# Modeling and Detection of Data Leakage Fraud

Nageswarrao.Vungarala[1], Manoj Kiran.Somidi[2], Krishnaiah.R.V.[3]
[1]*Department of CS, DRK College of Engineering& Technology, Ranga Reddy, Andhra Pradesh,India*
[2]*Department of SE, DRK Institute of Science & Technology, Ranga Reddy, Andhra Pradesh, India.*
[3]*Principal DRK Institute of Science & Technology, Ranga Reddy, Andhra Pradesh, India.*

***Abstract:*** *Protecting sensitive data when that is in rest or in transit is essential. This paper addresses a problem which is discussed here. A distributor is supposed to send his private data to other person or party through his trusted agents (human beings). When trusted agents leak data for monetary gains and that is found in some other person's laptop or over Internet, the distributor should be able to identify leakage and also establish the identity of the leaked person. For leakage detection traditionally watermarking is used effectively. However, watermarking causes the source content to be modified as watermarking content is embedded. The aim of this paper is to detect leakage without the need for modification of source content. This paper proposes data leakage strategies personalized for agents for improving the chances of leakage detection. We also use fake objects along with original objects to improve the probability of detecting leakage further. The empirical results tested with a prototype application revealed that the proposed strategies are efficient and can be used in real time applications.*

***Index Terms****–data leakage, allocation strategies, fake objects, data privacy, and leakage detection*

## I. Introduction

In businesses data transfer takes place through a variety of means including secondary storage media, network, mail, fax etc. Data transfer can also be made through trusted agents, human beings who can be relayed upon, also. When data is transferred through electronic means security of data is very important especially when it is sensitive and private in nature. Here the possible security threats to data are such as hacking, eavesdropping and VIRUS. Security mechanisms provided by OS, networking applications, email applications can be used to prevent exploitation of vulnerabilities. In this case the security of data is as good as your cryptographic algorithms or mechanisms one employs to protect data. However, it is interesting show sensitive data that is being carried by human beings (trusted agents) can be protected. When such data is compromised identifying the data leakage and also finding who is behind the leak out of all trusted agents is a challenging task.

One of the techniques used earlier is digital watermarking. Digital watermarking is an information hiding mechanism where the hidden content has no business meaning with the carrier content. It does mean that some content is embedded into cover image or any electronic source for identifying the original content whether tampered or not. The digital watermarking technology is widely used in protecting intellectual properties. However, a limitation here is that in the process of watermarking the original source or object gets modified. Close to this technology is Steganography where sensitive information is encrypted and embedded into cover media such as audio, video and image. Steganography is more secure when compared with only encryption techniques as adversaries can't even see the sensitive content as opposed to cryptography where cipher text can be viewed by adversaries though they can't decrypt the content. In case of Steganography also, the original object is being modified before transferring data to authorized recipient.

To overcome this drawback and keep the original object transmitted through trusted agents intact, we propose strategies in this paper that enable data leakage detection and identification of the agent who leaked the data. The data allocation strategies proposed in this paper enable the distributor to personalize the objects (associated with agents) without modifying them. To achieve this allocation strategies create fake objects that have agent's identity. The fake objects are mixed with original objects. This process is transparent to trusted agents and only known to the distributor. When objects are leaked by trusted agents and when they are found somewhere by distributor, the distributor will be capable of detecting the leakage and also relate the fraud with the agent who did that leakage. The sections below elaborate the problem model and also the detection strategies in detail.

## II. Related Work

Active research has been around on the topic data leakage detection for long time. However, the detection of leakage has been traditionally pertaining to electronic data transfer. When data is leaked by trusted people intentionally through electronic means, there are possible security measures such as authentication mechanisms. However, when data is given to a human being or trusted agent and asked him to hand it over to a genuine recipient, the agent may give it to other person as well. This is the problem and the solution is challenging. The literature on this kind of problem is relatively less. The research initially began with description of provenance problem as given in [4]. This problem is related to finding the lineage or origin of the data or tracing origin of data in other words. When lineage is traced, it is possible to identify guilty agents. The agents who leak the data must be having that data from a particular origin. When that origin or lineage is found correctly, the detection of the guilty is possible. The research conducted in this field is presented in a tutorial. A good overview of all research results in the area of lineage detection or data provenance is presented in the tutorial [5]. However, the suggested solutions in this tutorial are domain specific. For instance the scenarios are pertaining to data warehouses where historical data is stored permanently. More on the data provenance problems on data warehouses are provided in [6] and it assume some previous knowledge with respect to data creation and how the data is viewed. These researches so far related to data lineage or data provenance problems. There are other solutions in the literature pertaining to watermarking [7]. However, as it is well known, watermarking leaves the source objects modified for the purpose of protecting it. This is the phenomenon involved in watermarking as described in [10]. In this paper we present a data leakage detection approach without modifying the original objects. Watermarking with various cover media like voice, images and video are presented in literature [10], [8], [11], and [12]. The objects that are to be transferred are never modified in the proposed model. Instead some fake objects are mixed with original objects for the purpose of leakage detection. There are other approaches that are based on access control policies. The objects are securely transmitted and when they are leaked that leakage is found with access control policies and also embedding marks in relational data. These findings are described in [13] and [10]. In this paper we stick to the policy that the original objects should never be modified in order to detect leakage and also identification of the agent who leaked it.

