ISSN:2348-2079

Volume-5 Issue-2

International Journal of Intellectual Advancements and Research in Engineering Computations

Security and privacy enhancing in multi cloud

1. Ms.K.E.Eswari M.C.A., M.Phil, Associate Professor
2. Mr. T.Mohansasundaram Final MCA
Department of MCA, Nandha Engineering College,(Autonomous), Erode-52.
eswari.eswaramoorthy@nandhaengg.org, sunther744@gmail.com

Abstract— Using Cloud Storage, users can remotely store their data and enjoy the on-demand high quality applications and services from a shared pool of configurable computing resources, without the burden of local data storage and maintenance Moreover, users should be able to just use the cloud storage as if it is local, without worrying about the need to verify its integrity. Thus, enabling public audit ability for cloud storage is of critical importance so that users can resort to a third party auditor (TPA) to check the integrity of outsourced data and be worry-free. To securely introduce an effective TPA, the auditing process should bring in no new vulnerabilities towards user data privacy, and introduce no additional online burden to user. This project proposes a secure cloud storage system supporting privacy-preserving public auditing. We further extend our result to enable the TPA to perform audits for multiple users simultaneously and efficiently. In addition, it articulates performance optimization mechanisms for this scheme, and in particular present an efficient method for selecting optimal parameter values to minimize the computation costs of clients and storage service providers. It shows that the solution introduces lower computation and communication overheads in comparison with non-cooperative approaches.

Index Terms— Centrality, cloud security, fragmentation, replication, performance.

I.INTRODUCTION

The cloud computing paradigm has reformed the usage and management of the information technology infrastructure [7]. Cloud computing is characterized by on-demand self-services, ubiquitous network accesses, resource pooling, elasticity, and measured services [8]. The aforementioned characteristics

of cloud computing make it a striking candidate for businesses, organizations, and individual users for adoption [25]. However, the benefits of low-cost, negligible management (from a users perspective), and greater flexibility come with increased security concerns [7]. Security is one of the most crucial

aspects among those prohibiting the wide-spread adoption of cloud computing [14, 19]. Cloud security issues may stem due to the core technology's implementation (virtual machine (VM) escape, session riding, etc.), cloud service offerings (structured query language injection, weak authentication schemes, etc.), and arising from cloud characteristics (data recovery vulnerability, Internet protocol vulnerability, etc.) [5]. For a cloud to be secure, all of the participating entities must be secure. In any given system with multiple units, the highest level of the system's security is equal to the security level of the weakest entity [12]. Therefore, in a cloud, the security of the assets does not solely depend on an individual's security measures [5]. The neighboring entities may provide an opportunity to an attacker to bypass the users defenses.

1.1 Hey, You, Get Off Of My Cloud: Exploring Information Leakage In Third-Party Compute Clouds

In this paper [1] the authors Thomas Ristenpart, Eran Tromer, Hovav Shacham and Stefan savage stated that third-party cloud computing represents the promise of outsourcing as applied to computation. Services, such as Microsoft's Azure and Amazon's EC2, allow users to instantiate virtual machines (VMs) on demand and thus purchase precisely the capacity they require when they require it. In turn, the use of virtualization allows third-party cloud providers to maximize the utilization of their sunk capital costs by multiplexing many customer VMs across a

shared physical infrastructure. However, in this paper, the authors showed that this approach can also introduce new vulnerabilities.

It has become increasingly popular to talk of "cloud computing" as the next infrastructure for hosting data and deploying software and services. In addition to the plethora of technical approaches associated with the term, cloud computing is also used to refer to a new business model in which core computing and software capabilities are outsourced on demand to shared third-party infrastructure.

While this model, exemplified by Amazon's Elastic Compute Cloud (EC2) [9], Microsoft's Azure Service Platform [10], and Rackspace's Mosso [11] provides a number of advantages—including economies of scale, dynamic provisioning, and low capital expenditures—it also introduces a range of new risks. Some of these risks are self-evident and relate to the new trust relationship between customer and cloud provider.

For example, customers must trust their cloud providers to respect the privacy of their data and the integrity of their computations. However, cloud infrastructures can also introduce non-obvious threats from other customers due to the subtleties of how physical resources can be transparently shared between virtual machines (VMs).

