IMPLEMENTATION OF PRIVACY-PRESERVING PUBLIC AUDITING METHODS FOR SECURE CLOUD STORAGE

Mr. A. Mahesh Babu,
Research Scholar,
Department of Computer Science & Technology,
Sri Krishnadevaraya University, Anantapur. AP, India.

Dr. G.A. Ramachandra,
Associate Professor,
Department of Computer Science & Technology,
Sri Krishnadevaraya University, Anantapur. AP, India.

Dr. M. Suresh Babu,
Professor & Head,
Department of Computer Applications,
Madanapalle Institute of Technology & Science, Madanapalle, India.

ABSTRACT

Cloud computing is internet based computing which enables sharing of services. Many users place their data in the cloud. However, the fact that users no longer have physical possession of the possibly large size of outsourced data makes the data integrity protection in cloud computing a very challenging and potentially formidable task, especially for users with constrained computing resources and capabilities. So correctness of data and security is a prime concern. This paper studies the problem of ensuring the integrity and security of data storage in Cloud Computing. Security in cloud is achieved by signing the data block before sending to the cloud. Using Cloud Storage, users can remotely store their data and enjoy the on-demand high quality applications and services from a shared pool of configurable computing resources, without the burden of local data storage and maintenance. However, the fact that users no longer have physical possession of the outsourced data makes the data integrity protection in Cloud Computing a formidable task, especially for users with constrained computing resources. Moreover, users should be able to just use the cloud storage as if it is local, without worrying about the need to verify its integrity. Thus, enabling public auditability for cloud storage is of

critical importance so that users can resort to a third party auditor (TPA) to check the integrity of outsourced data and be worry-free. To securely introduce an effective TPA, the auditing process should bring in no new vulnerabilities towards user data privacy, and introduce no additional online burden to user. In this paper, we propose a secure cloud storage system supporting privacy-preserving public auditing. We further extend our result to enable the TPA to perform audits for multiple users simultaneously and efficiently. Extensive security and performance analysis show the proposed schemes are provably secure and highly efficient.

Key words: Cloud Storage, privacy preserving, public auditability.

Introduction

Cloud Computing has been envisioned as the next-generation information technology (IT) architecture for enterprises, 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 is 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, universal data access with independent geographical locations, and avoidance of capital expenditure on hardware, software, and personnel maintenances, etc. While Cloud Computing makes these advantages more appealing than ever, it also brings new and challenging security threats towards 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. Examples of outages and security breaches of noteworthy cloud services appear from time to time. Secondly, there do exist various motivations for CSP to

behave unfaithfully towards the cloud users regarding the status of their outsourced data. For examples, CSP might reclaim storage for monetary reasons by discarding data that has not been or is rarely accessed, or even hide data loss incidents so as to maintain a reputation.

In short, although outsourcing data to the cloud is economically attractive for long-term large-scale data storage, it does not immediately offer any guarantee on data integrity and availability. This problem, if not properly addressed, may impede the successful deployment of the cloud architecture.

1.1 Existing System:

Although the existing schemes aim at providing integrity verification for different data storage systems, the problem of supporting both public auditability and data dynamics has not been fully addressed. How to achieve a secure and efficient design to seamlessly integrate these two important components for data storage service remains an open challenging task in Cloud Computing.

Disadvantages:

Especially to support block insertion, which is missing in most existing schemes.

Proposed System:

• Client: an entity, which has large data files to be stored in the cloud and relies on the cloud for data maintenance and computation, can be either individual consumers or organizations.

• Cloud Storage Server (CSS): an entity, which is managed by Cloud Service Provider (CSP), has significant storage space and computation resource to maintain the clients' data.

• Third Party Auditor (TPA): an entity, which has expertise and capabilities that clients do not have, is *trusted* to assess and expose risk of cloud storage services on behalf of the clients upon request.

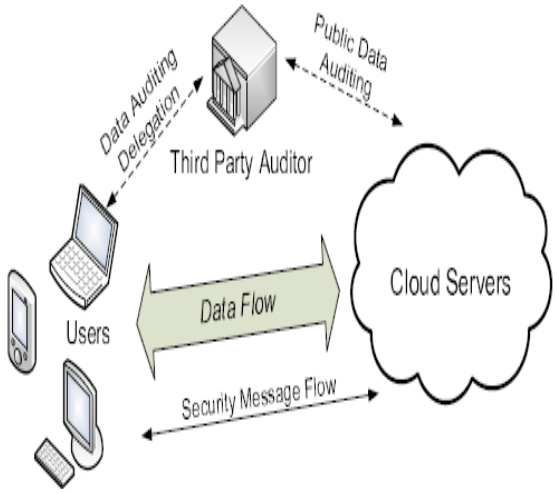
Advantages:

1) We motivate the public auditing system of data storage security in Cloud Computing, and propose a protocol supporting for fully dynamic data operations, especially to support block insertion, which is missing in most existing schemes;

2) We extend our scheme to support scalable and efficient public auditing in Cloud Computing. In particular, our scheme achieves auditing tasks from different users can be performed simultaneously by the TPA.