## III. Problem Description

This paper assumes a problem that is described here. A distributor has plenty of objects that are maintained in his trusted server. He is authorized to collect such objects from various organizations. He is also authorized to distribute those objects to a group of agents who are authorized. Such agents in this scenario are known as "trusted agents". Though they are known as trusted agents and authorized people to have data given by distributor and in turn give those data objects to intended recipients, there is possibility that agents may involve in data leakage illegally. It does mean that they hand over data objects to third parties by any means for monetary or other reasons. To elaborate this scenario, an example is described here. For a given company A, T contains records. Marketing agency U1 is hired by company A in order to perform online survey of customers. As survey needs can be satisfied by any arbitrary records, the U1 requests for 1000 customer records. At the same time company A may give contact to a billing agent U2 who needs customer records. U2 receives records based on the area for which he is responsible to collect bills. This means that his query is conditional while U1 can obtain data randomly. This way there are two models here. The first model is to pick records randomly and the second model is to select records based on a specific condition. The first model is known as random selection of objects while the second one is known as sample of objects.

## IV. Guilty Agents

Agents who involve in fraud i.e., distributing data objects given by distributor to third parties without permission are known as guilty agents. Agents who have such malicious practices are to be identified by the strategies followed in this paper. This paper specifically follows a data allocation strategy that involves fake objects including original objects. The fake objects contain identity information of agents to whom those objects are given along with real objects. However, the trusted agents believe that all objects given to them are genuine objects only. This paper assumes that the agents do not have knowledge that they receive fake objects from distributor along with original objects.

## V. Agent Guilt Model

The proposed guilt model facilitates the distributor to detect leakage of data objects and identify the agent who leaked it. Pr{Gi|S} where the S indicates set of objects and Gi indicates a particular guilty agent. Pr is the probability of guilt. Assume the following.

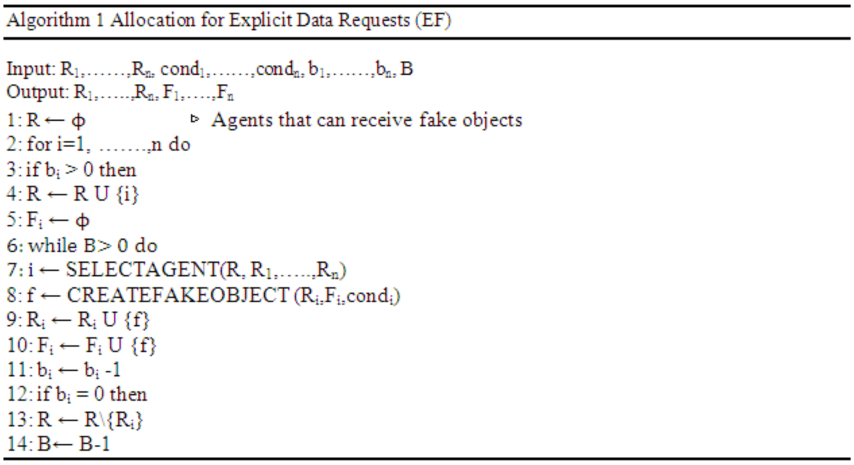
T = {t1, t2, t3}, R1{t1, t2, t3}, R2{t1, t2, t3}, S={t1, t2, t3}.

