IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 5, Issue 3, March 2016

Privacy Preservation and Detection of Sensitive Data Exposure over Cloud

Pooja Pawar[1], Supriya Palwe[2], Shweta Munde[3], Priyanka Gadhave[4], Mrs. Shikha Pachouly[5]

Department of Computer Engineering, AISSMSCOE, Pune[1, 2, 3, 4]

Professor, Computer Engineering Department, AISSMSCOE, Pune [5]

Abstract: Statistics from security firms, research institutions and government organizations show that the number of data-leak instances have grown rapidly increases years. Among various data-leak cases, human mistakes are one of the main causes of data loss. There exist solutions detecting inadvertent sensitive data leaks caused by human mistakes and to provide alerts for organizations. A common approach is to screen content in storage and transmission for exposed sensitive information. Such an approach usually requires the detection operation to be conducted in secrecy. However, this secrecy requirement is challenging to satisfy in practice, as detection servers may be compromised or outsourced. In this paper, we present a privacy preserving data-leak detection solution to solve the issue where a special set of sensitive data digests is used in detection. The advantage of our method is that it enables the data owner to safely delegate the detection operation to a semi honest provider without revealing the sensitive data to the provider. We describe how Internet service providers can over their customers DLD as an add-on service with strong privacy guarantees. The evaluation results show that our method can support accurate detection with very small number of false alarms under various data leak.

Keywords: Data leak, network security, privacy.

1. INTRODUCTION

Data leakage is the unauthorized transmission of data or information from within an organization to an external destination. In the course of doing business, sometimes sensitive data must be handed over to supposedly trusted third parties. Our goal is to detect when the distributor's sensitive data have been leaked by agents, and if possible to identify the agent that leaked the data. So we are using SHA-1 algorithm for data leakage detection. According to Risk Based Security the number of leaked sensitive data record increased dramatically during the last few years, deliberately planned attacks, inadvertent leaks and human mistakes lead to most of the data-leak incidents. Detecting and preventing data leaks require a set of complementary solutions, which may include data leak detection, data confinement stealthy malware detection, and policy enforcement.

The main aim behind this project is detection of the leakage of sensitive data .The data Owner of the organization load the sensitive data of organization ,create the chunks of the data, and calculate the hash of each chunk ,owner send that hash information to the public cloud server with the meta data. The actual data send on the private server for the further use of the organization, that data can have access by all the employee of that organization The hash value of each chunk send on the public that can be continuous comparing with the data on the Internet for security, when match found, the owner can get mail from the public server that your data has been leaked.

The owner the take appropriate action on that, what data can leaked. By who can come to know owner.

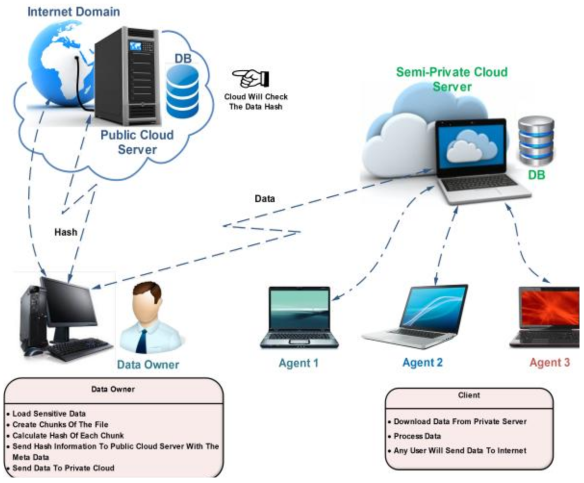
Internet Domain
DB
Public Cloud Server
Cloud Will Check The Data Hash
Semi-Private Cloud Server
DB
Data
Hash
Data Owner
Agent 1
Agent 2
Agent 3
Data Owner
• Load Sensitive Data
• Create Chunks Of The File
• Calculate Hash Of Each Chunk
• Send Hash Information To Public Cloud Server With The Meta Data
• Send Data To Private Cloud
Client
• Download Data From Private Server
• Process Data
• Any User Will Send Data To Internet

Fig1. Architecture

In this paper, section 2 includes the problem definition in detail, Section 3 covers work done in this area, Section 4 represents Modules and overview related with the data leakage detection system, Section 5 represents distribution and detection algorithm along with description optimization problem in the system**,** Section 6includes some experimental results and outcome of the algorithms, Section 7 includes conclusion of and list of references used to prepare this paper.

