DataBank: A Blueprint for efficient privacy-preserving personalized user data management worldwide

Anirban Mondal[1] Pankaj Garg[2] Masaru Kitsuregawa[1]

[1] Institute of Industrial Science
University of Tokyo
Japan
{anirban,kitsure}@tkl.iis.u-tokyo.ac.jp

[2] Intel Technology Group
Tokyo
Japan
pankaj.garg@intel.com

Abstract

The unprecedented increase in the complexity of user-related data coupled with the dramatically growing user dependence on such data motivates a strong need for a new kind of virtual (electronic) institution. We designate such an institution as DataBank. Analogous to a 'bank' in the real-world that provides services related to users' financial data, DataBank provides services associated with the valuable data of users. The main contributions of this work are as follows. First, we propose the blueprint for designing the DataBank and indicate the practical considerations, which must be taken into account, for DataBank to work effectively in the real world. Second, we propose an effective privacy-preserving strategy (for user data) involving instance-based information retrieval, which does *not* require any passwords. Third, we discuss the exciting potential possibilities for DataBank.

1 Introduction

Banks can be viewed as institutions which provide services related to users' financial data. However, the unprecedented increase in the complexity of user-related data coupled with the dramatically growing user dependence on such data suggests that **data** is itself one of the most important *assets* of users. This motivates a strong need for a new kind of virtual (electronic) institution for managing user data. We designate such an institution as *DataBank*. Analogous to a 'bank' in the real-world, which stores financial data and provides services related to users' financial data, DataBank acts as a repository for user data and provides services associated with the user data.

Submitted to COMAD 2005b.

Interestingly, the users of DataBank may be individuals or even organizations. Given that data is also one of the most important *assets* of any corporate organization, a large number of corporate organizations may possibly wish to entrust to DataBank the dual responsibilities of keeping their respective data safe and secure as well as highly available. As a single instance, loss of data in case of a pharmaceutical research company may not only translate to severe financial losses, but also potentially cause loss of the lives of those people who had been waiting to receive new drugs for their respective health disorders. Understandably, managing the data of organizations is significantly more complicated as compared to managing the data of individuals.

DataBank should be able to provide efficient support for various kinds of data. For individual users, the user data can be anything ranging from a customer's personal details (e.g., photograph, fingerprints, distinguishing marks on his/her body, date of birth, residential address, stock portfolio) to highly specialized information associated with his/her medical records. For corporate organizations, the data could be related to past business transactions, important information concerning employees, information related to various projects and so on. We believe that integrating and consolidating all the data pertaining to a user (or organization) can be highly beneficial especially if the user (or members of the organization) is allowed easy, seamless and secure access to data. In this regard, the main contributions of this work are as follows:

1. We propose the blueprint for designing DataBank and indicate the practical considerations, which must be taken into account, for DataBank to work effectively in the real world.

2. We propose an effective privacy-preserving strategy (for user data) involving instance-based information retrieval, which does *not* require any passwords.

3. We discuss the exciting potential possibilities for DataBank.

Notably, safeguarding user data privacy is critical for DataBank because providing data to DataBank puts the user at the risk of his/her data being leaked to potential fraudsters, thereby leading to possible misuse of the data. As a single instance, if a user's credit card information stored in DataBank subsequently falls in the hands of fraudsters, he/she could suffer severe financial losses.

The remainder of this paper is organized as follows. Section 2 discusses possible application scenarios for DataBanks, while Section 3 presents the design of DataBank. The proposed scheme for data classification is described in Section 4, while our proposed strategy for preserving user data privacy is discussed in Section 5. Section 6 discusses some of the possibilities of DataBanking and Section 7 discusses practical considerations associated with implementing a DataBank in the real world. Section 8 provides an overview of related work. Finally, we conclude in Section 9 with directions for future work.

2 Application Scenarios

To understand the tremendous usefulness of DataBank, let us now consider the following three scenarios.

1. *Natural disasters:* Throughout the course of history, natural disasters (e.g., earthquakes, volcanic eruptions and typhoons) have caused severe damage to life and property. For example, the earthquake in Kobe (Japan) on January 17, 1995 left 5470 people dead, 33000 people injured and 300000 people homeless in addition to destroying or severely damaging 319622 buildings and causing financial damage of around US $200 billion [4]. According to the USGS[11], there are 939 earthquakes every year on an average that have a magnitude of 5 or above on the Richter scale[1].

2. *Human-induced disasters:* Large scale disasters induced by human beings (e.g., wars, terrorist attacks) have caused phenomenal damage at various stages of history. For example, the 9/11 tragedy [9] caused the loss of nearly 3,000 lives, wounded scores of people, and destroyed or severely damaged a number of important buildings, including the twin towers of the World Trade Center.

3. *Loss of a customer's passport in a foreign country:* Suppose a citizen of Japan Mr. X makes a business-related trip to New York and unfortunately, he loses some of his travel-related documents (e.g., passport). This may potentially translate to a lot of trouble for Mr. X. Given the current state-of-the art, he would possibly need to request the Embassy of Japan in New York to issue him new copies of his lost travel documents, thereby possibly initiating time-consuming and cumbersome administrative procedures. Now suppose Mr. X has an account with DataBank. In this case, the Embassy officials would just need to verify the details of Mr. X's personal information (e.g., photograph, fingerprints, identification marks on his body) from DataBank. Once DataBank has performed the necessary validation procedures for Mr. X, Mr. X may be able to get new copies of his lost travel-related documents or temporary documents endorsed by the Embassy within a relatively short period of time. We also observe that survivors of plane-crashes may benefit considerably from DataBank because it can facilitate them in proving their respective identities even after they had lost their identification documents due to the plane-crash.

Notably, the above scenarios vary tremendously in their scale, scope, frequency, duration and resulting consequences. For example, scenarios 1 and 2 typically affect a much larger number of people than scenario 3. Moreover, scenarios 1 and 2 may differ in terms of duration and scope i.e., an earthquake may last for only a few minutes in a small geographical region, while a war may last for several months with a much larger scope involving several countries located in diverse geographical regions. Interestingly, despite the differences in these three scenarios, the underlying similarity in each of these three scenarios is that in each case, the data needs to be safe and secure as well as available. Additionally, these scenarios emphasize the extent of disaster control that needs to be in place today.

