International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Index Copernicus Value (2016): 79.57 | Impact Factor (2017): 7.296

A Survey on Security and Privacy in Cloud Computing

Priyanka V. Surnar[1], S. G. Swami[2]

[1, 2]SRTM University, Department of CSE, M. S. Bidve Engineering College, Latur, Maharashtra, India,

Abstract: ***Cloud computing provides a flexible and convenient way for data sharing, which brings various benefits for both the society and individuals. But there exists a natural resistance for users to directly outsource the shared data to the cloud server since the data often contain valuable information. Thus, it is necessary to place cryptographically enhanced access control on the shared data. Identity-based encryption is a promising cryptographically primitive to build a practical data sharing system. However, access control is not static. That is, when some user's authorization is expired, there should be a mechanism that can remove him/her from the system. Consequently, the revoked user cannot access both the previously and subsequently shared data. To this end, we propose a notion called revocable-storage identity-based encryption (RS-IBE), which can provide the forward/backward security of cipher text by introducing the functionalities of user revocation and cipher text update simultaneously. Furthermore, we present a concrete construction of RS-IBE, and prove its security in the defined security model. The performance comparisons indicate that the proposed RS-IBE scheme has advantages in terms of functionality and efficiency, and thus is feasible for a practical and cost-effective data-sharing system. Finally, we provide implementation results of the proposed scheme to demonstrate its practicability.***

Keywords: decryption key exposure, Cloud computing, data sharing, revocation, Identity-based encryption, cipher text update

1. Introduction

More and more, we are seeing technology moving to the cloud. It's not just a fad — the shift from traditional software models to the internet has steadily gained momentum over the last 10 years. Looking ahead, the next decade of cloud computing promises new ways to collaborate everywhere, through mobile devices. So what is cloud computing? Essentially, cloud computing is a kind of outsourcing of computer programs. Using cloud computing, users are able to access software and applications from wherever they are; the computer programs are being hosted by an outside party and reside in the cloud. This means that users do not have to worry about things such as storage and power, they can simply enjoy the end result.

Life before cloud computing

Traditional business applications have always been very complicated and expensive. The amount and variety of hardware and software required to run them are daunting. You need a whole team of experts to install, configure, test, run, secure, and update them. When you multiply this effort across dozens or hundreds of apps, it's easy to see why the biggest companies with the best IT departments aren't getting the apps they need. Small and midsize businesses don't stand a chance.

Cloud computing: a better way

With cloud computing, you eliminate those headaches that come with storing your own data, because you're not managing hardware and software — that becomes the responsibility of an experienced vendor like Sales force. The shared infrastructure means it works like a utility: You only pay for what you need, upgrades are automatic, and scaling up or down is easy. Cloud-based apps can be up and running in days or weeks, and they cost less. With a cloud app, you just open a browser, log in, customize the app, and start using it. Businesses are running all kinds of apps in the cloud, like customer relationship management (CRM), HR, accounting, and much more. Some of the world's largest companies moved their applications to the cloud with Sales force after rigorously testing the security and reliability of our infrastructure. As cloud computing grows in popularity, thousands of companies are simply rebranding their non-cloud products and services as "cloud computing." Always dig deeper when evaluating cloud offerings and keep in mind that if you have to buy and manage hardware and software, what you're looking at isn't really cloud computing but a false cloud.

Learn More about Platform as a Service

Infrastructure as a Service (IaaS)
A third party hosts elements of infrastructure, such as hardware, software, servers, and storage, also providing backup, security, and maintenance.

Software as a Service (SaaS)
Using the cloud, software such as an internet browser or application is able to become a usable tool.

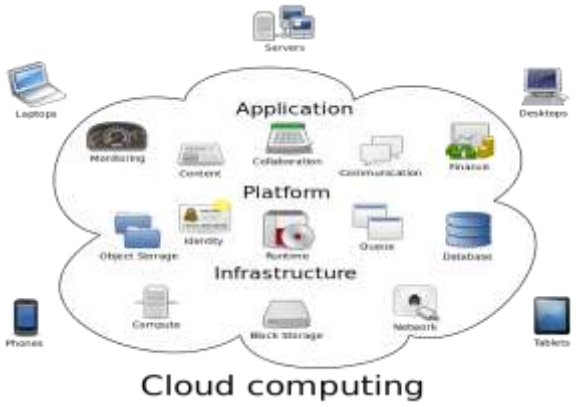
Platform as a Service (PaaS)
The branch of cloud computing that allows users to develop, run, and manage applications without having to get caught up in code, storage, infrastructure and so on.