In particular, to maximize efficiency multiple VMs may be simultaneously assigned to execute on the same physical server. Moreover, many cloud providers allow "multi-tenancy" — multiplexing the virtual machines of disjoint customers upon the same physical hardware. Thus it is conceivable that a customer's VM could be assigned to the same physical server as their adversary. This in turn, engenders a new threat — that the adversary might penetrate the isolation between VMs (e.g., via a vulnerability that allows an "escape" to the hypervisor or via side-channels between VMs) and violate customer confidentiality

This paper explores the practicality of mounting such cross-VM attacks in existing third-party compute clouds. The attacks they considered require two main steps: placement and extraction. Placement refers to the adversary arranging to place their malicious VM on the same physical machine as that of a target customer.

Using Amazon's EC2 as a case study, they demonstrated that careful empirical "mapping"can reveal how to launch VMs in a way that maximizes the likelihood of an advantageous placement. They found that in some natural attack scenarios, just a few dollars invested in launching VMs can produce a 40% chance of placing a malicious VM on the same physical server as a target customer.

Using the same platform they also demonstrated the existence of simple, low-overhead, "co-residence" checks to determine when such an advantageous placement has taken place. While they focused on EC2, they believed that variants of our techniques are likely to generalize to other services, such as Microsoft's Azure or Rackspace's Mosso [11], as they only utilized standard customer capabilities and do not require that cloud providers disclose details of their infrastructure or assignment policies.

Having managed to place a VM co-resident with the target, the next step is to extract confidential information via a cross-VM attack. While there are a number of avenues for such an attack, in this paper we focus on side-channels: cross-VM information leakage due to the sharing of physical resources (e.g., the CPU's data caches). In the multi-process environment, such attacks have been shown to enable extraction of RSA and AES secret keys. However, they are unaware of published extensions of these attacks to the virtual machine environment; indeed, there are significant practical challenges in doing so.

They showed preliminary results on cross-VM side channel attacks, including a range of building blocks (e.g., cache load measurements in EC2) and coarse-grained attacks such as measuring activity burst timing (e.g., for cross-VM keystroke monitoring). This point to the practicality of side-channel attacks in cloud-computing environments. Overall, their results indicated that there exist tangible dangers when deploying sensitive tasks to third-party compute clouds.

1.2 Cross-Vm Side Channels and their use to Extract Private Keys

In this paper [2] the authors YINQIAN ZHANG and ARI JUELS details the construction of an access-driven side-channel attack by which a malicious virtual machine (VM) extracts fine-grained information from a victim VM running on

the same physical computer. This attack is the first such attack demonstrated on a symmetric multiprocessing system virtualized using a modern VMM (Xen). Such systems are very common today, ranging from desktops that use virtualization to sandbox application or OS compromises, to clouds that co-locate the workloads of mutually distrustful customers. Constructing such a side-channel requires overcoming challenges including core migration, numerous sources of channel noise, and the difficulty of preempting the victim with sufficient frequency to extract fine-grained information from it.

This paper addresses these challenges and demonstrates the attack in a lab setting by extracting an ElGamal decryption key from a victim using the most recent version of the libgcrypt cryptographic library. Modern virtualization technologies such as Xen, HyperV, and VMWare are rapidly becoming the cornerstone for the security of critical computing systems. This reliance stems from their seemingly strong isolation guarantees, meaning their ability to prevent guest virtual machines (VMs) running on the same system from interfering with each other's execution or, worse, exfiltrating confidential data across VM boundaries.

The assumption of strong isolation underlies the security of public cloud computing systems [12] such as Amazon EC2, Microsoft Windows Azure, and Rackspace; military multi-level security environments [13]; home user and enterprise desktop security in the face of compromise [14]; and software-based trusted computing [15]

VM managers (VMMs) for modern virtualization systems attempt to realize this assumption by enforcing logical isolation between VMs using traditional access-control mechanisms. But such logical isolation may not be sufficient if attackers can circumvent them via side-channel attacks.

Concern regarding the existence of such attacks in the VM setting stems from two facts. First, in non-virtualized, cross-process isolation contexts, researchers have demonstrated a wide variety of side-channel attacks that can extract sensitive data such as cryptographic keys on single-core architectures [16]. The most effective attacks tend to be so-called "access-driven" attacks that exploit shared micro architectural components such as caches.

Second, Ristenpart et alexhibited coarser, cross-VM, access-driven side-channel attacks on modern symmetric multi-processing (SMP, also called multi-core) architectures. But their attack could only provide crude information (such as aggregate cache usage of a guest VM) and, in particular, is insufficient for extracting cryptographic secrets.

