ISSN No:-2456-2165

Data Leakage in Cloud System

Aditi Chaudhary[1], Chetna Gola[2] and Manshi[3]
[1,2,3]Indira Gandhi Delhi Technical University for Women, New Delhi 110006, India

Abstract:- The issue of data leakage is more prevalent in a cloud computing en- vironment. The data distributor transfers such sensitive data to the agents and some of the data gets leaked, the technique of finding a guilty agent talked about here is through a probability distribution model. The probability calculat- ed on the basis of the number of file downloads from the distributors by the agents is used in detection. To combat the leakage scenario, several precaution- ary measures can be taken out of which the technique of watermarking is dis- cussed here.

Keywords:- *Data leakage detection, guilty agent, probability distribution model, watermarking, watermarking techniques, Advanced Encryption Standard.*

I. INTRODUCTION

The unintentional or unintended disclo- sure of private or sensitive information to an unauthorized entity is referred to as data leakage, also known as data loss or data breach. A number of things, includ- ing as hacking, theft, unintentional expo- sure, or staff misbehavior, might cause this. Data leaking may cause great harm to people or organizations since it may lead to financial loss, damage to one's reputation, or legal culpability. Personal identifying information (PII), financial data, health records, and intellectual property are among the sensitive data that hackers and other bad actors frequently target.

In the context of cloud computing, da- ta leakage is the unintentional disclosure of private or sensitive information that has been stored there. This can happen when unauthorized users access, modify, or send data, or when there is a security breach inside the cloud infrastructure itself.

There are various ways in which data leakage can occur in a cloud computing environment, including:

- Insider threats: This term describes the possibility of sensitive data being dis- closed by authorized people who have access to it. These people could uninten- tionally or purposely reveal information.
- Insufficient access controls: Inadequate or incorrectly designed access controls may provide unauthorized people access to confidential information, which may cause data leaks.
- Attacks by malware: Viruses and Trojan horses are examples of malware that may enter cloud infrastructure and compro- mise sensitive data.
- Application programming interfaces (APIs) that are insecure might make pri- vate information accessible to outsiders.
- Insufficient encryption makes it possible for unauthorized users to intercept and read data that has not been adequately encrypted.

Data is altered and made "less sensi- tive" before being given to agents in the method known as perturbation [1]. For instance, one can substitute ranges for precise numbers or introduce random noise to particular properties. It's crucial in some circumstances, nevertheless, to keep the data from the original distributor intact. For instance, if we outsource our payroll, the outsourcer has to know the precise salary and client bank account information.

Medical researchers might want pre- cise patient data if they plan to treat pa- tients rather than just compute statistics. Watermarking has historically been used to identify leaks; for example, a special code is included into each disseminated copy. The leaker can be found if that copy is later found in the possession of an unauthorised person. In some circum- stances, watermarks can be quite helpful, but they once again need some alteration of the original material [2]. Also, if the data receiver is hostile, watermarks may occasionally be removed. In this article, we explore covert methods for identify- ing the loss of a collection of items or documents.

In particular, we investigate the fol- lowing case: The distributor hands over a batch of items to agents, but later finds some of those same items in an unap- proved location. (For instance, the infor- mation may be discovered on a website or acquired via a legal discovery proce- dure.)

Due to the extensive product offerings of leading IT security providers, the data leakage detection market is highly diver- sified. a broad range of enabling.

The data leaking problem has been addressed in the past with the use of firewalls, encryption, access control, identity management, machine learning content/context-based detectors, and other technologies. The competitive ad- vantages of creating a "one-stop-shop," all-encompassing data leakage detection suite lie primarily in the effective orches- tration of the aforementioned enabling technologies to deliver the highest level of protection by ensuring an ideal fit of particular data leakage detection technol- ogies with the "threat landscape" in which they operate. The variety of data states, users, and IT platforms, as well as the many types of leakage paths, charac- terise this environment.

The distributor may discontinue doing business with an agent or file legal action if there is "adequate evidence" that the agent leaked data. In this research, a model for judging an agent's "guilt" is developed. We also provide techniques for allocating items to agents in a way that increases the likelihood that a leaker will be found.

