ISSN:0975-9646

Security Provision in Publicly Auditable Secure Cloud Data Storage Services Using SHA-1 Algorithm

Akkala.Saibabu, T.Satyanarayana Murthy
Department of Computer Science, Vignan University, Vadlamudi-522213

Abstract: 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 over a network. Cloud computing entrusts services with a user's data, software and computation on a published application programming interface over a network Where data owners can remotely store their data in the cloud to enjoy on-demand high-quality applications and services from a shared pool of configurable computing resources. While data outsourcing relieves the owners of the burden of local data storage and maintenance, it also eliminates their physical control of storage dependability and security. So correctness of data and security is a prime concern. This work 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 sha-1 algorithm which is more secure compared to other algorithms. To ensure the correctness of data, we consider the task of allowing a 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.

Key words
Third party auditor, homomorphic authenticator, cloud server

I. INTRODUCTION:

Cloud computing, to put it simply, means internet computing. The internet is commonly visualized as clouds; hence the term "cloud computing" for computation done through the internet. With cloud computing users can access database resources via the internet from anywhere, for as long as they need, without worrying about any maintenance or management of actual resources. Besides, databases in cloud are very dynamic and scalable.

Cloud Computing is unlike grid computing, utility computing, or autonomic computing. In fact, it is a very independent platform in terms of computing. The best example of cloud computing is Google apps where any application can be accessed using a browser and it can be deployed on thousands of computer through the internet. It also provides facilities for users to develop, deploy and manage their applications on the cloud, which entails virtualization of resources that maintains and manages itself.

A. Service models

These services are broadly divided into three categories:

1. Saas (software as a service) -Saas is a model of software deployment where an application is hosted as a service provided to customers across the internet.

2. Paas (platform as a service)-the cloud provides hardware resources, typically virtual machines, which can be loaded with the users, operating system and software.

3. Iaas (infrastructure as a service) the cloud provides an infrastructure including platforms, networking, *etc.* on which applications can be placed.

II. PROBLEM STATEMENT:

While cloud computing makes these advantages more appealing than ever, it also brings new and challenging security threats to the outsourced data. Since cloud service providers (Csp) are separate administrative entities, data outsourcing actually relinquishes the owner's ultimate control over the fate of their data. As a result, the correctness of the data in the cloud is 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 still face a broad range of both internal and external threats to data integrity. Outages and security breaches of noteworthy cloud services appear from time to time. Amazons s3's recent downtime, Gmail's mass email deletion incident, and apple mobile me's post-launch downtime are all such examples. Second, for benefits of their own, there are various motivations for Csp's to behave unfaithfully toward cloud customers regarding the status of their outsourced data. In short, although outsourcing data into the cloud is economically attractive for the cost and complexity of long-term large scale data storage, it does not offer any guarantee on data integrity and availability. This problem, if not properly addressed, may impede successful deployment of the cloud architecture.

As Data owners no longer physically possess the storage of their data, traditional cryptographic primitives for the purpose of data security protection cannot be directly adopted[6,7]. In particular, simply downloading the data for its integrity verification is not a practical solution due to the high cost of input/output (I/O) and transmission across the network. Besides, it is often insufficient to detect data corruption only when accessing the data, as it does not give correctness assurance for unaccessed data and might be too late to recover the data loss or damage. Considering the large size of the outsourced data and the owner's constrained resource capability, the tasks of auditing the data correctness in a cloud environment can be formidable and expensive for data owners [6-8]. Moreover, from the system usability point of view, data owners should be able to just use cloud storage as if it is local, without worrying about the need to verify its integrity. Hence, to fully ensure data security and save data owners' computation resources, this work enables the publicly auditable cloud storage

services, where data owners can resort to an external third party auditor (TPA) to verify the outsourced data when needed. Third party auditing provides a transparent yet cost-effective method for establishing trust between data owner and cloud server. in fact, based on the audit result from a TPA, the released audit report would not only help owners to evaluate the risk of their subscribed cloud data services, but also be beneficial for the cloud service provider to improve their cloud based service platform . In a word, enabling public risk auditing protocols will play an important role for this nascent cloud economy to become fully established.