There are several types of PaaS. Every PaaS option is either public, private, or a hybrid mix of the two. Public PaaS is hosted in the cloud, and its infrastructure is managed by the provider. Private PaaS, on the other hand, is housed in onsite servers or private networks, and is maintained by the user. Hybrid PaaS uses elements from both public and private, and is capable of executing applications from multiple cloud infrastructures.

PaaS can be further categorized depending on whether it is open or closed source, whether it is mobile compatible (mPaaS), and what business types it caters to.

When choosing a PaaS solution, the most important considerations beyond how it is hosted are how well it integrates with existing information systems, which programming languages it supports, what application-building tools it offers, how customizable or configurable it is, and how effectively it is supported by the provider.

As digital technologies grow ever more powerful and available, apps and cloud-based platforms are becoming almost universally widespread. Businesses are taking advantage of new PaaS capabilities to further outsource tasks that would have otherwise relied on local solutions. This is all made possible through advances in cloud computing.



The goal of cloud computing is to allow users to take benefit from all of these technologies, without the need for deep knowledge about or expertise with each one of them. The cloud aims to cut costs, and helps the users focus on their core business instead of being impeded by IT obstacles.[35] The main enabling technology for cloud computing is virtualization. Virtualization software separates a physical computing device into one or more "virtual" devices, each of which can be easily used and managed to perform computing tasks. With operating system–level virtualization essentially creating a scalable system of multiple independent computing devices, idle computing resources can be allocated and used more efficiently. Virtualization provides the agility required to speed up IT operations, and reduces cost by increasing infrastructure utilization. Autonomic computing automates the process through which the user can provision resources on-demand. By minimizing user involvement, automation speeds up the process, reduces labor costs and reduces the possibility of human errors.[35]

To achieve secure computing auditing in cloud, one straightforward method is to double-check each result. The cloud providers may give the inputs and overall computing result to the auditor, which will follow an identical procedure to compute the result and then compare it with the one provided by the cloud providers. However, these schemes may lead to a waste of I/O and computation resources. Note that the data transferring bottlenecks rank in the top of the ten obstacles which may prevent the overall success of the cloud computing [1]. In [13], a Commitment-Based Sampling (CBS) technique is introduced in the conventional grid computing however it does not take the privacy issue into consideration. In this paper, we introduce a novel technique by integrating CBS with the designated verification technique.

The contributions of this paper can be summarized as follows.

Firstly, we model the security problems in cloud computing and define the concepts: *uncheatable cloud computation* and *privacy cheating discouragement* in our cloud computing, which are our design goals.

Secondly, we propose a basic protocol, **SecCloud**, to attain data storage security and computation auditing security as well as privacy cheating discouragement and an advanced protocol to achieve computation and communication efficiency improvement through batch verification.

Thirdly, we analyze and prove that **SecCloud** achieves our design goals and discuss how to minimize the computation cost by choosing the optimal sampling size.

Finally, we develop a cloud computing experimental environment **SecHDFS**and implement **SecCloud** as a test bed. Experiment results demonstrate the suitability of the proposed protocol.

The remainder of this paper is organized as follows. A brief review on the related work is given in Section 2. Section 3 describes the system architecture and security problems and presents design goals. Some necessary preliminary knowledge is given in Section 4. We propose an overview of our **SecCloud** in Section 5 and then present an advanced **SecCloud** with performance optimization in Section 6. Section 7 gives out detailed security analysis and discussion. Section 8 introduces the experiment environment **SecHDFS** and implement our **SecCloud** as a test bed. Finally, Section 9concludes the whole paper.