3) We prove the security of our proposed construction and justify the performance of our scheme through concrete implementation and comparisons.

Architecture:-



2.0 Problem definition:-

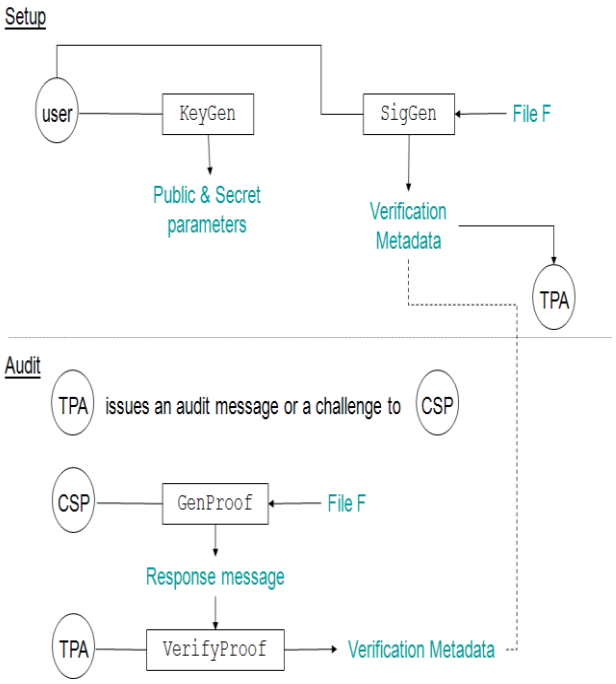
Project Enhancement:-

““Very efficient and dynamic Data Outsourcing on Cloud”

Public audit ability allows an external party, in addition to the user himself, to verify the correctness of remotely stored data public key based homomorphic linear authenticator A public auditing scheme consists of four algorithms (KeyGen, SigGen, GenProof, VerifyProof). KeyGen

is a key generation algorithm that is run by the user to setup the scheme. SigGen is used by the user to generate verification metadata, which may consist of MAC, signatures, or other related information that will be used for auditing. GenProof is run by the cloud server to generate a proof of data storage correctness, while VerifyProof is run by the TPA to audit the proof from the cloud server Running a public auditing system consists of two phases, Setup and Audit: • Setup: The user initializes the public and secret parameters of the system by executing KeyGen, and pre-processes the data file F by using SigGen to generate the verification metadata.

The user then stores the data file F and the verification metadata at the cloud server, and deletes its local copy. As part of pre-processing, the user may alter the data file F by expanding it or including additional metadata to be stored at server. • Audit: The TPA issues an audit message or challenge to the cloud server to make sure that the cloud server has retained the data file F properly at the time of the audit. The cloud server will derive a response message from a function of the stored data file F and its verification metadata by executing GenProof. The TPA then verifies the response via VerifyProof. A privacy-preserving public auditing system for data storage security in Cloud Computing. We utilize the homomorphic linear authenticator and random masking to guarantee that the TPA would not learn any knowledge about the data content stored on the cloud server during the efficient auditing process, which not only eliminates the burden of cloud user from the tedious and possibly expensive auditing task, but also alleviates the users' fear of their outsourced data leakage. Considering TPA may concurrently handle multiple audit sessions from different users for their outsourced data files, we further extend our privacy-preserving public auditing protocol into a multi-user setting, where the TPA can perform multiple auditing tasks in a batch manner for better efficiency. Extensive analysis shows that our schemes are provably secure and highly efficient.



3.0 Modules:

1. Public auditability for storage correctness assurance:

To allow anyone, the clients who originally stored the file on cloud servers, to have the capability to verify the correctness of the stored data on demand.

2. Dynamic data operation support:

To allow the clients to perform block-level operations on the data files while maintaining the same level of data correctness assurance. The design should be as efficient as possible so as to ensure the seamless integration of public auditability and dynamic data operation support.

3. Blockless verification:

No challenged file blocks should be retrieved by the verifier (*e.g.*, TPA) during verification process for efficiency concern.