Despite the clear potential for attacks, no actual demonstrations of fine-grained cross-VMside-channels attacks have appeared. The oft-discussed challenges [17] to doing so stem primarily from the facts that VMMs place more layers of isolation between attacker and victim than in cross-process settings, and that modern SMP architectures do not appear to admit fine-grained side-channel attacks (even in non-virtualized settings) because the attacker and victim are often assigned to disparate cores.

Of course a lack of demonstrated attack is not a proof of security, and so whether fine-grained cross-VM side-channel attacks are possible has remained an important open question. In this paper, the authors presented the development and application of a cross-VM side-channel attack in exactly such an environment. Like many attacks before, theirs an access- driven attack in which the attacker VM alternates execution with the victim VM and leverages processor caches to observe behavior of the victim.

However, they believed many of the techniques we employ to accomplish this effectively and with high fidelity in a virtualized SMP environment are novel. In particular, they provided an account of how to overcome three classes of significant challenges in this environment:

(i) inducing regular and frequent attacker-VM execution despite the coarse scheduling quanta used by VMM schedulers;

(ii) overcoming sources of noise in the information available via the cache timing channel, both due to hardware features (e.g., CPU power saving) and due to software ones (e.g., VMM execution); and

(iii) dealing with core migrations, which give rise to cache "readings" with no information of interest to the attacker (i.e., the victim was migrated to a core not shared by the attacker).

Finally, they customized their attack to the task of extracting a private decryption key from the victim and specifically show how to "stitch together" these intermittent, partial observations of the victim VM activity to assemble an entire private key. As they demonstrated in a lab testbed, their attack establishes a side-channel of sufficient fidelity that an attacker VM can extract a private ElGamal decryption key from a co-resident victim VM running Gnu Privacy Guard (GnuPG) a popular software package that implements the OpenPGP e-mail encryption standard.The underlying vulnerable code actually lies in the most recent version of the libgcrypt library, which is used by other applications and deployed widely.

Specifically, they showed that the attacker VM's monitoring of a victim's repeated exponentiations over the course of a few hours provides it enough information to reconstruct the victim's 457-bit private exponent accompanying a 4096-bit modulus with very high accuracy—so high that the attacker was then left to search fewer than 10, 000 possible exponents to find the right one.

They stressed, moreover, that much about their attack generalizes beyond ElGamal decryption (or, more generally, discovering private exponents used in modular exponentiations) in libgcrypt. In particular, their techniques for preempting the victim frequently for observation and sidestepping several sources of cache noise are independent of the use to which the side-channel is put. Even those components that they necessarily tune toward ElGamal private-key extraction, and the pipeline of components overall, should provide a roadmap for constructing side-channels for other ends. They thus believed that their work serves as a cautionary note for those who rely on virtualization for guarding highly sensitive secrets of many types, as well as motivation for the research community to endeavor to improve the isolation properties that modern VMMs provide to a range of applications.

1.3 Towards Ensuring Client-Side Computational Integrity (A Position Paper)

In this paper [5], the authors GEORGE DANEZIS and BENJAMIN LIVSHITS stated that privacy is considered one of the key challenges when moving services to the Cloud. Solution like access control is brittle, while fully homomorphic encryption that is hailed as the silver bullet for this problem is far from practical. But would fully homomorphic encryption really be such an e ective solution to the privacy problem? And can we already deploy architectures with similar security properties? They proposed one such architecture that provides privacy, integrity and leverages the Cloud for availability while only using cryptographic building blocks available today.

Cloud computing services promise scalable outsourcing of computations, networking and storage. Yet, offering such services as a commodity has met resistance due to privacy concerns. Users and consumer groups are increasingly sceptical of infrastructures that centralise personal data for processing. Yet, the drive to store personal data in clouds is not likely to end. Social trends such as consolidation of medical records with sevices such as Microsoft HealthVault or Google Health, or consolidation of personal finance data with Mint.com, as well as others see users record, store and process an increasing amount of data about their personal lives.

A second solution involves trusted hardware at the servers or the clients to ensure the correctness of processing as well as the confidentiality of the data. Often Trusted Computing Modules (TPM) present on most modern motherboards and even mobile hardware are relied upon, but these are not safe against adversaries with physical access to the module. Robust secure co-processors, such as the IBM4758 are expensive and slow compared with a modern computer and using those to perform all computations would deny most benefits of cloud computing.

