IJISET - International Journal of Innovative Science, Engineering & Technology, Vol. 1 Issue 7, September 2014.
www.ijiset.com
ISSN 2348 – 7968

Obfuscation Technique for Securing Data Sensitivity Indicators

Shalini Saxena, Rohit Singh
Computer Science Engineering Department
Career Point University
Kota, Rajasthan, India

Abstract

Cloud Computing is the recent trend in computing service provision, from being provided locally to being provided remotely and en masse, by third-party service providers. Functionality such as storage, processing is now offered on demand, as a pay-per-use service. Data, housed under the consumer's administrative and security domain, has now been extracted and placed under the domain of the Cloud Service Provider (CSP). The consumer has effectively lost control over how their data is being stored, shared and used, and also over the techniques used to protect their data. Moreover, a surreptitious employee of the service provider who has access to data for legitimate purposes might abuse this power for own means. Such potential threats to data stored in a cloud call for mechanisms for data management which can be dictated by the data owner. We propose here a method to include data owner's choice of treatment with the data object itself in a secure manner.

Keywords- *Data Treatment, Obfuscation, Service attribute, SLA indicator, Service Level Options, Tag.*

1. Introduction

Cloud computing is the new trend of computing where readily available computing resources are exposed 'as a service'. These computing resources are generally offered as pay-as-you-go plans and hence have become attractive to cost conscious customers. Apart from the cost, cloud computing also supports the growing concerns of carbon emissions and environmental impact since the cloud advocates better management of resources [2]. We see a growing trend of off-loading the previously in-house service systems to the cloud, based primarily on the cost and the maintenance burden. Such a move allows businesses to focus on their core competencies and not burden themselves with back office operations. As consumers move towards adopting such a Service-Oriented Architecture (SOA), the quality and reliability of the services become important aspects. However the demands of the service consumers vary significantly. It is not possible to fulfill all consumer expectations from the service provider perspective and hence a balance needs to be made via a negotiation process. At the end of the negotiation process, provider and consumer commit to an agreement. In SOA terms, this agreement is referred to as a Service Level Agreements (SLA). This SLA serves as the foundation for the expected level of service between the consumer and the provider. The QoS attributes that are generally part of an SLA (such as response time and throughput) however change constantly and to enforce the agreement, these parameters need to be closely monitored [2]. As more and more consumers delegate their tasks to cloud providers, SLA between consumers and providers emerge as a key aspect. As observed in [2], two main drawbacks of current SLA management practice are:

- a very legitimate need for a clear and formal methodology to handle SLAs in the context of cloud computing
- true to the paradigm of SOA, every functionality is provided as a service that may not necessarily come from the same provider. SLAs should be "inheritable" in part wherever the service is delegated by the cloud service provider to other provider.

2. Current scenario

The existing security-based solutions for cloud-based platform are either based on single tamper-proof hardware or homomorphic encryption. Hardware-based solution lacks scalability, while homomorphic encryption is very costly. Moreover, traditional defense in-depth security mechanism cannot be directly implemented in cloud-based platform due to the varying nature of its service and deployment model. However, the concept of multi-layered security mechanism can be proposed to secure the cloud-based platform [10].

At present, a customer can select the type of service he needs for private data storage through SLA. But what if the user wants different service for some of its data? Only possible solution in current set-up would be that user re-registers as a new user and signs a new SLA. The exercise would be repeated for every different set of services. The inconvenience increases when service level for certain data requires a change, needing logically a change in data owner. Practical situations do not match this logical scenario. Hence, we need a mechanism through which the SLA parameters can be attached to the data itself, instead of data-owner (or user) [9].

