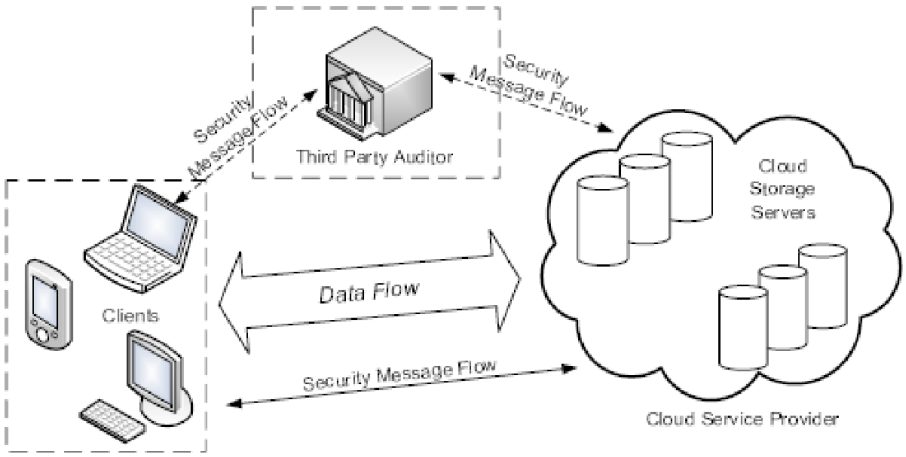
Enabling Public Auditability and Data Dynamics for Storage Security in Cloud Computing

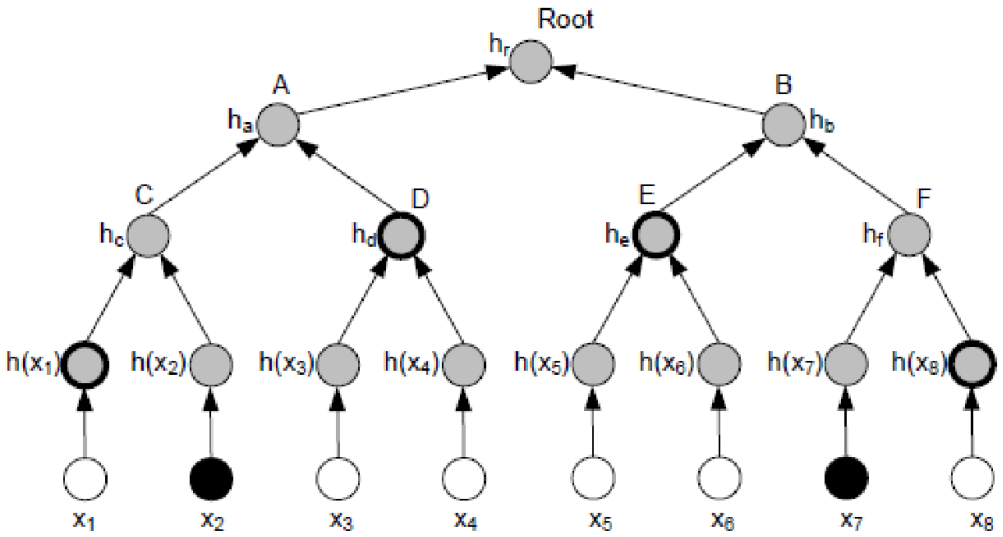
[1]Himanshu Pradhanr, [2]Dr. Neeta Sharma
[1]Student, [2]Proffessor
[1]School of Engineering and Technology
[1]Noida International University, Greater Noida, India

Abstract: Cloud Computing has been envisioned as the next-generation architecture of IT Enterprise. 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, we 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 is 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 paper achieves both. We 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, we improve the existing proof of storage models by manipulating the classic Merkle Hash Tree construction for block tag authentication. To support efficient handling of multiple auditing tasks, we further explore the technique of bilinear aggregate signature to extend our main result into a multi-user setting, where TPA can perform multiple auditing tasks simultaneously. Extensive security and performance analysis show that the proposed schemes are highly efficient and provably secure.

LIST OF FIGURES



MERKLE HASH TREE STRUCTURE



LIST OF ABBREVIATIONS

TPA - Third Party Auditor.

PDP - Provable Data Possession.

PoR - Proof of Retrievability.

CSP – Cloud Server Provider.

CSS - Cloud Storage Server

DH – Diffie-Hellman

MHT – Merkle Hash Tree.

RSA – Rivest, Samir, Adleman.

MAC – Message Authentication Code

PKC – Public Key Cryptography

BLS - Boneh–Lynn–Shacham signature scheme

I. INTRODUCTION

Several trends are opening up the era of Cloud Computing, which is an Internet-based development and use of computer technology. The ever cheaper and more powerful processors, together with the software as a service (SaaS) computing architecture, are transforming data centers into pools of computing service on a huge scale. The increasing network bandwidth and reliable yet flexible network connections make it even possible that users can now subscribe high quality services from data and software that reside solely on remote data centers. Moving data into the cloud offers great convenience to users since they don't have to care about the complexities of direct hardware management. The pioneer of Cloud Computing vendors, Amazon Simple Storage Service (S3) and Amazon Elastic Compute Cloud (EC2) are both well-known examples. While these internet-based online services do provide huge amounts of storage space and customizable computing resources, this computing platform shift, however, is eliminating the responsibility of local machines for data maintenance at the same time. As a result, users are at the mercy of their cloud service providers for the availability and integrity of their data. On the one hand, although the cloud infrastructures are much more powerful and reliable than personal computing devices, broad range of both internal and external threats for data integrity still exist. Examples of outages and data loss incidents of noteworthy cloud storage services appear from time to time. On the other hand, since users may not retain a local copy of outsourced data, there exist various incentives for cloud service providers (CSP) to behave unfaithfully towards the cloud users regarding the status of their outsourced data. For example, to increase the profit margin by reducing cost, it is possible for CSP to discard rarely accessed data without being detected in a timely fashion. Similarly, CSP may even attempt to hide data loss incidents so as to maintain a reputation. Therefore, although outsourcing data into the cloud is