2. PROBLEM DEFINITIONS

In organization, the owner give some data to agents for performing some operation on it But in this data some sensitive data is included. This sensitive data if leakages

DOI 10.17148/IJARCCE.2016.5370

then it is very harmful for the business it leaves the company unprotected and destroys the image and customers trust. This uncontrolled data leakage puts business in a vulnerable position. if it is not controlled then sensitive data is not securely distributed to the agents. Company is at serious risk so distributor must find out the guilty agent if the leaked from one or more agents, here the data allocation strategies that improve the probability of identifying agent are proposed. This method works if leaked data is obtained as it was distributed.

3. RELATED WORK

Privacy preserving data-leak detection solution to solve the issue where a special set of sensitive data digests is used in detection. The advantage of their method is that it enables the data owner to safely delegate the detection operation to a semi honest provider without revealing the sensitive data to the provider [1]. Data allocation strategies are used to improve the probability of identifying guilty third parties. Here implement and analyse a guilt model that detects the agents using allocation strategies without modifying the original data [4].A file distribution model aiming at data leakage prevention. Taking into consideration the guilt probability which is the probability of the user having leaked files, this model can select a file with the least overlap between obtained file sets of users, as the con-sequence the model can find out leakage sources with high probability. So that measures can be taken for information security further. The simulation experiments reveal that the model can distinguish malicious users and detect the leak source effectively [5]. Goal is to identify the guilty agent when distributor's sensitive data have been leaked by some agents. Perturbation and watermarking are techniques which can be helpful in such situations. Perturbation is a very useful technique where the data is modified and made less sensitive before being handed to agents [3].

4. MODELSAND OVERVIEW

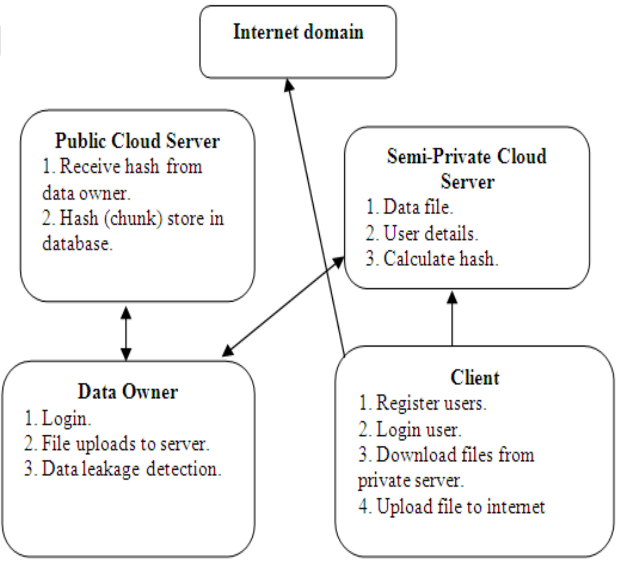
Internet domain
Public Cloud Server
1. Receive hash from data owner.
2. Hash (chunk) store in database.
Semi-Private Cloud Server
1. Data file.
2. User details.
3. Calculate hash.
Data Owner
1. Login.
2. File uploads to server.
3. Data leakage detection.
Client
1. Register users.
2. Login user.
3. Download files from private server.
4. Upload file to internet

Fig: 2 Implementation Diagram.

1. Login / Registration:
This is a module mainly designed to provide the authority to a user/agent in order to access the other modules of the project. Here a user/agent can have the accessibility authority after the registration

2. Data Distributor:
A data Distributor part is developed in this module. A data distributor has given sensitive data to a set of supposedly trusted agents (third parties). Some of the data is leaked and found in an unauthorized place (e.g., on the web or somebody's laptop). The distributor must assess the likelihood that the leaked data came from one or more agents, as opposed to having been independently gathered by other means.