3. Literature survey

3.1 Cloud Security Issues

Place The Notorious Nine: Cloud Computing Top Threats in 2013, experts identified the following nine critical threats to cloud security (ranked in order of severity)[5]:

1. Data Breaches

2. Data Loss

3. Account Hijacking

4. Insecure APIs

5. Denial of Service

6. Malicious Insiders

7. Abuse of Cloud Services

8. Insufficient Due Diligence

9. Shared Technology Issues

Data Loss is the issue being addressed here. Data Loss/Leakage Prevention (DLP) is a computer security term which is used to identify, monitor, and protect data in use, data in motion, and data at rest [6]. DLP is used to identify sensitive content by using deep content analysis to peer inside files and with the use if network communications. Thus, in order to ensure security of data, such mechanisms actually may violate privacy of the data owner.

Currently, existing DLP systems do not consider the privacy violation during the monitoring process. A DLP system considering privacy violation level is proposed in [3]. Yet, the proposed solution can only reduce the amount of privacy violation, but cannot totally forego it [3]. It would be better if sensitive data can be identified by looking at a tag attached to the data object like file, document, database etc.

3.2 Data Classification

A Data Classification Program is an extremely important first step to building a secure organization. Classifying data is the process of categorizing data assets based on nominal values according to its sensitivity (e.g., impact of applicable laws and regulations) [7].

Data classification can yield significant benefits, such as compliance efficiencies, improved ways to manage the organization's resources, and facilitation of migration to the cloud. Although data classification efforts can be complex undertakings and require risk assessment for successful implementation, quicker and simpler efforts can also yield benefits. The key to effective data management may well depend on effective data classification [4].Classifications are useful, because they compress a vast set of possibilities into a small set of categories. This makes it easier to decide what to do.

The data has to be classified into 4 broad categories:

1. Highly Sensitive: This refers to information that, if disclosed without authorization, would cause "exceptionally grave damage" to national security [7].

2. Sensitive: This is information that would cause "grave damage" to national security if made available to the public [7].

3. Confidential: This refers to information that would either "cause damage" or be "prejudicial" to national security if release [7].

4. Public: This is information that falls outside the classification scheme and may be publicly released [7].

4. METHODOLOGY

We need a mechanism through which all different sets of services that can be provided to a customer can be identified by looking at the data-tag and treated accordingly. Such process would not need a change of owner, rather identify within a logical owner different levels of services to be provided to different data. The only problem here is that if data would contain a tag to classify its "importance" (indicating the level of data treatment it requires) it might lead to security breach either during transit or by malicious insiders. Hence, we need to secure the tag either through encryption or obfuscation. We present here a mechanism to store the different levels of services provided by the cloud, within a single SLA [6]. The possible combinations of different levels of the services required by a customer are identified and arranged into a tree structure which transforms each set into a unique number. The number is tagged along with data. To secure the tag we have used a randomizing obfuscation method, which is effective and fast. The tag processing on cloud's end is very efficient thus involving practically no overhead.

4.1 Terminology

4.1.1 Service Level agreement

An SLA defines how the consumer will use the services and how the provider will deliver them [6]. The SLA should contain:

1) The list of services the provider will deliver and a complete definition of each service.

2) Metrics to determine whether the provider is delivering the service as promised and an auditing mechanism to monitor the service.

3) Responsibilities of the provider and the consumer and remedies available to both if the terms of the SLA are not met.

4) A description of how the SLA will change over time.

There can be two types of SLAs off-the-shelf agreements and customized-negotiated agreements. Customers with critical data needs will not be satisfied with off-the-shelf agreements, so a first step before going to the cloud is to determine how critical your data and applications are [9]. SLA parameters are specified by metrics. Metrics define how service parameters can be measured and are typically functions. There are at least two major types of metrics. 1) Resource metrics are retrieved directly from the provider resources and are used as is without further processing. For example, transaction counts. 2) Composite metrics represents a combination of several resource metrics, calculated according to a specific algorithm. For example transactions per hour combine the raw resource metrics of transaction count and up-time. Composite metrics are required when the consumers need insightful and contextual information where raw numbers do not suffice. A third metrics referred to as a business metric can also be defined. It relates SLA parameters to financial terms specific to a service customer [2]. Here, SLA indicator represents the sensitivity level and the relevant services to be provided accordingly.