economically attractive for the cost and complexity of long-term large-scale data storage, its lacking of offering strong assurance of data integrity and availability may impede its wide adoption by both enterprise and individual cloud users. In order to achieve the assurances of cloud data integrity and availability and enforce the quality of cloud storage service, efficient methods that enable on-demand data correctness verification on behalf of cloud users have to be designed. However, the fact that users no longer have physical possession of data in the cloud prohibits the direct adoption of traditional cryptographic primitives for the purpose of data integrity protection. Hence, the verification of cloud storage correctness must be conducted without explicit knowledge of the whole data files. Meanwhile, cloud storage is not just a third-party data warehouse. The data stored in the cloud may not only be accessed but also be frequently updated by the users, including insertion, deletion, modification, appending, etc. Thus, it is also imperative to support the integration of this dynamic feature into cloud storage correctness assurance, which makes the system design even more challenging. Last but not the least, the deployment of Cloud Computing is powered by data centers running in a simultaneous, cooperated and distributed manner. It is more advantages for individual users to store their data redundantly across multiple physical servers so as to reduce the data integrity and availability threats. Thus, distributed protocols for storage correctness assurance will be of most importance in achieving robust and secure cloud storage systems. However, such important area remains to be fully explored in the literature. Recently, the importance of ensuring the remote data integrity has been highlighted by the following research works under different system and security models. These techniques, while can be useful to ensure the storage correctness without having users possessing local data, are all focusing on single server scenario. They may be useful for quality-of-service testing, but does not guarantee the data availability in case of server failures. Although direct applying these techniques to distributed storage (multiple servers) could be straightforward, the resulted storage verification overhead would be linear to the number of servers. As a complementary approach, researchers have also proposed distributed protocols for ensuring storage correctness across multiple servers or peers. However, while providing efficient cross server storage verification and data availability insurance, these schemes are all focusing on static or archival data. As a result, their capabilities of handling dynamic data remain unclear, which inevitably limits their full applicability in cloud storage scenarios. In this paper, we propose an effective and flexible distributed storage verification scheme with explicit dynamic data support to ensure the correctness and availability of users' data in the cloud. We rely on erasure correcting code in the file distribution preparation to provide redundancies and guarantee the data dependability against Byzantine servers, where a storage server may fail in arbitrary ways. This construction drastically reduces the communication and storage overhead as compared to the traditional replication-based file distribution techniques. By utilizing the homomorphic token with distributed verification of erasure-coded data, our scheme achieves the storage correctness insurance as well as data error localization: whenever data corruption has been detected during the storage correctness verification, our scheme can almost guarantee the simultaneous localization of data errors, i.e., the identification of the misbehaving server(s). In order to strike a good balance between error resilience and data dynamics, we further explore the algebraic property of our token computation and erasure-coded data, and demonstrate how to efficiently support dynamic operation on data blocks, while maintaining the same level of storage correctness assurance. In order to save the time, computation resources, and even the related online burden of users we also provide the extension of the proposed main scheme to support third-party auditing, where users can safely delegate the integrity checking tasks to third-party auditors and be worry-free to use the cloud storage services. Our work is among the first few ones in this field to consider distributed data storage security in Cloud Computing. Our contribution can be summarized as the following three aspects:

1) Compared to many of its predecessors, which only provide binary results about the storage status across the distributed servers, the proposed scheme achieves the integration of storage correctness insurance and data error localization, i.e., the identification of misbehaving server(s).

2) Unlike most prior works for ensuring remote data integrity, the new scheme further supports secure and efficient dynamic operations on data blocks, including: update, delete and append.

3) The experiment results demonstrate the proposed scheme is highly efficient. Extensive security analysis shows our scheme is resilient against Byzantine failure, malicious data modification attack, and even server colluding attacks.

In this paper, we address this open issue and propose a secure and scalable fine-grained data access control scheme for cloud computing. Our proposed scheme is partially based on our observation that, in practical application scenarios each data file can be associated with a set of attributes which are meaningful in the context of interest. The access structure of each user can thus be defined as a unique logical expression over these attributes to reflect the scope of data files that the user is allowed to access. As the logical expression can represent any desired data file set, fine-grainedness of data access control is achieved. To enforce these access structures, we define a public key component for each attribute. Data files are encrypted using public key components corresponding to their attributes. User secret keys are defined to reflect their access structures so that a user is able to decrypt a cipher text if and only if the data file attributes satisfy his access structure.

II. CLOUD COMPUTING TECHNOLOGY

Cloud computing is the delivery of computing as a service rather than a product, whereby shared resources, software, and information are provided to computers and other devices as a utility (like the electricity grid) over a network.

Cloud computing provides computation, software, data access, and storage services that do not require end-user knowledge of the physical location and configuration of the system that delivers the services. Parallels to this concept can be drawn with the electricity grid, wherein end-users consume power without needing to understand the component devices or infrastructure required to provide the service.

Cloud computing is different from hosting services and assets at ISP data center. It is all about computing systems are logically at one place or virtual resources forming a Cloud and user community accessing with intranet or Internet. So, it means Cloud could reside in-premises or off-premises at service provider location. There are types of Cloud computing like 1. Public clouds 2. private Clouds 3. Inter-clouds or Hybrid Clouds, say Mr.B.L.V. Rao- CIO and IT Leaders and expert in cloud computing.

Cloud computing describes a new supplement, consumption, and delivery model for IT services based on Internet protocols, and it typically involves provisioning of dynamically scalable and often virtualized resources. It is a byproduct and consequence of the ease-of-access to remote computing sites provided by the Internet. This may take the form of web-based tools or applications that users can access and use through a web browser as if the programs were installed locally on their own computers.

Cloud computing providers deliver applications via the internet, which are accessed from web browsers, desktop and mobile apps, while the business software and data are stored on servers at a remote location. In some cases, legacy applications (line of business applications that until now have been prevalent in thin client Windows computing) are delivered via a screen-sharing technology, while the computing resources are consolidated at a remote data center location; in other cases, entire business applications have been coded using web-based technologies such as AJAX.

At the foundation of cloud computing is the broader concept of infrastructure convergence (or Converged Infrastructure) and shared services. This type of data center environment allows enterprises to get their applications up and running faster, with easier manageability and less maintenance, and enables IT to more rapidly adjust IT resources (such as servers, storage and networking) to meet fluctuating and unpredictable business demand.[6] [7]

Most cloud computing infrastructures consist of services delivered through shared data-centers and appearing as a single point of access for consumers' computing needs. Commercial offerings may be required to meet service-level agreements (SLAs), but specific terms are less often negotiated by smaller companies.

III. CLOUD WORKING PROGRESS:

Cloud computing has been changing how most people use the web and how they store their files. It's the structure that runs sites like Facebook, Amazon and Twitter and the core that allows us to take advantage of services like Google Docs and Gmail. But how does it work.