Here T represents set of objects available at distributor. R represents a set of objects available at agent and S indicates a set of objects available at target. The probability that an agent leaks objects to third parties is computed as follows.

$$Pr\{U_i \text{ leaked } t \text{ to } S\} = \begin{cases} \frac{1-p}{|V_t|}, & \text{if } U_i \in V_t \\ 0, & \text{otherwise} \end{cases}$$

## VI. Data Allocation Strategy

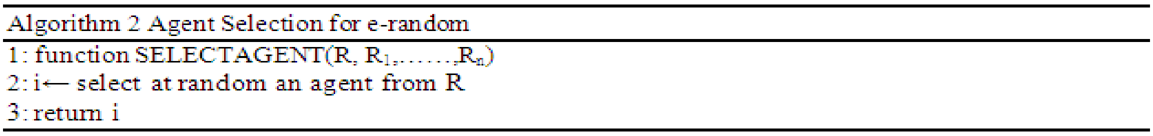
Data allocation to agents plays an important role in enhancing the probability of detection of leakage and establishing identity of agent who caused the leakage. For data objects allocation algorithms are proposed in [3] are used with some changes. The original algorithms provided in [3] are provided here.



```
Algorithm 1 Allocation for Explicit Data Requests (EF)

Input: $R_1,\ldots,R_n$, $cond_1,\ldots,cond_n$, $b_1,\ldots,b_n$, B
Output: $R_1,\ldots,R_n$, $F_1,\ldots,F_n$
R ← ϕ                ▹ Agents that can receive fake objects
for i=1, .......,n do
if $b_i > 0$ then
R ← R U {i}
$F_i$ ← ϕ
while B> 0 do
i ← SELECTAGENT(R, $R_1,\ldots,R_n$)
f ← CREATEFAKEOBJECT ($R_i$,$F_i$,$cond_i$)
$R_i$ ← $R_i$ U {f}
$F_i$ ← $F_i$ U {f}
$b_i$ ← $b_i$ -1
if $b_i$ = 0 then
R ← R\{$R_i$}
B← B-1
```

Listing 1 – Allocation for Explicit Data Requests

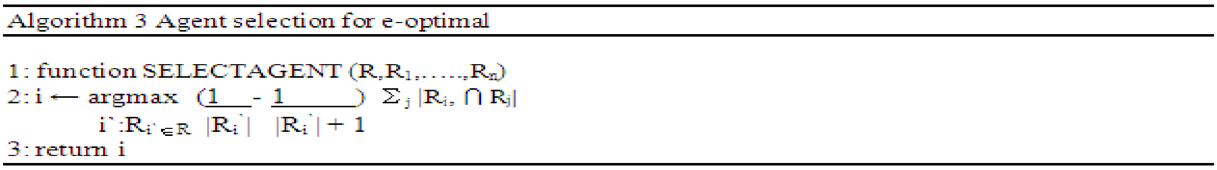
The algorithm 1 is a generic algorithm used by other algorithms for data objects allocation. It makes use of two important procedures namely SELECTAGENT() and CREATEFAKEOBJECT().



```
Algorithm 2 Agent Selection for e-random
function SELECTAGENT(R, $R_1,\ldots,R_n$)
i← select at random an agent from R
return i
```

Listing 2 – Agent Selection for e-random

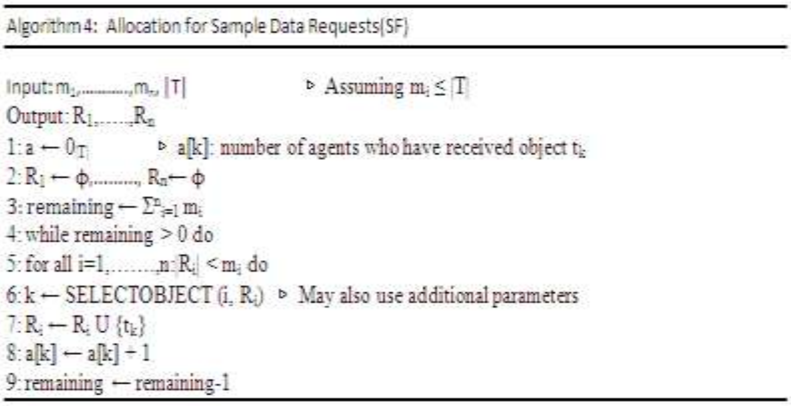
This algorithm in invoked by first algorithm. It is responsible to select an agent randomly for data allocation.



```
Algorithm 3 Agent selection for e-optimal

function SELECTAGENT (R,$R_1,\ldots,R_n$)
i ← $\underset{i':R_{i'} \in R}{\operatorname{argmax}} \left( \frac{1}{|R_{i'}|} - \frac{1}{|R_{i'}|+1} \right) \Sigma_j |R_{i'} \cap R_j|$
return i
```