III. PROPOSED SYSTEM:

Recently, great interest has been shown in ensuring remotely stored data integrity under different system and security models. Some of the work has already been promoting the development of public auditability for existing cloud data storage services. However, it is not feasible yet. on one hand, data owners are currently not sophisticated enough to demand risk assessment; on the other hand, current commercial cloud vendors do not provide such a third party auditing interface to support a public auditing service. This article is intended as a call for action, aiming to motivate further research on dependable cloud storage services and enable public auditing services to become a reality. We start by suggesting a set of systematically and cryptographically desirable properties that should apply to practical deployment for securing the cloud storage on behalf of data owners. We sketch a set of building blocks, including recently developed cryptographic primitives (e.g., homomorphic authenticator), to ensure these strong security properties, which could form the basis of a publicly auditable secure cloud data storage system.

IV. SYSTEM DESIGN & IMPLEMENTATION

The Cloud storage architecture consists of four different entities: owner, cloud server, user and TPA here the TPA is the trusted entity that has expertise and capabilities to assess cloud storage security on behalf of a data owner upon request. Under the cloud paradigm, the data owner may represent either the individual or the enterprise customer, who relies on the cloud server for remote data storage and maintenance, and thus is relieved of the burden of building and maintaining local storage infrastructure. in most cases cloud data storage services also provide benefits like availability, relative low cost and on demand sharing among a group of trusted users, such as partners in a collaboration team or employees in the enterprise organization. For simplicity, we assume a single writer/many readers scenario here.Only the data owner can dynamically interact with the Csp to update her stored data, while users just have the privilege of file reading.

With in the scope of this article, we focus on how to ensure publicly auditable secure cloud data storage services. As the data owner no longer possesses physical control of the data, it is of critical importance to allow the data owner to verify that his data is being correctly stored and maintained in the cloud. Considering the possibly large cost in terms of resources and expertise, the data owner may resort to a TPA for the data auditing task to ensure the storage security of her data, while hoping to keep the data private from the TPA. 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 owners during the auditing process. The TPA should be able to efficiently audit the cloud data storage without local copy of data and without any additional online burden for data owners. Besides, any possible leakage of an owner's outsourced data toward a TPA through the auditing protocol should be prohibited. The Cs is semi-trusted in the sense that most of the time it behaves properly and does not deviate from the prescribed protocol execution.

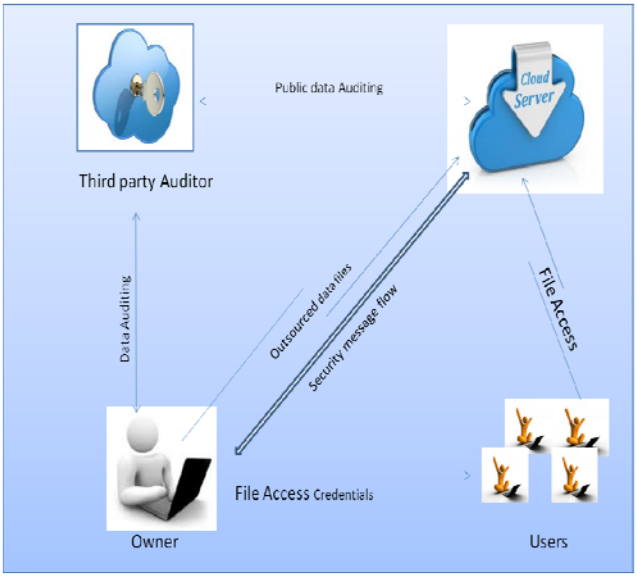

Figure.1the Architecture of cloud data storage services

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 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.

Selection of cloud service 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,

2.1) Any Where/anytime access - it promises "universal" access to high-powered computing and storage resources for anyone with a network access device.

2.2) Collaboration among users -cloud represents an environment in which users can develop software based services and from which they can deliver them.

2.3) Cost Benefits - the cloud promises to deliver computing power and services at a lower cost.

2.4) Storage as a universal service - the cloud represents a remote but scalable storage resource for users anywhere and everywhere.

V. SECURING THE CLOUD DATA AT CLOUD SERVER USING SHA-1 ALGORITHM

The secure hash algorithm (Sha) [2] was developed by the national institute of standards and technology (nist) and published as a federal information processing standard (fips 180) in 1993.