2. Related Work

Security and privacy issues in cloud computing has received extensive attentions recently. Generally speaking, the research work on cloud computing almost falls into the two cases: *cloud storage security* and *cloud computation security*. Cloud storage security mainly addresses the secure outsourced storage issue. In [2], Ateniese et al. first defined a model for *provable data possession* (PDP), which allowed a client that had data stored at an untrusted server to verify that the server possessed the original data without retrieving it. They utilized RSA-based homomorphic tags for auditing outsourced data, but they did not consider the dynamic data storage. In their later work, Ateniese et al. [3] proposed a partially dynamic version of the PDP scheme using symmetric key cryptography. However, it did not support public auditability. Juels et al. [22] proposed the definition of *proof of retrievability* (PoR), which used spot-checking and error-correcting codes to ensure both possession and retrievability for data file on archive service system. Wang et

al. [34] first achieved both public verifiability and dynamic data storage operations employing an Third Party Auditor and improving the proof of retrievability model by using classic Merkle Hash Tree [26] construction for BLS [8] based block tag authentication. Later, they proposed a scheme achieving privacy preserving public verifiability as well as the dynamic data storage operations in [33] by utilizing the public key based homomorphic authenticator and uniquely integrate it with random mask technique. The further work explored the technique of bilinear aggregate signature for TPA can verify data auditing in a complexity of $O(n)$. Erway et al. [15]proposed the first construction of dynamic provable data possession, which extended the PDP model in [2] to achieve provable updating stored data using rank-based authenticated skip lists.

Compared with secure cloud storage, secure cloud computation still receives less attentions. The related work include remote computation audit and verifiable computation. [17] proposed a ringer scheme in distributed computing where the supervisor sent to the participant some pre-computed results without disclosing the corresponding inputs. [27] presented a remote audit mechanism on an existing distributed computing model and provided efficient methods for verifying whether a remote host performed the assigned task. Similar to our prior work [35,16] introduced and formalized a notion of verifiable computation, which allows a weak client to outsource the computation of a function on various dynamically-chosen inputs to the help workers, which return the results as well as proofs that the computation was carried out correctly on the given input value. The primary constraint is that the verification of the proof should require substantially less computational effort than computing the function again from scratch. Further work about verifiable computation could be found in [9,30] which consider the efficiency of the outsourcing computation. The incentive issues of outsourcing computation have been considered in [4] to prevent from cheating.

3. Problem Formulation

In this section, we present the system architecture, model formulation and design goals.

3.1. System architecture of cloud computing

As shown in Fig. 1, we consider a general cloud computing model constituted of a number of *cloud servers*, S1,S2,…,SN, which are under the control of one or multiple *cloud service providers* (**CSP**). These cloud servers process plenty of computation resource and storage resource. **CSP** allocates these resources by means of customized Service Level Agreements [28]. For example, to perform a batch-processing tasks, by employing the existing programming abstraction techniques such as MapReduce [12] and its open-source counterpart Hadoop [5], **CSP** divides such a large task into multiple small sub-tasks and allows them parallelly executed across up to hundreds of cloud servers.

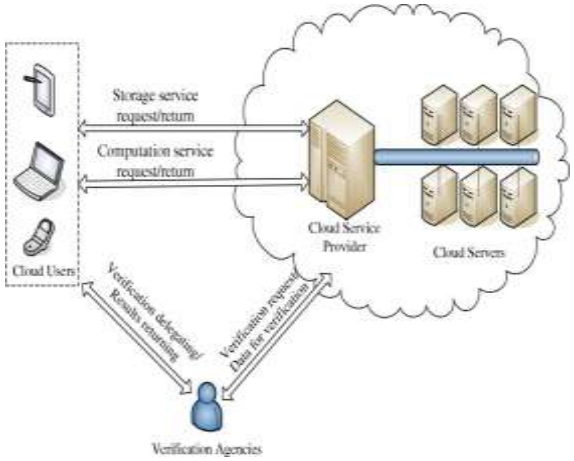


Figure 1: Cloud computing architecture in our protocol

We assume the *cloud user* (**CU**), such as a mobile phone, a laptop, and an apple ipad, which has lower computation resource and smaller storage resource than those of the cloud servers. Most of the communication are through wireless, even if an ordinary computer with limited hours (not 24-h) wired-connecting to the cloud servers. **CU** would submit storage service requests and computation service requests to **CSP** when it demands.