3. Data Allocation Module:
The main focus of our project is the data allocation problem as how can the distributor intelligently give data to agents in order to improve the chances of detecting a guilty agent.

4. Finding Guilty Agents Module:
The Optimization Module is the distributor's data allocation to agents has one constraint and one objective. The distributor's constraint is to satisfy agents requests, by providing them with the number of objects they request or with all available objects that satisfy their conditions. His objective is to be able to detect an agent who leaks any portion of his data. Here a count value (also called as fake objects) are incremented for any transfer of data occurrence when agent transfers data.

5. SECURE HASH ALGORITHM -1

Step 1: Initialize some variables There are _ve variables that need to be initialized.
- h0 = 01100111010001010010001100000001
- h1 = 11101111110011011010101110001001
- h2 = 10011000101110101101110011111110
- h3 = 00010000001100100101010001110110
- h4 = 11000011110100101110000111110000

Step 2: Pick a string Example use the string: 'A Test'.

Step 3: Break it into characters.
- A
- T
- e
- s
- t

Step 4: Convert characters to ASCII codes
- 65
- 32
- 84
- 101
- 115
- 116

Step 5: Convert numbers into binary
- 01000001
- 00100000
- 01010100

DOI 10.17148/IJARCCE.2016.5370

- 01100101
- 01110011
- 01110100

Step 6: Add '1' to the end.
- Put the numbers together: 01000001001000000101010001100101011100110111010 0
-Add the number '1' to the end: 01000001001000000101010001100101011100110111010 01
o

Step 7: Append '0's' to the end.
-
01000001001000000101010001100101011100110111010
0100000000000000000000000000000000000

Step 8: 'Chunk' the message.
01000001001000000101010001100101011100110111010
01000000000000000000000000000000000000000000000
000000000000000000000000000000000

Step 9: Break the 'Chunk' into 'Words'. Break each chunk up into sixteen
32-bit words
- 0: 01000001001000000101010001100101
8:00000000000000000000000000000000
- 1: 01110011011101001000000000000000
9:00000000000000000000000000000000
- 2: 00000000000000000000000000000000
10:00000000000000000000000000000000
- 3: 00000000000000000000000000000000
11:00000000000000000000000000000000
- 4: 00000000000000000000000000000000
12:00000000000000000000000000000000
- 5: 00000000000000000000000000000000
13:00000000000000000000000000000000
- 6: 00000000000000000000000000000000
14:00000000000000000000000000000000
- 7: 00000000000000000000000000000000
15:00000000000000000000000000110000

B.E. Project Seminar. 10/16
Step 10: 'Extend' into 80 words. We begin by selecting four of the current words. The ones we want are: [i-3], [i-8], [i-14] and [i-16]. That means forthe first time through the loop we want the words numbered: 13, 8, 2 and0.
- 0: 01000001001000000101010001100101
- 2: 00000000000000000000000000000000
- 8: 00000000000000000000000000000000
- 13: 00000000000000000000000000000000

Now that it has our words selected we will start by performing what' sknown as an 'XOR' or 'Exclusive OR' on them. In the end all four words will be XOR 'ed together, it first doing [i- 3]XOR[i-8] then XOR 'ing that by [i-14] and that again by [i-16]. Left rotate Perform a left bit rotation by a factor of one. B.E.

6. EXPERIMENT RESULTS

We present the time for detection per packet in fig.3 it shows SHA-1 run faster than MD-5.

SHA-1 algorithm has many features which shows its better than MD5 algorithm like the SHA-1 is used to compute message digest for message or data file that is provided as a input it consider message a data file to be bit string.

