PRIVACY – PRESERVING PUBLIC AUDITING FOR MULTIPLE CLOUD SERVICE PROVIDERS

***Mr. Arun K., **Dr. T. Senthil Prakash, *Ms. T. Malathi**

**II year ME (CSE), Shree Venkateshwara Hi-Tech Engg College, Gobi*
***Professor & HOD, Shree Venkateshwara Hi-Tech Engg College, Gobi*

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. However, the fact that users no longer have physical possession of the outsourced data makes the data integrity protection in Cloud Computing a formidable task, especially for users with constrained computing resources. 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 auditability 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. The application used for simulation is designed using Microsoft Visual Studio .Net 2005 as front end. The coding language used is Visual C# .Net. MS-SQL Server 2000 is used as back end database.

Keywords- Cloud computing, Virtual machine, and Multiparty computation.

I. INTRODUCTION

Cloud computing creates a large number of security issues and challenges. A list of security threats to cloud computing is presented in [8]. These issues range from the required trust in the cloud provider and attacks on cloud interfaces to misusing the cloud services for attacks on other systems. The main problem that the cloud computing paradigm implicitly contains is that of secure outsourcing of sensitive as well as business-critical data and processes. When considering using a cloud service, the user must be aware of the fact that all data given to the cloud provider leave the own control and protection sphere. Even more, if deploying data-processing applications to the cloud (via IaaS or PaaS), a cloud provider gains full control on these processes. Hence, a strong trust relationship between the cloud provider and the cloud user is considered a general prerequisite in cloud computing. Depending on the political context this

trust may touch legal obligations. For instance, Italian legislation requires that government data of Italian citizens, if collected by official agencies, have to remain within Italy. Thus, using a cloud provider from outside of Italy for realizing an e-government service provided to Italian citizens would immediately violate this obligation. Hence, the cloud users must trust the cloud provider hosting their data within the borders of the country and never copying them to an off-country location (not even for backup or in case of local failure) nor providing access to the data to entities from abroad. An attacker that has access to the cloud storage component is able to take snapshots or alter data in the storage. This might be done once, multiple times, or continuously. An attacker that also has access to the processing logic of the cloud can also modify the functions and their input and output data. Even though in the majority of cases it may be legitimate to assume a cloud provider to be honest and handling the customers' affairs in a respectful and responsible manner, there still remains a risk of malicious employees of the cloud provider, successful attacks and compromisation by third parties, or of actions ordered by a subpoena.

In this paper, the following four architectural patterns are distinguished: replication of applications allows to receive multiple results from one operation performed in distinct clouds and to compare them within the own premise. This enables the user to get an evidence on the integrity of the result. Partition of application System into tiers allows separating the logic from the data. This gives additional protection against data leakage due to flaws in the application logic. Partition of application logic into fragments allows distributing the application logic to distinct clouds. This has two benefits. First, no cloud provider learns the complete application logic. Second, no cloud provider learns the overall calculated result of the application. Thus, this leads to data and application confidentiality. Partition of application data into fragments allows distributing fine-grained fragments of the data to distinct clouds. None of the involved cloud providers gains access to all the data, which safeguards the data's confidentiality. Each of the introduced architectural patterns provides individual security merits, which map to different application scenarios and their security needs. Obviously, the patterns can be combined resulting in combined security merits, but also in higher deployment and runtime effort. The following sections present the four patterns in more detail and investigate their merits and flaws with respect to the stated security requirements under the assumption of one or more compromised cloud systems.

II. RELATED WORK

- To set different trust level is set to different cloud providers and encryption/decryption is varied based on the clouds computational capability.
- To take partial data of files from multiple mirror locations and send to selected client.
- To reduce the risk for data and applications in a public cloud is the simultaneous usage of multiple clouds.
- To handle Irrelevant size blocks of data among the multiple cloud service providers based on their computational capabilities.

- To replicate the applications that allows to receive multiple results from one operation performed in distinct clouds and to compare them within the own premise. This enables the user to get an evidence on the integrity of the result.
- To partition the application System into tiers that allows separating the logic from the data. This gives additional protection against data leakage due to flaws in the application logic.
- To partition the application logic into fragments that allows distributing the application logic to distinct clouds. This has two benefits. First, no cloud provider learns the complete application logic. Second, no cloud provider learns the overall calculated result of the application. Thus, this leads to data and application confidentiality.

  To partition the application data into fragments that allows distributing fine-grained fragments of the data to distinct clouds. None of the involved cloud providers gains access to all the data, which safeguards the data's confidentiality.