In case of scenarios 1 and 2, whole streets of buildings may sometimes be wiped out, thereby implying that any important data stored in these buildings would probably be lost foreover. Moreover, individuals typically keep important identification information (e.g., passport) and other important information (e.g., financial papers and legal agreements) in their respective houses, but if the houses themselves are destroyed due to some natural or human-induced disaster, it might take the individuals in question a lot of time to 'rebuild' their very existence. In other words, it would become extremely challenging for them to prove their respective identities without any officially/legally admissible identification papers. But if a person has an account with DataBank, all he needs to do is to log on to his DataBank account and everything he requires to prove his identity and 'rebuild' his life would be at his fingertips.

Even though it may be argued that such events fortunately do *not* occur everyday, it is important to realize that these events are very *real* and they happen

[1]The Richter scale is a means of measuring the magnitude of earthquakes.

to *real* people living in the *real* world, the implication being that these scenarios *cannot* be ignored just because they happen rarely. Given the tremendous dependence of human beings on data in today's world, safeguarding data is more important now than ever before. Hence, loss of data should necessarily be construed as *extremely* serious and therefore we should make every possible effort to ensure that the data is *always* safe and accessible under normal conditions and that the probability of loss of data is minimized as far as possible even in case of extreme situations. Observe that DataBank would also be useful in many instances of daily life primarily because it provides users with *instant* access to their data at any time from anywhere, thereby possibly adding greatly to the convenience of the users. As a single instance, DataBank may help in significantly reducing the need for users to carry huge number of papers containing data because the user can always access his data via DataBank.

3 Design of DataBank

This section discusses the design of DataBank. Just as international banks typically have many branch offices in different parts of the world, DataBank needs to have *dedicated servers* in geographically distributed parts of the globe for optimal coverage in order to maximize user satisfaction. In the interest of amenability, we divide the world into zones and place the dedicated servers in each of the zones.

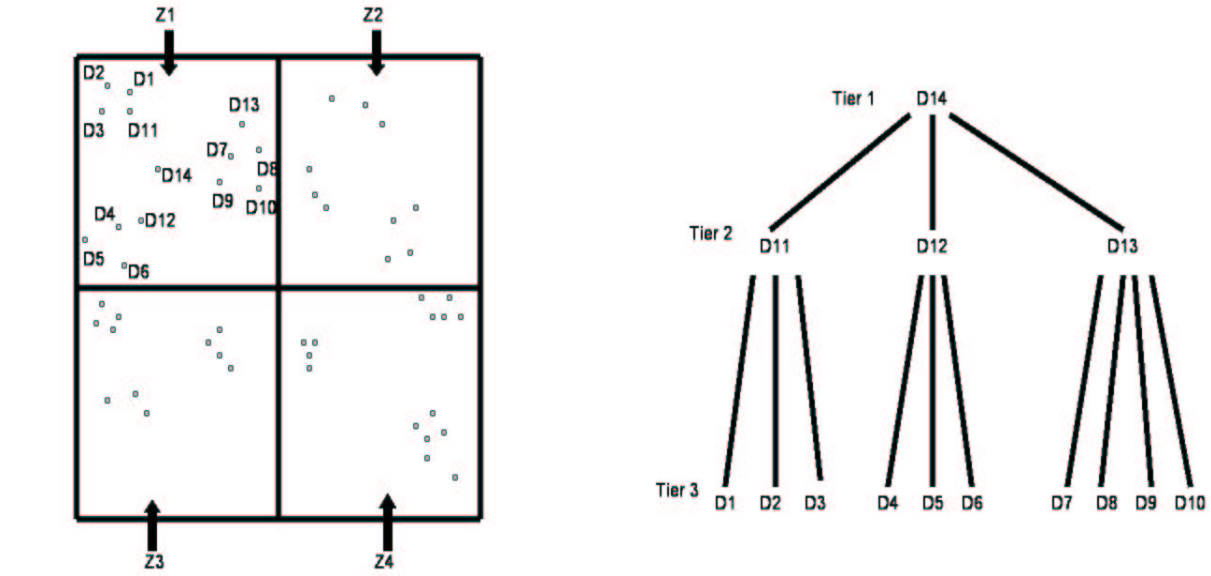
Dividing the world into zones

We divide the entire geographical region encompassed by the world into N *zones* where each zone is a geographical region such that the union of all the N zones encompass the geographic region covered by the whole world. In particular, the zones need *not* necessarily be equal in area especially because different parts of the world typically have significant differences in patterns of usage of IT (Information Technology) resources. Such differences in patterns of IT usage may be attributed to differences in IT infrastructures in different countries as well as skewed population densities in different parts of the world. Moreover, the number of dedicated servers need *not* necessarily be equal across all the zones.

Once the world has been divided into zones, deciding upon the placement of the dedicated servers within a particular zone is an important practical problem. If the number of users in a particular region exceeds a pre-specified threshold, a dedicated server should be placed in that region to speed up user transactions. As a single instance, if DataBank has a *dedicated server* in Tokyo, but not in Osaka, the data of the users in Osaka will need to be managed by the dedicated server in Tokyo, the implication being that the users in Osaka will experience delays in their transactions because of the time taken to transfer their data between Tokyo and Osaka. However, if the number of users in a particular region is too low to warrant a dedicated server in that region, it may *not* be a cost-effective option to place a dedicated server in that region. The number and size of transactions occurring at a particular region should also be taken into consideration when placing dedicated servers. For example, there may be only a few users in a specific region, but these few users may be performing a lot of big transactions (these users may possibly be big organizations). In such cases, it would be reasonable to place a dedicated server in the region under consideration.