ISSN No:-2456-2165

II. PROPOSED METHODOLOGY

To combat the issue of data leakage, watermarking technique is used. It in- cludes embedding a special code that is included into each disseminated copy. The leaker can be found if that copy is later found in the possession of an unau- thorised person. The approach developed here gives the distributor the power to maintain the records of the number of downloads of data by the agents across the cloud environment and then draw up a probability model and that is how a guilty agent can be identified by the dis- tributor.

The waterfall method has evolved into the incremental model. Via a series of incremental builds, the product is de- signed, implemented, integrated, and tested. Several system vendors and for- profit software firms adopt this well- liked approach of programme growth.

Although there may be delays in im- plementation, the incremental software development process may be useful for projects with clearly defined software demands. Early on, you must have the essential software features [3].

III. WATERMARKING MODEL

A method for preserving the copyright of the data owner is watermarking. It is a technique for inserting a special code into every distributor's copy. In essence, it is encryption applied to a particular piece of distributed data.

The information might take the form of a picture, a movie, or any other im- portant file. The watermark helps the business assert its ownership of particular data. This method introduces a minor pattern to the data, mostly tuples and sub- sets.

The tuple and subset attributes are algo- rithmically built to be managed by a private key that is only accessible to the data owner. This image represents the watermark.

A. Applications of Watermarking

Watermarking technique finds it's appli- cation in a variety of situations. A digital watermark is inserted onto a digital sig- nal or picture using the watermarking process. Data concealing, content authen-tication, and copyright protection are just a few uses for this information. The fol- lowing are a few uses for the watermark- ing technique:

- Copyright protection: Water- marking is frequently used to prevent unlawful use or distribu- tion of digital assets, such as photographs, audio files, and video files. The owner may readily identify and trace the content and take legal action against copyright infringement by including a distinctive wa- termark in it.
- Watermarking may be used to authenticate digital material, such as pictures or movies, and make sure that it hasn't been changed or tampered with. In forensic investigations and judi- cial actions, when the integrity of the material is crucial, this is very helpful.
- Watermarking can be used to conceal sensitive information in digital signals like text messages or images. This has a variety of uses, including steganography and digital watermarking.
- Digital forensics: To locate and trace the origin of illicit infor- mation or to confirm the veraci- ty of digital evidence, water- marking is employed in digital forensics.
- Watermarking can be used to trace how digital media, such as pictures, movies, and music, is used and distributed. In the ad- vertising sector, where busi- nesses may monitor how their adverts are being used across numerous media platforms, this is very helpful.Overall, watermarking is a versatile technique that can be used in various applications to protect, authenticate, and track digital content.

B. Techniques of Watermarking System

- **Embedding and Extraction.** To provide a watermark, an insignificant component of the fractional part of the pixel intensity value of the cover image is encoded using this technique. The use of a watermark in the tiny section has contributed to the authenticity of the cover image. The results show that imperceptibility is well preserved. This system has the added benefit of having a large capacity for watermarking.
- Consequently, utilizing this approach, big capacity watermarks may be successfully inserted and removed, which might be tremendously valuable for enterprises creating watermarking applications and digital information security goods. This technique employs embedding and extraction algorithms.
- **Wavelet Based Watermarking.** For embedding, multi- resolution data fusion is performed, in which both the picture and the watermark are translated into the discrete wavelet domain. Each wavelet decomposition level of the host image contains the watermark. The watermark is an average of the estimations from each resolution level of wavelet decomposition during detection. This approach is resistant to JPEG compression, additive noise, and filtering.
- **Robust Watermarking Technique.** In contrast to the LSB technique, the key to making a watermark resilient is to inte-grate it into the image's perceptually relevant components.

A good watermark is one that consid- ers the behavior of the human visual system. A scaling factor can be used to adjust the amount of energy in a water- mark for the spread spectrum-based wa- termarking process. Watermark energy should be robust enough to survive any assaults and distortions. Meanwhile, high watermark energy will have an impact on the visual quality of the watermarked image.

- **Invisible Watermarking Technique.** This technique presents a novel invisible robust watermarking scheme for embed- ding and extracting a digital watermark in an image. The novelty lies in deter- mining a perceptually important sub- image in the host image [4].

The invisible insertion of the water- mark is performed in the most significant region of the host image such that tam- pering with that portion with the inten- tion to remove or destroy will degrade the esthetic quality and value of the im- age. One feature of the algorithm is that this sub-image is used as a region of interest for the watermarking process and eliminates the chance of watermark re- moval.

