An Approach to Protect the Privacy of Cloud Data from Data Mining Based Attacks

Himel Dev, Tanmoy Sen, Madhusudan Basak and Mohammed Eunus Ali
Department of Computer Science and Engineering (CSE)
Bangladesh University of Engineering and Technology (BUET)
Dhaka-1000
Email: {himeldev, sen.buet, madhusudan.buet, mohammed.eunus.ali}@gmail.com

***Abstract*—Cloud computing has revolutionized the way computing and software services are delivered to the clients on demand. It offers users the ability to connect to computing resources and access IT managed services with a previously unknown level of ease. Due to this greater level of flexibility, the cloud has become the breeding ground of a new generation of products and services. However, the flexibility of cloud-based services comes with the risk of the security and privacy of users' data. Thus, security concerns among users of the cloud have become a major barrier to the widespread growth of cloud computing. One of the security concerns of cloud is data mining based privacy attacks that involve analyzing data over a long period to extract valuable information. In particular, in current cloud architecture a client entrusts a single cloud provider with his data. It gives the provider and outside attackers having unauthorized access to cloud, an opportunity of analyzing client data over a long period to extract sensitive information that causes privacy violation of clients. This is a big concern for many clients of cloud. In this paper, we first identify the data mining based privacy risks on cloud data and propose a distributed architecture to eliminate the risks.**

I. Introduction

Cloud computing facilitates end-users or small companies to use computational resources such as software, storage, and processing capacities belonging to other companies (cloud service providers). Cloud services include Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS) [2]. Big corporates like Amazon, Google and Microsoft are providing cloud services in various forms. Amazon Web Services (AWS) provides cloud services that include Amazon Elastic Compute Cloud (EC2), Simple Queue Service (SQS) and Simple Storage Service (S3) [9]. Google provides Platform as a Service (PaaS) known as Google App Engine (GAE) and facilitates hosting web applications [6]. Microsoft also provides cloud services in the form of Windows Azure, SQL Azure, Windows Intune etc. By using these services, users can exploit the benefit of mass storage and processing capacity at a low cost. Developers can use these services to avoid the mass overhead cost of buying resources, e.g., processors and storage devices [6].

Although cloud computing is a powerful means of achieving high storage and computing services at a low cost, it has not lived up to its reputation. Many potential users and companies yet lack interest in cloud based services [11]. One of the main reasons behind this lack of interest involves security issues. Cloud has several security issues involving assurance and confidentiality of data [6]. A user entrusting a cloud provider may lose access to his data temporarily or permanently due to an unlikely event such as a malware attack or network outage. On April 21, 2011, EC2's northern Virginia data center was affected by an outage and brought several websites down [29][3]. Problems caused by this outage lasted till April 25, 2011 [29]. Such an unlikely event can do significant harm to the users. Confidentiality of user data in the cloud is another big concern. Cloud has been giving providers an opportunity to analyze user data for a long time. In addition, outside attackers who manage to get access to the cloud can also analyze data and violate user privacy. Cloud is not only a source of massive static data, but also a provider of high processing capacity at low cost. This makes cloud more vulnerable as attackers can use the raw processing power of cloud to analyze data [11].

Various data analysis techniques are available now-a-days that successfully extract valuable information from a large volume of data. These analysis techniques are being used by cloud service providers. For example, Google uses data analysis techniques to analyze user behaviors and recommend search results [11]. Attackers can use these techniques to extract valuable information from the cloud. The recent trends of data analysis involve mining which is closely associated with statistical analysis of data [22]. Data mining can be a potential threat to cloud security considering the fact that entire data belonging to a particular user is stored in a single cloud provider. The single storage provider approach gives the provider opportunity to use powerful mining algorithms that can extract private information of the user. As mining algorithms require a reasonable amount of data, the single provider architecture suits the purpose of the attackers. This approach (single cloud storage provider) also eases the job of attackers who have unauthorized access to the cloud and use data mining to extract information. Thus the privacy of data in the cloud has become a major concern in recent years.

In this paper, we present an approach to prevent data mining based attacks on the cloud. Our system involves

distributing user data among multiple cloud providers to make data mining a difficult job to the attackers. The key idea of our approach is to categorize user data, split data into chunks and provide these chunks to the proper cloud providers. In a nutshell our approach consists of categorization, fragmentation and distribution of data. The categorization of data is done according to mining sensitivity. Mining sensitivity in this context refers to the significance of information that can be leaked by mining. Categorization allows to identify sensitive data and to to take proper initiatives to maintain privacy of such data. Fragmentation and distribution of data among providers reduce the amount of data to a particular provider and thus minimize the risk associated with information leakage by any provider. This distribution is done according to the sensitivity of data and the reliability of cloud providers. The reliability of a cloud provider is defined in terms of its reputation. A cloud provider is given a particular data chunk only if the provider is reliable enough to store chunks of such sensitivity. Distribution restricts an attacker from having access to a sufficient number of chunks of data and thus prevents successful extraction of valuable information via mining. Even if an attacker manages to access required chunks, mining data from distributed sources remains a challenging job [28]. The main challenge in this case is to correlate the data seen at the various probes [30]. In addition to prevent data mining, the proposed system ensures greater availability of data and optimizes cost.

In the remainder of the paper, we have identified data mining based security risks on cloud and demonstrated our distributed cloud architecture to eliminate the risks. We have discussed how applications will work under the given architecture. We have also discussed about the methods required to implement the system and the feasibility of the system.

II. Data Mining on Cloud

Data mining is one of the fastest growing fields in computer industry [14] that deals with discovering patterns from large data sets [22]. It is a part of knowledge discovery process and is used to extract human understandable information [8]. Mining is preferably used for a large amount of data [25][26] and related algorithms often require large data sets to create quality models [1].