We shall use Figure 1 to illustrate the design of DataBank when N=4 i.e., the world has been divided into 4 zones designated as $Z1$, $Z2$, $Z3$ and $Z4$ in Figure 1. Figure 1a depicts the physical locations of the dedicated servers in the four zones. For the sake of clarity, let us take a closer look at $Z1$. Figure 1b displays how the dedicated servers are hierarchically placed in $Z1$. In Figure 1, $D1$ to $D10$ represent dedicated servers which store and manage the data associated with the users of DataBank. In analogy with traditional multi-national banks, these dedicated servers may be regarded as being akin to the city branch-offices of traditional banks. $D11$ represents a dedicated server which stores meta-information concerning $D1$ to $D3$ and supervises the working of $D1$ to $D3$. Similarly, $D12$ and $D13$ stand for dedicated servers which store meta-information and supervise $D4$ to $D6$ and $D7$ to $D10$ respectively. Extending the analogy with traditional multi-national banks, $D11$, $D12$ and $D13$ may be considered as being akin to provincial branch-offices of traditional banks. In Figure 1, $D14$ is a dedicated server which is responsible for supervising the working of $D11$, $D12$ and $D13$ as well as for storing meta-information concerning $D11$, $D12$ and $D13$. $D14$ may be regarded as being akin to the head-office of a traditional multi-national bank for a specific geographical region. We shall henceforth designate this hierarchical tree structure associated with the functionality-based arrangement of the dedicated servers at a specific zone as the *DBank-tree*. Moreover, we designate the dedicated server at the root of the DBank-tree as the *'zone-head'*. In case of Figure 1, $D14$ is the zone-head.

Note that the dedicated servers are also placed in a similar hierarchical manner using the DBank-tree in all the other zones. We specifically observe that each dedicated server, except the dedicated servers at the leaf nodes of the DBank-tree, stores some information concerning the dedicated servers immediately below it in the DBank-tree. Moreover, any interaction between dedicated servers associated with different zones need to occur via the zone-heads of the zones under consideration. We wish to clarify that Figure 1 just provides an example of the proposed structure of DataBank where the number of tiers is 3. However, depending



(a) Physical locations of dedicated servers in the world

(b) DBank-tree for Zone 1

Figure 1: Proposed design of DataBank

upon the circumstances, the number of tiers can be any integer i where $i \geq 1$. (Theoretically, a particular zone may have only one dedicated server.) To generalize, the DBank-tree is a heirarchical tree structure, whose leaf nodes correspond to dedicated servers that store data associated with users. Non-leaf nodes of the DBank-tree contain entries of the form *(DS_{ID}, ptr, Info)* where DS_{ID} refers to the identifier of a particular dedicated server, *ptr* is a pointer to a child node in the DBank-tree and *Info* is the meta-information that the dedicated server DS_{ID} maintains about the dedicated servers at the child node pointed to by *ptr*. Additionally, the heights of the respective DBank-trees in different zones need *not* necessarily be the same.

Data Management

Given that replication of user data becomes a necessity to guarantee robustness of DataBank in the face of security threats, a particular user's data may be replicated at a number of different dedicated servers at different zones. Moreover, as noted in the application scenarios 1 and 2 discussed in Section 2, the data also needs to be safeguarded against security threats that can potentially wipe out whole streets of buildings. The implication is that the data of each user needs to be replicated at geographically distributed areas. However, there would be only one primary copy of a particular user's data in the entire system. This primary copy may reside at the dedicated server nearest to the address (e.g., the user's residential address or office address) that the user provides to DataBank when he/she opens an account with DataBank. The implication is that if a user accesses his/her DataBank account from a place, which is under the jurisdiction of a dedicated server other than the one in which his/her data is stored, the transaction needs to be performed via the dedicated server storing his data. Additionally, all updates need to be performed first on the primary copy and then the updates should be propagated to the replicas.

4 Data Classification Scheme

Intuitively we can understand that different kinds of data require to be handled differently. For example, static data does *not* require techniques for maintenance of replica consistency, while efficient protocols for maintaining the consistency of replicas become a necessity for dynamically changing data. To facilitate efficient data management, it becomes important to classify the data based on certain directions (viewpoints) so that we can develop a priority-based scheme to qualitatively and quantitatively decide the importance of a certain data item with respect to a certain direction. The higher the priority of a data item D in a given direction, the more we should maximize our efforts concerning D in that direction. For example, from the direction of security, non-sensitive data (e.g., music and/or video files) would be considered to be of low priority since these non-sensitive data can be reasonably expected to require less stringent security measures as compared to highly sensitive data (e.g., financial records) that should be regarded as being of high priority from the viewpoint of security. However, from the viewpoint of data dynamism, financial records pertaining to fixed-deposit investments would be regarded as being of low priority on the dynamism scale, while financial records associated with savings account may be considered to be of high priority on

the dynamism scale. Keeping these points in mind, we propose that the data should be classified according to sensitivity, dynamism and goals associated with the data.

Data Sensitivity: Access Control and Authorization

Each user has different kinds of data associated with himself/herself. Some of the data associated with a user may be *highly sensitive*, while other data related to the same user may be of low sensitivity. By *highly sensitive data*, we imply those data which we cannot afford to lose because loss of such data would lead to severe consequences e.g., loss of records of financial transactions. On the other hand, *data with low sensitivity* refers to those data, the loss of which causes only minor problems as opposed to any grave consequences e.g., loss of data concerning a person's height or weight. To facilitate effective access control and authorization, we propose a classification of data based on sensitivity. Here, we present a classification scheme that divides every user's data into three types.

- Type 1 information: This refers to a user's personal information which should be accessible *only* to the user himself and *nobody else* should be able to access this information. Examples of such information may include a user's financial and medical records.
- Type 2 information: This type of information should be accessible to the user as well as a selected group of people that the user has nominated for viewing this information. Examples of such information may be music and movie files or some photographs of certain important occasions which the user has decided to share with some of his friends.
- Type 3 information: This type of information is extremely general information which the user wants other people to view and hence, this information can be accessed by anyone. Examples of this kind of information include a user's website or his/her ideas on important international, social, religious and economic issues.

Note that we do *not* advocate that data with low sensitivity should be provided with little protection just because their loss is not as serious as that of the loss of highly sensitive data. In contrast, our recommendation is that *all* data should be provided with certain adequate baseline protection so that no data is lost. However, we believe that the more sensitive the data is, the more additional protection measures it must be provided with. This facilitates separation of concerns between highly sensitive data and data with low sensitivity, thereby enabling us to direct our best efforts towards safeguarding the more sensitive data.

