BIOINFO Security Informatics
ISSN: 2249-9423 & E-ISSN: 2249-9431, Volume 2, Issue 2, 2012, pp.-49-52.
Available online at http://www.bioinfo.in/contents.php?id=344

HOMOMORPHIC AUTHENTICATION WITH RANDOM MASKING TECHNIQUE ENSURING PRIVACY & SECURITY IN CLOUD COMPUTING

JACHAK K.B.*, KORDE S.K., GHORPADE P.P. AND GAGARE G.J.

Department of Information Technology Engineering, University of Pune, P.R.E.C. Loni-413736, MS, India
*Corresponding Author: Email- mr.kuldeep@hotmail.com

Received: March 15, 2012; Accepted: April 12, 2012

Abstract- Cloud computing may be defined as delivery of product rather than service. Cloud computing is a 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 article 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. Signing is performed using Boneh–Lynn–Shacham (BLS) algorithm which is more secure compared to other algorithms. To ensure the correctness of data, we consider an external auditor called as third party auditor (TPA), on behalf of the cloud user, to verify the integrity of the data stored in the cloud. By utilizing public key based homomorphic authenticator with random masking privacy preserving public auditing can be achieved. The technique of bilinear aggregate signature is used to achieve batch auditing. Batch auditing reduces the computation overhead. Extensive security and performance analysis shows the proposed schemes are provably secure and highly efficient.
Keywords- Batch Auditing, Bilinear Aggregate Signature, BLS Algorithm, Constrained Computing Resources, Data Integrity Protection, Homomorphic Authenticator, Outsourced Data, Third Party Auditor

Citation: Jachak K.B., et al (2012) Homomorphic Authentication with Random Masking Technique Ensuring Privacy and Security in Cloud Computing. BIOINFO Security Informatics, ISSN: 2249-9423 & E-ISSN: 2249-9431, Volume 2, Issue 2, pp.-49-52.

Introduction

Cloud computing is a general term for anything that involves delivering hosted services over the internet. These services are broadly divided into three categories: Infrastructure-as-a-Service (IaaS), Platform-as-a-Service (PaaS) and Software-as-a-Service (SaaS). A cloud service has three distinct characteristics that differentiate it from traditional hosting. It is sold on demand, typically by the minute or the hour it is elastic - a user can have as much or as little of a service as they want at any given time and the service is fully managed by the cloud service provider (the consumer needs nothing but a personal computer and Internet access).The advantage of cloud is cost saving. The prime disadvantage is security. Cloud computing is used by many software industries nowadays. Since the security is not provided in cloud, many companies adopt their unique security structure. Introducing a new and uniform security structure for all types of cloud is the problem we are going to tackle in this paper. Since the data placed in the cloud is accessible to everyone, security is not guaranteed.

To ensure security, cryptographic techniques cannot be directly adopted. Sometimes the cloud service provider may hide the data corruptions to maintain the reputation. To avoid this problem, we introduce an effective third party auditor (TPA) to audit the user's outsourced data when needed. The security is achieved by signing the data blocks. Signing is performed using BLS algorithm. We utilized public key based homomorphic authenticator with random masking to achieve privacy preserving auditing protocol. TPA performs the auditing task for each user i.e. single auditing. This increases the auditing time and computation overhead. The technique of Bilinear Aggregate Signature is used to achieve batch auditing.

Infrastructure Models

There are many considerations for cloud computing architects to make when moving from a standard enterprise application deployment model to one based on cloud computing.

- Software as a Service (SaaS)

Software as a service features a complete application offered as a service on demand. A single instance of the software runs on the cloud and services multiple end users or client organizations.

- Platform as a Service (PaaS)

Platform as a service encapsulates a layer of software and provides it as a service that can be used to build higher-level services.

- Infrastructure as a Service (IaaS)

Infrastructure as a service delivers basic storage and compute capabilities as standardized services over the network. Servers, storage systems, switches, routers, and other systems are pooled and made available to handle workloads that range from application components to high-performance computing applications.