ISSN No:-2456-2165

IV. GUILTY AGENT DETECTION

Sample (S) and explicit (E) requests are the two categories that are dealt with. Objects created by the distributor that are fake are those that are not part of set T. To boost the likelihood of finding agents who leak data, the objects—which are sent to agents along with T objects—are created to seem like actual things. The figure below represents the distributor exchange of files, where F represents Fake Tuples and F' for legit tuples.

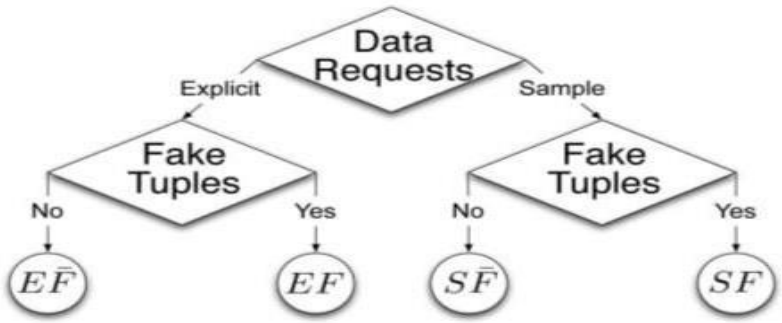

Fig. 1: Data Leakage

In order to increase his ability to iden- tify guilty agents, the distributor may be able to include fake objects in the data that is disseminated. Fake items, howev- er, may affect how agents function cor- rectly, therefore they might not always be permitted. But, in most applications, individual items are disrupted, for exam- ple, by adding random noise to critical salaries or applying a watermark to a picture. The concept of perturbing data to identify leakage is not new. In our situa- tion, introducing fictitious items disturbs the collection of distributor objects. In some situations, faking items could be less problematic than disturbing genuine ones [5].

Consider the scenario where the agents are hospitals and the distributed data objects are medical records. In this situation, even minor alterations to the patient records itself may not be desired. Since no patient matches these data and no one will ever be treated based on fraudulent information, the inclusion of a few bogus medical documents would be permissible. The usage of "trace" records in mailing lists served as inspiration for our use of fake objects.

In this instance, business A offers business B a mailing list to be used just once (e.g., to send advertisements). Business A adds trace data with addresses that are its own. As a result, A receives copies of every mailing sent using the acquired mailing list by firm B. These records are an example of a false item that aids in spotting data misuse. The data that the distributor sends to the agents includes fake objects that he builds and adds. We let the subset of false items that agent Ui gets to be Fi -Ri. So that agents cannot tell false things from real ones, as will be covered below, meticulous fabrication is required. The number of bogus objects that the distribu- tor can produce is frequently constrained. For instance, objects can have email addresses, and each bogus email account might need to have a real inbox created (otherwise, the agent may discover that the object is fake). The distributor may really keep an eye on the inboxes; if email starts to arrive from a source other than the agent who was given the ad- dress, it's clear that the information was leaked. The distributor can put a cap on the number of bogus items since setting up and maintaining email accounts uses resources. If a limit exists, we identify it using B fake objects.

Similar to this, the distributor could wish to restrict the quantity of false items that each agent receives in order to avoid raising red flags and to avoid impairing the agents' performance. As a result, we state that the distributor may deliver agent Ui up to two bogus objects.

Data distribution via the distributor has one purpose and one limitation. The distributor's restriction is to fulfil agents' demands, either by giving them the exact number of objects they ask for or by giving them all of the objects that are available and meet their requirements. His goal is to be able to identify any agent who releases any information about him. We view the restriction as being severe. The distributor is not allowed to refuse a request from an agent or provide them several disturbed versions of the same object. The only way to loosen the limitation is what we refer to as fake object distribution. Our detection goal is both ideal and unachievable. Only if the distributor withheld all data objects from any agent would detection be guaranteed (consider that in order to get "perfect" privacy and security, we must forego utility). Instead, we employ the goal of increasing the likelihood of finding a guilty agent who leaks all of his data objects.

A. Data Allocation Problem

The main focus of this project is the data allocation problem: how can the distribu- tor "intelligently" give data to agents in order to improve the chances of detecting a guilty agent?

- The distributor will transmit agents' data that has been encrypted with public and private keys; any agent wishing to download the data must first log in to the system and do so.
- To open the file in software because it is encrypted, the agent must input the secret key.
- The downloaded file can only be seen using the program; it cannot be accessed using the file system.

