International Journal of Advanced Science and Technology
Vol.135 (2020), pp.49-62
http://dx.doi.org/10.33832/ijast.2020.135.05

A FRAMEWORK FOR MALICIOUS AGENT DETECTION IN CLOUD COMPUTING ENVIRONMENT

Ishu Gupta[1*] and Ashutosh Kumar Singh[2]

Department of Computer Applications, National Institute of Technology, Kurukshetra, Haryana, India
[1]ishugupta23@gmail.com, [2]ashutosh@nitkkr.ac.in

***Abstract*— Data security is one of the major challenges encountered by cloud computing. The cloud data is shared among multiple entities which can be intentionally or unintentionally revealed out by any agent to the unauthorized recipient. Therefore, it has become a necessity to detect the malicious agent for protecting shared information. In this regard, we present a framework based on the probabilistic estimation that identifies the malicious agent for minimizing the likelihood of further leakage. In the proposed model, the data is distributed among multiple agents and the allocation is performed using 2-level trees. The parameters based on probability theory are computed for malicious agent identification when the data is leaked by any agent. The experimental results achieved average probability, average success rate, and detection rate up to 1, 0.98, 0.76 respectively for the various number of agents.**

***Keywords*— Data Leakage, Information Security, Leaker Detection, Threat Model, Probabilistic Approach**

1. INTRODUCTION

Cloud computing is gaining popularity among industries, academia, government and business communities due to its on demand facilities. It provides hardware infrastructure and application oriented software on pay per use basis which significantly minimizes the capital investment. The key features of cloud computing are flexibility, disaster recovery, automatic software and hardware updates, work from anywhere, security and competitiveness etc. [1-3]. Because of these features, around 77% of the enterprises are using the cloud services all over the world for various applications [4]. Data sharing in the cloud environment is an essential step among various stakeholders to elevate their business performances. However, the users or agents who receive the data for different purposes may misuse or leak this data that can cause a heavy loss to the various enterprises in terms of finance and reputation [5]. As a consequence, data leakage has become a major challenge in securing the cloud data [6-8]. According to a survey conducted by Gemalto breach level index, 4.5 billion records have been exposed worldwide due to 945 data breaches in the first half of 2018 which shows a rise of 133% compared to the same period in previous year [9].

The methods that deal with the malicious agent detection can be classified into two broad categories (i) watermarking (ii) probability based approaches. In watermarking [10-15] based methods, before handing over the data (text, image, audio, video, relational etc.) to

Received: August 21, 2019
Reviewed: January 12, 2020
Accepted: February 12, 2020
* Corresponding Author

ISSN: 2207-6360 IJAST

the agent, a secret code is embedded in the transferable document using watermark embedding process. If the critical data leaked by the agent is discovered from unauthorized place; malicious agent is detected by extracting the embedded watermark from the leaked document using watermark extraction process. But, if the watermarked is destroyed or tempered in the embedded data by malicious agent, then the leaker can't be identified [16]. However, the probability method is not affected by the aforementioned problem raised in watermarking based approaches and identify the leaker on the basis of allocated data. Different methods [16-20] has been proposed to distribute the data among agents during allocation process and several parameters are evaluated for the guilty agent identification. The proposed model allocates the shared data using the 2-level trees and detects the malicious agent responsible for leakage.

Rest of the paper is organized as follows: Section 2 provides the brief view of the proposed model along with basic definitions, symbols and threat model. Section 3 and 4 discuss about the data allocation and malicious agent detection in detail. The performance is supported by the numerous experimental analysis in Section 5 followed by conclusion in Section 6.

2. PROPOSED MODEL

The threat model used in the proposed framework consists three different entities data owner O^D, cloud server C^S, and agent A_j where data confidentiality is considered as the most serious threat. The entity agent is considered as an attacker and the objective of the model is to identify the malicious agent that leaks the data to the unauthorized third party. In addition to this, the action can be taken against the guilty party. Following are the basic definitions used for the proposed model:

Data Owner
A data owner O^D possesses the data $D = \{D_1, D_2, \ldots Dn\}$ to be stored in the cloud server C^S and is accountable for the distribution of D. O^D has to allocate D among various agents $A = A_1, A_2, \ldots, A_m\}$ with the intention that data must not be leaked by the agent. It is considered as a trusted party in the model.

Agents
An agent A_j acquires the data $Y_j \subseteq \mathcal{D}$ by requesting it from O^D and carry out required tasks via utilizing Y_j. This entity is intended as an untrusted party in the model.

Data Leakage
Accidentally or advertently disclosure of private or sensitive information to the unauthorized party is termed as data leakage. The sensitive or private data in the enterprises include financial, personal, medical information etc.

Malicious Agent
The entity is named as malicious agent M_A that leaked the data $D^L \subseteq D$ from its allocated dataset Y_j to the unauthorized third party which can misuse D^L.

Third Party
An entity that maliciously obtains the data and may maltreat is named as a third party. It belongs to the model in an indirect manner.

Cloud Server
The entity having considerable D to be stored that is provisioned by O^D named as cloud server C^S. The stored data is shared among $A_j \, \forall \, j = 1, 2, \ldots, m$; on their demands.

The complete process for malicious agent M_A detection is shown in Fig. 1. The requests are obtained from multiple agents $A_j; j = \{1, 2, \ldots, m\}$ and the data owner O^D allocates the data to the agent A_j The data allocation is performed using the 2-level trees

$T_1(A, D, E)$ and $T_2(D, A, E)$ in the model. If intended A_j leaks the allocated data Y_j either intentionally or unintentionally, then responsible M_A is to be recognized by applying the detection mechanism. The proposed model has two major steps: (i) data allocation and (ii) probability evaluation which are discussed in subsequent sections in detail. All the used notations and their corresponding descriptions are tabulated in Table I.