The relationship between data mining and cloud is worth to discuss. Cloud providers use data mining to provide clients a better service [27]. If clients are unaware of the information being collected, ethical issues like privacy and individuality are violated [26][12]. This can be a serious data privacy issue if the cloud providers misuse the information. Again attackers outside cloud providers having unauthorized access to the cloud, also have the opportunity to mine cloud data. In both cases, attackers can use cheap and raw computing power provided by cloud computing [11][17] to mine data and thus acquire useful information from data. According to the survey done by Rexer Analytics, 7% data miners use cloud to analyze data [16]. As cloud is a massive source of centralized data, data mining gives attackers a great advantage in extracting valuable information and thus violating clients' data privacy.

A. The Importance of Client Privacy

Client privacy is a tentative issue as all clients do not have the same demands regarding privacy. Some are satisfied with the current policy while others are quite concerned about their privacy. The proposed system is designed preferably for the clients belonging to the second category for whom privacy is a great concern. These clients may not afford the luxury of maintaining private storage while they are interested in spending a little more money on maintaining their privacy on the cloud. If the client itself is a company providing services to others, the violation of privacy of the client affects the privacy of its customers. Specially companies dealing with financial, educational, health or legal issues of people are prominent targets and leaking information of such companies can do significant harm to their customers. Information in this context refers to the financial condition of a customer, the likelihood of an individual getting a terminal illness, the likelihood of an individual being involved in a crime etc. Sometimes leaking information regarding a particular company leads to a national catastrophe. The events of TIA (Total Information Awareness) gathering financial, educational, health and other information about people in 2002 [23] and NSA obtaining customer records from phone companies and analyzing them to identify potential terrorists in May 2006 [23] can be considered as examples.

B. Data Mining: A Potential Threat to Privacy

The successful extraction of useful information via data mining depends on two main factors: proper amount of data and suitable mining algorithms. Various mining algorithms are used for numerous purposes. Some mining algorithms are good enough to extract information up to the limit that violates client privacy. For example, multivariate analysis identifies the relationship among variables and this technique can be used to determine the financial conditon of an individual from his buy-sell records, clustering algorithms can be used to categorize people or entities and are suitable for finding behavioral patterns [18], association rule mining can be used to discover association relationships among large number of business transaction records [18] etc. Analysis of GPS data is common nowadays and the results of such analysis can be used to create a comprehensive profile of a person covering his financial, health and social status [15]. Thus analysis of data can reveal private information about a user and leaking these sort of information may do significant harm. As more research works are being done on mining, improved algorithms and tools are being developed [14]. Thus, data mining is becoming more powerful and possessing more threat to cloud users. In upcoming days, data mining based privacy attack can be a more regular weapon to be used against cloud users.

III. Eliminating Cloud-mines

In this section, we first discuss the data mining based privacy threats to the single provider cloud architecture. Then give an overview of the state-of-the-art distributed approach to prevent data mining based privacy attacks on the cloud.

A. Existing System Threats

The current cloud storage system is a vulnerable one because data remain under a single cloud provider. This can lead to data loss in case of events like network outage, the cloud provider going out of business, malware attack etc. The current system also gives a great advantage to the attackers as they have fixed targets in the forms of cloud providers. If an attacker chooses to attack a specific client, then he can aim at a fixed cloud provider, try to have access to the client's data and analyze it. This eases the job of the attackers. As long as the entire data belonging to a client remain under a single cloud provider, both inside and outside attackers get the benefit of using data mining to a great extent. Inside attackers in this context refers to malicious employees at a cloud provider. Data mining models often require large number of observations and single provider architecture is a great advantage suiting the case as all the samples remain under the provider. Thus single provider architecture is the biggest security threat concerning data mining on cloud.

B. A Distributed Approach to the Cloud

To eliminate the disadvantage of storing all data of a client to the same provider, data can be split into chunks and distributed among multiple cloud providers. The advantage of this distributed system can be visualized when an attacker chooses a specific client but the distribution of data obliges him to target multiple cloud providers, making his job increasingly difficult. Mining based attacks on cloud involves attackers of two categories: malicious employees inside provider and outside attackers. Distribution of data chunks among multiple providers restricts a cloud provider from accessing all chunks of a client. Even if the cloud provider performs mining on chunks provided to the provider, the extracted knowledge remains incomplete. Again, mining data from distributed sources is challenging [28]. Specially correlating data from various sources is cumbersome [30] and often leads to unsuccessful mining. So outside attackers managing access to various providers can't use mining effectively. Aggarwal et al. [5] described partitioning of data across multiple databases in such a fashion to ensure that the exposure of the contents of any one database does not result in a violation of privacy. The distributed architecture for cloud redefines the partitioning of data in terms of preserving privacy from mining based attacks.

The distributed approach can take the form of Redundant Array of Independent Disks (RAID) technique used for traditional databases. RACS [4] uses the RAID concept to reduce the cost of maintaining the data on the cloud. It considers each cloud provider as a separate disk. RACS exploits the benefit of RAID on the cloud. One can choose a different RAID level for each client depending on the client's demand. For example, RAID level 6 can be used to ensure high assurance of data. It guarantees successful retrieval of data in case of a cloud provider being blocked by any unlikely event or going out of business. In summary, the distributed approach exploits two major benefits. First, it improves privacy by making the attacker's job complicated by increasing the number of targets and decreasing amount of data available at each target. Second, it ensures the greater availability of data.