4. Dynamic Data Operation with Integrity Assurance:

Now we show how our scheme can explicitly and efficiently handle fully dynamic data operations including data modification (M), data insertion (I) and data deletion (D) for cloud data

storage. Note that in the following descriptions, we assume that the file F and the signature _ have already been generated and properly stored at server. The root metadata R has been signed by the client and stored at the cloud server, so that anyone who has the client's public key can challenge the correctness of data storage.

5. Data Modification:

We start from data modification, which is one of the most frequently used operations in cloud data storage. A basic data modification operation refers to the replacement of specified blocks with new ones. At start, based on the new block the client generates the corresponding signature. The client signs the new root metadata R′ by sigsk(H(R′)) and sends it to the server for update. Finally, the client executes the default integrity verification protocol. If the Output is *TRUE*, delete sigsk(H(R′)),and generate duplicate file.

6. Batch Auditing for Multi-client Data:

As cloud servers may concurrently handle multiple verification sessions from different clients, given K signatures on K distinct data files from K clients, it is more advantageous to aggregate all these signatures into a single short one and verify it at one time. To achieve this goal, we extend our scheme to allow for provable data updates and verification in a multi-client system. The signature scheme allows the creation of signatures on arbitrary distinct messages. Moreover, it supports the aggregation of multiple signatures by distinct signers on distinct messages into a single short signature, and thus greatly reduces the communication cost while providing efficient verification for the authenticity of all messages.

4.0 Functional Requirements

Functional requirements should include functions performed by specific screens, outlines of work-flows performed by the system and other business or compliance requirements the system must meet.

Interface requirements

- Field accepts numeric data entry
- Field only accepts dates before the current date

- Screen can print on-screen data to the printer

Business Requirements

- Data must be entered before a request can approved
- Clicking the Approve Button moves the request to the Approval Workflow
- All personnel using the system will be trained according to internal training strategies

Regulatory/Compliance Requirements

- The database will have a functional audit trail
- The system will limit access to authorized users
- The spreadsheet can secure data with electronic signatures

Security Requirements

- Member of the Data Entry group can enter requests but not approve or delete requests
- Members of the Managers group can enter or approve a request, but not delete requests
- Members of the Administrators group cannot enter or approve requests, but can delete requests

The functional specification describes what the system must do; how the system does it is described in the Design Specification.

If a User Requirement Specification was written, all requirements outlined in the user requirement specification should be addressed in the functional requirements.

Non Functional Requirements:

All the other requirements which do not form a part of the above specification are categorized as Non-Functional Requirements.

A system may be required to present the user with a display of the number of records in a database. This is a functional requirement.

How up-to-date this number needs to be is a non-functional requirement. If the number needs to be updated in real time, the system architects must ensure that the system is capable of updating the displayed record count within an acceptably short interval of the number of records changing. Sufficient network bandwidth may also be a non-functional requirement of a system.

Other examples:

- Accessibility

- Availability
- Backup
- Certification
- Compliance
- Configuration Management
- Documentation
- Disaster Recovery
- Efficiency (resource consumption for given load)
- Effectiveness (resulting performance in relation to effort)
- Extensibility (adding features, and carry-forward of customizations at next major version upgrade)
- Failure Management
- Interoperability
- Maintainability
- Modifiability
- Open Source
- Operability
- Performance
- Platform compatibility
- Price
- Portability
- Quality (e.g. Faults Discovered, Faults Delivered, Fault Removal Efficacy)
- Recoverability
- Resilience
- Resource constraints (processor speed, memory, disk space, network bandwidth etc.)
- Response time
- Robustness
- Scalability (horizontal, vertical)
- Security

- Software, tools, standards etc.
- Stability
- Safety
- Supportability
- Testability
- Usability by target user community

Accessibility is a general term used to describe the degree to which a product, device, service, or environment is accessible by as many people as possible. Accessibility can be viewed as the "ability to access" and possible benefit of some system or entity. Accessibility is often used to focus on people with disabilities and their right of access to the system.

Availability is the degree to which a system, subsystem, or equipment is operable and in a committable state at the start of a mission, when the mission is called for at an unknown, i.e., a random, time. Simply put, availability is the proportion of time a system is in a functioning condition. Expressed mathematically, availability is 1 minus the unavailability.

A backup or the process of backing up refers to making copies of data so that these additional copies may be used to restore the original after a data loss event. These additional copies are typically called "backups."

Certification refers to the confirmation of certain characteristics of an object, system, or organization. This confirmation is often, but not always, provided by some form of external review, education, or assessment

Compliance is the act of adhering to, and demonstrating adherence to, a standard or regulation.