4.1.2 Service Attribute

Data related services like isolation, encryption, archival, back-up, redundancy etc have been addressed generally as a service attribute [2].

4.1.3 Service Level Option

Any service attribute has some different options out of which one can be selected by the customer. For example, in case of data-access service attribute, options could be read-only for all, read-only for group, read-only for individual, read/write for a group, read/write by owner only and so on. In case, the service is measurable using performance metrics, the service level option can be the range of the metric [2].

4.1.4 Data treatment

A combination of service attribute level options when applied to data, are collectively termed as data treatment. Data treatment is manifestation of the service asked for according to the service level option. For example, encrypting data for security, granting access to authorized user, isolation in multi-tenant environment, maintaining replicas for availability, taking back-up after certain duration etc.

4.1.5 Service Level Objectives (SLOs)

Service Level Options are a set of formal expressions. These formal expressions have the well known if...then structure. The antecedent, (if) contains conditions and the consequent, (then) contains actions. An action represents what a party has agreed to perform when the conditions are met [2].

4.2 Proposed Approach

We arrange the possible combinations of services into a tree structure, where each level denotes a type of service. Every node is numerically labeled such that no child can have a label value greater than its parent label value. The number of children of a node is governed by its label and level. These values indicate the type of service. A leaf node represents a set of services. A non-leaf node has all children arranged by increasing order of label values. We assumed that SLA has certain options for services like encryption, isolation level, sensitivity level, access protection, etc., resulting into total of N different combinations of services that user can subscribe to and structured the combinations of services into a labeled tree, such that every path from root to leaf has a unique sum of weighted labels of nodes.

Suppose the SLA contains k different attributes, then the tree has depth of k, each level representing one attribute. For example, let k be 4.

4.2.1 Computing the SLA Indicator

Any SLA indicator indicates an entire set of services that is represented by a leaf node in the tree structure too. Hence, a path from root to a leaf node is selected. Say a path from root to leaf consists of the sequence of nodes $(n_1, n_2, \dots, n_k)$ excluding the root. The indicator is a numerical value computed as

$$SI = \sum_{i=1}^{k} label_{n_i} * k^{i-1}$$

where $label_v$ refers to the numeric value of label of node v in the path.

Figure 1 shows all paths from root to leaf nodes along with the leaf nodes, representing the numeric weight specifying the SLA indicator.