Table I.

Notation	Description
O^D	Data owner
D_i	Data object
A_j	An agent
Y_j	Allocated dataset to agent A_j
M_A	Malicious agent
D^L	Leaked dataset
C^S	Cloud server
$T(U,V,E)$	2-level tree
D_i'	Encrypted document
Δ_{A_j}	Total number of request fulfilled of agent A_j
X_i	Agent set among which D_i is distributed
Ω_{D_i}	Total number of agents having D_i in Y_j
K	Key used for encryption
M_{A_j}	the event that agent A_j is malicious
Θ	Probability of guessing
$\bar{\Psi}^*$	Average success rate
$min\,\Psi^*$	Detection rate

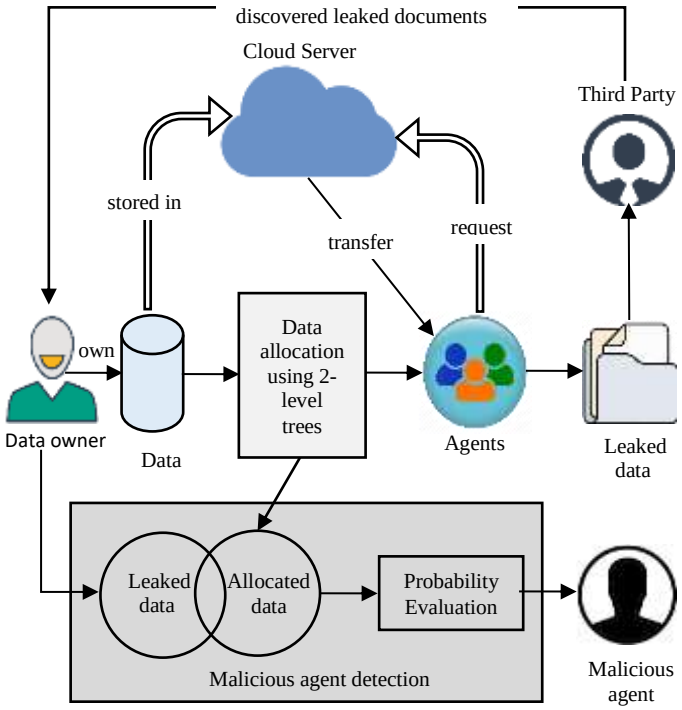


Fig. 1. Proposed Framework

3. DATA ALLOCATION

Agent A_j makes the request to C^S for required document D_i; $i \in \{1,\ 2,\ \ldots,\ n\}$. O^D verifies the previous record of A_j as well as availability of D_i in C^S. Once, A_j is found trustworthy and D_i is available, O^D encrypts D_i to ensure the more security to it and delivers the generated encrypted document D_i' to A_j. Similarly, the request of all $A_j \in A$ is fulfilled by O^D after examining their malicious record and availability of required $D_i \in D$ retained in the database.

Data allocation among various agents $A_j \in A$ is implemented using 2-level Trees T_1 and T_2. $T(U,\ V,\ E)$ is a 2-level tree where U is the root node, V represents the leaf nodes and E is the set of edges exist from U to V. An edge $e \in E$ exists between the two nodes iff T remains circuit free and connected. $T_1(A_j,\ D',\ E_1)$ provides the dataset $Y_j = D' \subseteq D$ received by $A_j \in A$ as shown in Fig. 2(a). This figure shows that an edge exists between A_j and D_i i.e. $e_{j,i} \in E_1$ iff A_j obtains the required D_i. The total number of requests accomplished of an agent $A_j \in A$ is given by $\Delta_{A_j} = \deg(A_j)$. $T_2(D_i,\ A',\ E_2)$ in Fig. 2(b) provides the allocation of $D_i \in D$ among the agents $A' \subseteq A$. There exists an edge between data D_i and an agent A_j i.e. $e_{i,j} \in E_2$ iff data object D_i is allocated to the agent A_j. $\mathrm{X}_i = A' \subseteq A$ is obtained from T_1 which is the set of agents having D_i in their allocated dataset Y_j. $\Omega_{D_i} = \deg(D_i)$ gives the total number of agents to whom data object D_i is allocated.

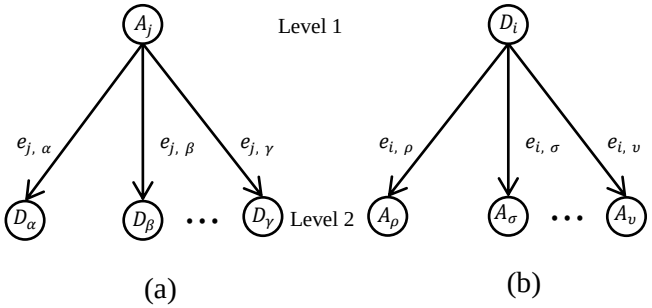


Fig. 2. 2-Level tree (a) $T_1(A_j,\ D',\ E_1)$ (b) $T_2(D_i,\ A',\ E_2)$

Example 1

Let the model consists of seven data objects $D = \{D_1,\ D_2,\ D_3,\ D_4,\ D_5,\ D_6,\ D_7\}$ and four agents $A = \{A_1,\ A_2,\ A_3,\ A_4\}$. The agents $A_j \forall\ j = 1, 2, 3, 4$; submit their requests for the required data objects to C^S. The data is distributed among agents as per their requests and the allocation is accomplished via T_1 and T_2. The allocated dataset to the agents $A_1,\ A_2,\ A_3,\ A_4$ are:

$Y_1 = \{D_2,\ D_4,\ D_5,\ D_6,\ D_7\}$,
$Y_2 = \{D_2,\ D_3,\ D_5,\ D_6\}$,
$Y_3 = \{D_1,\ D_3,\ D_5\}$,
$Y_4 = \{D_1,\ D_2,\ D_3,\ D_4,\ D_6,\ D_7\}$

The allocation of $D_1,\ D_2,\ D_3,\ D_4,\ D_5,\ D_6,\ D_7$ among the agents are:

$X_1 = \{A_3,\ A_4\}$,
$X_2 = \{A_1,\ A_2,\ A_4\}$,
$X_3 = \{A_2,\ A_3,\ A_4\}$,
$X_4 = \{A_1,\ A_4\}$,
$X_5 = \{A_1,\ A_2,\ A_3\}$,
$X_6 = \{A_1,\ A_2,\ A_4\}$,
$X_7 = \{A_1,\ A_4\}$

The request fulfilled of the agents A_1, A_2, A_3, A_4 are $\Delta_{A_1} = 5$, $\Delta_{A_2} = 4$, $\Delta_{A_3} = 3$ and $\Delta_{A_4} = 6$ respectively. The total number of agents to whom D_1, D_2, D_3, D_4, D_5, D_6, D_7 has given are $\Omega_{D_1} = 2$, $\Omega_{D_2} = 3$, $\Omega_{D_3} = 3$, $\Omega_{D_4} = 2$, $\Omega_{D_5} = 3$, $\Omega_{D_6} = 3$ and $\Omega_{D_7} = 2$ respectively. The pseudo code for data distribution is shown in Algorithm 1.

Algorithm 1 Data Allocation Algorithm

Input:

$C^S \leftarrow A_j(R(D_i))$ //multiple requests for $D_i \in D$ by various $A_j \in A$

Output:

$Y_j, \Delta_{A_j} \forall$ j $= \{1, 2, \ldots, m\}$

$X_i, \Omega_{D_i} \forall$ i $= \{1, 2, \ldots, n\}$

Begin
Initialize $T_1(A_j, D, E_1) : E_1 \leftarrow \emptyset$ and $T_2(D_i, A, E_2) : E_2 \leftarrow \emptyset$ //2-level tree
for i $= 1, 2, \ldots, n$ **do**
    $X_i \leftarrow \emptyset$
    $\Omega_{D_i} \leftarrow 0$
end for
for j $= 1, 2, \ldots, m$ **do**
    $Y_j \leftarrow \emptyset$
    $\Delta_{A_j} \leftarrow 0$
end for
while $!R$ **do**
    if $D_i \in DB$ **then** // DB −database
        if $e_{j,i} \notin E_1 \wedge e_{i,j} \notin E_2$ **then**
            $D_i \leftarrow select(R(D_i)) \in DB$
            $D_i' \leftarrow encrypt('K', 'D_i')$
            $e_{j,i} \in E_1$
            $e_{i,j} \in E_2$
            $A_j \leftarrow transfer(D_i')$
            $X_i \leftarrow X_i \cup \{A_j\}$
            $Y_j \leftarrow Y_j \cup \{D_i\}$
        end if
    end if
end while
for $i = 1, 2, \ldots, n$ **do**
    $\Omega_{D_i} = \deg(D_i)$
end for
for $j = 1, 2, \ldots, m$ **do**
    $\Delta_{A_j} = \deg(A_j)$
end for
end

4. MALICIOUS AGENT DETECTION

Let $M_{A_j}$is the event that agent A_j is the malicious agent M_A. In our model, there are two possibilities for obtaining the data by target t: (i) any single agent from the set X_i has leaked object D_i to t (ii) t retrieved the data object D_i by guess or through any other mean without intervention of any agent A_j. The probability to leak any data object D_i is equal $\forall A_j \in X_i$ if it is leaked by any agent $A_j \in X_i$, otherwise probability is Θ if it is obtained by t. It is

considered that decision of A_j to leak any data object D_i is autonomous to the leaking of other data object $D_{i'}$ $\forall D_i, D_{i'} \in D^L$ where $D_i \neq D_{i'}$. For the given leaked dataset D^L, the conditional probability $P\{M_{A_j} \mid D^L\}$ of the agent A_j to be malicious is computed by Eq. (1) where Θ is the probability of guessing the data object D_i and Ω_{D_i} signifies the number of A_j to whom the object D_i has been alloacted.

$$P\{M_{A_j} \mid D^L\} = 1 - \prod_{\forall D_i \in (D^L \cap \Upsilon_j)} \left(1 - \frac{(1-\Theta)}{\Omega_{D_i}}\right) \quad (1)$$

If A_j leaks all the data objects from its allocated set Υ_j such that $D^L = \Upsilon_j$, the probability $P\{M_{A_j} \mid \Upsilon_j\}$ of A_j for being M_A is computed. Difference function $\Psi^*_{(j,k)}(M_A)$ given in Eq. (2) maximizes the possibility of identifying M_A which is obtained in the form of a $m \times m$ matrix.

$$\Psi^*_{(j,k)}(M_A) = P\{M_{A_j} \mid \Upsilon_j\} - P\{M_{A_k} \mid \Upsilon_j\} \quad \forall j,k = \{1, 2, \ldots, m\} \quad (2)$$

To evaluate and analyze the performance of the proposed approach, two parameters $\bar{\Psi}^*$ and $min\ \Psi^*$ are calculated with the help of the matrix $\Psi^*_{(j,k)}(M_A)$ in Eq. (3) and Eq. (4) respectively. $\bar{\Psi}^*$ is the average success rate which is computed by taking the average of all the entries of the matrix $\Psi^*_{(j,k)}(M_A)$. $min\ \Psi^*$ renders the detection rate which represents the minimum entry of the matrix $\Psi^*_{(j,k)}(M_A)$. Algorithm 2 depicts the pseudo code for probability computation.