Related Work

We are going to tackle the problem of how to enable a privacy-preserving third-party auditing protocol, independent to data encryption in this paper. Besides, with the prevalence of Cloud Computing, a foreseeable increase of auditing tasks from different users may be delegated to TPA. As the individual auditing of these growing tasks can be tedious and cumbersome, a natural demand is then how to enable TPA to efficiently perform the multiple auditing tasks in a batch manner, i.e., simultaneously. To address these problems, our work utilizes the technique of public key based homomorphic authenticator [1, 3] which enables TPA to perform the auditing without demanding the local copy of data and thus drastically reduces the communication and computation overhead as compared to the straightforward data auditing approaches. By integrating the homomorphic authenticator with random mask technique, our protocol guarantees that TPA could not learn any knowledge about the data content stored in the cloud server during the efficient auditing process. The aggregation and algebraic properties of the authenticator further benefit our design for the batch auditing.

Existing System

We proposed few schemes for ensuring data integrity of stored data. We start by motivating our approach by highlighting the disadvantages of using two Basic Schemes.

Basic Scheme I

The cloud user pre-computes Message Authentication Code (MACs) i = MACsk(i||mi) of each block mi (i ∈{1, . . . , n}), sends both the data file F and the MACs {_i}1≤i≤n onto the cloud server, and releases the secret key (sk) to TPA. During the Audit phase, the TPA requests from the cloud server a number of randomly selected blocks and their corresponding MACs to verify the correctness of the data file. The insight behind this approach is that auditing most of the file is much easier than the whole of it. But with certain drawbacks like

The audit from TPA demands retrieval of user's data, which should be prohibitive because it violates the privacy-preserving guarantee.

Its communication and computation complexity are both linear with respect to the sampled data size, which may result in large communication overhead and time delay, especially when the bandwidth available between the TPA and the cloud server is limited.

Basic Scheme II

To avoid retrieving data from the cloud server, one may improve the above solution as follows: Before data outsourcing, the cloud user chooses s random message authentication code keys, pre-computes MACs, for the whole data file F, and publishes these verification metadata to TPA. The TPA can each time reveal a secret key to the cloud server and ask for a fresh keyed MAC for comparison, thus achieving privacy-preserving auditing, having certain drawbacks

The number of times a particular data file can be audited is limited by the number of secret keys that must be a fixed priori. Once all possible secret keys are exhausted, cloud user then has to retrieve data from the server in order to re-compute and re-publish new MACs to TPA.

The TPA has to maintain and update state between audits, i.e., keep a track on the possessed MAC keys. Considering the potentially large number of audit delegations from multiple users, maintaining such states for TPA can be difficult and error prone.

Proposed System

Technical contribution in this paper is summarized as follows:

- It supports an external auditor to audit the user's outsourced data without learning knowledge on the data content.
- Achieves batch auditing where multiple delegated auditing tasks from different users can be performed simultaneously by the TPA.
- Also supports dynamic operations on data blocks i.e. data update, append and delete.

Cloud computing components are classified as:

i) Cloud User (CU)
ii) Cloud Service Provider (CSP) & Cloud Server (CS)
iii) Third party Auditor (TPA)

Now let's get to know the component working for cloud computing in detail.

i) Cloud User (CU)

Cloud user who has large amount of data files to be stored in the cloud; 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.

ii) Cloud Service Provider (CSP) & Cloud Server (CS)

Services made available to users on demand via the Internet from a cloud computing provider's servers as opposed to being provided from a company's own on-premises servers. Cloud services are designed to provide easy, scalable access to applications, resources and services, and are fully managed by a cloud services provider.

A cloud service can dynamically scale to meet the needs of its users, and because the service provider supplies the hardware and software necessary for the service, there's no need for a company to provision or deploy its own resources or allocate IT staff to manage the service. Examples of cloud services include online data storage and backup solutions, Web-based e-mail services, hosted office suites and document collaboration services, database processing, managed technical support services and more; and the person or authority who manages it is called as Cloud Service provider.

iii) Third party Auditor (TPA)

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 o 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.