Configuration management (CM) is a field that focuses on establishing and maintaining consistency of a system's or product's performance and its functional and physical attributes with its requirements, design, and operational information throughout its life.

Documentation may refer to the process of providing evidence ("to document something") or to the communicable material used to provide such documentation (i.e. a document). Documentation may also (seldom) refer to tools aiming at identifying documents or to the field of study devoted to the study of documents and bibliographies

Disaster recovery is the process, policies and procedures related to preparing for recovery or continuation of technology infrastructure critical to an organization after a natural or human-induced disaster.

Disaster recovery planning is a subset of a larger process known as business continuity planning and should include planning for resumption of applications, data, hardware, communications (such as networking) and other IT infrastructure

Extensibility (sometimes confused with forward compatibility) is a system design principle where the implementation takes into consideration future growth. It is a systemic measure of the ability to extend a system and the level of effort required to implement the extension. Extensions can be through the addition of new functionality or through modification of existing functionality. The central theme is to provide for change while minimizing impact to existing system functions.

Interoperability is a property referring to the ability of diverse systems and organizations to work together (inter-operate).

The term is often used in a technical systems engineering sense, or alternatively in a broad sense, taking into account social, political, and organizational factors that impact system to system performance.

Maintenance is the ease with which a software product can be modified in order to:

- correct defects
- meet new requirements
- make future maintenance easier, or
- cope with a changed environment;

Open source describes practices in production and development that promote access to the end product's source materials—typically, their source code

Operability is the ability to keep equipment, a system or a whole industrial installation in a safe and reliable functioning condition, according to pre-defined operational requirements.

In a computing systems environment with multiple systems this includes the ability of products, systems and business processes to work together to accomplish a common task.

Computer performance is characterized by the amount of useful work accomplished by a computer system compared to the time and resources used.

Depending on the context, good computer performance may involve one or more of the following:

- Short response time for a given piece of work
- High throughput (rate of processing work)
- Low utilization of computing resource(s)
- High availability of the computing system or application
- Fast (or highly compact) data compression and decompression
- High bandwidth / short data transmission time

Price in economics and business is the result of an exchange and from that trade we assign a numerical monetary value to a good, service or asset

Portability is one of the key concepts of high-level programming. Portability is the software-code base feature to be able to reuse the existing code instead of creating new code when moving software from an environment to another. When one is targeting several platforms with the same application, portability is the key issue for development cost reduction.

Quality is the common element of the business definitions is that the quality of a product or service refers to the perception of the degree to which the product or service meets the customer's expectations. Quality has no specific meaning unless related to a specific function and/or object. Quality is a perceptual, conditional and somewhat subjective attribute.

Reliability may be defined in several ways:

- The idea that something is fit for purpose with respect to time;
- The capacity of a device or system to perform as designed;
- The resistance to failure of a device or system;
- The ability of a device or system to perform a required function under stated conditions for a specified period of time;
- The probability that a functional unit will perform its required function for a specified interval under stated conditions.
- The ability of something to "fail well" (fail without catastrophic consequences

Resilience is the ability to provide and maintain an acceptable level of service in the face of faults and challenges to normal operation.

These services include:

- supporting distributed processing
- supporting networked storage
- maintaining service of communication services such as
    - video conferencing
    - instant messaging
    - online collaboration
- access to applications and data as needed

Response time perceived by the end user is the interval between

(a) The instant at which an operator at a terminal enters a request for a response from a computer and

(b) The instant at which the first character of the response is received at a terminal.

In a data system, the system response time is the interval between the receipt of the end of transmission of an inquiry message and the beginning of the transmission of a response message to the station originating the inquiry.

Robustness is the quality of being able to withstand stresses, pressures, or changes in procedure or circumstance. A system or design may be said to be "robust" if it is capable of coping well with variations (sometimes unpredictable variations) in its operating environment with minimal damage, alteration or loss of functionality.

The concept of scalability applies to technology and business settings. Regardless of the setting, the base concept is consistent - The ability for a business or technology to accept increased volume without impacting the system.

In telecommunications and software engineering, scalability is a desirable property of a system, a network, or a process, which indicates its ability to either handle growing amounts of work in a graceful manner or to be readily enlarged.

Security is the degree of protection against danger, loss, and criminals.

Security has to be compared and contrasted with other related concepts: Safety, continuity, reliability. The key difference between security and reliability is that security must take into account the actions of people attempting to cause destruction.

Security as a state or condition is resistance to harm. From an objective perspective, it is a structure's actual (conceptual and never fully knowable) degree of resistance to harm.

Stability - it means much of the objects will be stable over time and will not need changes.