Finally, and from an academic viewpoint most interestingly, cryptography is presented as a solution to privacy concerns. Convincing solutions have been presented for secure storage of encrypted data, as well as a restricted set of operations on encrypted data (such as searching an index). The

practicality of those is debatable on the basis of cost.

At the same time, fully homomorphic encryption seems to be hailed as the holy grail, that will "solve" the privacy problem. An increasing number of publications at major cryptography conferences are looking at constructions for such schemes or applications of such schemes to secure outsourcing of computations. In general these schemes promise the ability to perform computations on encrypted data.

They decompose any computation to an equivalent logic circuit, and implement the basic gates in terms of the "plus" and "multiply" operations. The circuit results in a ciphertext encoding the result of the computation that is sent back to the user for decryption.

There are two key problems with this approach: first, no practical fully homomorphic encryption schemes exists yet second, as we will argue, even if fully homomorphic encryption was available at the cost of other cryptographic operations today, it would still be inefficient for most computations and could be replaced with a simpler architecture that is already realisable at a low cost today.

They devoted the remaining of this paper in describing a cryptographic architecture that could be made available today to solve aspects of the problem of privacy in the cloud at a relatively similar cost as if homomorphic encryption was used. While the network overheads of the proposed approach will be higher, its advantage is that it can be deployed today.

1.4 Towards User Centric Data Governance and Control in the Cloud

In this paper [6], the authors STEPHAN GROß and ALEXANDER SCHILL stated that cloud computing, i. e. providing on-demand access to virtualised computing resources over the Internet, is one of the current mega-trends in IT. Today, there are already several providers offering cloud computing infrastructure (IaaS), platform (PaaS) and software (SaaS) services. Although the cloud computing paradigm promises both economical as well as technological advantages, many potential users still have reservations about using cloud services as this would mean to trust a cloud provider to correctly handle their data according to previously negotiated rules.

Furthermore, the virtualisation causes a location independence of offered services which could interfere with domain specific legislative regulations. In this paper, we present an approach of putting the cloud user back into power when migrating data and services into and within the cloud. They outlined their work in progress, that aims at providing a platform for developing flexible service architectures for cloud computing with special consideration of security and non-functional properties.

The recent progress in virtualising storage and computing resources combined with service oriented architectures (SOA) and broadband Internet access has led to a renaissance of already known concepts developed in research fields like grid, utility and autonomic computing. Today, the term cloud computing describes different ways of providing on-demand and pay-per-use access to elastic virtualised computing resource pools

These resources are abstracted to services so that cloud computing resources can be retrieved as infrastructure (IaaS), platform (PaaS) and software (SaaS) services respectively. The pay-per-use model of such service oriented architectures includes Service Level Agreements (SLA) negotiated between service provider and user to establish guarantees for required non-functional properties including mandatory security requirements. The (economical) advantages of this approach are fairly obvious: One saves costly investments for procuring and maintaining probably underused hardware and at the same time gains new flexibility to react on temporal higher demands.

Nevertheless, there are reasonable reservations about the deployment of cloud computing services, e.g. concerning data security and compliance. Most of these concerns result from the fact, that cloud computing describes complex socio-technical systems with a high number of different kinds of stakeholders following different and possibly contradicting objectives. From a user's perspective, one has to hand over the control over his data and services when entering the cloud, i.e. the user has to trust that the cloud provider behaves in compliance with the established SLA.

However, to actu- ally agree on a specific SLA a user first has to assess his organizational risks related to security and resilience . Current solutions that restrict the provision of sensible services to dedicated private, hybrid or socalled national clouds do not go far enough as they reduce the user's flexibility when scaling in or out and still force him to trust the cloud provider. Furthermore, private clouds intensify the vendor lock-in problem. Last but not least, there is no support for deciding which services and data could be safely migrated to which cloud.

Instead they demanded new methods and technical support to put the user in a position to benefit from the advantages of cloud computing without giving up the sovereignty over his data and applications. In their current work, they followed a system oriented approach focussing on technical means to achieve this goal.