Before we dig further into how does cloud computing work, first let's understand what the term "cloud"refers to. The concept of the cloud has been around for a long time in many different incarnations in the business world. It mostly means a grid of computers serving as a service-oriented architecture to deliver software and data.

Most websites and server-based applications run on particular computers or servers. What differentiates the cloud from the way those are set up is that the cloud utilizes the resources from the computers as a collective virtual computer, where the applications can run independently from particular computer or server configurations. They are basically floating around in a "cloud of resources", making the hardware less important to how the applications work.

With broadband internet, the need to have the software run on your computer or on a company's site is becoming less and less essential. A lot of the software that people use nowadays are completely web-based. The cloud takes advantage of that to bring it to the next level. The understand how does cloud computing work, imagine that the cloud consists of layers — mostly the back-end layers and the front-end or user-end layers. The front-end layers are the ones you see and interact with. When you access your email on Gmail for example, you are using software running on the front-end of a cloud. The same is true when you access your Facebook account. The back-end consists of the hardware and the software architecture that fuels the interface you see on the front end.

Because the computers are set up to work together, the applications can take advantage of all that computing power as if they were running on one particular machine. Cloud computing also allows for a lot of flexibility. Depending on the demand, you can increase how much of the cloud resources you use without the need for assigning specific hardware for the job, or just reduce the amount of resources assigned to you when they are not necessary.

The transition from being very 'personal hardware dependent' to a world where resources are shared among the masses is creeping up on us slowly and unobtrusively. Very many people have already transitioned to using a cloud environment for most of their time in front of the computer without even realizing it.

Sure, most of us still use some version of Microsoft Office or QuickBooks that was installed on our computers, but even those kinds of software are now offering an online version that can be used instead. The possibility of being able to access your data and software wherever you need it makes this transition very appealing to most people.

Are there problems with this concept? Of course, there are. If for some reason your internet goes down, your access to your data also disappears. There are security concerns with the data and the risk that companies will use proprietary formats for the files and that require that you pay for a certain service monthly or you may lose access to your own data permanently.

So, choose wisely when picking a service to use with your important data and make sure it can be downloaded if needed, but also enjoy the flexibility those services provide. The wave of the future is in the clouds".

IV. PUBLIC CLOUD:

Public cloud describes cloud computing in the traditional mainstream sense, whereby resources are dynamically provisioned to the general public on a fine-grained, self-service basis over the Internet, via web applications/web services, from an off-site third-party provider who bills on a fine-grained utility computing basis.

V. HYBRID CLOUD

Hybrid cloud is a composition of two or more clouds (private, community, or public) that remain unique entities but are bound together, offering the benefits of multiple deployment models. Briefly it can also be defined as a multiple cloud system which are connected in a way that allows programs and data to be moved easily from one deployment system to another.

VI. PRIVATE CLOUD

Private cloud is infrastructure operated solely for a single organization, whether managed internally or by a third-party and hosted internally or externally.

They have attracted criticism because users "still have to buy, build, and manage them" and thus do not benefit from lower up-front capital costs and less hands-on management, essentially "[lacking] the economic model that makes cloud computing such an intriguing concept"

As cloud computing is achieving increased popularity, concerns are being voiced about the security issues introduced through adoption of this new model. The effectiveness and efficiency of traditional protection mechanisms are being reconsidered as the characteristics of this innovative deployment model differ widely from those of traditional architectures.

The relative security of cloud computing services is a contentious issue that may be delaying its adoption. Issues barring the adoption of cloud computing are due in large part to the private and public sectors unease surrounding the external management of security-based services. It is the very nature of cloud computing-based services, private or public, that promote external management of provided services. This delivers great incentive among cloud computing service providers in producing a priority in building and maintaining strong management of secure services. Security issues have been categorized into sensitive data access, data segregation, privacy, bug exploitation, recovery, accountability, malicious insiders, management console security, account control, and multi-tenancy issues. Solution to various cloud security issues vary through cryptography, particularly public key infrastructure (PKI), use of multiple cloud providers, standardization of APIs, improving virtual machine support and legal support

VII. CHARACTERISTICS OF CLOUD COMPUTING

Characteristics Cloud computing is cost-effective. Here, cost is greatly reduced as initial expense and recurring expenses are much lower than traditional computing. Maintenance cost is reduced as a third party maintains everything from running the cloud to storing data. Cloud is characterized by features such as platform, location and device independency, which make it easily adoptable for all sizes of businesses, in particular small and mid-sized. However, owing to redundancy of computer system networks and storage system cloud may not be reliable for data, but it scores well as far as security is concerned. In cloud computing, security is tremendously improved because of a superior technology security system, which is now easily available and affordable. Yet another important characteristic of cloud is scalability, which is achieved through server virtualization. Some of the most important five key characteristics are,

7.1 On-demand Self Service

A consumer can unilaterally provision computing capabilities, such as server time and network storage, as needed automatically without requiring human interaction with each service provider.

7.2 Broad Network Access

Capabilities are available over the network and accessed through standard mechanisms that promote use by heterogeneous thin or thick client platforms.

7.3 Resource Pooling

The provider's computing resources are pooled to serve multiple consumers using a multi-tenant model with different physical and virtual resources dynamically assigned and reassigned according to consumer demand. There is a sense of location independence in that the customer generally has no control or knowledge over the exact location of the provided resources but may be able to specify location at a higher level of abstraction (e.g., country, state, or data center). Examples of resources include storage, processing, memory, network bandwidth, and virtual machines

7.4 Measured Service

Cloud systems automatically control and optimize resource use by leveraging a metering capability at some level of abstraction appropriate to the type of service. Resource usage can be monitored, controlled, and reported providing transparency for both the provider and consumer of the utilized service

7.5 Selection of Provider

A good service provider is the key to good service. So, it is imperative to select the right service provider. One must make sure that the provider is reliable, well-reputed for their customer service and should have a proven track record in IT- related ventures. As cloud computing has taken hold, there are six major benefits that have become clear, 1) Anywhere/anytime access - It promises "universal" access to high-powered computing and storage resources for anyone with a network access device. 2) Collaboration among users -cloud represents an environment in which users can develop software-based services and from which they can deliver

them. 3) Storage as a universal service - the cloud represents a remote but scalable storage resource for users anywhere and everywhere. 4) Cost benefits - the cloud promises to deliver computing power and services at a lower cost.