Similar to existing secure storage auditing schemes, we also assume the existence of a number of *verification agencies* (**VAs**), which are chosen and trusted by **CU** and responsible for auditing the cloud services on data storage and computation. **VAs** are expected to have more powerful computation and storage capability to perform the auditing operations than those of **CU**.

3.2. Adversarial models

In our assumption, the adversary A could corrupt a small set of cloud servers and control these servers to launch various cheating attacks as A's wish. Obviously, according to the different goals, these attacks are summarized as follows.

Storage-cheating attack model. When the attacks towards data storage security in the cloud, for example, the adversary would arbitrarily modify the stored data to compromise the data integrity (malicious case) or reveal the confidential data to purchase interest (interest-purchasing case) or in both of cases. In the malicious case, the compromised cloud servers would simply reply to the cloud users' storage queries with a random number. It is a great challenge for cloud users due to lacking the physical possession of the potentially large size of outsourced data. We assume that if the request data set is X, the honest returned data set is X' and the invalid returned data set is $X - X'$.

Computation-cheating attack model. When the attacks towards data computation security of the cloud, we assume that a complicated computation request, denoted F, is comprised of a set of subtasks $\{f_1, f_2, \ldots, f_n\}$[1] and each of subtasks may involve the input data xpi located at position p_i in the cloud server. Thus, the expected computation result is R={f1(xp1),f2(xp2),…,fn(xpn)} for further computing $F(R)$. The adversary would cheat by the

following three ways: The adversary partially computes fi(xpi) for some *i*s ($1 \leqslant i \leqslant n$) and returns random numbers for the rest, but claims to have completed all the computation; the adversary sophisticatedly takes other x^ where x^∉X and leads to much lower computational cost; even claims to use the correct data *x* but the original data *x* is just missing. We denote the set $\{f_i\}$ as F' in which the subtasks f_i are carried out honestly and the set $F - F'$ where the subtasks are not carried out honestly.

Privacy-cheating attack model. When the attacks towards privacy issue of the cloud, which can be viewed as another kind of storage-cheating attack, we assume that the adversary may compromise cloud users' privacy by leaking their confidential data to others, e.g. healthy condition to public or auction price to business competitors, which would lead to serious consequences. To provide data confidentiality, one straightforward approach is to save encrypted the data in the cloud servers. However, such an approach may prevent the regular cloud computation from being further processed.[2] Besides, if the data are stored in a plaintext in the cloud servers, the adversary in the interest-purchasing case may break into and sell/publish the sensitive data to the public. Furthermore, we assume that to sell the sensitive data, the adversary should provide the corresponding proofs to demonstrate the authenticity of the stored data and computing results to convince others.

3.3. Secure cloud computing

a) 3.3.1. Uncheatable cloud computation

To formally define the security model in the cloud computing, we introduce two concepts *Secure Computation Confidence* (**SCC**) and *Secure Storage Confidence*(**SSC**) to indicate the trust level of computation security and storage security, respectively. **SCC** is defined as $|F'|/|F|$ and **SSC** is formalized as $|X'|/|X|$. In both cases, cloud computation or cloud storage is regarded as fully trusted if **SCC** (**SSC**) equals 1. Otherwise, it is semi-trusted.

Definition 1 Uncheatable cloud computation. Let Pr[*CheatingSuccessfully*] be the probability that an adversary with the trust level of **SCC** and **SSC** could successfully cheat without being detected by sampling based verifiers. We denote the computation is uncheatable, if for arbitrary sufficiently small positive number ϵ, there exists a sampling size *t* such that the following conditions always satisfies:
(1)∈Pr[CheatingSuccessfully]=Pr[SCC,SSC,t]<∈.

3.3.2. Privacy-cheating discouragement

To discourage the adversary from leaking the cloud users' sensitive data, we introduce a novel privacy-cheating discouragement model where the adversary wants to illegally sells the cloud users' sensitive data to others. Similar to software sales [21], the software vendor may embed a digital signature in its products to allow the users to authenticate them. Such an authentication could be strictly limited to paying customers rather than the illegitimate users to avoid software piracy. Therefore, it is required that any storage and computation auditing should be authorized by the cloud users. In other words, this could discourage adversaries from leaking users' private data. To achieve this target, we introduce the following definition.