Sha-1 produces a hash value of 160 bits. in 2002, nist produced a revised version of the standard, fips 180-2, that defined three new versions of sha, with hash value lengths of 256, 384, and 512 bits, known as sha-256, sha-384, and sha-512 as shown in table 1 these new versions have the same underlying structure and use the same types of modular arithmetic and logical binary operations as sha-1. Shortly thereafter, a research team described an attack in which two separate messages could be found that deliver the same sha-1 hash using 2^{69} operations, far fewer than the 2^{80}

	sha-1	sha-256	sha-384	sha-512
message digest size	160	256	384	512
message size	$<2^{64}$	$<2^{64}$	$<2^{128}$	$<2^{128}$
block size	512	512	1024	1024
word size	32	32	64	64
number of steps	80	64	80	80
security	80	128	192	256

notes :
1.all sizes are measured in bits
2. security refers to the fact that a birthday attack on a message digest of size *n* produces a collision with a work factor of approximately $2^{n/2}$

Table.1 Comparison of Sha Parameters

1. Sha-1 Logic

Sha-1 is a part of the fips 180-2: secure hash standard. It is very widely used in public-key cryptography, especially in message authentication schemes. Sha-1 calculates a 160-bit *h* for a *b*-bit *m*. the algorithm consists of the following steps:

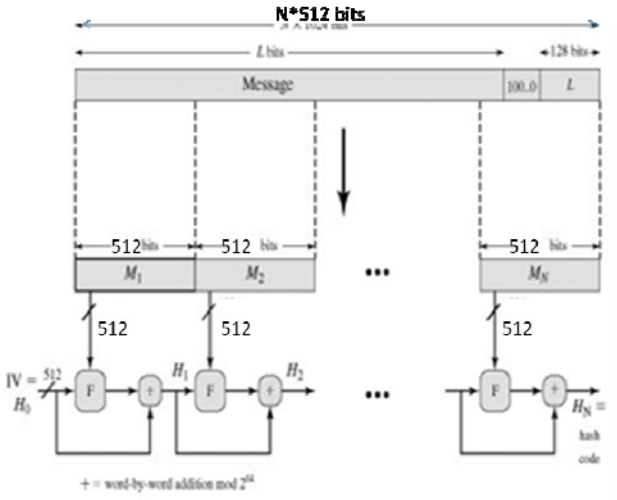

Figure 2 Sha logic

1. **Appending Padding Bits**

The *b*-bit *m* is padded in the following manner: a single 1-bit is added into the end of *m*, after which 0-bits are added until the length of the message is congruent to 448, modulo 512.

2. **Appending Length**

A 64-Bit representation of *b* is appended to the result of the above step. Thus, the resulted message is a multiple of 512 bits.

3. B**uffer Initialization**

A 512-bit buffer is used to hold intermediate and final results of the hash function. The buffer can be represented as eight 64-bit registers (a, b, c, d, e, f, g, h). These registers are initialized to the following 64-bit integers:

a = 6a09e667f3bcc908
b = bb67ae8584caa73b
c = 3c6ef372fe94f82b
c = a54ff53a5f1d36f1
e = 510e527fade682d1
f = 9b05688c2b3e6c1f
g = 1f83d9abfb41bd6b
h = 5be0cdi9137e2179

These values are stored in big-endian format, which is the most significant byte of a word in the low-address (leftmost) byte position. These words were obtained by taking the first sixty-four bits of the fractional parts of the square roots of the first eight prime numbers

4: Process Message In 512-Bit (128-Word) Blocks.

The Heart of the algorithm is a module that consists of 80 rounds; this module is labeled f in figure 2 .contents of the buffer. At input to the first round, the buffer has the value of the intermediate hash value, *hi*-1. each round *t* makes use of a 64-bit value w_t derived from the current 512-bit block being processed (*mi*) these values are derived using a message schedule described subsequently. Each round also makes use of an additive constant *kt* where $0 \leq t \leq 79$ indicates one of the 80 rounds. These words represent the first sixty-four bits of the fractional parts of the cube roots of the first eighty prime numbers. The constants provide a "randomized" set of 64-bit patterns, which should eliminate any regularity in the input data. The output of the eightieth round is added to the input to the first round ($\boldsymbol{h_{i-1}}$) to produce *hi*. The addition is done independently for each of

the eight words in the buffer with each of the corresponding words in $\mathbf{h_{i-1}}$ using addition modulo 2^{32}

5. Output.

After all *n* 512-bit blocks have been processed; the output from the *n*th stage is the 160-bit message digest

***h*0 = IV**

***hi* = sum32(*hi*-1, abcdefgh*i*)**

md* = *hn

Where

iv	initial value of the abcdefgh buffer, defined in step 3
$abcdefgh_i$	the output of the last round of processing of the i^{th} message block
n	the number of blocks in the message (including padding and length fields)
sum32	addition modulo 2^{32} performed separately on each word of the pair of inputs
md	final message digest value