Safety is the state of being "safe", the condition of being protected against physical, social, spiritual, financial, political, emotional, occupational, psychological, educational or other types or consequences of failure, damage, error, accidents, harm or any other event which could be considered non-desirable. This can take the form of being protected from the event or from exposure to something that causes health or economical losses. It can include protection of people or of possessions Supportability (also known as serviceability) is one of the aspects of RASU (Reliability, Availability, Serviceability, and Usability)). It refers to the ability of technical support personnel to install, configure, and monitor products, identify exceptions or faults, debug or isolate faults to root cause analysis, and provide hardware or software maintenance in pursuit of solving a problem and restoring the product into service. Incorporating serviceability facilitating features typically results in more efficient product maintenance and reduces operational costs and maintains business continuity.

Testability, a property applying to an empirical hypothesis, involves two components: (1) the logical property that is variously described as contingency, defeasibility, or falsifiability, which means that counter examples to the hypothesis are logically possible, and (2) the practical feasibility of observing a reproducible series of such counter examples if they do exist. In short it refers to the capability of an equipment or system to be tested

Usability is a term used to denote the ease with which people can employ a particular tool or other human-made object in order to achieve a particular goal. In human-computer interaction and computer science, usability often refers to the elegance and clarity with which the interaction with a computer program or a web site is designed.

5.0 Related Work

Ateniese *et al.* are the first to consider public auditability in their defined "provable data possession" (PDP) model for ensuring possession of data files on untrusted storages. Their scheme utilizes the RSA based homomorphic linear authenticators for auditing outsourced data and suggests randomly sampling a few blocks of the file. However, the public auditability in their scheme demands the linear combination of sampled blocks exposed to external auditor. When used directly, their protocol is not provably privacy preserving, and thus may leak user data information to the auditor. Juels *et al.* describe a "proof of retrievability" (PoR) model, where spot-checking and error-correcting codes are used to ensure both "possession" and "retrievability" of data files on remote archive service systems. However, the number of audit challenges a user can perform is fixed a priori, and public auditability is not supported in their main scheme. Although they describe a straightforward Merkle-tree construction for public PoRs, this approach only works with encrypted data. Dodis *et al.* give a study on different variants of PoR with private auditability. Shacham *et al.* design an improved PoR scheme built from BLS signatures with full proofs of security in the security model defined in. Similar to the construction in, they use publicly verifiable homomorphic linear authenticators that are built from provably secure BLS signatures.

Based on the elegant BLS construction, a compact and public verifiable scheme is obtained. Again, their approach does not support privacy-preserving auditing for the same reason as. Shah *et al*, propose allowing a TPA to keep online storage honest by first encrypting the data then sending a number of pre-computed symmetric-keyed hashes over the encrypted data to the auditor. The auditor verifies both the integrity of the data file and the server's possession of a previously committed decryption key. This scheme only works for encrypted files, and it suffers from the auditor statefulness and bounded usage, which may potentially bring in online burden to users when the keyed hashes are used up. In other related work, Ateniese *et al.* propose a partially dynamic version of the prior PDP scheme, using only symmetric key cryptography but with a bounded number of audits. In, Wang *et al.* consider a similar support for partial dynamic data storage in a distributed scenario with additional feature of data error localization. In a

subsequent work, Wang *et al.* propose to combine BLS-based HLA with MHT to support both public auditability and full data dynamics. Almost simultaneously, Erway *et al.* developed a skip lists based scheme to enable provable data possession with full dynamics support. However, the verification in these two protocols requires the linear combination of sampled blocks just as, and thus does not support privacy preserving auditing. While all the above schemes provide methods for efficient auditing and provable assurance on the correctness of remotely stored data, none of them meet all the requirements for privacy preserving public auditing in cloud computing. More importantly, none of these schemes consider batch auditing, which can greatly reduce the computation cost on the TPA when coping with a large number of audit delegations.

6.0 Conclusion

In this paper, we propose a privacy-preserving public auditing system for data storage security in Cloud Computing. We utilize the homomorphic linear authenticator and random masking to guarantee that the TPA would not learn any knowledge about the data content stored on the cloud server during the efficient auditing process, which not only eliminates the burden of cloud user from the tedious and possibly expensive auditing task, but also alleviates the users' fear of their outsourced data leakage. Considering TPA may concurrently handle multiple audit sessions from different users for their outsourced data files, we further extend our privacy-preserving public auditing protocol into a multi-user setting, where the TPA can perform multiple auditing tasks in a batch manner for better efficiency. Extensive analysis shows that our schemes are provably secure and highly efficient.

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.