They identified security as a major obstacle that prevents someone to transfer his resources into the cloud. In order to make sound business decisions and to maintain or obtain security certifications, cloud customers need assurance that providers are following sound security practices and behave according to agreed SLAs [3]. Thus, their overall goal is the development of a flexible open source cloud platform that integrates all necessary components for the development of user-controlled and -monitored secure cloud environments. This platform should contain the following components:

1. Mechanisms to enable an user controlled migration of resources and data into the cloud. These mechanisms should support (semi) automatic configuration of cryptographic algorithms to simplify the enforcement of a user's security requirements as well as the dynamic selection of cloud providers that best fit the user's requirements and trust assumptions. Thus, they need a formalized way to acquire a user's requirements. Furthermore, they need to integrate the user's private resources and different cloud providers in our cloud platform, e.g. by using wrapper mechanisms or standardized interfaces.

2. A sound and trustworthy monitoring system for cloud services that is able to gather all relevant information to detect or even predict SLA violations without manipulations by the cloud provider under control. To support the configuration of the monitoring system, there should be some mechanism that derives relevant monitoring objectives from negotiated SLAs. Thus, we need a formalized language for machine-readable SLA focusing on the technical details of a cloud computing environment.

3. The proposed cloud platform should be adaptive, i.e. it should provide mechanisms to react on SLA violations detected by the monitoring system in order to mitigate the resulting negative effects. These mechanisms should include migration tools to transparently transfer resources to another cloud provider as well as adaptation tools that leaves the resources at the chosen provider but transforms them to further meet the user's non-functional and security requirements.

1.5 Sepia: Privacy-Preserving Aggregation of Multi-Domain Network Events and Statistics

In this paper [7] the authors MARTIN BURKHART, MARIO STRASSER, DILIP MANY and XENOFONTAS DIMITROPOULOS stated that secure multiparty computation (MPC) allows joint privacy-preserving computations on data of multiple parties. Although MPC has been studied substantially, building solutions that are practical in terms of computation and communication cost is still a major challenge.

In this paper, they investigated the practical usefulness of MPC for multi-domain network security and monitoring. They first optimized MPC comparison operations for processing high volume data in near real-time. They then designed privacy-preserving protocols for event correlation and aggregation of network traffic statistics, such as addition of volume metrics, computation of feature entropy, and distinct item count.

Optimizing performance of parallel invocations, they implemented their protocols along with a complete set of basic operations in a library called SEPIA. We evaluate the running time and bandwidth requirements of their protocols in realistic settings on a local cluster as well as on PlanetLab and show that they work in near real-time for up to 140 input providers and 9 computation nodes. Compared to implementations using existing general-purpose MPC frameworks, their protocols are significantly faster, requiring,

for example, 3 minutes for a task that takes 2 days with general-purpose frameworks. This improvement paves the way for new applications of MPC in the area of networking. Finally, they ran SEPIA's protocols on real traffic traces of 17 net-works and show how they provide new possibilities for distributed troubleshooting and early anomaly detection.

A number of network security and monitoring problems can substantially benefit if a group of involved organizations aggregates private data to jointly perform a computation. For example, IDS alert correlation, e.g., with DOMINO requires the joint analysis of private alerts. Similary, aggregation of private data is useful for alert signature extraction, collaborative anomaly detection multi-domain traffic engineering], detecting traffic discrimination and collecting network performance statistics.

All these approaches use either a trusted third party, e.g., a university research group, or peer-to-peer techniques for data aggregation and face a delicate privacy versus utility tradeoff Some private data typically have to be revealed, which impedes privacy and prohibits the acquisition of many data providers, while data anonymization, used to remove sensitive information, complicates or even prohibits developing good solutions. Moreover, the ability of anonymization techniques to effectively protect privacy is questioned by recent studies .

Adopting MPC techniques to network monitoring and security problems introduces the important challenge of dealing with voluminous input data that require online processing. For example, anomaly detection techniques typically require the online generation of traffic volume and distributions over port numbers or IP address ranges.

Such input data impose stricter requirements on the performance of MPC protocols than, for example, the input bids of a distributed MPC auction .In particular, network monitoring protocols should process potentially thousands of input values while meeting near real-time guarantees. This is not presently possible with existing general-purpose MPC frameworks. In this work, they designed, implemented, and evaluated SEPIA (Security through Private Information Aggregation), a library for efficiently aggregating multi-domain network data using MPC.

The foundation of SEPIA is a set of optimized MPC operations, implemented with performance of parallel execution in mind. By not enforcing protocols to run in a constant number of rounds, they were able to design MPC comparison operations that require up to 80 times less distributed multiplications and, amortized over many parallel invocations, run much faster than constant-round alternatives.