VIII. ADVANTAGES OF CLOUD COMPUTING

8.1 Lower IT costs

It costs are decreased on several areas: - Applications are no longer run on the desktop Personal Computer (PC), but are run in the cloud. This means that the PC does not need the processing power or hard disk space as demanded by traditional desktop software. - Powerful servers and the like are no longer required. The computing power of the cloud can be used to replace or supplement internal computing resources. - Organizations no longer have to purchase computing resources to handle the capacity peaks. Peaks are easily handled by the cloud. - Payment for most cloud computing services is based on a pay-as-you-go model. This means that customers only pay for what they use. - The IT staff does not have to install and maintain the software on every desktop in the organization.

8.2 Fewer maintenance issues

Fewer maintenance issues with less hardware on hand in the organization, the maintenance costs are accordingly decreased. Also, software is run in the cloud, not on the PC. So, there is no software for the IT staff to maintain. Also, organizations do not have to face the choice between obsolete software and high upgrade costs. The service provider upgrades the software in the cloud, so whenever the customer logs in to the cloud, the latest version is loaded, with no need to pay for or download an upgrade.

8.3 Increased computing power

No longer is the computing power limited to the power of the desktop PC. The power of the entire cloud is at the disposal of the user. This means that bigger tasks can be performed in the cloud than on the desktop.

8.4 Unlimited storage capacity

The cloud offers virtually limitless storage capacity. Improved compatibility between operating systems and documents Documents can be shared with computers that run different operating systems such as Windows, Apple's MAC OS, Linux or UNIX.

8.5 Easier group collaboration

One of the most important advantages to many users of cloud computing is the easy collaboration on documents and projects. Cloud computing no longer requires the correspondence of documents from one user to another, for example by e-mail, and work on them sequentially. Cloud computing allows simultaneous access to documents, and edits in the document are updated real-time.

8.6 Universal access to documents

Documents are stored in the cloud. This means that documents can be accessed from anywhere, as long as a computer and an Internet connection is available.

8.7 Other Advantages

1)Cloud Computing provides the Flexible Architecture to share the application (Software) as well as the other network resources(hardware).
2)Resource Sharing is the main theme.
3)Highly Virtualized and Standardized infrastructures.
4)No need to install or update any software or hardware.
5) It can be accessed from any browser.

IX. DISADVANTAGES OF CLOUD COMPUTING

9.1 Requires a constant Internet connection

Cloud computing is impossible without the connection to the Internet. Internet is needed to access both documents and applications. If no Internet connection is available, this means that no work can be done.

9.2 Does not work well with low-speed connections

Web-based applications and large documents require both a lot of bandwidth to download. With a low-speed connection, such as dial-up, it might take a while to even change pages in a document. Web-based applications have to send everything back and forth from the PC to the cloud, from the interface of the application to the document that is being edited. Even on a fast connection, cloud computing can be slower than accessing a similar application on a desktop PC.

9.3 Features might be limited

For now, web-based applications are not as full-featured as their fellow desktop applications. This might be a big disadvantage for advanced users. Stored data might not be secure All data is stored in the cloud, and thus outside the sphere of control. As shown in the previous section, this provides a lot of advantages. However, safety cannot be guaranteed. Cloud systems can be hacked and documents can be accessed by unauthorized users. This has a big impact on privacy and trust – which is the focus of this research project.

9.4 No physical or local backup

The data is only stored in the cloud. In the off chance that data goes missing, this means that the data cannot be restored by (traditional) local backup systems.

9.5 Drawbacks of cloud computing:

Clouds pose more than just legal problems; there are technical ones, too, according to Bob Liebert, analyst at Enterprise Strategy Group."We say about virtualization that it's hard to manage an environment where your applications are playing hide and seek and your hardware is lying to you," Liebert says. "It's even more with clouds. You're having to try to manage someone else's hardware that's lying to you."

There is no single "cloud" involved in cloud computing, Liebert says. All the SaaS and infrastructure-services providers use different technology and different standards, meaning every vendor relationship will be different. You can't just tool up one application or business process for "the cloud" and be ready to go.

You also can't just move applications to the cloud and expect them to run, even with the best virtualization technology, To move any significant corporate processing into a cloud environment requires at least the same amount of work IT would have to do to move the same workload from its existing servers to new virtual or physical servers, including reconfiguring connections to network and storage resources, Wolf says. Keeping track of what happens after the workloads move often means using a completely different set of management applications that integrate imperfectly, if at all, with a company's existing management applications, Liebert says. IBM, HP, BMC and other data-center systems-management vendors are adding cloud-management functions as quickly as possible in order to try to appeal to customers who have never dealt with them before

9.6 Reliable Distributed Systems:

An understanding of the techniques used to make distributed computing systems and networks reliable, fault-tolerant and secure will be crucial to those who design and deploy the next generation of mission-critical applications and Web Services.

Reliable Distributed Systems reviews and describes the key concepts, principles and applications of modern distributed computing systems and architectures. This self-contained book consists of five parts. The first covers introductory material, including the basic architecture of the Internet, simple protocols such as RPC and TCP, object-oriented architectures, operating systems enhancements for high performance, and reliability issues. The second covers the Web, with a focus on Web Services technologies, Microsoft's .NET and the Java Enterprise Edition. The last three parts look at a number of reliability and fault-tolerance issues and techniques, with an emphasis on replication applied in Web Services settings.

Handling failures is an important theme in distributed systems design. Failures fall into two obvious categories: hardware and software. Hardware failures were a dominant concern until the late 80's, but since then internal hardware reliability has improved enormously. Decreased heat production and power consumption of smaller circuits, reduction of off-chip connections and wiring, and high-quality manufacturing techniques have all played a positive role in improving hardware reliability. Today, problems are most often associated with connections and mechanical devices, i.e., network failures and drive failures.

Building a reliable system that runs over an unreliable communications network seems like an impossible goal. We are forced to deal with uncertainty. A process knows its own state, and it knows what state other processes were in recently. But the processes have no way of knowing each other's current state. They lack the equivalent of shared memory. They also lack accurate ways to detect failure, or to distinguish a local software/hardware failure from a communication failure.

Distributed systems design is obviously a challenging endeavor. How do we do it when we are not allowed to assume anything, and there are so many complexities? We start by limiting the scope. We will focus on a particular type of distributed systems design, one that uses a client-server model with mostly standard protocols. It turns out that these standard protocols provide considerable help with the low-level details of reliable network communications, which makes our job easier. Let's start by reviewing client-server technology and the protocols.