B. Optimization Module

Data distribution via the distributor has one purpose and one limitation. The dis- tributor's restriction is to fulfil agents' demands, either by giving them the exact number of objects

they ask for or by giving them all of the objects that are available and meet their requirements. His goal is to be able to track down any agent who releases any of his data and lock the file to prevent further distribu- tion or leaks [6].

C. Optimization Module

If we are aware that the material was leaked, we must determine the likelihood that the agent is guilty.

V. IMPLEMENTATION AND ALGORITHMS

A. Key Operations

- Distributor sends file
- Agent can download file
- Agent can request
- Distributor finds the probability
- Admin or distributor can block the guilty agent

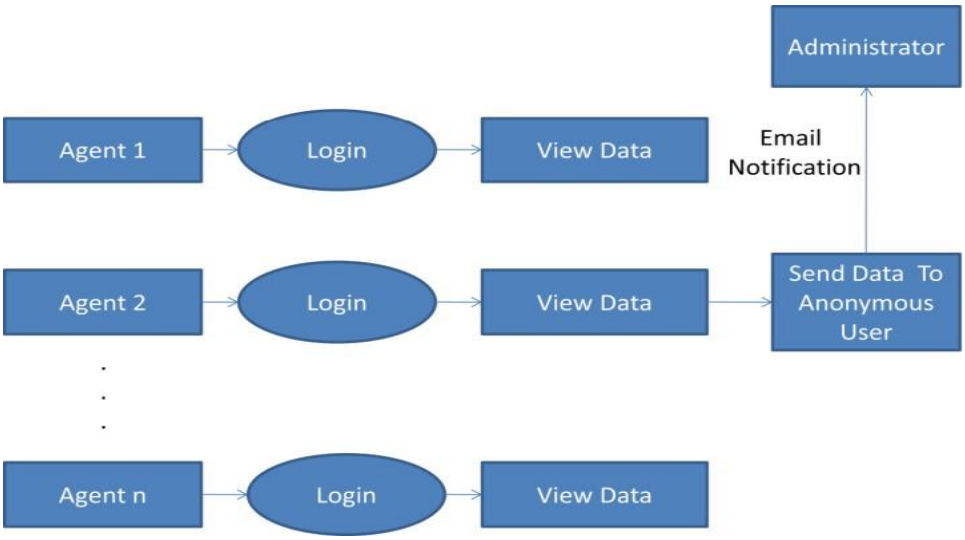

Fig. 2: Data Leakage and Detection

B. Terminologies

- DLD is the system such that DLD={A,D,T,U,R,S,U*,C,M,F}.
- {A} is the Administrator who controls entire operations performed in the Soft- ware
- {D} is the Distributor who will send data T to different agents U.
- T is the set of data object that are sup- plied to agents. T can be of any type and size, e.g., they could be tuples in a rela- tion, or relations in a database. T= {t1 ,t2
- ,t3 ,...tn }
- U is the set of Agents who will receive the data from the distributor A U={u1 ,u2
- ,u3 ,...un }
- R is the record set of Data objects which is sent to agents R = {t1,t3 ,t5..tm} R is a Subset of T
- S is the record set of data objects which are leaked. S={t1 ,t3 ,t5..tm} S is a Subset of T
- U* is the set of all agents which may have leaked the data U*={u1 ,u3 ,...um} U* is a subset of U
- C is the set of conditions which will be given by the agents to the distributor. C={cond1 ,cond2 ,cond3 ,...,condn }
- M is set of data objects to be send in Sample Data Request algorithm M={m1
- m2 ,m3 ,...,mn }

C. Sample Request Algorithm

Input: - m1,m2…….mN ,|T| As- suming mi<=|T|
Output: - R1,R2…..Rn

- 1: a ← 0|T| a[k]:number of agents who have received object tk
- 2: R1 ← ∅, . . . , Rn ← ∅ 3: remaining ← $\sum_{i=1}^{n} m$ 4: while remaining > 0 do
- 5: for all i = 1, . . . , n : |Ri| < mi do 6: k ← SELECTOBJECT(i,Ri)
- May also use additional Parameters 7: Ri ← Ri ∪ {tk}
- 8: a[k] ← a[k] + 1
- 9: remaining ← remaining – 1

Thus we get the set of data objects (R1,R2….Rn) to be send to the particular Agent.(U1,U2…..Un). In lines 1 and 2 of Algorithm 4, the main loop in lines 4-9 is executed while there are still data objects (remaining > 0) to be allocated to agents.