In addition, they introduced SEPIA's entropy and distinct count protocols that compute the entropy of traffic feature distributions and find the count of distinct feature values, respectively. These metrics are used frequently in traffic analysis applications. In particular, the entropy of feature distributions is used commonly in anomaly detection, whereas distinct count metrics are important for identifying scanning attacks, in firewalls, and for anomaly detection. They implemented these protocols along with a vector addition protocol to support additive operations on time-series and histogram

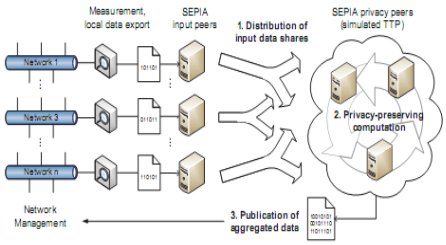

FIG 2.1 DEPLOYMENT SCENARIO FOR SEPIA

A typical setup for SEPIA is depicted in Fig. 2.1 where individual networks are represented by one input peer each. The input peers distribute shares of secret input data among a (usually smaller) set of privacy peers using Shamir's secret sharing scheme.

The privacy peers perform the actual computation and can be hosted by a subset of the networks running input peers but also by external parties. Finally, the aggregate computation result is sent back to the networks. They adopted the semi-honest adversary model, hence privacy of local input data is guaranteed as long as the majority of privacy peers is honest. Their evaluation of SEPIA's performance shows that SEPIA runs in

near realtime even with 140 input and 9 privacy peers. Moreover, we run SEPIA on traffic data of 17 networks collected during the global Skype outage in August 2007 and show how the networks can use SEPIA to troubleshoot and timely detect such anomalies.

Finally, they discussed novel applications in network security and monitoring that SEPIA enables. In summary, this paper made the following contributions:

1. They introduced efficient MPC comparison operations, which outperform constant-round alternatives for many parallel invocations.

2. They designed novel MPC protocols for event correlation, entropy and distinct count computation.

3. They introduced the SEPIA library, in which they implemented their protocols along with a complete set of basic operations, optimized for parallel execution. SEPIA is made publicly available.

4. They extensively evaluated the performance of SEPIA on realistic settings using synthetic and real traces and show that it meets near real-time guarantees even with 140 input and 9 privacy peers.

5. They ran SEPIA on traffic from 17 networks and show how it can be used to troubleshoot and timely detect anomalies, exemplified by the Skype outage.

They concluded that the aggregation of network security and monitoring data is crucial for a wide variety of tasks, including collaborative network defense and cross-sectional Internet monitoring. Unfortunately, concerns regarding privacy prevent such collaboration from materializing. They investigated the practical usefulness of solutions based on secure multiparty computation (MPC).

For this purpose, they designed optimized MPC operations that run efficiently on voluminous input data. They implemented these operations in the SEPIA library along with a set of novel protocols for event correlation and for computing multi-domain network statistics, i.e., entropy and distinct count. Their evaluation results clearly demonstrate the efficiency and scalability of SEPIA in realistic settings. With COTS hardware, near real-time operation is practical even with 140 input providers and 9 computation nodes.

Furthermore, the basic operations of the SEPIA library are significantly faster than those of existing MPC frameworks and can be used as building blocks for arbitrary protocols. They believed that their work provides useful insights into the practical utility of MPC and paves the way for new collaboration initiatives.

Their future work includes improving SEPIA's robustness against host failures, dealing with malicious adversaries, and further improving performance, using, for example, polynomial set representations. Furthermore, in collaboration with a major systems management vendor, they had started a project that aims at incorporating MPC primitives into a mainstream traffic profiling product.

1.6 Twin Clouds: Secure Cloud Computing with Low Latency (Full Version)

In this paper [8] the authors SVEN BUGIEL, STEFAN NURNBERGER, AHMAD-REZA SADEGHI and THOMAS SCHNEIDER stated that Abstract. Cloud computing promises a cost effective enabling technology to outsource storage and massively parallel computations. However, existing approaches for provably secure outsourcing of data and arbitrary computations are either based on tamper-proof hardware or fully homomorphic encryption. The former approaches are not scalable, while the latter ones are currently not efficient enough to be used in practice.

They proposed an architecture and protocols that accumulate slow secure computations over time and provide the possibility to query them in parallel on demand by leveraging the benefits of cloud computing. In their approach, the user communicates with a resource-constrained Trusted Cloud (either a private cloud or built from multiple secure hardware modules) which encrypts algorithms and data to be stored and later on queried in the powerful but untrusted Commodity Cloud. They split their protocols such that the Trusted Cloud performs security-critical pre-computations in the setup phase, while the Commodity Cloud computes the time-critical query in parallel under encryption in the query phase.