IV. System Architecture

In this section we discuss our proposed system architecture that prevents data mining based privacy attacks on the cloud. Our system consists of two major components: *Cloud Data Distributor* and *Cloud Providers*. The Cloud Data Distributor receives data in the form of files from clients, splits each file into chunks and distributes these chunks among cloud providers. Cloud Providers store chunks and responds to chunk requests by providing the chunks.

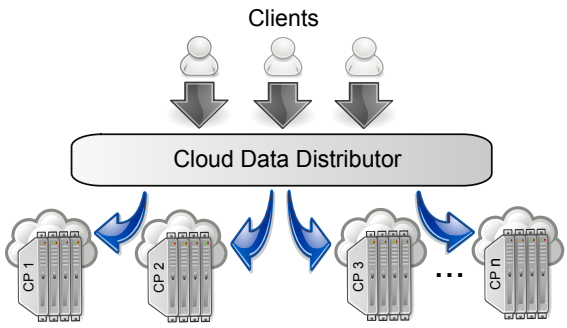

Fig. 1. System Architecture

A. Cloud Data Distributor

Cloud Data Distributor is the entity that receives data (files) from clients, performs fragmentation of data (splits files into chunks) and distributes these fragments (chunks) among Cloud Providers. It also participates in data retrieving procedure by receiving chunk requests from clients and forwarding them to Cloud Providers. Clients do not interact with Cloud Providers directly rather via Cloud Data Distributor. This entity deals with Cloud Providers as an agent of clients.

To upload data, clients deliver files to the Cloud Data Distributor. Each file is given a privacy level chosen by the client indicating its mining sensitivity. Here mining sensitivity of a file refers to the significance of information that can be leaked through mining the data in the file. The proposed system suggests 4 sensitivity levels of privacy: PL 0, 1, 2, 3. These 4 levels indicate public data (data accessible to everyone including the adversary), low sensitive data (data that do not reveal any private or protected information but can be used to find patterns), moderately sensitive data (protected data that can be used to extract non-trivial financial, legal,

health information of a company or an individual), highly sensitive data or private data (data that can be used to extract personal information of an individual or private information of a company, revealing which can prove disastrous) respectively. The higher the privacy level of a file, the more sensitive the data inside the file. After receiving files from clients, the Cloud Data Distributor partitions each file into chunks with each chunk having the same privacy level of the parent file. The total number of chunks for each file is notified to the client so that any chunk can be asked by the client by mentioning the filename and serial no. Serial no. corresponds to the position of the chunk within the file.

Inside the Cloud Data Distributor each chunk is given a unique virtual *id* and this *id* is used to identify the chunk within the Cloud Data Distributor and Cloud Providers. This virtualization conceals the identity of a client from the provider. After assigning *id*, the Cloud Data Distributor distributes chunks among Cloud Providers. A provider storing a particular chunk with a virtual *id* has no idea about the real owner (client) of the chunk. Cloud Data Distributor maintains privacy level (4 level privacy similar to files) for each provider. Privacy level of a provider indicates its reliability. The higher the privacy level, the more trustworthy the provider. A chunk is given to a provider having equal or higher privacy level compared to the privacy level of the chunk. While distributing chunks, the distributor applies Redundant Array of Independent Disks (RAID) strategy [4]. This ensures availability and integrity of data in case of outages. The default choice is RAID level 5. In case of higher assurance, RAID level 6 is used. The cloud data distributor also maintains a cost level (4 cost levels and the higher the cost level, the more costly the provider) for each cloud provider indicating its storage cost (cost of data stored per GB-Month) and in case of equal privacy level, the one with a lower cost level is given preference.

To ensure greater dimension of privacy, the Cloud Data Distributor may add misleading data into chunks depending on the demand of clients. The positions of misleading data bytes are also maintained by the distributor and these misleading bytes are removed while providing the chunks to the clients.

To perform distribution and retrieval of data (chunks), the Cloud Data Distributor needs to maintain information regarding providers, clients and chunks. Hence, it maintains three types of tables describing the providers, the clients and the chunks. Each of these tables are described below:

1) Cloud Provider Table: Each entry of this table contains information regarding a particular cloud provider. These informations include the cloud provider's name, its privacy level PL, its cost level CL, count of chunks given to this provider and the list of *id*s corresponding to the chunks given to this provider.

TABLE I
CLOUD PROVIDER TABLE

Cloud Provider	PL	CL	Count	Virtual *id* list
CP1	3	3	57538	{41367, ...}
CP2	3	2	92654	{57643, ...}
CP3	3	2	96456	{88653, ...}
CP4	3	3	52387	{78540, ...}

2) Client Table: The entries of this table contain information regarding clients. These information include client's name, set of pairs combining a password and a privacy level associated with this password, total number of chunks of this client (Count), a set of quadruples consisting of filename, serial no., privacy level and Chunk Table index for each chunk belonging to this client. The pair ⟨password, PL⟩ is used for access control which associates a group of users with a ⟨password, PL⟩ pair at client side.

TABLE II
CLIENT TABLE

Client	(pass, PL)	Count	(filename, sl, PL, idx)
CL1	(98pX, 3) (m98r, 0)	29586	(cf11, 0, 3, 0) (cf11, 1, 3, 1) ....
CL2	(cv67, 3) (H7y5, 1)	34567	(cf21, 0, 3, 2) (cf21, 1, 3, 3) ....

3) Chunk Table: The entries of this table contain information regarding data chunks. These information include the virtual *id*, privacy level (PL), Cloud Provider Table index of the current cloud provider storing the chunk (CP), Cloud Provider Table index of the snapshot provider (SP) (if any), set of positions of misleading data bytes (M) (if any) for all chunks. Snapshot of a chunk refers to the state of the chunk before the chunk is modified. That is, snapshot provider stores the pre-state and cloud provider stores the post-state of a chunk after each modification.