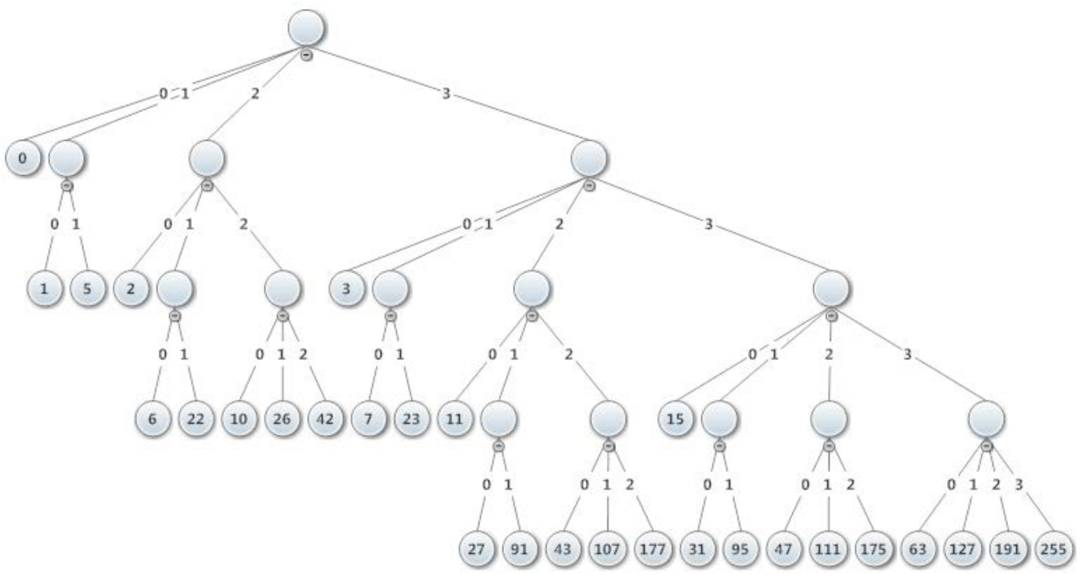

Fig 1: Tree Representation showing SLA Indicators.

4.2.2 Obfuscation Scheme

Input is the k numbers $\langle n_1, n_2, \dots, n_k \rangle$ indicating the type of service for the different attributes of SLA.

Step 1: Construct an array of size k, each element itself is a string of $\log_2 k$ bits, that is $\langle b_{11}b_{12} \dots b_{1\log k}b_{21}b_{22} \dots b_{2\log k} \dots b_{k1}b_{k2} \dots b_{k\log k} \rangle$, such that $\langle b_{i1}b_{i2} \dots b_{i\log k} \rangle$ is binary representation of number n_i within $\log_2 k$ bits.

Step 2: Treating entire array as one string of bits, convert it to equivalent decimal number. This is called Tag.

Step 3: Choose a secret nonce, s, to append the tag. Thus, $Tag' = Tag + s$

Nonce could be a timestamp, or sent in a cookie, or a shared secret. Value to be sent to the CSP is Tag', which is not directly related to the original SLA indicator.

4.2.3 Tag Processing

CSP receives data along with Tag', which should be processed to obtain SLA. Data is treated according to the SLA during storage. Thus, SLA indicator is always processed but never stored with data. This processing essentially reverses the process of obfuscation to obtain service level options.

Step 1: Subtract secret s from Tag' to obtain Tag, that is $Tag = Tag' - s$

Step 2: Convert Tag to string of bits, as $\langle a_1, a_2, \dots, a_{k\log k} \rangle$

Step 3: Group the string into k groups, as

$\langle a_1, \dots, a_{\log k}, a_{\log k+1}, \dots, a_{2\log k}, \dots, a_{(k-1)\log k}, \dots, a_{k\log k} \rangle$

Step 4: Each group is converted to equivalent decimal value to obtain k values, $value_1, value_2, \dots, value_k$

Step 5: Compute $SI = \sum_{i=1}^{k} value_i * k^{i-1}$

SI denotes the SLA indicator.

Lemma: Maximum value of SLA indicator can be $(k^k - 1)$ for any k.

Proof: The number of levels in a tree is k, hence length of path from root to a leaf node is k. Largest label value possible is $k - 1$. Thus, maximum value of SLA indicator can be obtained as

$$SI_{max} = \sum_{i=1}^{k} (k-1) * k^{i-1} = (k^k - 1)$$

Hence proved.

5. IMPLEMENTATION RESULTS

The scheme was implemented on a single system by treating the user and CSP as separate threads. The run-time for different steps of the proposed scheme were recorded to observe the growth of run-time with value of k. Table 1 shows the growth of runtimes of various operations with varying values of ‘k’. Here, k represents the maximum value of SLA attributes provided by the CSP.

The values of timing are in microseconds.

Table 1 Time taken by various operations for different values of k

k	Time CBSTR (in µsec)	Time BTDEC (in µsec)	Time PROC (in µsec)
4	35.005	120.403	10.26
5	35.306	124.024	11.166
6	35.307	124.929	11.769
7	35.608	126.136	12.674
8	35.91	127.042	13.278
9	36.211	127.645	18.71

CBSTR Conversion to Binary String, BTDEC Binary to Decimal Conversion, PROC Processing carried out at data retrieval.

The recorded values were plotted to produce a graph. Figure 2 represents the graphical interpretation of the results.