In distributed systems, there can be many servers of a particular type, e.g., multiple file servers or multiple network name servers. The term service is used to denote a set of servers of a particular type. We say that a binding occurs when a process that needs to access a service becomes associated with a particular server which provides the service. There are many binding policies that define how a particular server is chosen. For example, the policy could be based on locality (a Unix NIS client starts by looking first for a server on its own machine); or it could be based on load balance (a CICS client is bound in such a way that uniform responsiveness for all clients is attempted).

A distributed service may employ data replication, where a service maintains multiple copies of data to permit local access at multiple locations, or to increase availability when a server process may have crashed. Caching is a related concept and very common in distributed systems. We say a process has cached data if it maintains a copy of the data locally, for quick access if it is needed again. A cache hit is when a request is satisfied from cached data, rather than from the primary service. For example, browsers use document caching to speed up access to frequently used documents.

Caching is similar to replication, but cached data can become stale. Thus, there may need to be a policy for validating a cached data item before using it. If a cache is actively refreshed by the primary service, caching is identical to replication. As mentioned earlier, the communication between client and server needs to be reliable. You have probably heard of TCP/IP before. The Internet Protocol (IP) suite is the set of communication protocols that allow for communication on the Internet and most commercial networks. The Transmission Control Protocol (TCP) is one of the core protocols of this suite. Using TCP, clients and servers can create connections to one another, over which they can exchange data in packets. The protocol guarantees reliable and in-order delivery of data from sender to receiver.

The IP suite can be viewed as a set of layers, each layer having the property that it only uses the functions of the layer below, and only exports functionality to the layer above. A system that implements protocol behavior consisting of layers is known as a protocol

stack. Protocol stacks can be implemented either in hardware or software, or a mixture of both. Typically, only the lower layers are implemented in hardware, with the higher layers being implemented in software.

X. CLOUD DEPLOYMENT MODELS:

The selection of cloud deployment model depends on the different levels of security and control required. The Private cloud infrastructure is operated solely for a single organization with the purpose of securing services and infrastructure on a private network. This deployment model offers the greatest level of security and control, but it requires the operating organization to purchase and maintain the hardware and software infrastructure, which reduces the cost saving bene_ts of investing in a cloud infrastructure. Rackspace, Eucalyptus, and VMware6 are example providers of private cloud solutions. A Community cloud infrastructure is shared by several organizations and supports a specific community that has shared concerns. It may be established where organizations have similar requirements and seek to share cloud infrastructure. Example of community cloud is Google's Gov Cloud. Public clouds provide services and infrastructure over the Internet to the general public or a large industry group and is owned by an organization selling cloud services. Major public cloud providers are Google and Amazon. These clouds offer the greatest level of efficiency in shared resources; however, they are also more vulnerable than private clouds.
A Hybrid cloud infrastructure, as the name suggests, is a composition of private, public, and/or community clouds possibly through multiple providers. Reasoning for hybrid cloud infrastructure is to increase security, better management or failover purposes. For some it may not be feasible to place assets in a public cloud, therefore many opt for the value of combining different cloud deployment models. The drawbacks of a hybrid cloud however is the requirements of managing multiple different security platforms and communication protocols.

XI. CLOUD APPLICATION UNDER TEST PLAN

This section and the following describe the test plan, the test design, and the test case specification for a distributed image rendering application running in the Amazon
EC2 cloud environment. Jata was described and the application features Following the test plan in this section the test design is covered the test case specification is covered. A brief description of the test procedure Note that the IEEE format is only used as basis for the structure of this thesis text. The following sections might omit details that are not relevant to the research purpose of this thesis, but might be needed in a conventional test documentation. The thesis also adds information to support the thesis work, such as source code listings, as is normally not done in a conventional software test. documentation.
The testing focuses on the application management, i.e. the communication with workers and the distribution of rendering tasks in order to verify that testing can be performed within a cloud computing environment. The integration between the web interface layer and the request manager is also under test. The testing is done by simulating an HTTP request to the web layer which needs to interpret the request correctly in order to start the application management. The rendering scenario which forms the basis of the testing is shown in Application output testing is done in the way that a dummy non-rendered image is assembled and verified. The dummy image does not represent the input scene information. The application management uses the scene _le however in deciding the number of buckets (tasks) needed for the job. The input scene _le used requires 300 rendering tasks to be handled resulting in 300 image parts. The disk file size of the composed dummy image is verified in a simple manner to make sure all parallel test components have been issued the precise number of rendering tasks and returned dummy image parts accordingly.
The input scene file for the rendering process is preloaded to the application manager's node, so no input parameters are needed in the stimulation message. The scene file is used to determine the job's tasks for the rendering workers. After loading the scene _le, the request manager sends 'run' (initialize) message to each available worker node (interactions 2 and 3), announcing that they should take part in a rendering job. After receiving a initialization message, the workers themselves are responsible for making calls to the request manager asking for the next bucket in line to render (interactions 4 and 5). In the Jata test case implementation the worker OSGi service has been switched out for a test double [42]. The test double can be described as a stub that forwards rendering information from the request manager to the test system. It is also responsible for sending fake image parts back to the request manager instead of using Sunflow.
This section describes the cloud application that is used as the system under test (SUT) for the case study in this thesis. Sun_ow, multi processing image rendering application, was selected as the application under test for its parallel functionality. Sunflow is an open source rendering system for photo-realistic image synthesis. It is written in Java and built around a flexible ray tracing core and an extensible object-oriented design.
The reason for using parallelization and concurrent computing application is to explore to the full, with advanced testing techniques, the challenges proposed in testing distributed systems within a cloud environment. Image rendering is an extremely time-consuming computing task, therefore rendering systems like Sun flow have been created to split up the task into smaller subtasks for parallel computing. For a typical rendering job, one master node is needed for the application management and several worker nodes for solving individual rendering tasks.
The control part or the distribution management of the application is therefore a good candidate for testing within a cloud environment. One cloud instance is needed for the master node and an arbitrary number of other instances for the workers, where the focus of the testing is on the communication of the distributed components involved.
Architectural scripting is a way to model the dynamic aspect of runtime and deployment-time software architecture. The notion of architectural scripting and the exploration of its theoretical and practical utility was introduced by Ingstrup and Hansen in Modeling Architectural Change: Architectural scripting and its applications to reconfiguration.