Round Function

Let us look in more detail at the logic in each of the 80 steps of the processing.

t_1=h+ch (e, f, g) + ($\sum_1^{160}$e) +w_t+k_t
t2= ($\sum_1^{160}$ a) + maj (a, b, c)
a=t_1+t_2
b=a
c=b
d=c
e=d+t_1
f=e
g=f
h=g

Where *t* =step number; 0≤ *t* ≤79
Ch (*e, f, g)* = (*e* and *f*) $\oplus$ (not *e* and g)
The conditional function: *if e* then *f* else *g*

Maj (*a*, *b*, *c*) = (*a* and *b*) $\oplus$ (*a* and *c*) $\oplus$ (*b* and *c*) *the function is true only of the majority*

$$\left(\sum_0^{512} a\right) = \mathrm{ROTR}^{28}(a) \oplus \mathrm{ROTR}^{34}(a) \oplus \mathrm{ROTR}^{39}(a)$$

$$\left(\sum_1^{512} e\right) = \mathrm{ROTR}^{14}(e) \oplus \mathrm{ROTR}^{18}(e) \oplus \mathrm{ROTR}^{41}(e)$$

Rotr(*x*) = circular right shift (rotation) of the 64-bit argument *x* by *n* bits

Wt = a 64-bit word derived from the current 512-bit input block

kt = a 64-bit additive constant

+ = addition modulo 2^{32}

VI. USING HOMOMORPHIC AUTHENTICATOR

To Significantly reduce the arbitrarily large communication overhead for public auditability without introducing any online burden on the data owner, we resort to the homomorphic authenticator technique [7, 10].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. Specifically, a data file is divided into n blocks mi (i =1… n), and each block mi has a corresponding homomorphic authenticator σi computed as its metadata to ensure the integrity. Every time it must be verified that the cloud server is honestly storing the data, the data owner or TPA can submit challenges

Chal = {(i, vi)} for sampling a set of randomly selected blocks, where {vi} can be arbitrary weights. due to the nice property of the homomorphic authenticator, server only needs to response a linear combination of the sampled data blocks μ = ∑ivi.mi, as well as an aggregated authenticator σ=πiσivi both computed from {mi,σi,vi}i∈chal. once the response of μ and σ is verified by tpa, then high probabilistic guarantee on large fraction of cloud data correctness can be obtained.1 because off-the-shelf error-correcting code technique can be adopted before data outsourcing [6, 10], large fraction of correct cloud data would be sufficient to recover the whole data. Note that for typical choices of block size |mi| and file block number n, where |mi|>> log (n), the response μ and σ are (essentially) about the same size as individual block mi and σi. This means almost constant communication overhead, independent of file size, for each auditing can be achieved. Moreover, since the TPA could regenerate the fresh random sampling challenges, unbounded auditing is achieved too, which means no additional on-line burden would be incurred towards data owner. However, despite the desirable properties, this approach only works well for encrypted data. When directly applied to unencrypted data, it still leaks bits information towards TPA, as discussed next.

VII. CAPABLE OF HANDLING MULTIPLE CONCURRENT TASKS

With the establishment of privacy-preserving public auditing in cloud computing, a tpa may concurrently handle auditing delegations on different owners' requests. The individual auditing of these tasks in a sequential way can be tedious and very inefficient for a TPA. Given *k* auditing delegations on *k* distinct data files from *k* different owners, it is more advantageous for a TPA to batch these multiple tasks together and perform the auditing one time, saving computation overhead as well as auditing time cost. Keeping this natural demand in mind, we note that two previous works [10, 13] can be directly extended to provide batch auditing functionality by exploring the technique of bilinear aggregate signature [18]. Such a technique supports the aggregation of multiple signatures by distinct signers on distinct messages into a single signature and thus allows efficient verification for the authenticity of all messages. basically, with batch auditing the *k* verification equations

(for k auditing tasks) corresponding to k responses $\{\mu, \sigma\}$ from a cloud server can now be aggregated into a single one such that a considerable amount of auditing time is expected to be saved. A very recent work [15] gives the first study of batch auditing and presents mathematical details as well as security reasonings. Note that the aggregated verification equation in batch auditing only holds when all the responses are valid, and fails with high probability when there is even one single invalid response in the batch auditing. To further sort out these invalid responses, a recursive divide-and-conquer approach (binary search) can be utilized. Specifically, if the batch auditing fails, we can divide the collection of responses into two halves, and recurse the batch auditing in halves. preliminary results in [15] shows that compared to individual auditing, batch auditing indeed helps reduce the Tap's computation cost, as more than 11 and 14 percent of per-task auditing time is saved when the sampling block set is set to be 460 and 300, respectively. Moreover, even if up to 18 percent of 256 different responses are invalid, batch auditing still performs faster than individual verification.