$$\bar{\Psi}^* = \frac{\sum_{\substack{j,k = \{1,2,\ldots,m\} \\ j \neq k}} \Psi^*_{(j,k)}(M_A)}{m(m-1)} \quad (3)$$

$$min\ \Psi^* = \min_{\substack{j,k = \{1,2,\ldots,m\} \\ j \neq k}} \Psi^*_{(j,k)}(M_A) \quad (4)$$

Algorithm 2 Probability Computation Algorithm

Input: D^L
Output: M_A detection

```
Begin
for j = 1, 2, . . . , m do
Find average P{M_Aj | D^L} and average P{M_Aj | Υ_j}   using (1)
end for
for all j = 1, 2, . . . , m do
    for all k = 1, 2, . . . , m do
        Compute Ψ*_(j,k)(M_A) using (2)
    end for
end for
Compute Ψ̄* and min Ψ* by employing (3) and (4) respectively
Detect M_A
end
```

Let the leaked dataset $D^L = \{D_2, D_4, D_5, D_6, D_7\}$ is found at the unauthorized place in *example* 1, the probability of the agents A_1, A_2, A_3, A_4 are computed using Eq. (1), to estimate M_A for $\Theta = 0.1$. The values of computed probabilities are:

$P\{M_{A_1} \mid D^L\} = 0.8962,$
$P\{M_{A_2} \mid D^L\} = 0.657,$
$P\{M_{A_3} \mid D^L\} = 0.3,$
$P\{M_{A_4} \mid D^L\} = 0.8518$

The probability of the agents A_1, A_2, A_3, A_4 to leak their dataset Y_1, Y_2, Y_3, Y_4 respectively are computed and given as:

$P\{M_{A_1} \mid Y_1\} = 0.8962,$
$P\{M_{A_1} \mid Y_1\} = 0.7599,$
$P\{M_{A_3} \mid Y_3\} = 0.7305,$
$P\{M_{A_4} \mid Y_4\} = 0.9429$

$\Psi^*_{(j,k)}(M_A)$ is evaluated for all $j, k = \{1, 2, 3, 4\}$ using Eq. (2) as depicted in Matrix Ξ. Furthermore $\bar{\Psi}^* = 0.2344$ and $min\ \Psi^* = 0.0444$ are calculated using Eq. (3) and Eq. (4) respectively.

$$\Xi = \begin{pmatrix} 0 & 0.2392 & 0.5962 & 0.0444 \\ 0.1029 & 0 & 0.2499 & 0.1029 \\ 0.4305 & 0.2205 & 0 & 0.1155 \\ 0.0915 & 0.28593 & 0.32793 & 0 \end{pmatrix}$$

5. PERFORMANCE EVALUATION

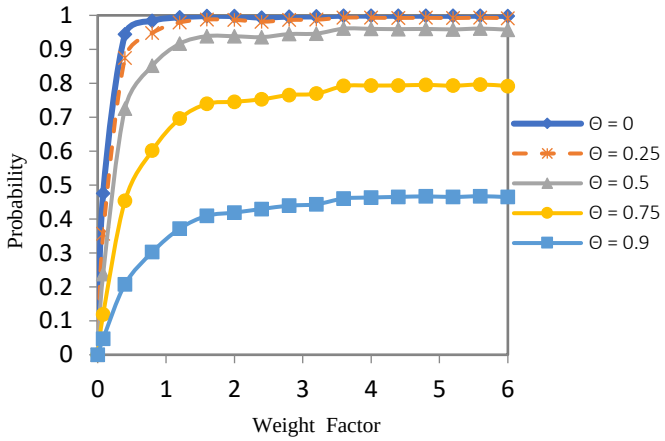
The experiments are conducted using C/C++ on a machine equipped with Intel® coreTM I5 processor with 2.60 GHz clock speed and 8 GB RAM. To analyze the performance, $|D| = 500$ is considered while the numbers of agents vary for different circumstances. The performance of three parameters: (i) probabilities for M_A detection (ii) average success rate (iii) detection rate have been evaluated against the weight factor W_F. Weight factor can be defined as the ratio of requests fulfilled of all the agents to the total number of data objects as given in Eq. (5).

$$W_F = \frac{\sum_{j=1}^{m} \Delta_{A_j}}{|D|} \tag{5}$$

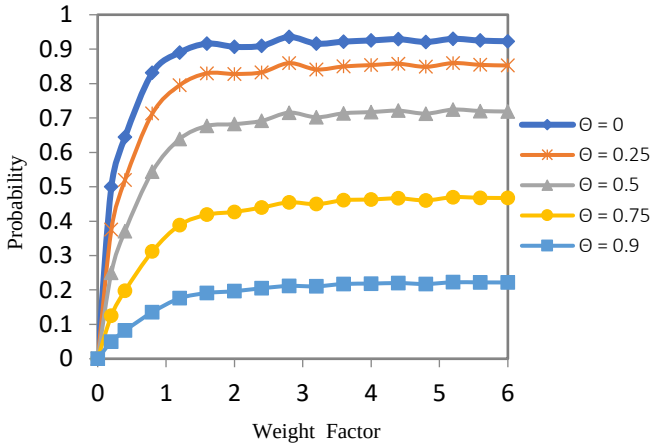
The value of W_F varies between 1 to 6 throughout the experiments. There are two possible ways to alter W_F: (i) number of agents are unchanged but their request size change (ii) number of agents change with their fixed request size. Furthermore, the number of requests can be same or differ for all the agents. Therefore, three circumstances are arisen for the experimental analysis: (a) number of agents are fixed i.e. $|A| = 40$ and request size changes in the range $(1 - 75)$ for different scenario while considering the same number of requests of all the agents (b) fixed number of agents i.e. $|A| = 100$ with same request size in every scenario in the range $(1 - 30)$ (c) number of agents vary from 2 to 80 in all scenarios with its different request from the range $(30 - 50)$.

