to directed graphs, is introduced. The motivations for developing this two-step procedure are twofold. First, the computation would be cumbersome, if not impossible, when dealing with the complete database.

Second, in many situations, the analyst is not interested in studying all the relationships between all elements of the database, but only a subset of them. Moreover, if the whole set of elements in the database is analyzed, the resulting mapping would be averaged out by the numerous relationships and elements we are not interested in for instance, the principal axis would be completely different. It would therefore not exclusively reflect the relationships between the elements of interest. Therefore, reducing the Markov chain by stochastic complementation allows to focus the analysis on the elements and relationships we are interested in. Interestingly enough, when dealing with a bipartite graph (i.e., the database only contains two tables linked by one relation), stochastic complementation followed by a basic diffusion map is exactly equivalent to simple correspondence analysis.

On the other hand, when dealing with a starschema database (i.e., one central table linked to several tables by different relations), this two-step procedure reduces to multiple correspondence analysis. The proposed methodology therefore extends correspondence analysis to the analysis of a relational database. In short, this paper has three main contributions. A two-step procedure for analyzing weighted graphs or relational databases is proposed. . It is shown that the suggested procedure extends correspondence analysis. . A kernel version of the diffusion map distance, applicable to directed graphs, is introduced.

JTAM (Joint Threshold Administrative Model)

Our interactive response policy language makes it very easy for the database administrators to specify appropriate response actions for different circumstances depending upon the nature of the anomalous request. The two main issues that we address incontext of such response policies are that of policy matching, and policy administration. For the policy matching problem, two algorithms that efficiently search the policy database for policies that match an anomalous request. We also extend the PostgreSQL DBMS with our policy matching mechanism, and report experimental results. The experimental evaluation shows that our techniques are very efficient. The other issue that we address is that of administration of response policies to prevent malicious modifications to policy objects from legitimate users. We propose a novel Joint Threshold Administration Model (JTAM) that is based on the principle of separation of duty. The key idea in JTAM is that a policy object is jointly administered by at least k database administrator (DBAs), that is, any modification made to a policy object will be invalid unless it has been authorized by at least k DBAs. We present design details of JTAM which is based on a cryptographic threshold signature scheme, and show how JTAM prevents malicious modifications to policy objects from authorized users.

TESTING SYSTEMTESTING

System testing of software or hardware is testing conducted on a complete, integrated system to evaluate the system's compliance with its specified requirements. System testing falls within the scope of black box testing, and as such, should require no knowledge of the inner design of the code or logic.

SOFTWARE TESTING

Software Testing is an empirical investigation conducted to provide stakeholders with information about the quality of the product or service under test, with respect to the context in which it is intended to operate. Software Testing also provides an objective, independent view of the software to allow the business to appreciate and understand the risks at implementation of the software. Test techniques include, but are not limited to, the process of executing a program or application with the intent of finding software bugs. Software Testing can also be stated as the process of validating and verifying that a software program/application/product (1) meets the business and technical requirements that guided its design and development; (2) works as expected; and (3) can be implemented with the same characteristics.

UNIT TESTING

The primary goal of unit testing is to take the smallest piece of testable software in the application, isolate it from the remainder of the code, and determine whether it behaves exactly as you expect. Each unit is tested separately before integrating them into modules to test the interfaces between modules. Unit testing has proven its value in that a large percentage of defects are identified during its use.

INTEGRATION TESTING

Integration testing (sometimes called Integration and Testing) is the activity of software testing in which individual software modules are combined and tested as a group. It occurs after unit testing and before system testing. Integration testing takes as its input modules that have been unit tested, groups them in larger aggregates, applies tests defined in an integration test plan to those aggregates, and delivers as its output the integrated system ready for system testing.

ACCEPTANCE TESTING

Acceptance testing by the system provider is often distinguished from acceptance testing by the customer (the user or client) prior to accepting transfer of ownership. In such environments, acceptance testing performed by the customer is known as user acceptance testing (UAT). This is also known as end-user testing, site (acceptance) testing, or field (acceptance) testing.