For every given data item d_i, the user is allowed to decide the category in which he/she wishes to place d_i. Hence, the classification scheme is subjective to the user's interpretation of the classification categories. For example, a particular user Mr. Y may decide to place his curriculum vitae in the Type 1 category because he regards it as highly personal as well as confidential information, while another user Mr. Z, who is currently searching for a job, may place his curriculum vitae in the Type 3 category because he wants prospective employers to view his curriculum vitae. We would also like to point out that the above classification scheme is only a representative one and many other classification schemes with possibly more detailed categories can also be adopted, depending upon users' requirements and demands. In essence, the above classification scheme just indicates how access control can be performed when the relative importance of different parts of the data pertaining to the same user varies to a significant extent.

Dealing with differences in dynamism of data

Interestingly, some of the data associated with a particular user may typically remain static over time, while some of the user's data may change dynamically. Relatively static data and highly dynamic data need to be dealt with differently because unlike highly dynamic data, relatively static data does *not* require efficient mechanisms for replica consistency maintenance and concurrency control. To address the differences in dynamism of the data, we present a scheme that classifies data into either one of the following categories based on the span of time for which the data can be reasonably expected to remain static.

- *Relatively static data:* Relatively static data refers to data that can be usually expected to remain static and even though updates/changes to this kind of data are theoretically possible, such updates are extremely rare in the real world. Examples of relatively static data include a user's name, date of birth, place of birth, DNA information and his/her passport number. Understandably, even though it is possible for a user to change his/her name via legal procedures, such changes are usually seldom and may happen (if at all) possibly only once in several years for a vast majority of the users. Similarly, a user's passport number may also change if a person changes his citizenship or for other reasons, but once again, this kind of change usually happens, if at all, once in several years.
- *Dynamic data:* Dynamic data implies those data that are likely to change quite frequently. An example of dynamic data is the financial data associated with a user's stock portfolio because prices of shares vary from day to day, thereby indicating

that the data is likely to change everyday or even a number of times in each day depending upon the financial markets.

Handling differences in data-oriented goals: 24 by 7 availability vs archival storage

It is of paramount importance to identify the goals of users in storing their data in DataBank. While some users need DataBank to provide them with secure archival storage, others may need DataBank for access to their data at any time from any place.

Recall that the users of DataBank may be individuals or even organizations. Now let us consider the case of a traditional bank being a user of DataBank. A moment's thought indicates that protecting the records of all the users' transactions is much more important to the bank than maintaining 24 by 7 availability of the users' data. Even if a few records of the transactions of a particular user of the bank are lost, the consequences would be extremely serious for the bank under consideration as it would put the bank's very credibility at stake. In contrast, if a particular user is unable to access his/her data a few times, the user would probably be angry, but the consequences would clearly not be as serious as the loss of possibly even a few records of the user's transactions. Note that we are *not* suggesting that availability is *not* important for a bank, but what we are trying to state is that preserving *all* the records of a user's transactions is much more important to a bank than providing 24 by 7 data availability to its users.

Now let us consider the case of an air-traffic control organization, which is a user of DataBank. An air-traffic control organization typically comprises a set of skilled personnel operating on air-traffic control machines. Incidentally, an air-traffic control system needs to work with real-time data. If the personnel associated with the air-traffic control system do *not* have access to real-time data for even a few minutes, the consequences could be disastrous in that airplane crashes may occur, thereby possibly resulting in loss of human lives. On the other hand, loss of some records of the past would probably *not* lead to such severe consequences. Hence, 24 by 7 data availability is much more critical to an air-traffic control system than archival storage. Once again observe that we are *not* trying to imply that archival storage is *not* important for the data associated with an air-traffic control system, but what we are trying to indicate is that maintenance of high availability is much more important in this specific case as compared to archival storage.

The summary of our proposed data classification scheme is depicted in Table 1.

5 Privacy-preserving data storage management in DataSafe

This section discusses how the user data are stored and managed in a privacy-preserving manner in DataBank.

Creation of unique user identifiers (UIDs)

Each user of DataBank is assigned a unique identifier *UID*. UID is created by concatenating *uid* with *En*. For the k^{th} user, who registered with DataBank, *En* is assigned the value of k, thereby ensuring the uniqueness of *En*. Now let us see how *uid* is created for each user.

When a new user X creates an account with DataBank, he/she is required to fill up a registration form that requires some of his/her personal details which are highly likely to remain relatively static over time e.g., country of birth, date of birth and passport number. X's registration form is stored at DataBank. The data in X's registration form consists of a finite maximum number N_F of fields. Let us refer to these fields as f_1, f_2, f_3,....,f_{N_F}. Notably, some of these fields are *compulsory* (i.e., users *have to* enter data in these fields), while other fields may be left blank by the user. Now a pre-defined set of n *compulsory* fields are selected and then X's *uid* is created according to a pre-defined scheme *Sch* as follows.

$$uid_X = ExtrConcatOrder\ (\ d_1, d_2, ..., d_{n-1}, d_n\ ) \quad (1)$$

where d_i denotes the data in the i^{th} selected compulsory field and *ExtrConcatOrder* is a function which first extracts the part of a given field's data d_i as specified by the scheme *Sch*, and then performs string concatenation of these extracted parts in the *order* that *Sch* specifies, for creating the *uid*. Notably, the same scheme *Sch* is used for generating the *UID* of every user.

Figure 2 depicts an illustrative example to indicate how *uid*s of two different users, Y and Z, are created. In Figure 2, the selected compulsory fields are *code of country of birth* (CCode), *date of birth* (DOB) and *zipcode of permanent address* (ZIPN). The scheme for creating the *uid*s for both Y and Z uses the first digit of CCode, the first four digits of DOB and the last three digits of ZIPN.

For simplicity, we have shown only three compulsory fields in the illustrative example of Figure 2. However, in practice, *Sch* would use a larger number of compulsory fields for creating *uid*s. Moreover, the data in the compulsory fields can be encrypted for facilitating added privacy. Note that it is possible for multiple users to have the same *uid* depending upon their data in the compulsory fields. However, the uniqueness of *En* guarantees the uniqueness of *UID*.