5.1. AVERAGE PROBABILITY

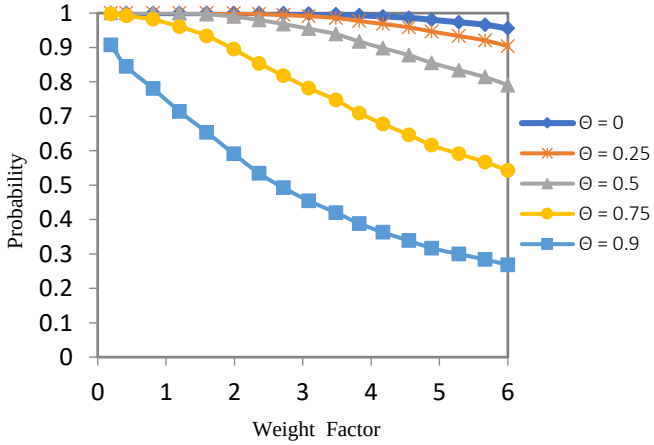
Fig. 3 shows the probability $P\{M_{A_j} \mid D^L\}$ for all three aforementioned circumstances when the leaked data set is given, where 250 data objects have been taken in D^L for $\Theta = 0, 0.25, 0.5, 0.75, 0.9$. The distinct values of Θ and W_F provide the different probabilities to identify M_A. It can be observed from Fig. 3(a, b) that probability increases with increment in W_F, but decreases with increment in Θ. Furthermore, we notice that the probability to detect M_A decays when the number of agents raise. However, in Fig 3(c), the probability reduces with augmentation in both W_F and Θ.



(a)



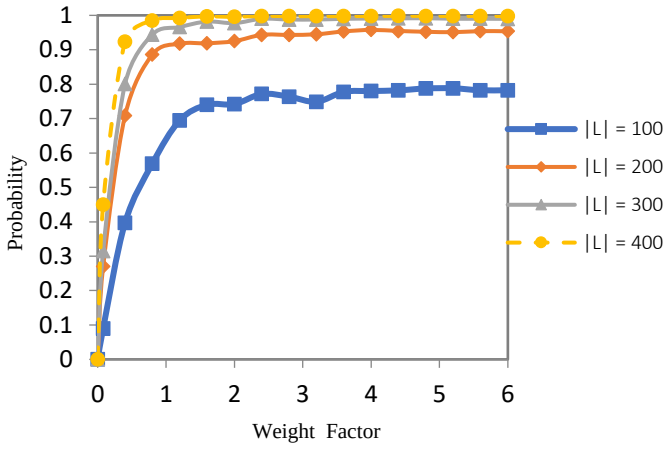
(b)



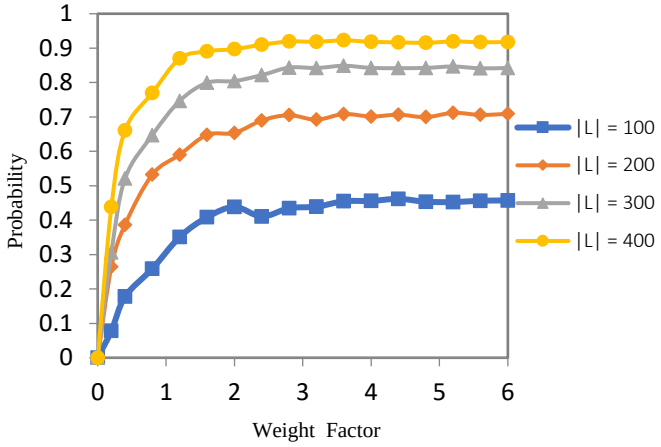
(c)

Fig. 3. Evaluation of $P\{M_{A_j} \mid D^L\}$ at $|L| = 250$ (a) $|A| = 40$ (b) $|A| = 100$ (c) $|A| = (2 - 80)$

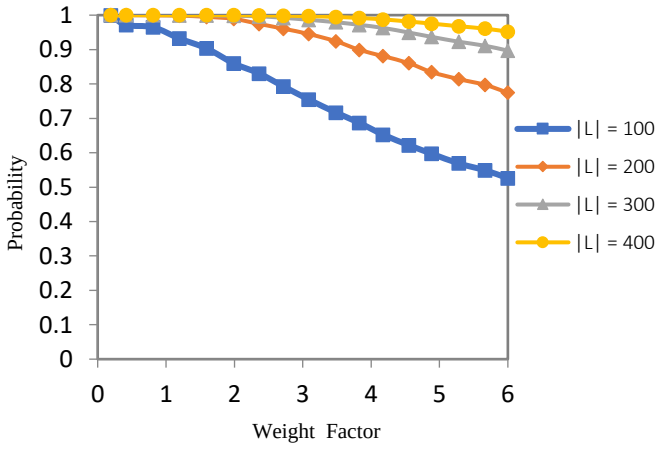
Fig. 4 shows the probability $P\{M_{A_j} \mid D^L\}$ when the data in D^L changes for the fixed value of Θ. For all three aforementioned circumstances, $|L| = \{100,\ 200,\ 300,\ 400\}$ and $\Theta = 0.4$ has been considered. In this figure, the probability enhances with increment of data in D^L.