Skulason presents Cloud-ASL, an external domain specific language (DSL) which enables architectural operations and architectural scripting in cloud computing environments to create and initialize cloud instances. To model and test the Cloud-ASL, a working architectural software prototype was implemented and named Turnip. The Turnip software is an extension of Sunflow to enable its distributed ray-tracing features within a cloud computing environment. The objective with the Sun flow extension was to create a case study application to use with Cloud-ASL, mainly for launching and destroying worker cloud instances.
In Turnip, Sun flow is adapted to run the different rendering workers on separate computers in a cloud. The distribution algorithm is based on Helios3, which supports
Sun flow distributed rendering computations on a grid service implemented with Jini4 technology. Turnip is used in this thesis case study as the SUT with focus on testing the application management features.

XII. TPA (THIRD PARTY AUDITOR)

The third-party auditor (TPA), who has expertise and capabilities that cloud users do not have and is trusted to assess the cloud storage service security on behalf of the user upon request. Users rely on the CS for cloud data storage and maintenance. They may also dynamically interact with the CS to access and update their stored data for various application purposes. The users may resort to TPA for ensuring the storage security of their outsourced data, while hoping to keep their data private from TPA. We consider the existence of a semi-trusted CS as does. Namely, in most of time it behaves properly and does not deviate from the prescribed protocol execution. However, during providing the cloud data storage-based services, for their own benefits the CS might neglect to keep or deliberately delete rarely accessed data files which belong to ordinary cloud users. Moreover, the CS may decide to hide the data corruptions caused by server hacks or Byzantine failures to maintain reputation. We assume the TPA, who is in the business of auditing, is reliable and independent, and thus has no incentive to collude with either the CS or the users during the auditing process. TPA should be able to efficiently audit the cloud data storage without local copy of data and without bringing in additional on-line burden to cloud users
To securely introduce an effective third-party auditor (TPA), the following two fundamental requirements have to be met:
TPA should be able to efficiently audit the cloud data storage without demanding the local copy of data, and introduce no additional on-line burden to the cloud user.
The third-party auditing process should bring in no new vulnerabilities towards user data privacy.

XIII. MULTIPLE BATCH AUDITING

TPA may concurrently handle multiple auditing delegations upon different users' requests. The individual auditing of these tasks for 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 TPA to batch these multiple tasks together and audit at one time [4]. Keeping this natural demand in mind, we propose the technique of bilinear aggregate signature, which supports the aggregation of multiple signatures by distinct signers on distinct messages into a single signature and thus provides efficient verification for the authenticity of all messages. Using this signature aggregation technique and bilinear property, we can now aggregate K verification equations into a single one, so that the simultaneous auditing of multiple tasks can be achieved.

XIV. PROBLEM DEFINITION

One of the biggest concerns with cloud data storage is that of data integrity verification at untrusted servers. For example, the storage service provider, which experiences Byzantine failures occasionally, may decide to hide the data errors from the clients for the benefit of their own. What is more serious is that for saving money and storage space the service provider might neglect to keep or deliberately delete rarely accessed data files which belong to an ordinary client. Consider the large size of the outsourced electronic data and the client's constrained resource capability, the core of the problem can be generalized as how can the client find an efficient way to perform periodical integrity verifications without the local copy of data files.

XV. PROBLEM OBJECTIVE

In order to solve the problem of data integrity checking, many schemes are proposed under different systems and security models. In all these works, great efforts are made to design solutions that meet various requirements: high scheme efficiency, stateless verification, unbounded use of queries and retrievability of data, etc. Considering the role of the verifier in the model, all the schemes presented before fall into two categories: private auditability and public auditability. Although schemes with private auditability can achieve higher scheme efficiency, public auditability allows anyone, not just the client (data owner), to challenge the cloud server for correctness of data storage while keeping no private information. Then, clients are able to delegate the evaluation of the service performance to an independent third-party auditor (TPA), without devotion of their computation resources. In the cloud, the clients themselves are unreliable or may not be able to afford the overhead of performing frequent integrity checks. Thus, for practical use, it seems more rational to equip the verification protocol with public auditability, which is expected to play a more important role in achieving economies of scale for Cloud Computing. Moreover, for efficiency consideration, the outsourced data themselves should not be required by the verifier for the verification purpose.

XVI. LITERATURE SURVEY:INTRODUCING EFFECTIVE THIRD-PARTY AUDITING (TPA) FOR DATA STORAGE SECURITY IN CLOUD

Disadvantages:
The most promising one we believe is a model in which public verifiability is enforced. Public verifiability, allows TPA to audit the cloud data storage without demanding users' time, feasibility or resources. An interesting question in this model is if we can construct a scheme to achieve both public verifiability and storage correctness assurance of dynamic data.

Advantages:

1. Append Operation in the cloud.
2. Update operation in the cloud.
3. Delete operation in the cloud.

XVII. AN EFFICIENT REMOTE DATA POSSESSION CHECKING IN CLOUD STORAGE

Advantages:

- An efficient remote data possession checking (RDPC) scheme is proposed.
- It almost satisfies all the requirements for cloud storage.
- First, it is efficient in terms of computation and communication.
- Second, it allows verification without the need for the challenger to compare against the original data, and it can be verified by comparing only the responds returned by the storage server.
- Users need to store only two secret keys and several random numbers.
- Finally, based on Euler's theorem, a challenge updating method is proposed. The efficiency of the scheme makes it ideally suited for use in cloud storage.

Disadvantages:

- The paper doesn't consider data updating which will be the future works.
- In addition, we will apply the scheme to a practical system.

XVIII. PRIVATE EDITING USING UNTRUSTED CLOUD SERVICES

Advantages:

- The contents of the file are protected (both confidentiality and optionally integrity) even against attacks from a possibly malicious cloud service provider.
- The extension has minimal impact on the existing functionality of the cloud application and requires no cooperation from the application provider.
- The incurred runtime and bandwidth costs are acceptable for typical uses.
- We achieve this by using a new data structure that supports variable-length blocks in an incremental encryption scheme.

Disadvantages:

- It is a light-weight component.
- The techniques cannot provide the highest level of privacy, especially against a malicious adversary with control over the client application.

XIX. PRIVACY-PRESERVING PUBLIC AUDITING FOR DATA STORAGE SECURITY IN CLOUD COMPUTING

Disadvantages:

- Batch auditing: There are K users having K files on the same cloud They have the same TPA. Then, the TPA can combine their queries and save in computation time.
- Data dynamics :The data on the cloud may change according to applications.