Listing 3 – Agent selection for e-optimal

This algorithm is meant for selecting agent using optimized function. This is invoked by the first algorithm as and when required.



```
Algorithm4: Allocation for Sample Data Requests(SF)

Input: $m_1,\ldots,m_n$, |T|                ▹ Assuming $m_i \leq |T|$
Output: $R_1,\ldots,R_n$
a ← $0_{|T|}$          ▹ a[k]: number of agents who have received object $t_k$
$R_1$ ← ϕ,........, $R_n$← ϕ
remaining ← $\Sigma^n_{i=1} m_i$
while remaining > 0 do
for all i=1,.......,n:|$R_i$| < $m_i$ do
k ← SELECTOBJECT (i, $R_i$)  ▹ May also use additional parameters
$R_i$ ← $R_i$ U {$t_k$}
a[k] ← a[k] + 1
remaining ← remaining-1
```

Listing 4 – Allocation for Sample Data Requests

When data objects are to be allocated to agents the sample data request allocation is done using this algorithm.

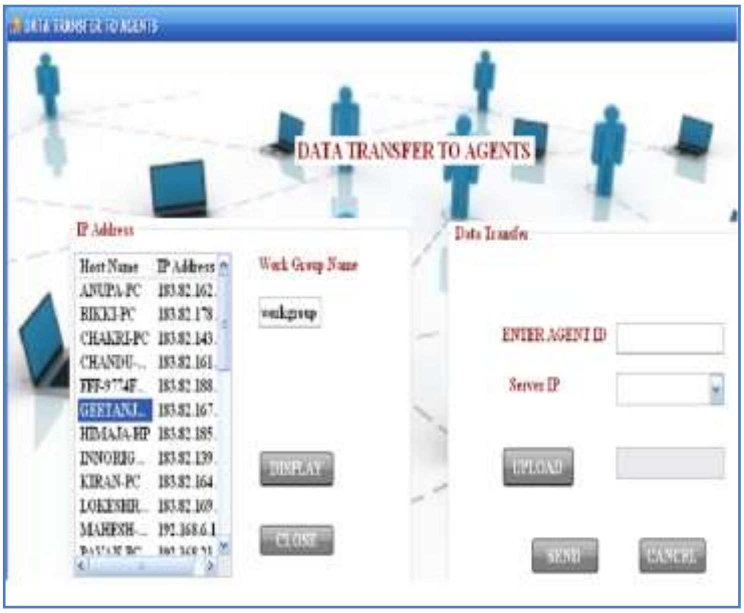
```
Algorithm 5: Object Selection for s-random

1: function SELECTOBJECT (i,$R_i$)
2: k ← select at random an element from set {k' | $t_k \notin R_i$}
3: return k
```

Listing 5 – Object Selection for e-random

This algorithm is meant for allocating objects randomly. From the set of given objects one object is selected randomly.

## VII. Emperical Results

To test and evaluate the proposed guilt model that helps in data leakage detection, a prototype application is developed in Java programming language. It environment used include Java, PC with Windows 7 OS, Eclipse IDE. The proposed application facilitates two users work in collaboration. They are distributor and agent. The distributor is having privileges to have data of various organizations and he is authorized to distribute it to the intended recipients through a group of trusted agents. The agents who interact with the system can make request for objects and get objects from the distributor. The communication model is distributed so as to enable remote agents also can get data objects from a distributor. Once objects are received from distributor, they are supposed to share them with only intended recipients. When any agent performs fraud activity by leaking data to third parties who are not authorized to receive it, such data objects are known as leaked objects. When distributor finds such leaked objects, he can detect the agent information who leaked the objects. The prototype application is as shown in fig. 1.



