International Journal of Research in Engineering, Science and Management
Volume-3, Issue-2, February-2020
www.ijresm.com | ISSN (Online): 2581-5792

# Smart Decision-Making Utility Comprising Better Security and Fairness at Airport Terminal

Anitha Moses[1], Sharon A. Ratna[2], S. Sushma[3], K. P. Lavanya[4]

[1]*Associate Professor, Department of Computer Science and Engineering, Panimalar Engineering College, Chennai, India*

[2,3,4]*Student, Department of Computer Science and Engineering, Panimalar Engineering College, Chennai, India*

***Abstract*: As profiling and automated processing of information emerge as enablers for more efficient, risk-based and smarter security, growing concerns on ethics and privacy are reflected on the adapting regulatory and legal framework, as expressed in relevant sections of the General Data Protection Regulation (GDPR). In this context and by examining the airport checkpoint as the most challenging and regulated security case we propose a solution monitoring the fairness of intelligent surveillance systems of an airport and any critical infrastructure. The embedded algorithms receive input from distributed sensors and high-level information and infer suspicious incidents and visitors' trustfulness level.**

***Keywords*: Airport terminal automation, GDPR, Information processing, Profiling, Security, Smart decision making.**

## 1. Introduction

The Web has evolved into a data-rich repository containing a large amount of structured content spread across millions of sources. The usefulness of Web data increases exponentially (e.g., building knowledge bases, Web-scale data analytics) when it is linked across numerous sources. Structured data on the Web resides in Web databases and Web tables. Web data integration is an important component of many applications collecting data from Web databases, such as Web data warehousing (e.g. Google and Bing Shopping; Google Scholar), data aggregation (e.g. product and service reviews), and meta searching.

Integration systems at Web scale need to automatically match records from different sources that refer to the same real-world entity, find the true matching records among them and turn this set of records into a standard record for the consumption of users or other applications. There is a large body of work on the record matching problem and the truth discovery problem. The record matching problem is also referred to as duplicate record detection, record linkage, object identification, entity resolution, or deduplication and the truth discovery problem is also called as truth finding or fact finding - a key problem in data fusion.

Record normalization is important in many application domains. For example, in the research publication domain, although the integrator website, such as Cite seer or Google Scholar, contains records gathered from a variety of sources using automated extraction techniques, it must display a normalized record to users. Otherwise, it is unclear what can be presented to users: (I) present the entire group of matching records or (ii) simply present some random record from the group, to just name a couple of ad-hoc approaches. Either of these choices can lead to a frustrating experience for a user, because in (I) the user needs to sort/browse through a potentially large number of duplicate records, and in (ii) we run the risk of presenting a record with missing or incorrect pieces of data.

Record normalization is a challenging problem because different Web sources may represent the attribute values of an entity in different ways or even provide conflicting data. Conflicting data may occur because of incomplete data, different data representations, missing attribute values, and even erroneous data.

## 2. On-demand cloud access of data

The on-demand cloud access and data sharing can greatly reduce data management cost, storage flexibility, and capacity. However, data owners have deep concerns when sharing data on the cloud due to security issues. Once uploaded and shared, the data owner inevitably loses control over the data, opening the door to unauthorized data access. A critical issue for data owners is how to efficiently and securely grant privilege level-based access rights. Data owners are becoming more interested in selectively sharing information with data users based on different levels of granted privileges. The desire to grant level-based access results in higher computational complexity and complicates the methods in which data is shared on the cloud. Research in this field focuses on finding enhanced schemes that can securely, efficiently and intelligently share data on the cloud among users according to granted access levels.

### *A. Drawbacks of existing system*

1. Selectively sharing data files on the cloud becomes a burden on the data owner as the hierarchy grows.
2. This method ensures that no unprivileged data user will gain access to any part of the data file even if that user is able to download the ciphertexts from the cloud.

3. The challenge is to provide the data owners with an efficient, secure and privilege-based method that allows them to selectively share their data files among multiple data users.
4. However, data owners have deep concerns when sharing data on the cloud due to security issues. Once uploaded and shared, the data owner inevitably loses control over the data, opening the door to unauthorized data access.