Incidentally, users may need to update their data, thereby implying that the data in the compulsory fields and/or the optional fields may change over time. In

Directions	Classes	Examples
Access control and authorization	Type 1	Medical records
	Type 2	Photographs shared with friends
	Type 3	Information that one provides on one's website
Dynamism of the data	Relatively Static	Passport data
	Dynamic	Financial data
Goal associated with the data	Archival Storage	Data of Financial transactions
	24 by 7 availability	Data of Air-traffic control

Table 1: Summary of proposed data classification scheme

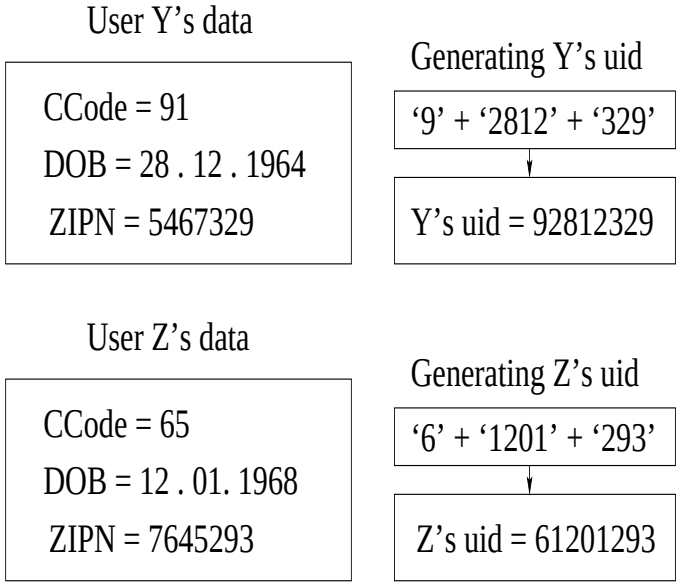


Figure 2: Illustrative example of *uid* creation

case the optional data fields change, no action needs to be taken. However, if the data changes in *any* of the compulsory fields of a user X (even compulsory fields that are not used in the pre-defined scheme Sch), we treat X as a new user. We assign X a new value of En and re-compute his/her *uid* based on the fresh data in the compulsory fields, thereby generating a new *UID* for X. Then we issue queries on DataBank to retrieve all the data stored by X and then we re-insert X's data into DataBank and associate this data with the new *UID*, following which the old *UID* is deleted from DataBank. Since we create *UID*s based on compulsory data fields that have a high probability of remaining static over time, the maintenance overhead associated with generating a new *UID* can be expected to be practically negligible. As an example, it is possible for the passport number of a specific user to change, but such changes occur (if at all) very infrequently.

User data storage in DataBank

Recall that user data (in the registration form) consists of N_F fields, which we refer to as f_1 to f_{N_F}. We maintain N_F different tables for storing user data. Let us designate these tables T_1, T_2,...., T_{N_F}. Each field of a given user X's data is stored in a separate table as follows. We hash the value of X's *UID* onto a value i such that $0< i \leq N_F$ and assign the first field f_1 of X's data to be stored at table T_i. An example of a simple hash function for performing this is $i = (uid \text{ modulo } N_F) + 1$. Notably, more complicated hash functions can also be used for hashing *UID*s. The table identifier *tid* for accessing data in X's f_1 is simply X's *uid* concatenated with X's En.

We arrange the numbers 1 to N_F in a circular modulo ring. The second field f_2 of the user data is placed in the table which corresponds to the next number (traversing clockwise) in the circular modulo ring, the third field f_3 is placed in the table that corresponds to the next number (traversing clockwise) and so on. In general, the i^{th} field of the user data is assigned to the table corresponding to the i^{th} number (traversing clockwise) from the number corresponding to the *uid* of the user. The table identifier *tid* for accessing X's data in f_2 is obtained by adding 1 to X's *uid* and then concatenating with X's En. Similarly, the *tid* for accessing the third field of X's data is obtained by adding 2 to X's *uid* and then concatenating with X's En. In general, given a specific user and a particular table, the *tid* for accessing that user's j^{th} field's data is formed by adding the user's *uid* to (j - 1), and then concatenating with the user's En. Figure 3 shows the algorithm for storing a particular user's data at the different tables.

Observe that our privacy-preserving scheme stores different fields of a given user's data at different tables and uses different identifiers (i.e., *tid*s) in each of these tables for accessing the same user's data. Hence, even if a potentially malicious user manages to gain unauthorized access in some way, he would still need to know the scheme Sch for creating *uid*s, the En for a given user, as well as the scheme for generating *tid*s in order to be able to understand the contents of the tables. Notably, this information is known only to DataBank system administrators, thereby ensuring effectiveness of our proposed privacy-preserving scheme. We stipulate that only a subset of this information should be known to each system administrator, the implication being that multiple system administrators

need to collaborate to retrieve the data and that no single system administrator can compromise privacy.

Algorithm Store_Data
Inputs: 1) d_1 to d_n, user data in the N_F fields
2) User's *uid*
3) Pre-defined scheme for hashing *uid*s $UScheme$
4) En, the user's entry number in the system
5) $Circle$, a modulo ring with numbers 1 to N_F
Output: User data stored successfully in DataBank

Create N_F empty tables, namely T_1 to T_{N_F}, each table having two columns called *tid* and *data*
Use $UScheme$ to hash *uid* onto an integer *val*

for $j = 1$ to N_F
Store d_j in table T_{val}
Compute *tid* as the concatenation of (*uid*+ (j-1)) and En
val = next number in $Circle$ traversed clockwise