Definition 2 Privacy cheating discouragement

Let *InfoLeak* denote the event that valid information is leaked by the adversary. The cloud computing is privacy cheating discouragement, if for a sufficiently small positive number ϵ, the following equation holds in a polynomial time *t*:
(2)∈Pr[InfoLeak]<∈.

3.4. Design goals

The proposed protocol is expected to achieve the following security and performance goals:

Data storage security: To make sure that the data are securely stored in cloud, the proposed protocol should enable that **CU** and **VA** could fetch and audit the pre-stored data effectively.

Data computation security: To achieving secure computation, the proposed protocol should ensure that the computation be audited by **CU** and **VA**. Considering the fact that some cloud users suffer from computation and transmission constraints, **VA**'s verification is a promising approach for securing cloud computation.

Privacy cheating discouragement: The proposed scheme should ensure that only designated verification parties (e.g., **CSP** or **VAs**) could verify the stored data or computing results, which can discourage the **CSP** from compromising users' privacy, even if the cloud servers are compromised by the attackers.

Efficiency: The computation and transmission overhead of the auditing should be reduced, as is best to meet the minimum.

4. Conclusion

In this paper, we have proposed, **SecCloud**, a privacy-cheating discouragement and secure-computation auditing protocol for data security in the cloud. To the best of our knowledge, it is the first work that jointly considers both of data storage security and computation auditing security in the cloud. We have defined the concepts of uncheatable cloud computation and privacy-cheating discouragement and proposed **SecCloud** to achieve the security goals. To improve the efficiency, different users' requests can be concurrently handled through the batch verification. By the extensive security analysis and performance simulation in our developed **SecHDFS**, it is showed that our protocol is effective and efficient for achieving a secure cloud computing. In our future work, we continue to consider some detailed computation such as linear program computation and data mining and formalize these security models in the cloud computing. In addition, we also focus on the privacy preserving issues in the above computation. Furthermore, we plan to implement them in the real cloud platform such as EC2 and OpenStack..

References

[1] K. Aas and L. Eikvil, "Text Categorization: A Survey," Technical Report Raport NR 941, Norwegian Computing Center, 1999.
[2] Ning Zhong, Yuefeng Li, and Sheng-Tang Wu, "Effective Pattern Discovery for Text Mining" VOL. 24, NO. 1, JANUARY 2012.
[3] R. Baeza-Yates and B. Ribeiro-Neto, Modern Information Retrieval. Addison Wesley, 1999.
[4] Y. Li, C. Zhang and J.R. Swan, "An Information Filtering Model on the Web and Its Application in Jobagent," Knowledge-Based Systems, vol. 13, no. 5, pp. 285-296, 2000.
[5] S. Robertson and I. Soboroff, "The Trec 2002 Filtering Track Report," TREC, 2002, trec.nist.gov/pubs/trec11/papers/OVER. FILTERING.ps.gz.
[6] D.D. Lewis, "Feature Selection and Feature Extraction for Text Categorization," Proc. Workshop Speech and Natural Language, pp. 212-217, 1992.
[7] F. Sebastiani, "Machine Learning in Automated Text Categorization," ACM Computing Surveys, vol. 34, no. 1, pp. 1-47, 2002.
[8] S.-T. Wu, Y. Li, and Y. Xu, "Deploying Approaches for Pattern Refinement in Text Mining," Proc. IEEE Sixth Int'l Conf. Data Mining (ICDM '06), pp. 1157-1161, 2006
[9] S.-T. Wu, Y. Li, Y. Xu, B. Pham and P. Chen, "Automatic Pattern- Taxonomy Extraction for Web Mining," Proc. IEEE/WIC/ACM Int'l Conf. Web Intelligence (WI '04), pp. 242-248, 2004.
[10] R. Agrawal and R. Srikant, "Fast Algorithms for Mining Association Rules in Large Databases," Proc. 20th Int'l Conf. Very Large Data Bases (VLDB '94), pp. 478-499, 1994.