(a)



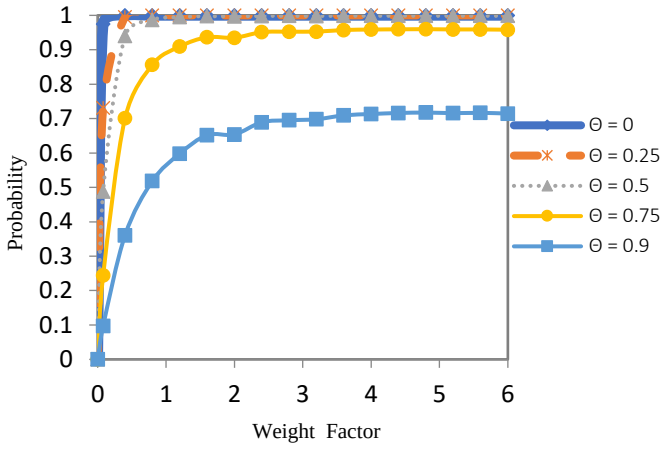
(b)



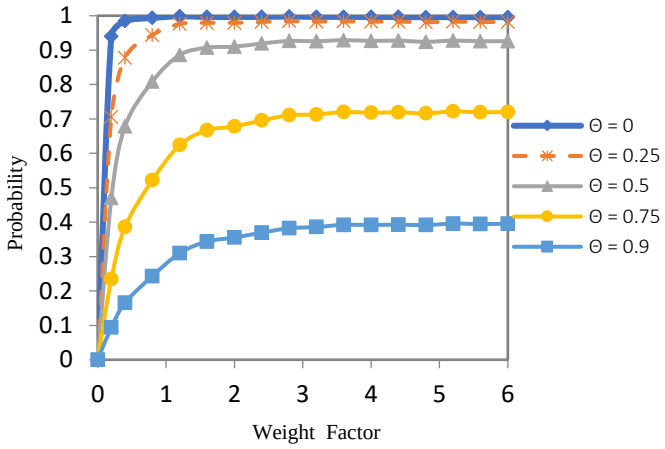
(c)

Fig. 4. Evaluation of $P\{M_{A_j} \mid D^L\}$ at $\Theta = 0.4$ (a) $|A| = 40$ (b) $|A| = 100$ (c) $|A| = (2 - 80)$

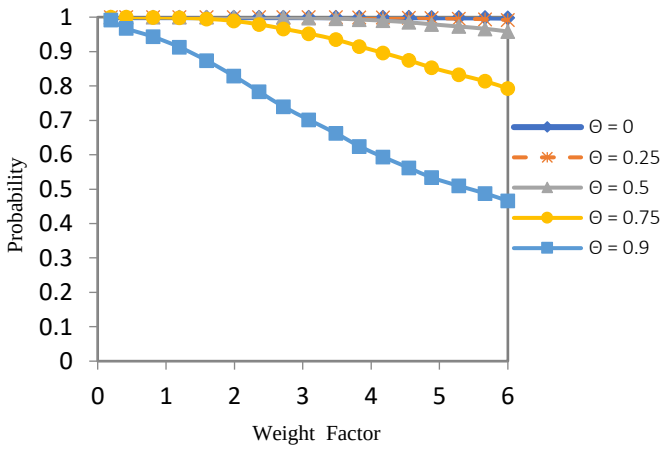
Fig. 5 represents the average probability $P\{\mathrm{M}_{\mathrm{A_j}} \mid Y_j\}$ when all the allocated dataset Y_j is leaked by all the agents. The probability to identify the malicious agent is high even for large value of W_F and Θ.



(a)



(b)



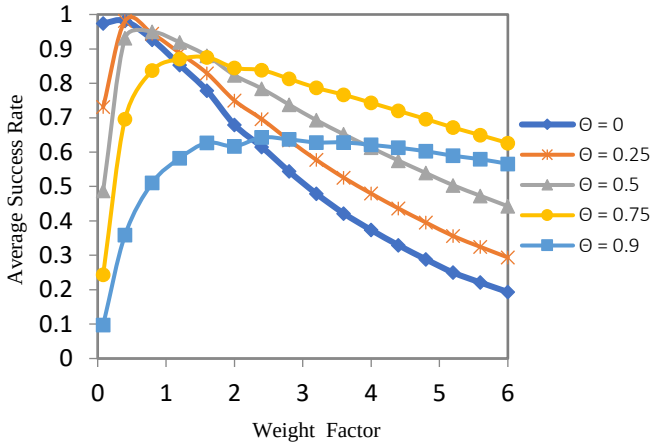
(c)

Fig. 5. Evaluation of $P\{M_{A_j} \mid Y_j\}$ (a) $|A| = 40$ (b) $|A| = 100$ (c) $|A| = (2 - 80)$

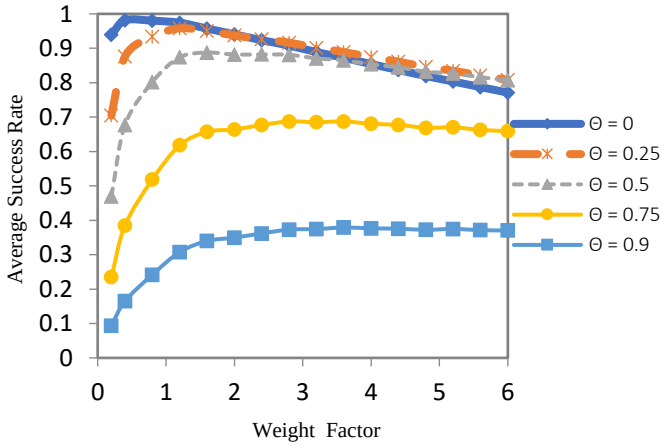
From Fig. (3-5), probability decays with increment of Θ in all the circumstances as chances of guessing the data increases. For the fixed number of agents in the model, probability rises with increment in W_F as overlapping of allocated dataset minimizes. When the data in leaked dataset increases (Fig. (4, 5)), the probability of detecting M_A also escalates as we obtain more evidences against M_A.