end

Figure 3: Algorithm for storing a user's data in DataBank

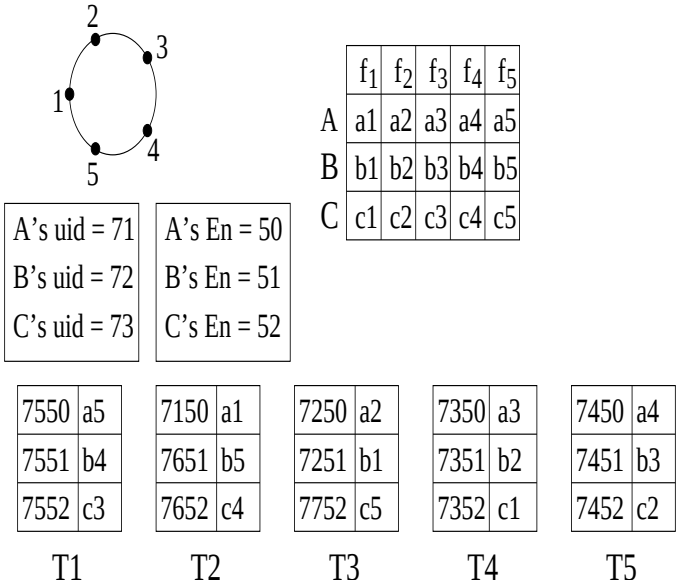


Figure 4: Illustrative example of data storage in DataBank

Figure 4 depicts an illustrative example indicating how the above scheme stores the data of three different users, namely A, B and C, in DataBank. In Figure 4, $N_F = 5$, hence user data has five fields f_1 to f_5. A's data for these five fields is denoted by a1 to a5, B's data is represented as b1 to b5 for these five fields and C's data for these fields is c1 to c5. We use a simple hash function $i = (uid \text{ modulo } N_F) + 1$ for this illustrative example. Figure 4 shows five tables $T1$ to $T5$. As the figure indicates, A's *uid* is 71, hence the value of i for A is 2. Hence, A's first field's data (i.e., a1) is stored in $T2$ with the *tid* for accessing a1 being 7150, formed by concatenating A's *uid* and A's En. Since 3 comes after 2 (in the clockwise direction) in the modulo ring, A's second field's data (i.e., a2) is stored at table $T3$. The *tid* for accessing a2 is formed by adding 1 to A's *uid* and then concatenating with A's En, thereby obtaining 7250. Similarly, a3 is stored at $T4$ with its *tid* being formed by adding 2 to A's *uid* and then concatenating with A's En to obtain 7350. The other entries in the tables $T1$ to $T5$ are populated in a similar manner.

Instance-based user information retrieval

We propose an instance-based information retrieval strategy for authenticating users. In case a user X needs to retrieve some of his data from DataBank, he/she issues a query to DataBank. DataBank will ask him several questions from the registration form and see whether his current answers match with that of the answers that he provided in his registration form at the time of registration. If the answers match, the user's identity is authenticated and verified by DataBank.

Using the same scheme that was deployed for creating *uid*s and finding out the user's En value from the user, DataBank computes the *UID* of X. Using the same hashing scheme that was used for hashing *uid*s, we hash the *uid* onto a value *val*. Now the first field of X's data can be found from the table T_{val}. The second field of X's data can be found in the next table i.e., the table corresponding to the next number in the modulo circle comprising numbers 1 to N_F. In general, the i^{th} field of X's data can be found from the table that corresponds to the i^{th} number after *val* in the modulo circle. The algorithm for query processing in DataBank is depicted in Figure 5.

Observe that our proposed instance-based information retrieval scheme requires the user to remember only his own personal information and the value of En for his DataBank account. (En may be regarded as a user's DataBank account number.) This is in contrast with existing services (e.g., email sites, online stores) which typically require users to remember their account numbers as well as *passwords*, which are often typically cumbersome and difficult to remember. Interestingly, similar to our approach, existing services also pose questions to users, who have forgotten their passwords, either to help them remember their passwords or to send them their passwords in case they are able to answer the questions correctly. However, in these cases, access to user data still occurs via passwords, which is different from our approach, which has no notion of passwords.

Algorithm Process_Query
Inputs: 1) User's *uid*
2) Pre-defined scheme for hashing *uids* $UScheme$
3) En, the user's entry number in the system
4) $Circle$, a modulo ring with numbers 1 to N_F
Output: User data retrieved successfully from DataBank

Use $UScheme$ to hash *uid* onto an integer *val*

for $j = 1$ to N_F
Compute *tid* as the concatenation of (*uid*+ (j-1)) and En
Retrieve d_j in table T_{val} using *tid*
val = next number in $Circle$ traversed clockwise

end

Figure 5: Algorithm for authenticating a user

6 Possibilities of DataBanking

DataBanks open a world of exciting possibilities for information sharing among the customers. This section looks at some of the possibilities of DataBanking which may potentially enhance customer satisfaction significantly.

1. Peer-to-peer (P2P) applications: Interestingly, a DataBank can also potentially perform the functions performed by existing P2P systems. P2P interactions and P2P functionalities in DataBanking apply to data of Type 3 (discussed earlier in Section 4) since data of Type 3 can be shared between users. For example, a user may wish to find out whether anyone in the DataBank has a particular document, song or video. For this purpose, efficient mechanisms for search and replication need to be in place. We can draw upon existing research associated with P2P networks.

2. *Data mining:* Recall that users of DataBank may be individuals or corporate organizations. An investment company may be interested in finding out patterns of customer investments in order to provide improved investment options from the customers' perspective. An insurance company may be interested in mining insurance patterns to provide potential customers with better insurance options, thereby boosting their own business in the process. A research organization may be interested to look at the data of other research organizations to determine the amount of investment in terms of time and money to complete an important piece of research e.g., pharmaceutical companies, software research companies. The whole point is that given the vast expanse of information in DataBank, efficient data mining techniques need to be in place to exploit this information as optimally as possible. Mining this information may provide interesting patterns, which may facilitate individuals as well as organizations in making important decisions.

3. *Services associated with providing alerts to customers:* With so many thousands of Gigabytes of data being in circulation in the world of today, it is often extremely difficult to locate interesting data in an efficient and timely manner. Observe that DataBank stores and manages a lot of data associated with different users, the implication being that it is possible for a particular user of DataBank to be interested in the services provided by other users of DataBank and/or the data of other users of DataBank. Now let us look at the following two examples to understand how DataBank can alert its users so as to improve user convenience.

   - *Introductory services:* Suppose a particular user A is a company which manufactures and repairs ships and A wishes to find out which users of DataBank are suppliers of ships' parts and also which users deal in shipping insurance possibly because A may be interested in doing business with some of these companies. Assume users B, C, D and E of DataBank are companies which specialize in supplying ship parts and users F, G and H are insurance companies which specifically deal with ship-related insurance. Interestingly, in this case, DataBank is able to provide A with the valuable opportunity to select which companies it wishes to enter into business dealings with.