TABLE III
CHUNK TABLE

virtual *id*	PL	CP *index*	SP *index*	M
41367	3	0	NA	{12, ...}
57643	3	2	NA	{19, ...}
88653	3	1	NA	{27, ...}
78540	3	3	NA	{21, ...}

B. Cloud Providers

The second entity refers to the cloud storage providers. The main tasks of Cloud Providers are: storing chunks of data, responding to a query by providing the desired data, and removing chunks when asked. All these are done using virtual *id* which is known as key for Amazons simple storage service (S3) [9]. Providers receive chunks from the distributor and store them. Each provider is considered as a separate disk storing clients' data. The cloud provider responds to the query of the distributor by providing data. Providers

also receive remove requests from the distributor and acts accordingly by removing the corresponding chunk.

Number of cloud service providers is rapidly increasing and some are providing better services than the other. Some cloud providers have a reputation of being very trustworthy while some offer very cheap services. It is wise to make a trade off between security and cost by providing regular data to cheaper providers while sensitive data to secured providers.

C. Architectural Issues

The first thing to consider in system architecture is that a single data distributor can create a bottleneck in the system as it can be the single point of failure. To eliminate this, multiple distributors of cloud data can be introduced. In case of multiple data distributors, for each client, a specific distributor will act as the primary distributor that will upload data, whereas other distributors will act as secondary distributors who can perform the data retrieval operations. Figure 2 shows the extended system architecture with multiple distributors of data.

The next architectural issue is the reliability of the Cloud Data Distributor implemented at a third party server. To solve this, the Cloud Data Distributor can be implemented at client side by using CAN [24] or CHORD [19] like hash tables that will map each ⟨filename,chunk Sl⟩ pair to a Cloud Provider. A downloadable list of Cloud Providers can be used to generate the Cloud Provider Table. Client will also have to maintain a Chunk Table for his chunks. This approach has some limitations. Client will require some memory where the tables will reside. The next issue to consider is the number of privacy levels. Our proposed system suggests but is not limited to 4 privacy levels. Number of privacy levels can be increased or decreased based on requirements.

V. Application Architecture

The application architecture of the proposed system is motivated by the google file system. The Google File System is a scalable distributed file system for large distributed data-intensive applications [13].

When a client runs an application using files, the application can request for individual chunk by providing (client name, password, filename, sl no.) or for all chunks of a file by providing (client name, password, filename). In both the cases the password will have to be privileged enough to ask for the particular chunk(s).

If the privilege level of the password is greater than or equal to the privilege level of the chunk(s), the Cloud Data Distributor uses the chunk *index* field in the client table to identify the corresponding chunk(s) in the chunk table. The chunk table provides the virtual *id* of the corresponding chunk(s). It also provides the cloud provider *index* which identifies the corresponding provider entry/entries in the cloud provider table. The entry/entries of the cloud provider table provide(s) information regarding the provider(s) storing the chunk(s). After identifying the cloud provider(s), the Cloud Data Distributor uses the virtual *id(s)* as the key to obtain the required chunk(s) from the corresponding provider(s). Then the chunk(s) is(are) passed to the application. Consider a scenario from Figure 3 where a chunk request to Cloud Data Distributor is made using the quadruple (Bob, x9pr, file1, 0). Bob is listed as a client on Client Table and the password x9pr is listed under Bob. The privacy level of the password

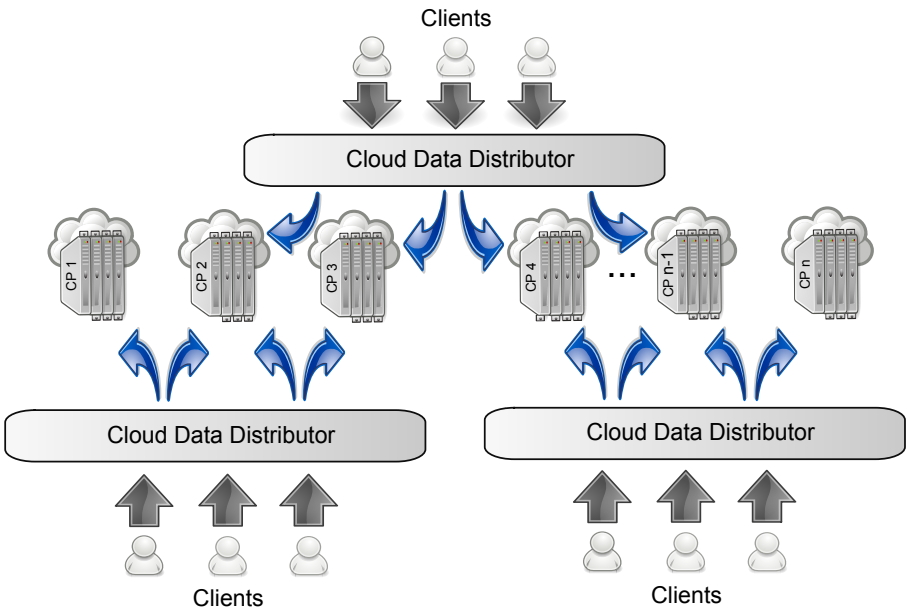

Fig. 2. Extended System Architecture

x9pr is 1 and the privacy level of chunk 0 of file1 is also 1. As the privacy level of the password and the chunk is equal, the password is privileged enough to ask for the chunk. Now, the chunk index of chunk 0 of file1 is listed as 0 at Client Table. So the Cloud Data Distributor checks the 0th entry of Chunk Table which reveals the virtual id of the chunk, 10986. It also provides the current provider index 6 which in turn reveals the identity of the cloud provider from the Provider Table. The sixth entry of Cloud Provider Table is Earth. So, a chunk request to cloud provider Earth is made using 10986 as key. Upon receiving the chunk from Earth, the Cloud Data Distributor provides the chunk to the seeker. Consider another scenario where a request is made using quadruple (Bob, aB1c, file1, 0). The password aB1c is listed under Bob and its privacy level is 0. As the privacy level of the requested chunk is 1, the password is not privileged enough to access the chunk. Hence its request is denied.