5.2. AVERAGE SUCCESS RATE

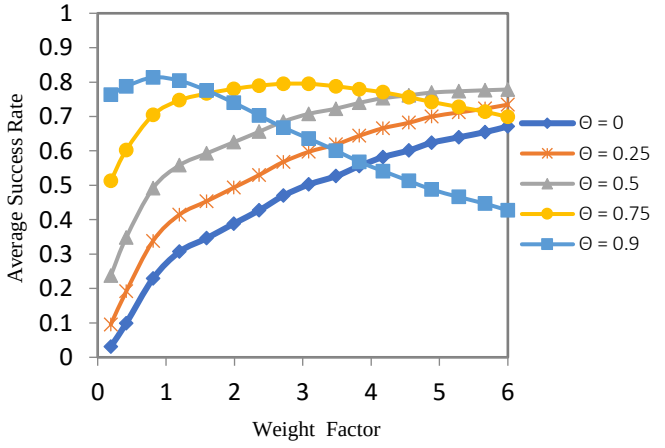
Fig. 6 portrays the parameter $\bar{\Psi}^*$ with respect to Θ and W_F. In Fig. 6 (a, b), it is noticed that $\bar{\Psi}^*$ initially increases and then decreases or remains constant with respect to W_F. Moreover, for larger number of agents, $\bar{\Psi}^*$ is lesser as depicted in Fig. 6 (b). In Fig. 6 (c), $\bar{\Psi}^*$ increases initially and then it starts decreasing with enhancement in W_F, but it increases for the lower value of Θ.



(a)



(b)



(c)

Fig. 6. Evaluation of $\bar{\Psi}^*$ (a) $|A| = 40$ (b) $|A| = 100$ (c) $|A| = (2 - 80)$

5.3. DETECTION RATE

Fig. 7 depicts the parameter $min\ \Psi^*$ with respect to W_F for different values of Θ. In Fig. 7 (a), the detection rate decreases for $\Theta = 0,\ 0.25,\ 0.5$ and increases or almost constant for $\Theta = 0.75,\ 0.9$ with increment in W_F. However, $min\ \Psi^*$ keeps increasing for all Θ in Fig. 7 (b) because of large number of agents. In the last case (Fig. 7 (c)), $min\ \Psi^*$ escalates with increment in Θ except $\Theta = 0.9$.

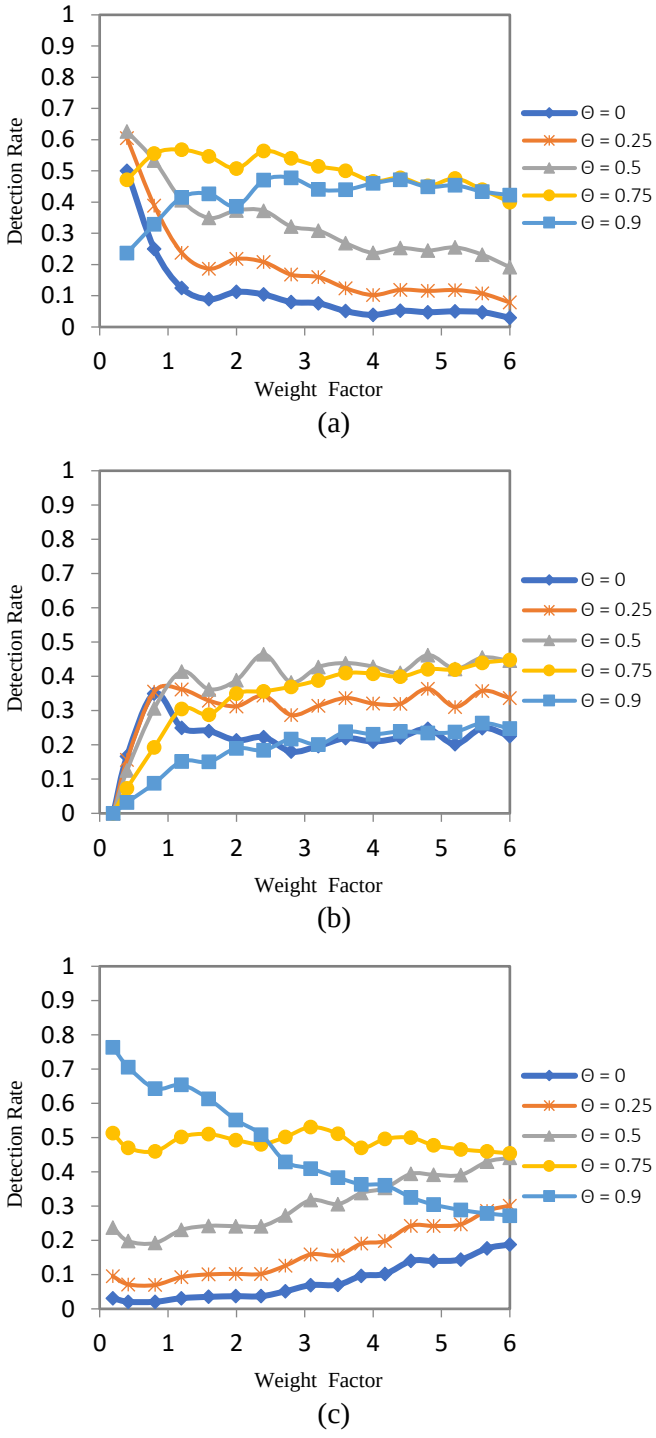


Fig. 7. Evaluation of $min\ \Psi^*$ (a) $|A| = 40$ (b) $|A| = 100$ (c) $|A| = (2 - 80)$

In Fig. (6-7), $\bar{\Psi}^*$ and $\min \Psi^*$ increase initially and then decrease with increment in both W_F and Θ. Because the data overlapping minimizes initially and after a threshold it starts increasing during the allocation of dataset. It is also noticed that $\Theta = 0.9$ shows distinct nature as chances of guessing the data are high instead of leaking it.