     Even in the absence of DataBank, A could possibly still have managed to establish business contacts with suppliers of ship parts and also shipping insurance companies via the Internet or via other means. Observe that choosing business partners from among the users of DataBank can be reasonably expected to be much more secure than selecting business partners arbitrarily from the Internet primarily because DataBank has the ability to function as the regulatory authority for the data of its users and also all the users of DataBank have to be properly registered with all the registration details of the users being verified rigorously by governmental and law enforcement authorities, thereby reducing the probability of scams and frauds, which have plagued the Internet significantly over the last decade. Additionally, choosing business partners from DataBank is convenient and easy since all that A has to do is to submit a query to DataBank asking what type of users A is interested in and the rest

of the process of finding interesting users for A's set of preferences is handled completely by DataBank itself.

Even though the above example concerned corporate organizations, DataBank can be equally beneficial to individuals. As a single instance, a user X of DataBank may be interested in buying a product K, while there may be many users of DataBank which sell product K. In this case also, the DataBank is able to provide the useful service of introducing X to potential sellers of the product which X is interested in.

- *Real-time alerts:* Some DataBank users may wish to know when the value of their stock portfolio falls below a certain threshold or when a stock in which they have invested (or plan to invest) rises or falls. DataBank can provide this service by issuing continuous queries on the data of the stock market to monitor current stock rates and issue alerts to its users whenever any of the trigger conditions specified by the users is satisfied. Observe that real-time alerts to users can be provided by other services even in the absence of DataBank, but our point here is that DataBank is *also* able to provide this important service.

4. *The 'paperless' concept:* Interestingly, the electronic data handling performed by DataBank may probably reduce the use of papers, the objective being to eventually have 'paperless' offices and administrative procedures. Imagine the tremendous savings in cost if everything starting from daily office work to highly complex bureaucratic administrative procedures could be performed via using electronic data without using any papers. Understandably, governments all over the world would also be able to save plenty of money by adopting paperless offices. Notably, electronic data backup is relatively easy as compared to making copies of important papers. Additionally, electronic data is much easier to update than data on paper especially in cases where geographically distributed replicas need to be in place (for dealing with emergencies such as earthquakes). Moreover, electronically updating the replicas of the data is faster as well as cheaper as compared to sending updates to the data via postal services. Clearly, browsing files on a computer is also much easier than browsing through archived files of paper in a huge library building. With the cost of data storage decreasing, the option of electronic data handling is becoming even more attractive.

   Also, as a pleasant side-effect, people would *not* need to carry any important paper documents and the question of losing their important data would *not* even arise because DataBank maintains a legally admissible copy of the data which is also highly available. This would indeed add significantly to the comfort of the users. What we wish to imply is that we envisage DataBank not only as an institution for protecting the data and keeping the data highly available, but our vision is that eventually the aim of DataBank should be to replace traditional paper-based procedures by procedures involving *only* electronic data. Understandably, we expect it to take time and investments, but in the long run, the savings will far outweigh the investments made to build and maintain DataBank.

7 Discussion

DataBank represents revolutionary changes in the landscape of customer data management worldwide. In contrast to traditional brick-and-mortar banks, which typically deal with only financial issues, the scope of DataBank is much wider because it is expected to deal with not only financial information, but also all the relevant information associated with its users. Hence, intuitively we can understand that legal issues, security issues and regulatory aspects, which are typically addressed by traditional banks, become significantly more complex in case of DataBank as compared to traditional banks. The increase in the complexity of these challenges associated with DataBank partly arises from electronic data handling and partly due to the fact that DataBank transcends geographical boundaries (i.e., transactions occur via the Internet), thereby facilitating 'borderless' transactions.

From the legal perspective, new legal definitions need to be adopted to recognize digital signatures as a legally admissible identification mechanism. Moreover, given the presence of 'borderless' transactions, new agreements and collaborations between different countries need to be in place in order to incorporate the presence of these 'borderless' transactions. For example, if a person Mr. Z in country A performs a fraudulent transaction (e.g., 'hacking' an account and stealing money and/or accessing information in an unauthorized manner) on another person Mr. Y's DataBank account in country B, countries A and B need to collaborate to bring Mr. Z to justice. Interestingly, this is a general problem concerning the Internet because Internet law is still in its infancy, but fortunately many countries have been making significant efforts to prevent criminal activities on the Internet.

Just as traditional brick-and-mortar banks need to interact with each other for financial transactions across banks, different Databanks would also need to interact with each other. We shall use the term *'inter-DataBanking'* to imply interactions between two or

more different DataBanks. For inter-DataBanking to be effective in practice, issues associated with federated databases need to be adequately addressed i.e., different DataBanks can be reasonably expected to maintain their data using different schemas for their databases as well as different semantics. This is essentially an interesting as well as a challenging problem which arises in cases of all the applications involving federated databases, the implication being that we can draw upon existing research in federated databases to address this problem. However, it is important to remember that the problem in case of DataBanks would be significantly more complicated as compared to that of the problems concerning traditional federated databases partly because of the wide variety of data that DataBanks would be required to handle and partly due to the fact that the scale and scope of DataBanks can be expected to be much larger than that of existing federated databases. To deal with the issue of difference in semantics of the data stored by different DataBanks, standardization would be required to precisely define the semantics associated with the data. We can draw upon the results of XML-related research to handle semantic issues. However, the most effective way of performing inter-DataBanking is still an open question and it is clear that the answer to this question requires further research.

The meticulous reader might have observed that we have referred to DataBank as an *institution* rather than a *service*, the primary reason being that there are two options for implementing DataBank in the real world.

1. Creation of DataBank as a new institution
2. Signing of collaborative agreements with major multi-national banks such that these multinational banks extend their services to the creation and maintenance of DataBank.

Both the above approaches seem to be fairly reasonable for implementing DataBank. Our own opinion concerning this issue is that DataBank should be created as a new institution especially because we believe that that user demands on DataBank as well as services provided by DataBank may typically be expected to increase tremendously over the next decade, the implication being that during the next decade, the services provided by traditional brick-and-mortar banks may be just a minor fraction of the services provided by DataBank. Moreover, for reasons of cost-effectiveness, we advocate that any DataBank implementation should also try to use as much as possible of the existing technological infrastructures of traditional banks.