CONCLUSION:

From this Cloud storage all the dynamic auditing details and data storage system will be analysed for similarity and interlinking between different individual databases. Joint Threshold Administrative tool implies authenticated and secured database management.

Future Work:

Recommending this proposal to the world bank to control black money . Then the bank should increase the speed of uploading the registration process in cloud.

REFERENCE:

1] P. Mell and T. Grance, "The NIST Definition of Cloud Computing," technical report, Nat'l Inst. of Standards and Technology,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, "A View of Cloud Computing," Comm. ACM, vol. 53, no. 4, pp. 50-58, 2010.

[3] T. Velte, A. Velte, and R. Elsenpeter, Cloud Computing: A PracticalApproach, first ed., ch. 7. McGraw-Hill, 2010.

[4] V. Kher and Y. Kim, "Securing Distributed Storage: Challenges, Techniques, and Systems," Proc. ACM Workshop Storage Security and Survivability (StorageSS), V. Atluri, P. Samarati, W. Yurcik, L. Brumbaugh, and Y. Zhou, eds., pp. 9-25, 2005.

[5] Y. Zhu, H. Wang, Z. Hu, G.-J.Ahn, H. Hu, and S.S. Yau, "Dynamic Audit Services for Integrity Verification of Outsourced Storages in Clouds," Proc. ACM Symp. Applied Computing, W.C. Chu, W.E. Wong, M.J. Palakal, and C.-C.Hung, eds., pp. 1550-1557, 2011.

[6] C. Wang, Q. Wang, K. Ren, and W. Lou, "Privacy-Preserving Public Auditing for Data Storage Security in Cloud Computing," Proc. IEEE INFOCOM, pp. 525-533, 2010.

[7] M.A. Shah, M. Baker, J.C. Mogul, and R. Swaminathan, "Auditing to Keep Online Storage Services Honest," Proc. 11th USENIX Workshop Hot Topics in Operating Systems (HOTOS), G.C. Hunt, ed., 2007.

[8] G. Ateniese, S. Kamara, and J. Katz, "Proofs of Storage from Homomorphic Identification Protocols," Proc. Int'l Conf. Theory and Application of Cryptology and Information Security: Advances in Cryptology, M. Matsui, ed., pp. 319-333, 2009.

[9] 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 Distributed Systems, vol. 22, no. 5, pp. 847-859, May 2011.

[10] M. Naor and G.N. Rothblum, "The Complexity of Online Memory Checking," J. ACM, vol. 56, no. 1, article 2, 2009.

[11] J. Li, M.N. Krohn, D. Mazie`res, and D. Shasha, "SecureUntrusted Data Repository (SUNDR)," Proc. Sixth Conf. Symp. Operating Systems Design Implementation, pp. 121-136, 2004.

[12] Y. Zhu, H. Hu, G. Ahn, and M. Yu, "Cooperative Provable Data Possession for Integrity Verification in Multi-Cloud Storage," IEEE Trans. Parallel and Distributed Systems, vol. 23, no. 12, pp. 2231-2244, Dec. 2012.

[13] K. Yang and X. Jia, "Data Storage Auditing Service in Cloud Computing: Challenges, Methods and Opportunities," World Wide Web, vol. 15, no. 4, pp. 409-428, 2012.

[14] B. Schroeder and G.A. Gibson, "Disk Failures in the Real World: What Does an MTTF of 1,000,000 Hours Mean to You?" Proc. USENIX Conf. File and Storage Technologies, pp. 1-16, 2007.

[15] C. Wang, K. Ren, W. Lou, and J. Li, "Toward Publicly Auditable Secure Cloud Data Storage Services," IEEE Network, vol. 24, no. 4, pp. 19-24, July/Aug. 2010.