6. CONCLUSION

A malicious agent identification in cloud environment by exploiting the property of probability theory is discussed in this paper. Proposed model used the 2-level trees for the data allocation among various agents. These trees are utilized for the computation of various parameters for detection of guilty agent. The probability of detecting malicious agent, average success rate and detection rate are high even though weight factor and probability of guessing are high that shows the efficiency of model. Future efforts could be made to improve the security of the most sensitive information via considering the threshold value and complex threat model.

REFERENCES

[1] Buyya, R., Broberg, J., and Goscinski, A., "Cloud Computing: Principles and Paradigms", vol. 87, John Wiley & Sons, 2010.
[2] Gupta, I. and Singh, A. K., "An Integrated Approach for Data Leaker Detection in Cloud Environment", Journal of Information Science and Engineering, 2019.
[3] Gupta, I. and Singh, A. K., "A Confidentiality Preserving Data Leaker Detection Model for Secure Sharing of Cloud Data using Integrated Techniques", 7th International Conference on Smart Computing and Communication Systems (ICSCC), IEEE, Sarawak, Malaysia, pp. 1-5, 2019.
[4] Forbes, "State of Enterprise Cloud Computing 2018", [online] Available: https://www.forbes.com/sites/louiscolumbus/2018/08/30/state-of-enterprise-cloud-computing-2018/#2e5312c2265e, August, 2018.
[5] Cheng, L., Liu, F., and Yao, D., "Enterprise Data Breach: Causes, Challenges, Prevention, and Future Directions", WIREs Data Mining and Knowledge Discovery, vol. 7, pp. 1-14, September/October 2017.
[6] Shu, X., Yao, D., and Bertino, E., "Privacy-Preserving Detection of Sensitive Data Exposure", IEEE Transactions on Information Forensics and Security, vol. 10, no. 5, pp. 1092-1103, May 2015.
[7] Shu, X., Zhang, J., Yao, D., and Feng, W.-C., "Fast Detection of Transformed Data Leaks", IEEE Transactions on Information Forensics and Security, vol. 11, no. 3, pp. 528-542, March 2016.
[8] Kaur, K., Gupta, I., and Singh, A. K., "A Comparative Study of the Approach Provided for Preventing the Data Leakage", International Journal of Network Security & its Applications (IJNSA), vol. 9, no. 5, pp. 21-33, September 2017.
[9] Breach Level Index, "A Global Database of Public Data Breaches", [online] Available: https://www.gemalto.com/press/pages/data-breaches-compromised-4-5-billion-records-in-first-half-of-2018.aspx, October, 2018.
[10] Shehab, M., Bertino, E., and Ghafoor, A., "Watermarking Relational Databases using Optimization-Based Techniques", IEEE Transactions on Knowledge and Data Engineering, vol. 20, no. 1, pp. 116-129, 2008.
[11] Backes, M., Grimm, N., and Kate, A., "Data Lineage in Malicious Environments", IEEE Transactions on Dependable and Secure Computing, vol. 13, no. 2, pp. 178-191, March/April 2016.
[12] Guo, Y., Au, O. C., Wang, R., Fang, L., and Cao, X., "Halftone Image Watermarking by Content Aware Double-Sided Embedding Error Diffusion", IEEE Transactions on Image Processing, vol. 27, no. 7, pp. 3387-3402, 2018.
[13] Mareen, H., Praeter, J. De, Wallendael, G. Van, and Lambert, P., "A Novel Video Watermarking Approach Based on Implicit Distortions", IEEE Transactions on Consumer Electronics, vol. 64, no. 3, pp. 250-258, August 2018.
[14] Liu, Z., Huang, Y., and Huang, J., "Patchwork-Based Audio Watermarking Robust Against De-synchronization and Recapturing Attacks", IEEE Transactions on Information Forensics and Security, vol. 14, no. 5, pp. 1171-1180, May 2019.
[15] Hu, D., Zhao, D., and Zheng, S., "A New Robust Approach for Reversible Database Watermarking with Distortion Control", IEEE Transactions on Knowledge and Data Engineering, vol. 31, no. 6, pp. 1024-1037, June 2019.
[16] Papadimitriou, P. and Garcia-Molina, H., "Data Leakage Detection", IEEE Transactions on Knowledge and Data Engineering, vol. 23, no. 1, pp. 51-63, January 2011.
[17] Gupta, I. and Singh, A. K., "A Probability Based Model for Data Leakage Detection using Bigraph", 7th International Conference on Communication and Network Security (ICCNS-2017), ACM, Tokyo, Japan, 2017.

[18] Gupta, I. and Singh, A. K., "A Probabilistic Approach for Guilty Agent Detection using Bigraph after Distribution of Sample Data", Procedia Computer Science, vol. 125, pp. 662-668, 2018.
[19] Gupta, I., and Singh, A. K., "Dynamic Threshold based Information Leaker Identification Scheme", Information Processing Letters, vol. 147, pp. 69-73, 2019.
[20] Gupta, I., Singh, N., and Singh, A. K., "Layer-based Privacy and Security Architecture for Cloud Data Sharing", Journal of Communication Software and Systems (JCOMSS), vol. 15, no. 2, pp. 173-185, 2019.