8 Related Work

Integration and consolidation of data from geographically distributed locations has motivated a lot of research efforts. In particular, the concept of GRID computing[8] has some similarities with our proposal concerning DataBank in that both address data integration and consolidation over wide-area networks. GRID computing essentially relates to the massive integration and virtualization of geographically distributed computing resources, thereby allowing the user to see a unified single system image. However, the main difference between our proposal and existing works on GRIDs is that GRIDs have been specifically designed for use by organizations, while DataBanks need to be designed for use by individual users too. Notably, in contrast with existing GRID systems where significant amount of programming expertise is often required to perform jobs, it is clear that we cannot expect a DataBank user to do a lot of programming to get his/her job done because that would make DataBank less attractive to most individual users. In other words, keeping in mind that user-friendliness is absolutely critical to the success of DataBank, customers should be able to get their DataBank-related jobs done with just a few mouse-clicks.

Representative examples of important ongoing GRID computing projects include the Earth Systems GRID (ESG)[1], the NASA Information Power GRID (IPG)[10], the GRID Physics Network (GriPhyN)[3] and the European DataGrid[2]. While the ESG project aims at facilitating detailed analysis of huge amounts of climate data by a geographically distributed community via high bandwidth networks, the IPG project attempts to improve existing systems in NASA for solving complex scientific problems efficiently. The GriPhyN project and the European DataGrid project both aim at employing GRID systems for improving scientific research which require efficient distributed handling of data in the petabyte range. Existing works [7, 13] have noted the demanding I/O needs of GRID applications. While the proposal in [7] discusses the design of a data GRID for data-intensive petabyte applications, the work in [13] proposes the binding of execution and storage sites together into I/O communities that participate in the wide area system. The proposal in [12] describes a data-movement system (Kangaroo) which makes opportunistic use of resources (disks and networks), while hiding network storage devices behind memory and disk buffers such that background processes handle data movements. It aims at availability and reliability by sacrificing consistency guarantees.

The Protein Data Bank (PDB) [6] is the single worldwide repository for the processing and distribution of 3-D biological macromolecular structure data i.e., the structure data of large molecules of proteins and nucleic acids. Data processing in the PDB consists of data deposition, annotation and validation, one of the key goals being to maintain the consistency and accuracy of the archive. An integrated system of het-

erogeneous databases has been created that store and organize the structural data. Data depositors to the PDB comprise a very diverse group of researchers in biology, chemistry and computer science, educators and students at all levels, who have varying expertise in the techniques of X-ray crystal structure determination, cryoelectron microscopy and theoretical modelling. Further details concerning the PDB can be found in [5].

The World Bank Group [14] makes available data profiles from the World Development Indicators database. The profiles cover 208 countries (184 World Bank members and 24 other economies with populations of more than 30,000) and 18 country groups reported in the World Development Indicators book, CD-ROM and the World Bank Atlas. The World Development Indicators database is aimed at performing cross-country comparisons. Various kinds of statistical information about different countries concerning education levels, gender ratios, health and nutrition details, population summary , social and economic factors, and computers and IT (Information Technology) usage are provided by the World Bank Group.

Observe that existing data banks such as the Protein Data Bank and the World Bank Group are typically domain-specific and are designed to be used by organizations as opposed to individual users. The scope of our proposed DataBank is much broader because in contrast to these data banks that typically store detailed information concerning a specific topic, DataBank aims at storing federated information about a wide variety of topics concerning a specific user.

9 Conclusion

In this paper, we have introduced the concept of a DataBank. Given the dramatically increasing user dependence on data in today's world, we believe that DataBank, which is an institution for storing and managing user data securely, is the need of the hour. In particular, we wish to emphasize that DataBank goes well beyond the traditional financial resource handling performed by traditional banks to storing and managing all the important information associated with a customer. Moreover, DataBank allows its users to access their respective data from *anywhere* at *anytime* without requiring them to bother about the complex processes which mediate their access to their respective data. To safeguard user data privacy, we have proposed a privacy-preserving scheme for DataBank. But there are still several open questions which need to be researched extensively for DataBank to work efficiently and effectively in practice. For example, detailed issues concerning replication such as the placement of replicas (for optimizing response times of users) and efficient protocols for maintenance of replica consistency need to be investigated. Additionally, concurrency control mechanisms for DataBank also need to be considered in detail. We intend to address these open questions in the near future. Given that DataBank opens several novel and interesting avenues for research, we encourage experts from academia as well as from industry to research and debate upon the open questions concerning DataBank, thereby ensuring the success of DataBank in the real world.

References

[1] The Earth Systems GRID Project. *http://www.earthsystemgrid.org/.*

[2] The European Datagrid Project. *http://eu-datagrid.web.cern.ch/eu-datagrid/.*

[3] The Griphyn Project Webpage. *http://www.griphyn.org/index.php.*

[4] The Kobe Earthquake Disaster. *http://www.disinc.com/kobe.htm.*

[5] Protein Data Bank. *http://www.rcsb.org/pdb/.*

[6] H.M. Berman, J. Westbrook, Z. Feng, G. Gilliland, T.N. Bhat, H. Weissig, I.N. Shindyalov, and P.E. Bourne. The Protein Data Bank. *Nucleic Acids Research*, 28:235–242, 2000.

[7] A. Chervenak, I. Foster, C. Kesselman, C. Salisbury, and S. Tuecke. The Data GRID: Towards an architecture for the distributed management and analysis of large scientific datasets. *Proc. Network Storage Symposium*, 1999.

[8] I. Foster and C. Kesselman. The GRID: Blueprint for a new computing infrastructure. *Morgan-Kaufmann*, 1999.

[9] 9/11 incident. http://en.wikipedia.org/wiki/.

[10] NASA IPG. http://www.ipg.nasa.gov/.

[11] USGS Earthquake Hazards Programme. http://wwwneic.cr.usgs.gov/neis/eqlists/sig_2004.html.

[12] D. Thain, J. Basney, S.C. Son, and M. Livny. The Kangaroo approach to data movement on the GRID. *Proc. HPDC*, 2001.

[13] D. Thain, J. Bent, A. Arpaci-Dusseau, R. Arpaci-Dusseau, and M. Livny. Gathering at the well: Creating communities for GRID I/O. *Proc. SC*, 2001.

[14] The World Bank Group Website. http://www.worldbank.org/data/countrydata/countrydata.html.