III. PROBLEM STATEMENT

The first step in the software development life cycle is the identification of the problem. As the success of the system depends largely on how accurately a problem is identified.

At present, all the data is encrypted/ decrypted with same encryption method, so that all data is given same importance and so existing system is less secure. So in proposed system partial data of files are taken from multiple mirror locations and send to selected client and different trust level is set to different cloud providers and encryption/decryption is varied based on the clouds computational capability.

The following drawbacks are identified from the existing system.

- 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.

IV. EXPLORING INFORMATION LEAKAGE IN THIRD-PARTY COMPUTE CLOUDS

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.

Using the Amazon EC2 service as a case study, they showed that it is possible to map the internal cloud infrastructure, identify where a particular target VM is likely to reside, and then instantiate new VMs until one is placed co-resident with the target. They explored how such placement can then be used to mount cross-VM side-channel attacks to extract information from a target VM on the same machine.

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 [5] 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 [2], 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.

V. SEPIA: PRIVACY- PRESERVING

AGGREGATION OF MULTI-DOMAIN

NETWORK EVENTS AND STATISTICS

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.

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 [7], multi-domain traffic engineering [6], detecting traffic discrimination [3], and collecting network performance statistics [4].

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 [3]. 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 [1].

One possible solution to this privacy-utility tradeoff is MPC. For almost thirty years, MPC [2] techniques have been studied for solving the problem of jointly running computations on data distributed among multiple organizations, while provably preserving data privacy without relying on a trusted third party. In theory, any computable function on a distributed dataset is also securely computable using MPC techniques [7]. However, designing solutions that are practical in terms of running time and communication overhead is non-trivial. For this reason, MPC techniques have mainly attracted theoretical interest in the last decades. Recently, optimized basic primitives, such as comparisons [6], make progressively possible the use of MPC in real-world applications, e.g., an actual sugar-beet auction [3] was demonstrated in 2009.

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 [8]. 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.

On top of these comparison operations, they designed and implemented novel MPC protocols tailored for network security and monitoring applications. The event correlation protocol identifies events, such as IDS or firewall alerts, that occur frequently in multiple domains. The protocol is generic having several applications, for example, in alert correlation for early exploit detection or in identification of multi-domain network traffic heavy-hitters. 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 histograms.

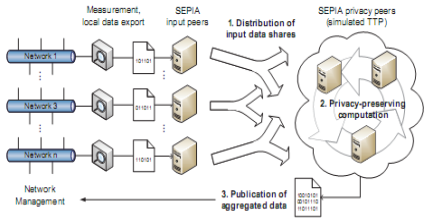

Fig.1. Deployment Scenario for Sepia

A typical setup for SEPIA is depicted in Fig. 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:

- They introduced efficientMPC comparison operations, which outperform constant-round alternatives for many parallel invocations.
- They designed novel MPC protocols for event correlation, entropy and distinct count computation.
- 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.
- 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.

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] [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.

AUTHORS BIOGRAPHY

Mr.Arun.K Pursuing ME (CSE) degree in Shree Venkateshwara Hi- Tech Engineering College, Erode, India in 2014-2016 and BE(CSE) degree from the Shree Venkateshwara Hi- Tech Engineering College, Erode, India in 2008-2012. He is attended National 1 Workshop. Him research interests include Software engineering, Cloud Computing and Data Mining.

Dr.T.Senthil Prakash received the Ph.D. degree from the PRIST University, Thanjavur, India in 2013 and M.E(CSE) degree from Vinayaka Mission's University, Salem , India in 2007 and M.Phil.,MCA.,B.Sc(CS) degrees from Bharathiyar University, Coimbatore India, in 2000,2003 and 2006 respectively, all in Computer Science and Engineering. He is a Member in ISTE New Delhi, India, IAENG, Hong Kong..IACSIT, Singapore SDIWC,

USA. He has the experience in Teaching of 10+Years and in Industry 2 Years. Now He is currently working as a Professor and Head of the Department of Computer Science and Engineering in Shree Venkateshwara Hi-Tech Engineering College, Gobi, Tamil Nadu, and India. His research interests include Data Mining, Data Bases, Artificial Intelligence, Software Engineering etc.,He has published several papers in 17 International Journals, 43 International and National Conferences.

Ms.T.Malathi Pursuing ME (CSE) degree in Shree Venkateshwara Hi-Tech Engineering College, Erode, India in 2014-2016 and BE(CSE) degree from Avinashilingam University for Women, Coimbatore, India in 2007-2011 .She published 2 National Conferences, 1 Workshop. Her research interests include Software engineering, Cloud Computing and Data Mining.