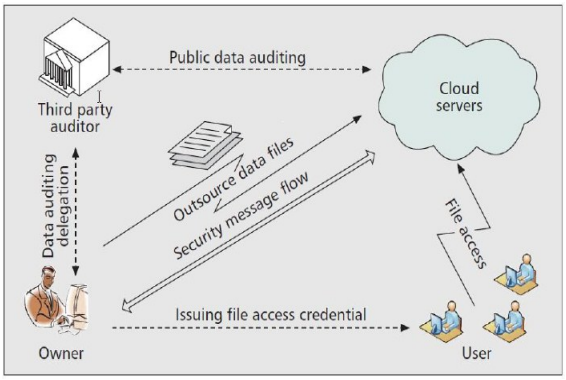

Fig. 1- Architecture for Cloud Storage Devices

Privacy Preserving Public Auditing Module

We use the technique to uniquely integrate the homomorphic authenticator with random masking technique. In our system, the linear combination of sampled blocks in the server's response is masked with randomness generated by a pseudo random function (PRF).With random masking, the TPA no longer has all the necessary information to build up a correct group of linear equations and therefore cannot derive the user's data content, no matter how many linear combinations of the same set of file blocks can be collected. Meanwhile, due to the algebraic property of the homomorphic authenticator, the correctness validation of the block-authenticator pairs will not be affected by the randomness generated from a PRF. A public auditing scheme consists of four algorithms (KeyGen, SigGen, GenProof, and 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 loud server to generate a proof of data storage correctness, while, VerifyProof is run by the TPA to audit the proof from the cloud server. We propose a batch signature scheme based on the BLS signature algorithm.

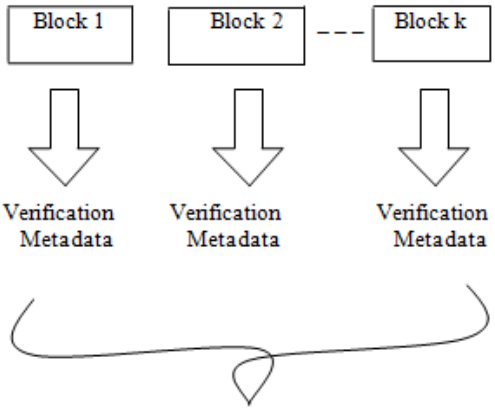

Fig. 2- Homomorphic Authenticator

The BLS signature scheme uses a cryptographic primitive called pairing, which can be defined as a map over two cyclic groups G1 and G2. The BLS signature scheme consists of three phases:

a) In the key generation phase, a sender chooses a random integer $x \in Zp$ and computes $y= g1x \in G1$. The private key is x and public key is y.

b) Given a message $m \in \{0.1\}$ in the signing phase, the sender first computes $h=h(m) \in G1$, where h () is a hash function, and then computes $\sigma = hx \in G1$.The signature of m is σ.

c) In the verification phase, the receiver first computes $h= h(m) \in G1$ and then check whether $e(h,y)=e(\sigma,g1)$. If the verification succeeds, then the message m is authentic.

So the advantage we secure is generation of a very short signature and also can resolve communication overhead.

Utilizing Homomorphic Authenticators

To significantly reduce the arbitrarily large communication overhead for public audit ability without introducing any online burden on the data owner, we resort to the homomorphic authenticator technique [6, 7].

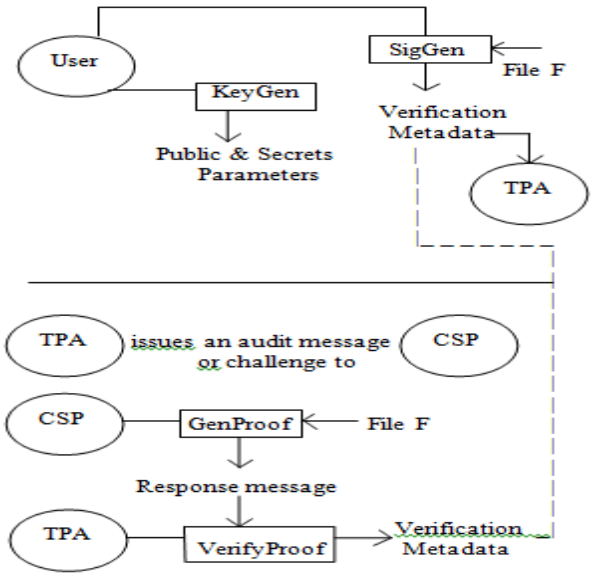

Fig. 3- Setup and Audit Phase

Homomorphic authenticators are unforgeable metadata generated from individual data blocks, which can be securely aggregated in such a way to assure a verifier that a linear combination of data blocks is correctly computed by verifying only the aggregated authenticator. Using this technique requires additional information encoded along with the data before outsourcing. If enough linear combinations of the same blocks are collected, the TPA can simply derive the sampled data content by solving a system of linear equations.