## 3. Proposed System

In this paper, a Privilege-based Multilevel Organizational Data-sharing scheme (P-MOD) is proposed. It builds on concepts presented in [1] to solve the problems of sharing data within organizations with complex hierarchies. The main contributions presented in this paper can be summarized.

Attribute-Based Encryption (ABE) schemes later emerged to provide more versatility when sharing data. These schemes integrate two types of constructs: attributes and access policies. Access policies are statements that join attributes to express which users of the system are granted access and which users are denied. ABE schemes were introduced via two different approaches.

### A. *Advantages of proposed system*

1. We present multiple data file partitioning techniques and propose a privilege-based access structure that facilitate data sharing in hierarchical settings.
2. We formally prove the security of P-MOD and show that it is secure against adaptively chosen plaintext attacks under the Decisional Bilinear Daffier-Hellman (DBDH) assumption.
3. We present a performance analysis for P-MOD and compare it to three existing schemes that aim to achieve similar hierarchical goals.
4. We implement P-MOD and conduct comprehensive simulations under various scenarios using the real U.S. Census Income data set. We also compare our results to simulations we have conducted for two other schemes under the same conditions.

### B. *System Diagram of proposed system*

The proposed system in this paper is to be used by both the airport authorized admin and the user. The modules proposed are based on the below given system diagram.

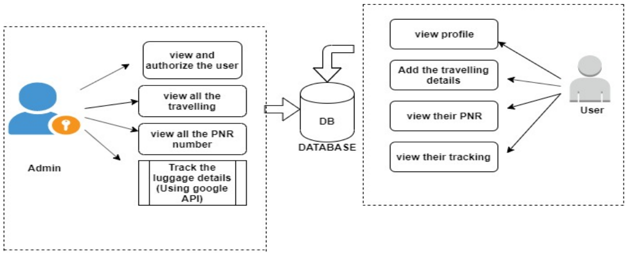

Fig. 1. System Diagram

## 4. Proposed Modules

In this paper we propose two modules for the system. This includes,

1) Airport Admin
2) Passenger

### A. *Airport Admin*

1. The admin has to view and authorize the user then only the user can login into the account.
2. The admin can view all the travelling details of the users from source and destination.
3. The admin can view all the PNR number details for the users.
4. The admin can also track the luggage details where it is arrived (Using the google API maps).
5. The Logout.

### B. *Passenger*

- The user can view their profile.
  - The user should add the travelling details.
  - The user can view their PNR.
  - The user can view their tracking.
- The user can view their position based on the latitude and longitude location
  - Logout.

## 5. System Requirements

### A. *Hardware Requirements*

- System: Pentium IV 2.4 GHz.
- Hard Disk: 40 GB.
- Floppy Drive: 1.44 Mb.
- Monitor: 15 VGA Colour.
- Mouse: Logitech.
- RAM: 512 Mb.

### B. *Software Requirements*

- Operating System: Windows XP
- Programming Language: JAVA
- Java Version: JDK Latest version

## 6. Software Environment

### A. *Java programming language*

The above section says how to prepare a subsection. Just copy and paste the subsection, whenever you need it. The numbers will be automatically changes when you add new subsection. Once you paste it, change the subsection heading as per your requirement.

- Simple
- Architecture neutral
- Object oriented
- Portable
- Distributed
- High performance

- Interpreted
- Multithreaded
- Robust
- Dynamic

A *platform* is the hardware or software environment in which a program runs. We've already mentioned some of the most popular platforms like Windows 2000, Linux, Solaris, and MacOS. Most platforms can be described as a combination of the operating system and hardware. The Java platform differs from most other platforms in that it's a software-only platform that runs on top of other hardware-based platforms.

The Java platform has two components:

- The *Java Virtual Machine* (Java VM)
- The *Java Application Programming Interface* (Java API)

*1) IP datagram's*

The IP layer provides a connectionless and unreliable delivery system. It considers each datagram independently of the others. Any association between datagram must be supplied by the higher layers. The IP layer supplies a checksum that includes its own header. The header includes the source and destination addresses. The IP layer handles routing through an Internet. It is also responsible for breaking up large datagram into smaller ones for transmission and reassembling them at the other end.