Currently, there is no guarantee that security objectives stated in Service Level Agreements (SLA) are indeed fulfilled. Consequently, when using the cloud, the client is forced to blindly trust the provider's mechanisms and configuration [46].

However, this is accompanied by the risk of data leakage and industrial espionage due to a malicious insider at the provider or due to other customers with whom they share physical resources in the cloud [47]. Example applications that need to protect sensitive data include, but are not limited to, processing of personal health records or payroll databases. Access usually occurs not very frequently, but needs to be processed very fast while privacy of the data should be preserved.

Furthermore, in a multi-client scenario, cryptography alone is not sufficient and additional assumptions have to be made such as using tamper-proof hardware [50]. Still, secure hardware which provides a shielded execution environment does not scale well as it is expensive and relatively slow.

Their approach. They proposed a model for secure computation of arbitrary functions with low latency using two clouds (twins). The resource-constrained Trusted Cloud is used for pre-computations whereas the untrusted, but powerful Commodity Cloud is used to achieve low latency (cf. Fig. 2).

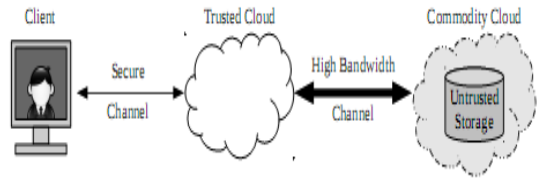

Fig 2.2 Twin Clouds Model with Client, Trusted Cloud, and Commodity Cloud

Their approach allows separating the computations into their security and performance aspects: security-critical operations are performed by the Trusted Cloud in the Setup Phase, whereas performance-critical operations are performed on encrypted data in parallel by the Commodity Cloud in the Query Phase. Analogous to electricity, this can be seen as a battery that can be charged over night with limited amperage and later provides energy rapidly during discharge.

In the Setup Phase, the Trusted Cloud encrypts the outsourced data and programs using Garbled Circuits (GC) which requires only symmetric cryptographic operations and a constant amount of memory. In the time-critical Query Phase, the Trusted Cloud verifies the results computed by the Commodity Cloud under encryption.

Their proposed solution is transparent as the Client uses the Trusted Cloud as a proxy that provides a clearly defined interface to manage the outsourced data, programs, and queries. They minimized the communication over the secure channel (e.g., SSL/TLS) between Client and Trusted Cloud

Their proposed solution has several advantages over previous proposals (cf. x2):

1. Communication Efficiency. They minimized the communication between the client and the Trusted Cloud as only a program, i.e., a very compact description of the function, is transferred and compiled on-the-fly into a circuit.

2. Transparency. The client communicates with the Trusted Cloud over a secure channel and clear interfaces that abstract from the underlying cryptography.

3. Scalability and Low Latency. Their approach is highly scalable as both clouds can be composed from multiple nodes. In the Query Phase, the Trusted Cloud performs only few computations (independent of the function's size).

4. Multiple Clients. Their protocols can be extended to multiple clients such that the Commodity Cloud securely and non-interactively computes on the clients' input data.

EXISTING SYSTEM

The existing system utilizes the technique of public key based homomorphic linear authenticator (or HLA for short), which enables Third Party Auditor 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 HLA with random masking, the protocol guarantees that the 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.

Various prime numbers are assigned as tags for each segment of file which is stored in server. Each segment is having two prime numbers each of which belongs to a different prime order. The third party auditor knows the prime numbers in a random manner. During verification, the third party auditor sends the numbers as random challenge and if the numbers are matched with tags then the file integrity is said to be verified.

Drawbacks

- All the nodes are treated equally and weak capable nodes also require huge computations.
- All the mirror nodes store the file with same encryption mechanism.
- Unauthorized data leakage still remains a problem due to the potential exposure of decryption keys.
- Only single cloud provider environment is considered.

PROPOSED SYSTEM

The proposed system includes all the existing system approach which covers multiple cloud service provider environments. In addition, size blocks of data are being processed with varying size nature in different cloud locations having same copy of data.

The data blocks is stored and retrieved in different cloud locations based on the storage and computational capability. Thus the proposed system explores such issue to provide the support of variable-length block verification.