Santos et al. [21] proposed a trusted cloud computing platform (TCCP) for ensuring the confidentiality and integrity of computations that are outsourced to IaaS services. The combination of our proposed system and the TCCP ensures the privacy of cloud data in case of outsourced storage and processing.

VI. System Design

To implement our proposed system, we need to implement the following functionalities.

- Distribute Data
- Retrieve Data

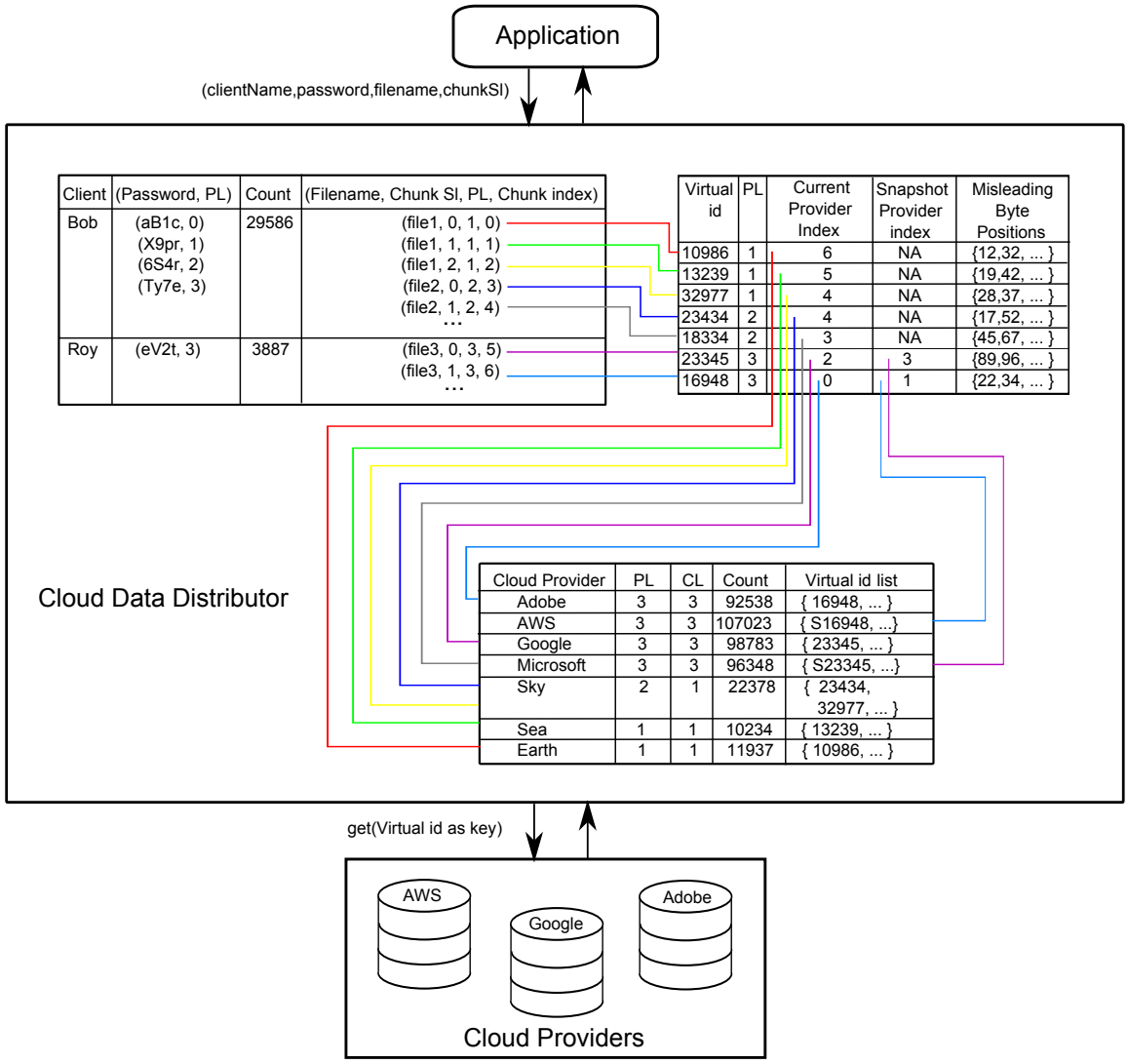

Fig. 3. Application Architecture

- Remove Data

Next, we consider a set of abstract functions that can implement the above procedures successfully.

The distribution of data among multiple Cloud Providers can be implemented using two functions and these functions are implemented inside the Cloud Data Distributor.

- *chunks[ ] split(file)*: receives a file from a client and splits the file into chunks. The chunk size is fixed for a particular privilege level. The higher the privilege level, the lower the chunk size. A unique virtual *id* is attached to each chunk. This *id* conceals the actual client identity from Cloud Providers, thus maintaining the client identity private to the Cloud Data Distributor.
- *void distribute(chunks[ ])*: accepts chunks of data from the split method described above and distributes these chunks among Cloud Providers in a random way. Same chunk can be provided to multiple Cloud Providers depending on the clients' requirement. Here requirement indicates the degree of assurance the client demands.

The data retrieving process can be implemented using the following functions inside the Cloud Data Distributor.