*2) UDP*

UDP is also connectionless and unreliable. What it adds to IP is a checksum for the contents of the datagram and port numbers. These are used to give a client/server model - see later.

*3) TCP*

TCP supplies logic to give a reliable connection-oriented protocol above IP. It provides a virtual circuit that two processes can use to communicate.

### *B. ODBC*

Microsoft Open Database Connectivity (ODBC) is a standard programming interface for application developers and database systems providers. Before ODBC became a *de facto* standard for Windows programs to interface with database systems, programmers had to use proprietary languages for each database they wanted to connect to. Now, ODBC has made the choice of the database system almost irrelevant from a coding perspective, which is as it should be. Application developers have much more important things to worry about than the syntax that is needed to port their program from one database to another when business needs suddenly change.

The advantages of this scheme are so numerous that you are probably thinking there must be some catch. The only disadvantage of ODBC is that it isn't as efficient as talking directly to the native database interface. ODBC has had many detractors make the charge that it is too slow. Microsoft has always claimed that the critical factor in performance is the quality of the driver software that is used. In our humble opinion, this is true. The availability of good ODBC drivers has improved a great deal recently. And anyway, the criticism about performance is somewhat analogous to those who said that compilers would never match the speed of pure assembly language. Maybe not, but the compiler (or ODBC) gives you the opportunity to write cleaner programs, which means you finish sooner.

## 7. Feasibility Study of the proposed system

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

Three key considerations involved in the feasibility analysis are,

- Economic feasibility
- Technical feasibility
- Social feasibility

### *A. Economic feasibility*

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

### *B. Technical feasibility*

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement; as only minimal or null changes are required for implementing this system.

### *C. Social feasibility*

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

## 8. Performance Analysis

Table 1
Performance Analysis

| Criteria | Time taken (in Minutes – approx.) | | |
|---|---|---|---|
| | Existing System | Proposed Model | Closes mins before takeoff |
| Login | | 5 to 7 | 15 |
| Authorization | | 10 to 15 | 15 |
| Bording pass collection | 5 to 30 | auto | 45 |
| Baggage Checkin | 5 to 30 | auto | 30 |
| Passport verification | 15 to 40 | auto | 30 |
| Security Check | 10 to 20 | 10 to 20 | 30 |
| Finding the gate | 10 to 20 | 5 to 7.5 | 10 to 20 |
| Estimated wait time | 47.5 | 29.85 | |
| Time saved: 17.65 (mins) | | | |
| % of improvement: 62.84 | | | |

## 9. Conclusion

In this paper, we studied the problem of record normalization over a set of matching records that refer to the same real-world entity. We presented three levels of normalization granularities (record-level, field-level and value component level) and two forms of normalization (typical normalization and complete normalization). For each form of normalization, we proposed a computational framework that includes both single-strategy and multi-strategy approaches. We proposed four single-strategy approaches: frequency, length, centroid, and feature-based to select the normalized record or the normalized field value.

## References

[1] M. L. Wick, K. Rohanimanesh, K. Schultz, and A. McCallum, “A unified approach for schema matching, co-reference and canonicalization,” in SIGKDD, 2008, pp. 722–730.

[2] K. C.-C. Chang and J. Cho, “Accessing the web: From search to integration,” in SIGMOD, 2006, pp. 804–805.

[3] M. J. Cafarella, A. Halevy, D. Z. Wang, E. Wu, and Y. Zhang, “Webtables: Exploring the power of tables on the web,” PVLDB, vol. 1, no. 1, pp. 538–549, 2008.

[4] W. Meng and C. Yu, Advanced Metasearch Engine Technology. Morgan & Claypool Publishers, 2010.

[5] A. Gruenheid, X. L. Dong, and D. Srivastava, “Incremental record linkage,” PVLDB, vol. 7, no. 9, pp. 697–708, May 2014.