Likewise, the privacy level for all cloud providers is analyzed by trusted authority and security degree and performance is quantified for encryption algorithms.

Advantages

The proposed system has following advantages:

- Partial data of files are taken from multiple mirror locations and send to selected client.
- Suitable for very large size files.
- Irrelevant size blocks of data are handled among the multiple cloud service providers based on their computational capabilities.

Different trust level is set to different cloud providers and encryption/decryption is varied based on the clouds computational capability.

C*ONCLUSION*

Through this project, the problem of secure communication is eliminated. In addition, the application required less working experience in systems to run the software. The application is tested well so that the end users use this software for their whole operations. It is believed that almost all the system objectives that have been planned at the commencement of the software development have been net with and the implementation process of the project is completed. A trial run of the system has been made and is giving good results the procedures for processing is simple and regular order. The process of preparing plans been missed out which might be considered for further modification of the application. The project effectively stores and retrieves the records from the cloud space database server. The records are encrypted and decrypted whenever necessary so that they are secure.

REFERENCES

[1] T. Ristenpart, E. Tromer, H. Shacham, and S. Savage, “Hey, You, Get Off of My Cloud: Exploring Information Leakage in Third-Party Compute Clouds,” Proc. 16th ACM Conf. Computer and Comm. Security (CCS ’09), pp. 199-212, 2009.

[2] Y. Zhang, A. Juels, M.K.M. Reiter, and T. Ristenpart, “Cross-VM Side Channels and Their Use to Extract Private Keys,” Proc. ACM Conf. Computer and Comm. Security (CCS ’12), pp. 305-316, 2012.

[3] J. Somorovsky, M. Heiderich, M. Jensen, J. Schwenk, N. Gruschka, and L. Lo Iacono, “All Your Clouds Are Belong to Us: Security Analysis of Cloud Management Interfaces,” Proc. Third ACM Workshop Cloud Computing Security Workshop (CCSW ’11), pp. 3-14, 2011.

[4] S. Bugiel, S. Nu ¨ rnberger, T. Po ¨ppelmann, A.-R. Sadeghi, and T.Schneider, “AmazonIA: When Elasticity Snaps Back,” Proc. 18th ACM Conf. Computer and Comm. Security (CCS ’11), pp. 389-400, 2011.

[5] G. Danezis and B. Livshits, “Towards Ensuring Client-Side Computational Integrity (Position Paper),” Proc. ACM Cloud Computing Security Workshop (CCSW ’11), pp. 125-130, 2011.

[6] S. Groß and A. Schill, “Towards User Centric Data Governance and Control in the Cloud,” Proc. IFIP WG 11.4 Int’l Conf. Open Problems in Network Security (iNetSeC), pp. 132-144, 2011.

[7] M. Burkhart,M. Strasser, D. Many, and X. Dimitropoulos, “SEPIA: Privacy-Preserving Aggregation of Multi-

Domain Network Events and Statistics," Proc. USENIX Security Symp., pp. 223-240, 2010.

[8] D. Hubbard and M. Sutton, "Top Threats to Cloud Computing V1.0," Cloud Security Alliance, http://www. cloudsecurityalliance.org/topthreats, 2010.

[9] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. H. Katz, A. Konwinski, G. Lee, D. A. Patterson, A. Rabkin, I. Stoica, and M. Zaharia. A view of cloud computing. Commun. ACM, 53(4):50–58, 2010.

[10]R. Meushaw and D. Simard. A network on a desktop. NSA Tech Trend Notes, 9(4), 2000. http://www.vmware.com/pdf/TechTrendNotes.pdf.

[11] P. England and J. Manferdelli. Virtual machines for enterprise desktop security. Information Security Technical Report, 11(4):193 – 202, 2006.

[12] T. Garfinkel, B. Pfaff, J. Chow, M. Rosenblum, and D. Boneh. Terra: a virtual machine-based platform for trusted computing. In ACM Symposium on Operating Systems Principles, pages 193–206. ACM, 2003.

[13] O. Acii¸cmez. Yet another microarchitectural attack: Exploiting I-cache. In ACM Workshop on Computer Security Architecture, pages 11–18, October 2007.

[14] T. Ristenpart, E. Tromer, H. Shacham, and S. Savage. Hey, you, get off of my cloud: Exploring information leakage in third-party compute clouds. In 16th ACM Conference on Computer and Communications Security, page 199–212,2009.