DATA TRANSFER TO AGENTS

IP Address

| Host Name | IP Address |
|---|---|
| ANUPA-PC | 183.82.162. |
| BIKKI-PC | 183.82.178. |
| CHAKRI-PC | 183.82.143. |
| CHANDU-... | 183.82.161. |
| FFF-9774F... | 183.82.188. |
| GEETANJ... | 183.82.167. |
| HIMAJA-HP | 183.82.185. |
| INNORIG... | 183.82.139. |
| KIRAN-PC | 183.82.164. |
| LOKESHR... | 183.82.169. |
| MAHESH-... | 192.168.6.1 |

Work Group Name: workgroup

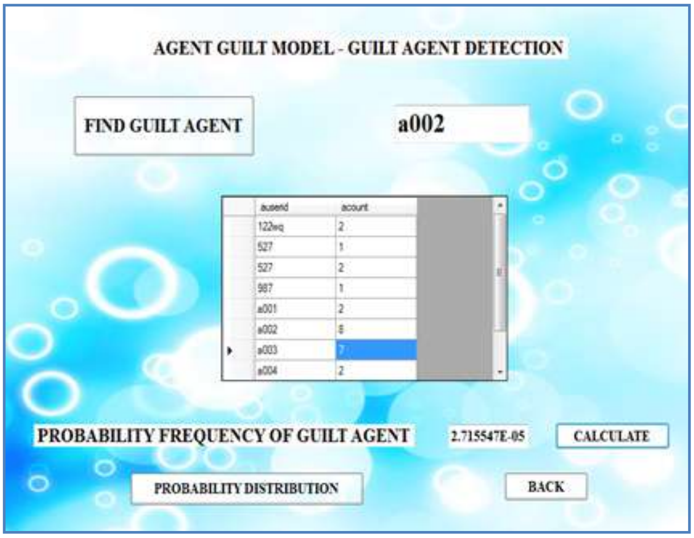
Data Transfer

ENTER AGENT ID

Server IP

Fig. 1 – Distributor shares objects to agents

Once agents receive data from distributor's server system, they can have data objects with them. However, the data objects they possess contain fake objects that appear like real objects. Due to lack of this knowledge, agents may try to share such data objects to third parties for monetary or other gains. In this case the data is leaked to third parties. When distributor finds such data, he will make use of the proposed application and can trace the identity of agent who leaked it.



AGENT GUILT MODEL - GUILT AGENT DETECTION

FIND GUILT AGENT

a002

| ausend | acount |
|---|---|
| 122wq | 2 |
| 527 | 1 |
| 527 | 2 |
| 987 | 1 |
| a001 | 2 |
| a002 | 8 |
| a003 | 7 |
| a004 | 2 |

PROBABILITY FREQUENCY OF GUILT AGENT 2.715547E-05

Fig. 2 – Simulation of Agent Guilt Model

Simulation of agent's guilt model which is based on the fake objects and the underlying algorithm in allocating objects to agents, reveal the identity of the agent who probably leaked data. The hypothesis i.e., "distributor who shares data objects with trusted agents can trace the identity of the agents when they leak those objects illegally" has been problem experimentally. The guilt probability distribution is visualized in fig. 3.

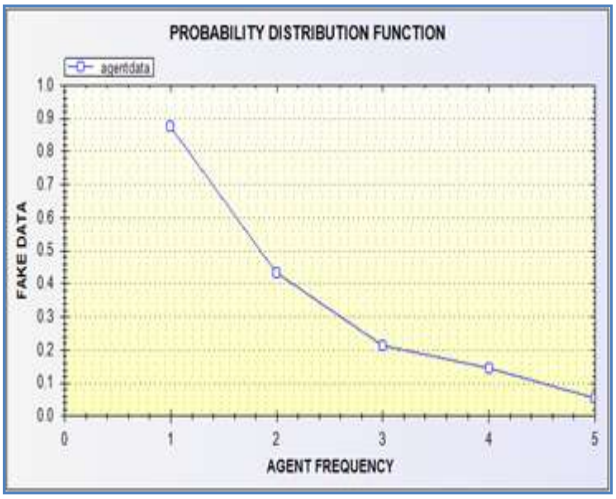


Fig. 2 – Probability Distribution of Guilt Model

As can be seen in fig. 2, it is evident that agents are involved in data leakage and the distributor of the data objects is capable of detecting leakage and also identifies the agent who performed the leakage.