- *chunk get_chunk(client name, password, filename, sl no.)*: accepts a chunk request from a client, fetches chunk from the corresponding Cloud Provider and provides it to the client.
- *chunks[] get_file(client name, password, filename)*: accepts a file request from a client, fetches chunks associated with the file from the corresponding Cloud Providers and provides them to the client.
- *chunk get(virtual id as key)*: asks Cloud Provider for a particular chunk. This method is used by the *get_chunk()* and *get_file()* methods described above .

The removal of data can be done by implementing the following functions.

- *remove_chunk(client name, password, filename, sl no.)*: accepts a chunk removal request from a client, forwards the request to the corresponding provider.
- *remove_file(client name, password, filename)*: accepts a file removal request from a client, forwards the request to the corresponding providers.
- *remove(virtual id as key)*: asks Cloud Provider to remove a particular chunk. This method is used by the *remove_chunk()* and *remove_file()* methods described above .

The methods described above can be implemented using *put()*, *get()* and *delete()* method associated with SOAP or REST-based interface for S3 [9].

VII. Feasibility

This section focuses on the effectiveness of the proposed system in preventing data mining. Certain factors such as distribution of chunks, maintaining privacy levels, reducing chunk size, addition of misleading data contribute to this regard. This section also highlights the comparison between encryption and fragmentation as a medium of preserving privacy.

A. Distribution of Chunks

Let us consider an example scenario where a company named Hercules has entrusted a cloud service provider named Titans with its data which includes its history of tender bidding.

TABLE IV
Hercules Bidding History

Year	Company	Materials	Production	Maintenance	Bid
2001	Greece	$1300	$600	$3200	$18111
2002	Rome	$1400	$600	$3300	$18627
2002	Greece	$1900	$800	$3200	$19337
2004	Rome	$1700	$900	$3500	$20078
2005	Greece	$1700	$700	$3100	$18383
2006	Rome	$1800	$800	$3300	$19600
2009	Greece	$1500	$1000	$3600	$20320
2010	Rome	$1700	$900	$3700	$20667
2010	Greece	$1800	$700	$3500	$19937
2011	Rome	$2100	$800	$3700	$21135
2011	Greece	$1900	$1100	$3600	$20945
2011	Rome	$2000	$1000	$3700	$21199

Now, a malicious employee of Titans whose name is Hera has performed some multivariate analysis (linear multiple regression using MATLAB [10]) on the data and has found that the bidding price has been near $(1.4 * Materials + 1.5 * Production + 3.1 * Maintenance) + 5436$ \$ irrespective of the company. If Hera reveals this information to Hydra (a rival of Hercules), Hercules may lose the next bidding.

Now, if Hercules distributes his data equally among 3 providers Titans, Spartans and Yagamis, Hera gets the first four rows of the above table. Multivariate analysis (linear multiple regression using MATLAB [10]) leads to the equation $(1.8 * Materials + 0.8 * Production + 3.4 * Maintenance) + 4489$ \$. Analyzing second set of data combining next four rows leads to the equation $(3.0 * Materials + 4.7 * Production + 2.2 * Maintenance) + 3089$ \$. Finally, analyzing 3rd set of data combining last four rows leads to the equation $(2.4 * Materials + 1.5 * Production + 1.7 * Maintenance) + 8753$ \$. All of these equations are misleading. It is hard to predict the bidding price for next year and thus impossible to beat the Greek superhero. So the example shows a case when distribution of data can prevent data mining.

Distribution of data affects almost all mining algorithms. Regression analysis involving many variables requires many sample cases. Fragmentation of data reduces the number of samples available and thus affect the result. The effect of fragmentation is also evident in case of clustering algorithms as entities may move from their original cluster to other clusters. Prediction algorithms may reveal misleading results as they lack numbers of observations.

B. Maintaining Privacy Level

The proposed system identifies sensitivity of data and maintains 4 privacy levels based on sensitivity. Categorizing data helps to take better initiatives for sensitive data. The proposed system maintains certain properties such as providing data with higher sensitivity to more secured providers, splitting such data into smaller chunks to reduce the risk associated with mining and ensure greater dimension of privacy. Data with privacy level zero are public data. Such data can be split into larger chunks compared to sensitive data. Thus the system minimizes the overhead associated with splitting.

C. Reducing Chunk Size

Mining is strongly associated with large data sets and algorithms often require a large amount of data [26][25]. So splitting data into smaller chunks restricts mining to a great extent. Smaller chunks contain insufficient data. So analyzing such chunks leads to mining failure. The proposed system splits sensitive data into smaller chunks compared to regular data. Thus, it minimizes the privacy risk associated with sensitive data.

D. Addition of Misleading Data

The proposed system provides scope of adding misleading data. Addition of misleading data affects mining results depending on their positions within data. Such data often lead to mining failure.

Misleading data enhances security, but it has some overhead associated with retrieving data. So this approach of enhancing security should be used only when data is not accessed frequently.

E. Encryption vs Fragmenation

Aggarwal et al. [5] described the comparison between encryption and splitting as the medium of maintaining privacy.

Existing proposals of secure database system relies mostly on encryption methods. Data is being encrypted in the trusted client side before it is being stored in the cloud. But encryption has a large disadvantage in the form of overhead associated with query processing [5]. The client has to fetch the whole database, then decrypt it and run queries. Another approach can be running queries on the encrypted data and to post-process the results on the client side. But this approach requires the encryption function to be weak. Weak encryption functions that allows efficient queries leaks too much information which is detrimental to privacy and stronger functions are practically much expensive [5].