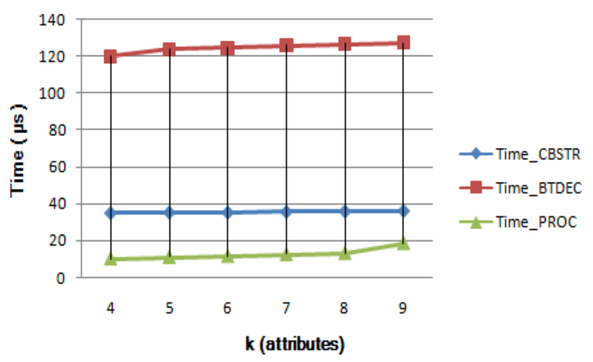

Fig 2: Graphical Representation of Results

6. Key Aspects of Proposed Scheme

The proposed scheme has following features:

- Unforgeability – The SLA indicators cannot be forged if value of k is not known. Even with the knowledge of k, it is difficult to guess a valid value of SLA. The valid values of a tag is any SLA indicator value, which varies from 0 to $(k^k - 1)$, but does not cover entire number space between 0 to $(k^k - 1)$, since there are very few possible valid SLA indicator values. Also, the random element of “nonce” makes it impossible to forge a desired SLA indicator.
- Fast – Most of the security schemes rely on encryption [8]. Our proposal uses data obfuscation which is much faster than encryption. Run-time is almost constant for k<15.
- Protects against both kinds of attacks: eavesdropping and malicious insiders. Eavesdropper cannot deduce from the obfuscated tag value the actual “class” of data object. Malicious insiders have the knowledge of SLA indicator, yet, if the value of k and nonce can be secured through other means, such adversary will not be able to associate the actual tag value with the data object.
- Negligible computation overhead for tag calculation and processing – The time required for tag preparation at the user end and processing at the cloud end is almost constant for values of k<15, which is generally used.

7. Conclusion

In spite of its promises, cloud computing has not been adopted by many enterprises due to the questions over security and privacy. The data owner wants to control how the data stored on cloud is used, shared, managed and protected. We have proposed an idea how this can be ensured to the data owner on per data item. Data owner has a set of options to choose from; the choice assures the data treatment while keeping the same hidden from malicious insiders. Cloud security threats can be dealt with only through formal models. We have proposed such a formalism which can be incorporated into existing SLAs.

REFERENCES

[1] Richard E. Mackey, Data Loss Prevention. A SEARCH SECURITY.COM E-BOOK..

[2] Pankesh Patel, Ajith H.Ranabahu, Amit P.Sheth, 2009. Service Level Agreement in Cloud Computing. Knoesis Center, Wright State University CORE Scholar.

[3] Jinhyung Kim, Jun Hwang, Hyung-Jong Kim, July, 2012. Privacy Level Indicating Data Leakage Prevention System. International Journal of Security and Its Applications Vol. 6, No. 3.

[4] Frank Simorjay, 2014. Data Classification for Cloud Readiness, Microsoft Trustworthy Computing.

[5] Top Threats Working Group, February 2013. The Notorious Nine: Cloud Computing Top Threats in 2013, Cloud Security Alliance (CSA).

[6] Security Guidance for Critical Areas of Focus in Cloud Computing V3.0, 2011. Cloud Security Alliance (CSA).

[7] Matt Jach, July 2012. Data Loss Prevention: Refreshing data security to meet an evolving threat environment. CDWG.com/dlpguide.

[8] SecaaS Implementation Guidance, 2012. Category 2: Data Loss Prevention, Cloud Security Alliance (CSA).

[9] Wayne Jansen, Timothy Grance, January 2011. Guidelines on Security and Privacy in Public Cloud Computing, National Institute of Standards and Technology.

[10] Sanjaya Dahal, 2012. Security Architecture for Cloud Computing Platform. KTH, School of Information and Communication Technology (ICT), KTH Publication Database.