Advantages:
We utilize the homomorphic authenticator and random masking to guarantee that 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.

ENCRYPTED DOMAIN PROCESSING FOR CLOUD PRIVACY
Author **D. A. Rodr´ıguez-Silva, F. J. Gonz´alez-Casta˜no,**
L. Atkinson-Orellana, A. Fern´andez-Cordeiro
Year**:2010**

Abstract:
Cloud security comprises access control and end-to-end security based on flow or message-level privacy.
The many traditional solutions for service-oriented architectures can be easily applied to Cloud environments. Regarding privacy, current schemas mainly focus on sending user information in a secure way to Cloud servers.
Cloud simply handles data storage and on-line data encryption guarantees privacy.
This is a complex context for Cloud Computing security due to the need to distribute resources among different locations to maximize performance.
Despite the many advantages of Cloud Computing such as scalability, flexibility and cost savings, there are no guarantees that stored data will not be accessed by unauthorized entities, such as the Cloud provider itself or malicious attackers.

Existing System:
The practical and provably semantically secure cryptosystems that present a privacy homomorphism typically allow for the execution of one arithmetic operation directly on ciphertexts, without the need for decryption or interaction with a trusted decryption party.
When the operation takes place for the first time, the client initializes the object module EncryptedProcessClient for the execution of that operation.
It provides transparent encryption of the data prior to their transmission to the server, and decryption of the data received from the server prior to their presentation.
It provides access to the arithmetic operations on the encrypted data received from the client or stored on the server, applying suitable privacy homeomorphisms.

Proposed System:
We propose executing server-side operations in the encrypted domain, so that both the operands and the results are opaque to the server.
The user can deliberately assume the risks of placing information on the Cloud, there exist activities in which law regulates data protection.
There are proposals of additional security layers to protect the user from data mishandling by Cloud providers.

ENSURING DATA STORAGE SECURITY IN CLOUD COMPUTING

Author: **Cong Wang, Qian Wang, and Kui Ren**
Year **: June 2011**

Abstract:
The ever cheaper and more powerful processors, together with the software as a service (SaaS) computing architecture, are transforming data centers into pools of computing service on a huge scale.
Moving data into the cloud offers great convenience to users since they don't have to care about the complexities of direct hardware management.
The internet-based online services do provide huge amounts of storage space and customizable computing resources, this computing platform shift, however, is eliminating the responsibility of local machines for data maintenance at the same time.
These techniques, while can be useful to ensure the storage correctness without having users possessing data, cannot address all the security threats in cloud data storage, since they are all focusing on single server scenario and most of them do not consider dynamic data operations.
Cloud Computing is powered by data centers running in a simultaneous, cooperated and distributed manner. Individual user's data is redundantly stored in multiple physical locations to further reduce the data integrity threats.

Existing System:
To ensure storage correctness under dynamic data update is hence of paramount importance. This dynamic feature also makes traditional integrity insurance techniques futile and entails new solutions.
The storage correctness without having users possessing data, cannot address all the security threats in cloud data storage, since they are all focusing on single server scenario and most of them do not consider dynamic data operations.
Cloud Computing inevitably poses new challenging security threats for number of reasons. Firstly, traditional cryptographic primitives for the purpose of data security protection cannot be directly adopted due to the users' loss control of data under Cloud Computing.
The data stored in the cloud may be frequently updated by the users, including insertion, deletion, modification, appending, reordering, etc. To ensure storage correctness under dynamic data update is hence of paramount importance.
Compared to many of its predecessors, which only provide binary results about the storage state across the distributed servers, the challenge-response protocol in our work further provides the localization of data error.

Proposed system:
Unlike most prior works for ensuring remote data integrity, the new scheme supports secure and efficient dynamic operations on data blocks, including: update, delete and append.
Extensive security and performance analysis show that the proposed scheme is highly efficient and resilient against Byzantine failure, malicious data modification attack, and even server colluding attacks.
The straightforward and trivial way to support these operations is for user to download all the data from the cloud servers and re-compute the whole parity blocks as well as verification tokens.
The user can always ask servers to send back blocks of the rows specified in the challenge and regenerate the correct blocks by erasure correction.

We believe that data storage security in Cloud Computing, an area full of challenges and of paramount importance, is still in its infancy now, and many research problems are yet to be identified.
We believe is a model in which public verifiability is enforced. We can construct a scheme to achieve both public verifiability and storage correctness assurance of dynamic data.

Project Background

Cloud computing is emerging at the convergence of three major trends — service orientation, virtualization and standardization of computing through the Internet. Cloud computing enables users and developers to utilize services without knowledge of, expertise with, nor control over the technology infrastructure that supports them. The concept generally incorporates combinations of the following:
Infrastructure as service (IaaS)
Platform as a service (PaaS)
Software as a service (SaaS)

Users avoid capital expenditure (CapEx) on hardware, software, and services when they pay a provider only for what they use. Consumption is billed on a utility (e.g. resources consumed, like electricity) or subscription (e.g. time based, like a newspaper) basis with little or no upfront cost**.**

Cloud Vendors

There are many companies who are into the market offering various ranges of services on Cloud Computing. The major players are VMware, Sun Microsystems, Rackspace US, IBM, Amazon, Google, Microsoft, and Yahoo. Cloud services are also being adopted by individual users through large enterprises including VMware, General Electric, and Procter & Gamble. The vendor hosts and manages the infrastructure required with the respective technology.

Cloud as a Service to Customer

The cloud computing that are evolving as a service in the cloud are being provided by big enterprises with a heavy investment with resource and technology which are accessed by others via the internet. The resources are accessed in this manner as a service – often on a subscription basis. The users of the services being offered often have very little knowledge of the technology being used. The users also have no control over the infrastructure that supports the technology they are using.
There are six different forms that have been consolidated so far to understand how the services are being provided to the customers:

SAAS

This type of cloud computing deliver a single application through the browser to thousands of customers using a multitenant architecture. On the customer side, it means no upfront investment in servers or software licensing; on the provider side, with just one app to maintain, costs are low compared to conventional hosting. SaaS is also common for HR apps and has even worked its way up the food chain to ERP, with players such as Workday. And some who could have predicted the sudden rise of SaaS desktop applications, such as Google Apps and Zoho Office.