The concept of quantam computing is also a threat to the encryption based system. Quantam computers have been shown to have exponential speedups and it implies that a quantam computer could break RSA, Diffie-Hellman and elliptic curve cryptography [7]. So encryption based security has its limitations.

On the other hand, splitting or fragmentation of data also ensures privacy but at much lower cost compared to encryption. It does not have a large overhead associated with query processing like encryption. The fragmentation approach involves splitting a file into chunks and distributing the chunks among various cloud providers. As chunks are pieces of information, no one can access the information as a whole. This approach exploits the benefit of parallel query processing as various fragments can be accessed simultaneously. Some optimized methods of fragmentation can be used like storing the chunks in the locations where they are frequently used (for multi national companies).

Concerned clients can also use encryption along with fragmentation. But encryption is not an alternative to fragmentation, rather it is a complement. Clients can also use partial encryption along with fragmentation, that involves partitioning data and encrypting a portion of it.

VIII. Experimental Evaluation

We have implemented a prototype of our proposed system using Java modules. In our prototype, we have implemented Cloud Data Distributor and Cloud Providers. We have tested the consistency of the system and have monitored its performance (Distribution time).

We have also applied various mining algorithms on various sizes of data. To determine the effect of fragmentation on mining, we have applied binary clustering algorithm on GPS data before and after fragmentation.

A. Experiment Setup

We have used PCs having 2.93 GHz Intel Core 2 Duo processor with 1.87 GB usable memory, running Windows XP service pack 2 edition, as Cloud Providers. Again we have used PCs having 3.09 GHz Intel Xenon processor with a memory of 4 GB, running Windows Server Edition, as Cloud Data Distributor.

The binary clustering of GPS data is done using MATLAB. The GPS data is collected from 30 people living in Dhaka city and using an Android application that provides location based service.

B. Performance Analysis

The dendrogram plot of the hierarchical binary cluster tree of 30 users based on GPS is shown at Figure 4, Figure 5 and Figure 6. Figure 4 corresponds to the clustering of users using more than 3000 observations and Figure 5 and Figure 6 corresponds to clustering using 500 observations. The results obtained using these two approaches (Clustering of entire data, clustering of fragmented data) are different and it is evident from the figures. Many entities have moved from their original cluster to other clusters due to fragmentation of data.

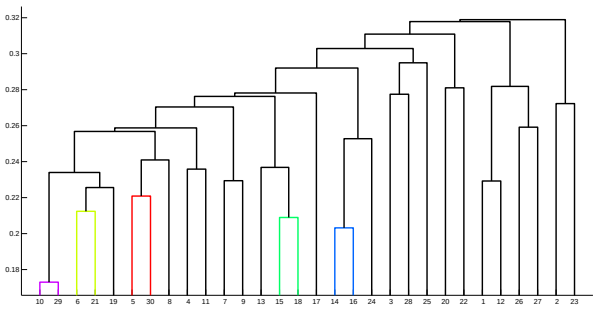

Fig. 4. Dendrogram Plot of Entire GPS Data

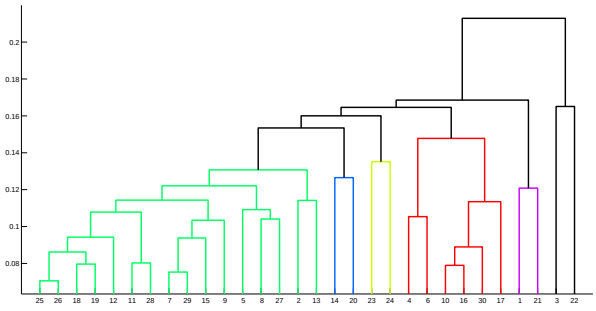

Fig. 5. Dendrogram Plot of Fragmented GPS Data

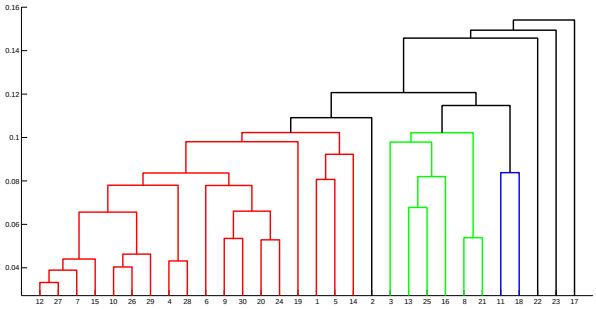

Fig. 6. Dendrogram Plot of Fragmented GPS Data

IX. Related Work

Cloud computing is a rapidly growing field receiving a great amount of attention. Various proposals are given to ensure greater assurance of data in the cloud. One of these proposals involves the concept of multiple Cloud Providers [6]. Another proposal involves multiple Cloud Providers concept combined with the Redundant Array of Independent Disks (RAID) technique to reduce the cost of switching providers and to provide greater assurance of data [4]. The latter strategy introduces RACS, a proxy that distributes storage load over many providers [4]. The Cloud Data Distributor is somewhat similar to RACS in the sense of distributing data among multiple providers. However, RACS focuses on reducing the cost of switching, whereas, our focus is on ensuring the privacy of cloud data. Moreover, RACS is tightly coupled to the S3 model, whereas our system model is open to any cloud architecture.

Recently some works have been done involving data mining on cloud [20][17][27]. Roy et al. [20] proposed a MapReduce-based system to provide security and privacy guarantees for distributed computations on sensitive data. They focus on protecting data privacy during computations [20]. Other proposals tend to use data mining to improve cloud service [17][27].