In each iteration of this loop (lines 5- 9), the algorithm uses function SELECTOBJECT() to find a random object to allocate to agent Ui. This loop iterates over all agents who have not received the number of data objects they have requested.

D. Explicit Request Algorithm

Input:-R1, . . .,Rn ;cond1,. . . , condn ; b1, . . . , bn, B
Output:- R1, . . .,Rn ; F1, . . . ,Fn

- 1: R ← ∅ Agents that can receive fake objects
- 2: for i = 1, . . . , n do 3: if bi > 0 then
- 4: R ← R ∪ {i}
- 5: Fi ← ∅
- 6: while B > 0 do
- 7: i ← SELECTAGENT(R,R1, . . . , Rn)
- 8: f ← CREATEFAKEOBJECT(Ri, Fi,
- condi)
- 9: Ri ← Ri ∪ {f}
- 10: Fi ← Fi ∪ {f} 11: bi ← bi − 1
- 12: if bi = 0 then 13: R ← R\{Ri}
- 14: B ← B – 1

ISSN No:-2456-2165

Thus we get the record set (R1,R2….Rn) for the particular condi- tion(cond1….condn) which has been given by the agents(U1,U2….Un).

In lines 1-5, Algorithm 1 finds agents that are eligible to receiving fake objects in O(n) time. Then, in the main loop in lines 6-14, the algorithm creates one fake object in every iteration and allocates it to random agent. The main loop takes O(B) time. Hence, the running time of the algorithm is O(n + B).

E. Encryptio Algorithm (AES Algorithm)

- *Key Expansion*
  Round keys are de- rived from the cipher key using Rijndael's key schedule.

- *Initial Round*
- Add Round Key—each byte of the state is combined with the round key using bitwise XOR.

- *Rounds*
- Sub Bytes—a non-linear substitution step where each byte is replaced with another according to a lookup table.
- Shift Rows—a transposition step where each row of the state is shifted cyclically a certain number of steps.
- Mix Columns—a mixing operation which operates on the columns of the state, combining the four bytes in each column.
- Add Round Key

- *Final Round (no MixColumns)*
- SubBytes
- ShiftRows
- AddRoundKey

VI. CONCLUSION

We can identify and prevent data leakage by employing algorithms and techniques developed from the research of data leak- age. In an ideal world, there would be no need to give sensitive information to agents who may inadvertently or pur-posefully disclose it. Even if we had to pass up sensitive information, in an ideal world, each thing would be watermarked, allowing us to track its origins with full confidence.

ACKNOWLEDGEMENT

We gratefully acknowledge the numerous informative conversations we had with Ms. Charu Gupta, Assistant Professor, Information and Technology Department at Indira Gandhi Delhi Technical Univer- sity for Women (IGDTUW) as we devel- oped the concepts in this work.

REFERENCES

[1.] Kumar, R., 2017. Data Leakage Detection. Global Journal of Computer Science and Technolo- gy, 17(E4), pp.13-20.
[2.] Papadimitriou, P. and Garcia- Molina, H., 2010. Data leakage detection. IEEE Transactions on knowledge and data engi- neering, 23(1), pp.51-63.
[3.] Vaidya, C., Khobragade, P. and Golghate, A., 2016. Data Leakage Detection and Security in Cloud Computing. GRD Journals-Global Research Development Journal for Engineering, 1(12), pp.137-140.
[4.] Data leakage detection - acad- publ.eu (no date). Available at: https://acadpubl.eu/jsi/2018-119-10/articles/10b/50.pdf (Accessed: March 6, 2023).
[5.] IOSR Journal of Computer Engi- neering (IOSR-JCE) e-ISSN: 2278-0661, p- ISSN: 2278-
[6.] 8727Volume 8, Issue 4 (Jan. - Feb. 2013), PP 79-
[7.] 84/www.iosrjournals.org
[8.] Papadimitriou, P. and Garcia- Molina, H., 2010. Data leakage detection. IEEE Transactions on knowledge and data engi- neering, 23(1), pp.51-63.