## VIII. Conclusion

In this paper we present a solution to a data leakage problem. We consider a hypothetical scenario where a distributor who has data of his own or other organizations shares it with trusted agents. When the trusted agents in turn leak it to third parties we called it data leakage problem. The solution to this problem is challenging as the trusted agents may use various means of leaking data. The hypothesis here is that when leaked data is found anywhere, the distributor of that data can identify the guilty agent. To prove this hypothesis strategies are devised. One such strategy is adding fake objects to original objects without modifying the original objects. The fake objects appear like real objects and agents and it is assumed that agents can't distinguish between fake and real objects. When data is found over Internet or any where, the distributor should be able to detect the agent who leaked it based on the probability estimation. This is possible as the fake objects are having some information associated with agents. It does mean that when agent is given real objects along with fake objects, the fake objects are created in such a way that, they are having information embedded pertaining to the agent to whom those objects are given. Thus the algorithm can detect the probability of data leakage and also the agent who leaked it. The empirical results revealed that the hypothesis has been proven successfully.

## References

[1] R. Agrawal and J. Kiernan. Watermarking relational data bases In VLDB ’02: Proceedings of the 28th international conference on Very Large Data Bases, pages 155–166. VLDB Endowment, 2002.

[2] B. Pfitzmann, “Information Hiding Terminology,” *Proc. First Int’l Workshop Information Hiding, Lecture Notes in Computer Science No. 1,174*, Springer-Verlag, Berlin,1996, pp. 347-356.

[3] Panagiotis Papadimitriou, Member, IEEE, Hector Garcia-Molina, Member, IEEE. Data Leakage Detection. IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, VOL. 22, NO. 3, MARCH 2010.

[4] P. Buneman, S. Khanna, and W. C. Tan. Why and where: A characterization of data provenance. In J. V. den Bussche and V. Vianu, editors, *Database Theory - ICDT 2001, 8th International Conference, London, UK, January 4-6, 2001, Proceedings*, volume 1973 of *Lecture Notes in Computer Science*, pages 316–330. Springer, 2001.

[5] P. Buneman and W.-C. Tan. Provenance in databases. In *SIGMOD ’07: Proceedings of the 2007 ACM SIGMOD international conference on Management of data*, pages 1171–1173, New York, NY, USA, 2007. ACM.

[6] Y. Cui and J. Widom. Lineage tracing for general data warehouse transformations. In *The VLDB Journal*, pages 471–480, 2001.

[7] F. Guo, J. Wang, Z. Zhang, X. Ye, and D. Li. *Information SecurityApplications*, pages 138–149. Springer, Berlin / Heidelberg, 2006.An Improved Algorithm to Watermark Numeric Relational Data.

[8] S. Jajodia, P. Samarati, M. L. Sapino, and V. S. Subrahmanian.Flexible support for multiple access control policies. *ACM Trans.Database Syst.*, 26(2):214–260, 2001.[10] Y. Li, V. Swarup, and S. Jajodia.

[9] P. Bonatti, S. D. C. di Vimercati, and P. Samarati. An algebra for composing access control policies. *ACM Trans. Inf. Syst. Secur.*, 5(1):1–35, 2002.

[10] S. Jajodia, P. Samarati, M. L. Sapino, and V. S. Subrahmanian.Flexible support for multiple access control policies. *ACM Trans.Database Syst.*, 26(2):214–260, 2001.

[11] Y. Li, V. Swarup, and S. Jajodia. Fingerprinting relationaldatabases: Schemes and specialties. *IEEE Transactions on Dependableand Secure Computing*, 02(1):34–45, 2005.

[12] R. Sion, M. Atallah, and S. Prabhakar. Rights protection for relational data. In *SIGMOD ’03: Proceedings of the 2003 ACMSIGMOD international conference on Management of data*, pages 98–109, New York, NY, USA, 2003. ACM.

[13] P. Bonatti, S. D. C. di Vimercati, and P. Samarati. An algebra for composing access control policies. *ACM Trans. Inf. Syst. Secur.*, 5(1):1–35, 2002.

## About Authors:

| | |
|---|---|
| | Nageswarrao Vungarala is a student of DRK College of Engineering and Technology, Ranga Reddy, Andhra Pradesh, India. He has received M.C.A degree and M.Tech Degree in Computer Science. His main research interest includes Data Mining and Cloud Computing |
| | Manoj Kiran Somidi is a student of DRK Institute of Science & Technology, Ranga Reddy, Andhra Pradesh, India. He has received B.Tech Degree in Computer Science and Engineering and M.Tech Degree in Software Engineering. His main research interest includes Software Engineering, Data Mining. |
| | Dr.R.V.Krishnaiah is working as Principal at DRK INSTITUTE OF SCINCE & TECHNOLOGY, Hyderabad, and AP, INDIA. He has received M.Tech Degree (EIE&CSE) and Ph.D. His main research interest includes Data Mining, Software Engineering. |