X. Conclusion and Future Work

Ensuring security of cloud data is still a challenging problem. Cloud service providers as well as other third parties use different data mining techniques to acquire valuable information from user data hosted on the cloud. In this paper, we have discussed the impact of data mining on cloud and have proposed a distributed structure to eliminate mining based privacy threat on cloud data. Our approach combining categorization, fragmentation and distribution, prevents data mining by maintaining privacy levels, splitting data into chunks and storing these chunks of data to appropriate cloud providers.

Although the proposed system provides an effective way to protect privacy from mining based attacks, it introduces performance overhead when client needs to access all data frequently, e.g. client needs to perform a global data analysis on all data. The analysis may have to access data from multiple locations, with a degraded performance. In future, we look forward to improve our system by reducing such overhead.

References

[1] Oracle data mining concepts 11g release 1 (11.1), may 2008.

[2] Introduction to Cloud Computing Architecture by Sun Microsystems, Inc., june 2009.

[3] Amazon Web Services: Overview of Security Processes, may 2011.

[4] H. Abu-Libdeh, L. Princehouse, and H. Weatherspoon. Racs: a case for cloud storage diversity. In *ACM SoCC*, pages 229–240, 2010.

[5] G. Aggarwal, M. Bawa, P. Ganesan, H. Garcia-molina, K. Kenthapadi, R. Motwani, U. Srivastava, D. Thomas, and Y. Xu. Two can keep a secret: A distributed architecture for secure database services. In *In Proc. CIDR*, 2005.

[6] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. H. Katz, A. Konwinski, G. Lee, D. A. Patterson, A. Rabkin, I. Stoica, and M. Zaharia. Above the clouds: A berkeley view of cloud computing. Technical report, EECS Department, University of California, Berkeley, 2009.

[7] P. Asst.Prof .PSV Vachaspati. Quantum attack resistant cloud. *World of Computer Science and Information Technology Journal*, 1:283–288, 2011.

[8] M. Bramer. *Principles of Data Mining*. Springer, 2007.

[9] M. Brantner, D. Florescu, D. A. Graf, D. Kossmann, and T. Kraska. Building a database on s3. In J. T.-L. Wang, editor, *ACM*, pages 251–264, 2008.

[10] S. H. Brown. Multiple linear regression analysis: A matrix approach with matlab. *Alabama Journal of Mathematics*, 2009.

[11] R. Chow, P. Golle, M. Jakobsson, E. Shi, J. Staddon, R. Masuoka, and J. Molina. Controlling data in the cloud : Outsourcing computation without outsourcing control. pages 85–90, 2009.

[12] C. Clifton and D. Marks. Security and privacy implications of data mining. In *ACM SIGMOD Workshop*, pages 15–19, 1996.

[13] S. Ghemawat, H. Gobioff, and S.-T. Leung. The google file system. *SIGOPS Oper. Syst. Rev.*, 37(5):29–43, Oct. 2003.

[14] M. Kantardzic. *Data Mining: Concepts, Models, Methods and Algorithms*. John Wiley & Sons, Inc., 2002.

[15] W. Karim. The Privacy Implications of Personal Locators: Why You Should Think Twice Before Voluntarily Availing Yourself to GPS Monitoring. *Washington University Journal of Law and Policy*, 14:485–515, 2004.

[16] P. Karl Rexer. 2010 data miner survey highlights the views of 735 data miners, 2010.

[17] L. Li and M. Zhang. The strategy of mining association rule based on cloud computing. In *IEEE Computer Society*, pages 475–478, 2011.

[18] Y. Liu, J. Pisharath, W. keng Liao, G. Memik, A. Choudhary, and P. Dubey. Performance evaluation and characterization of scalable data mining algorithms abstract.

[19] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker. A scalable content-addressable network. *SIGCOMM Comput. Commun. Rev.*, 31(4):161–172, Aug. 2001.

[20] I. Roy, S. T. V. Setty, A. Kilzer, V. Shmatikov, and E. Witchel. Airavat: security and privacy for mapreduce. In *Proceedings of the 7th USENIX conference on Networked systems design and implementation*, NSDI'10, pages 20–20, Berkeley, CA, USA, 2010. USENIX Association.

[21] N. Santos, K. P. Gummadi, and R. Rodrigues. Towards trusted cloud computing. In *HOTCLOUD*. USENIX, 2009.

[22] M. J. Shaw, C. Subramaniam, G. W. Tan, and M. E. Welge. Knowledge management and data mining for marketing. *Decis. Support Syst.*, 31(1):127–137, 2001.

[23] D. J. Solove. 'I've Got Nothing to Hide' and Other Misunderstandings of Privacy. *Social Science Research Network Working Paper Series*, 44, July 2007.

[24] I. Stoica, R. Morris, D. Karger, M. F. Kaashoek, and H. Balakrishnan. Chord: A scalable peer-to-peer lookup service for internet applications. *SIGCOMM Comput. Commun. Rev.*, 31(4):149–160, Aug. 2001.

[25] L. Torgo. *Data Mining with R: Learning with Case Studies*. Chapman & Hall/CRC, 2010.

[26] L. Van Wel and L. Royakkers. Ethical issues in web data mining. *Ethics and Inf. Technol.*, 6:129–140, 2004.

[27] J. Wang, J. Wan, Z. Liu, and P. Wang. Data mining of mass storage based on cloud computing. In *IEEE Computer Society*, pages 426–431, 2010.

[28] G. M. Weiss. Data mining in the real world: Experiences, challenges, and recommendations. In *DMIN*, pages 124–130, 2009.

[29] Wikipedia. Amazon elastic compute cloud — Wikipedia, the free encyclopedia, 2012. [Online; accessed 10-May-2011].

[30] Q. Yang and X. Wu. 10 challenging problems in data mining research. *International Journal of Information Technology and Decision Making*, 5(4):597–604, 2006.