Batch Auditing Module

With the establishment of privacy-preserving public auditing in Cloud Computing, 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. Batch auditing not only allows TPA to perform the multiple auditing tasks simultaneously, but also greatly reduces the computation cost on the TPA side. Hence, supporting data dynamics for privacy-preserving public risk auditing is also of paramount importance. Our main scheme can be adapted to build upon the existing work to support data dynamics, including block level operations of modification, deletion and insertion. We can adopt this technique in our design to achieve privacy-preserving public risk auditing with support of data dynamics.

Additional Security Considerations

Assessment with respect to the system and data being considered for Cloud resources should be conducted to ensure all risks are identified, understood, and accounted for. There are many tools available, such as the Privacy Impact Assessment (PIA) and Federated Privacy Impact Assessment (F-PIA) with which an organization or organizations can demonstrate privacy requirements at different phases of design and delineate their data protection efforts. In a Cloud environment there are new and different threats (both to privacy and security), and evaluators need different guidance on how to test Cloud delivery mechanisms against these threats. Regulators and standards bodies should thus update and modify their existing, computing-related regulations and standards to account for the differences in architecture and operation in a Cloud Environments. These new standards and regulations should be principles-based and media-neutral, in order to accommodate the required technical flexibility. Further, Cloud-based organizations may wish to engage these groups in order to assist in the development of appropriate principles – such input will almost certainly be crucial to the success of these new models. Organizations must rethink their established software development, validation, certification, and accreditation processes in response to the need to push or pull applications in the Cloud. They may thus need to re-design their Software Development Life Cycle (SDLC) – building privacy in and looking to solutions or evaluator techniques that extend beyond the trusted perimeter. Similar to security, designing privacy into a system is the best way to achieve effective protections. Early and comprehensive integration of privacy and security into the software design phase should be the focus.

Future Aspect

The day is not too far when applications will cease to be aware of physical hardware. Much like plugging in a microwave in order to power it doesn't require any knowledge of electricity, one should be able to plug in an application to the cloud in order to receive the power it needs to run, just like a utility. As an architect, you will manage abstract compute, storage and network resources instead of physical servers. Applications will continue to function even if the underlying physical hardware fails or is removed or replaced. Applications will adapt themselves to fluctuating demand patterns by deploying resources instantaneously and automatically, thereby achieving highest utilization levels at all times. Scalability, Security, High availability, Fault-tolerance, Testability and Elasticity will be configurable properties of the application architecture and will be an automated and intrinsic part of the platform on which they are built. Best practices in cloud computing architectures will continue to evolve and as researchers, we should focus not only on enhancing the cloud but also on building tools, technologies and processes that will make it easier for developers and architects to plug in applications to the cloud easily.

Conclusion

We have proposed a framework of a very light-weight and provably secure provable data possession scheme. It surpasses prior work on several counts, including storage, bandwidth and computation overheads as well as the support for dynamic operations. We have used a challenge-response protocol to ensure the intactness of data. This approach causes minimum overhead and also minimizes the bandwidth use. Our framework fully supports dynamic operations on data block which are very efficient. So the intactness of data is verified and along with that it provides protection against server colluding attacks which are more difficult to deal with. To summarize, the work described is an important step forward towards practical provable data possession techniques. We expect that the salient features of our scheme (very low cost and support for dynamic outsourced data)
make it attractive for realistic applications.

References

[1] Cong Wang, Qian Wang, Kui Ren,Wenjing Lou (2009) *Ensuring Data Storage Security in Cloud Computing*.
[2] Cong Wang, Qian Wang, Kui Ren, Wenjing Lou (2010), *Privacy Preserving Public Auditing for Data Storage Security in Cloud Computing*.
[3] Balakrishnan S., Saranya G., Shobana S., Karthikeyan S. (2010)*Introducing Effective Third Party Auditing (TPA) for Data Storage Security in Cloud.*
[4] Wang Q., Wang C., Li J., Ren K. and Lou W. (2009) *ESORICS'09*.
[5] Cloud Security Alliance (2009) *Security Guidance for Critical Areas of Focus in Cloud Computing.*
[6] Ateniese G. et al. (2007) *ACM CCS '07*, 598–609.
[7] Shacham H. and Waters B. (2008) *Asia-Crypt '08*, 5350, 90–107.
[8] Geelan J. (2009) *The Future of Cloud Computing. "Cloud Computing Journal*.
[9] Wang Q. et al. (2009) *ESORICS '09*, 355–70.
[10]Erway C. et al. (2009) *ACM CCS '09*, 213–22.
[11]Wang C. et al. (2010) *IEEE INFOCOM '10*.