VIII. FURTHER CHALLENGES

In the above sections we have described some suggested requirements for public auditing services and the state of the art that fulfills them. However, this is still not enough for a publicly auditable secure cloud data storage system, and further challenging issues remain to be supported and resolved

1. Performance

Performance is always an important concern for practical system deployment. although there is evidence that the overhead for auditing based on homomorphic authenticators will be manageable [13–15], we have yet to demonstrate that the cost of authenticator precomputation and transfer of a realistic personal device is acceptable.

2. Protect Data Privacy

Data privacy protection has always been an important aspect of a service level agreement for cloud storage services. Thus, the implementation of a public auditing protocol should not violate the owner's data privacy. in other words a tpa should be able to efficiently audit the cloud data storage without demanding a local copy of data or even learning the data content.

3. Support

Data Dynamics as a cloud storage service is not just a data warehouse; owners are subject to dynamically updating their data via various application purposes. The design of auditing protocol should incorporate this important feature of data dynamics in cloud computing.

IX. CONCLUSION:

Cloud computing today is the beginning of "network based computing" over internet in force. It is the technology of the decade and is the enabling element of two totally new computing models, the client-cloud computing and the terminal-cloud computing. The public auditability for cloud data storage security is of critical importance so that users can resort to an external audit party to check the integrity of outsourced data when needed. 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. We utilize sha-1 algorithm for data security in cloud and uniquely combine the public key based homomorphic authenticator with random masking to achieve the privacy-preserving public cloud data auditing system. we believe that the security in cloud computing, an area full of challenges and of paramount importance, is still in its infancy now but will attract enormous amounts of research effort for many years to come.

REFERENCES

[1] Cloud computing for the federal community by hannah wald http://www.scribd.com/doc/86884517/vol13-no2-cloud-computing

[2] journal of theoretical and applied information technology. © 2005 - 2009 jatit. all rights reserved. www.jatit.org. 71. cloud computing: an overview

[3] cryptography and network security principles and practices, 4th ed - william stallings http://www.filecrop.cm/cryptography-and-network-security-william-stallings.html

[4] amazon.com, "amazon s3 availability event: july 20, 2008," july 2008; http://status.aws.amazon.com/s3-20080720.html

[5] m. krigsman, "apple's mobileme experiences post-launch pain," july 2008; http://blogs.zdnet.com/projectfailures/?p=908

[6] a. juels, j. burton, and s. kaliski, "pors: proofs of retrievability for large files," proc. acm ccs '07, oct. 2007, pp. 584–97.

[7] g.ateniese et al., "provable data possession at untrusted stores," proc. Acm ccs '07, oct. 2007, pp. 598–609.

[8] m. a. shah et al., "auditing to keep online storage services honest," proc. usenix hotos '07, may 2007.

[9] g. ateniese et al., "scalable and efficient provable data possession," proc. securecomm '08, sept. 2008.

[10] cloud security alliance, (2009) "security guidance for critical areas of focus in cloud computing," 9, [online] available : http://www.cloudsecurityalliance.org.

[11] p. mell, t. grance, (2009) "draft nist working definition of cloud computing," referenced on june. 3rd, 2009 online at http://csrc.nist.gov/ groups/sns/cloud-computing/index.html.

[12]j. dean and s. ghemawat. mapreduce: simplified data processing on large clusters. in osdi'04: proceedings of the 6th conference on symposium on opearting systems design & implementation, pages 10–10, berkeley, ca, usa, 2004. usenix association.

[13] q. wang et al., "enabling public verifiability and data dynamics for storage security in cloud computing," proc. esorics '09, sept. 2009, pp. 355–70.

[14] c. erway et al., "dynamic provable data possession," proc. acm ccs '09, nov. 2009, pp. 213–22.

[15] c. wang et al., "privacy-preserving public auditing for storage security in Cloudcomputing,"proc.ieeeinfocom'10, mar. 2010.