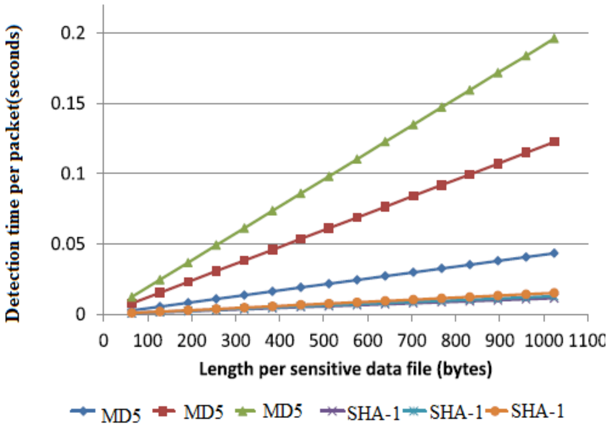

Fig 3.Comparision graph of MD5 and SHA1

A hash function is simply an algorithm that takes a string of any length and reduces it to a unique fixed length string. The length of the message is the number of bits in the message (the empty message has length 0).If the number of bits in a message is a multiple of 8, for compactness we can represent the message in hex. The purpose of message padding is to make the total length of a padded message a multiple of 512 The SHA1 sequentially processes blocks of 512 bits when computing the message digests. As a summary, a "1" followed by m "0"s followed by a 64-bit integer are appended to the end of the message to produce a padded message of length 512 * n. The 64-bit integer is l, the length of the original message. The padded message is then processed by the SHA1 as n 512-bit blocks.

7. CONCLUSION

It is possible to access the likelihood that an agent is responsible for a leak based on hashed value of original data and leaked data. It can improve the distributor’s chances of identifying a leaker. Data leakage detection system model is very useful as compared to existing watermarking model. For future work, we plan to focus on designing a mechanism for data-leak detection on Email, Email Attachment; multiple domains can also be considered.

OUR CONTRIBUTION

In previous paper he owner may be able to add fake objects to the owner data in order to improve his effectiveness in detecting guilty agents. However, fake objects may impact the correctness of what agents do, so they may not always be allowable. The idea of perturbing data to detect leakage is not new. However, in most cases, individual objects are perturbed. Adding random noise to sensitive salaries or adding a water-mark to an image is example of perturbation. But it has drawback that watermark can destroyed in some condition. So instance of this we use verification module to check data leakage .In

DOI 10.17148/IJARCCE.2016.5370

which the verification module continuously check the leak data and owners data chunk by chunk .After detection of leak data the owner come to know that data is leaked through mail.

ACKNOWLEDGEMENT

Apart from our own, the success of this report depends largely on the encouragement and guidelines of many others. We are especially grateful to our guide **Prof. S.J. Pachouly, B.A. Patil and Prof D. P. Gaikwad,** Head of Computer Engineering Department, AISSMSCOE who has provided guidance, expertise and encouragement. We are thankful to the staff of Computer Engineering Department for their cooperation and support. We would like to put forward our heartfelt acknowledgement to all our classmates, friends and all those who have directly or indirectly provided their over whelming support during this project work and the development of this report.

REFERENCES

[1] Xiaokui Shu, Danfeng Yao, Member, IEEE, and Elisa Bertino, Fellow, IEEE, Privacy-Preserving Detection of Sensitive Data Exposure, IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY, VOL. 10, NO.5, MAY 2015.

[2] Hector Garcia-Molina, "Data Leakage Detection", IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, VOL. 23, NO. 1, JANUARY 2011.

[3] Rupesh Mishra, D.K. Chitre, Data Leakage and Detection of Guilty Agent, International Journal of Scientifically and Engineering Research, Volume 3, Issue 6, June-2012 1 ISSN 2229-5518.

[4] Rudragouda G Patil Dept of CSE, the Oxford College of Engg, Bangalore. Development of Data leakage Detection Using Data Allocation Strategies, International Journal of Computer Applications in Engineering Sciences [ISSN:2231-4946]

[5] Yin Fan and Wang Lina , A Distribution Model for Data Leakage Prevention,2013 International Conference on Mechatronics Sciences, Electric Engineeringand Computer (MEC) Dec 20-22, 2013, Shenyang, China.

[6] B. Wang, S. Yu, W. Lou, and Y. T. Hou, Privacy-preserving multi key word fuzzy search over encrypted data in the cloud, in Proc. 33th IEEE Conf. Computer. Commun., Apr./May 2014, pp. 21122120.

DOI 10.17148/IJARCCE.2016.5370