Existing System

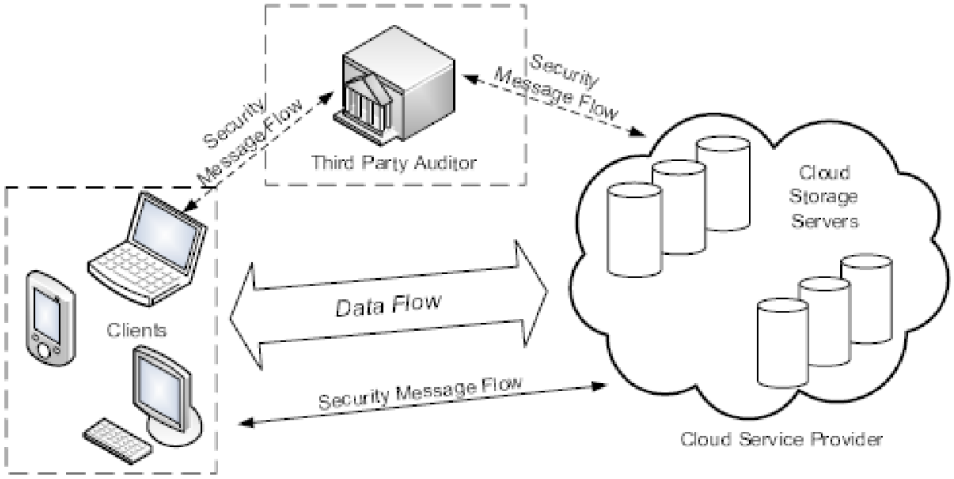
To securely introduce an effective third-party auditor (TPA), the following two fundamental requirements have to be met in the Existing system

- TPA should be able to efficiently audit the cloud data storage with demanding the local copy of data.
- On-line burden to the cloud user.
- Data Security and integrity is less.
- The third-party auditing process should bring in new vulnerabilities towards user data privacy

Proposed System

We utilize the public key based homomorphic authenticator and uniquely integrate it with random mask technique to achieve a privacy-preserving public auditing system for cloud data storage security while keeping all above requirements in mind.
Extensive security and performance analysis show the proposed schemes are provably secure and highly efficient.
We also show how to extent our main scheme to support batch auditing for TPA upon delegations from multi-users.

Architectural Representation



Module Specification

- Key Generation
- Assigning Key to File
- Data Storage on Cloud Server
- TPA Integrity Verification
- Dynamic Data Verification
- Batch Auditing

Module 1: Key generation

- Choose the multiple data.
- Store the data into form of Merkle-Hash Tree Structure.
- Count the number of files.
- Create the Secret keys For Authentication.

Module 2: Assigning Keys

- Map the keys to files.
- Encrypt the files using that corresponding keys.
- Store the keys and data in a hash table.
- Because accessing the data using index is less complexity.
- We cannot do search the whole data, just we search index of the data. So, the process will be very speed.

Module 3: Data Storage in cloud server

- Store that encrypted files in a different location in a cloud server.
- The requester only having that corresponding keys.
- The requester gives those keys to the Third-Party Auditor.
- Then the TPA will use that keys and checks the data verification.
- But the TPA cannot see the original data. Only checks the validation using Signature scheme in cryptography.

Module 4: Integrity verification

- Decrypt the each and every file in a cloud server.
- Combine all the files.
- Check the data size and the size will be same of original data.
- If any data loss occurs for technical problem in a particular file, then put the corresponding encrypt file in that location.
- We cannot loss security, because we store the entire file in a encrypt format.

Module 5: Dynamic Data

In this module, we are doing some operations in cloud server during run time.

- Data Modification.
- Data insertion.

- Data Deletion.

Module 6: Batch Auditing

- In a cloud server, lot of users stores their files.
- So, each user validates their data using batch system.
- For that purpose, we use some scheduling algorithms and priority algorithms for avoiding technical problems (i.e.) bottleneck, deadlock.
- So, the auditing time will be very less.

XX. CONCLUSION

To ensure cloud data storage security, it is critical to enable a third-party auditor (TPA) to evaluate the service quality from an objective and independent perspective. Public auditability also allows clients to delegate the integrity verification tasks to TPA while they themselves can be unreliable or not be able to commit necessary computation resources performing continuous verifications. Another major concern is how to construct verification protocols that can accommodate dynamic data files. In this paper, we explored the problem of providing simultaneous public auditability and data dynamics for remote data integrity check in Cloud Computing. Our construction is deliberately designed to meet these two important goals while efficiency being kept closely in mind. To achieve efficient data dynamics, we improve the existing proof of storage models by manipulating the classic Merkle Hash Tree (MHT) construction for block tag authentication. To support efficient handling of multiple auditing tasks, we further explore the technique of bilinear aggregate signature to extend our main result into a multi-user setting, where TPA can perform multiple auditing tasks simultaneously. Extensive security and performance analysis show that the proposed scheme is highly efficient and provably secure.

References

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. Saint Malo, France: Springer- Verlag, 2009, pp. 355–370.

[2] 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. New York, NY, USA: ACM, 2007, pp. 598–609.

[3] A. Juels and B. S. Kaliski, Jr., "Pors: proofs of retrievability for large files," in Proc. of CCS'07. New York, NY, USA: ACM, 2007, pp. 584–597.

[4] H. Shacham and B. Waters, "Compact proofs of retrievability," in Proc. of ASIACRYPT'08. Melbourne, Australia: Springer-Verlag, 2008, pp. 90–107.

[5] K. D. Bowers, A. Juels, and A. Oprea, "Proofs of retrievability: Theory and implementation," Cryptology ePrint Archive, Report 2008/175, 2008.

[6] M. Naor and G. N. Rothblum, "The complexity of online memory checking," in Proc. of FOCS'05, Pittsburgh, PA, USA, 2005, pp. 573–584.

[7] E.-C. Chang and J. Xu, "Remote integrity check with dishonest storage server," in Proc. of ESORICS'08. Berlin, Heidelberg: Springer-Verlag, 2008, pp. 223–237.

[8] M. A. Shah, R. Swaminathan, and M. Baker, "Privacy-preserving audit and extraction of digital contents," Cryptology ePrint Archive, Report 2008/186, 2008.

[9] A. Oprea, M. K. Reiter, and K. Yang, "Space-efficient block storage integrity," in Proc. of NDSS'05, San Diego, CA, USA, 2005.