Implementing Security Technique on Generic Database

Gaurav Dubey Vikram Khurana Shelly Sachdeva

Department of Computer Science and Engineering
Jaypee Institute of Information Technology, Noida
gd.jiit@gmail.com, vikram0602@gmail.com, shelly.sachdeva@jiit.ac.in

***Abstract*— Database maintenance has become an important issue in today's world. Addition or alteration of any field to an existing database schema cost high to a corporation. Whenever new data types are introduced or existing types are modified in a conventional relational database system, the physical design of the database must be changed accordingly. For this reason, it is desirable that a database should be flexible and allow for modification and addition of new types of data without having to change the physical database schema. The generic model is designed to allow a wide variety of data to be accommodated in a general purpose set of data structures. This generic mechanism for data storage has been used in various information systems such as banking, defense, e-commerce and especially healthcare domain. But, addressing security on generic databases is a challenging task. To the best of our knowledge, applying security on generic database has not been addressed yet. Various cryptographic security techniques, such as hashing algorithms, public and private key algorithms, have already been applied on a database. In this paper, we are proposing an extra layer of security to the existing databases, through Negative Database technique. The advantages of the negative database approach on generic database has been demonstrated and emphasized. Correspondingly, the complexity of the proposed algorithm has been computed.**

Keywords—Negative Database; Generic Database; Database security; Information Security; Security and Privacy.

I. INTRODUCTION

The complexity of databases is increasing rapidly, where the design of an efficient database is not possible. There is a more general database (DB) for this kind of problem known as generic database, which handles most of the complex problems with ease without changing the basic structure of the database. It's based on a basic EAV (Entity Attribute Value) model where all type of data can be stored in a single table without worrying about which type of data has to be stored and where it has to be stored. It is a more general database that can be used for any purpose. Its application is widely seen in the health care systems where the structure of the table is not strictly defined as defining a database is not that easy and a requirement of slight addition in the structure of the table results in several empty fields in the table whereas generic database handles it with ease.

Security is the major concern. Considering the current trends of usability, the need of protecting this type of database is becoming mandatory. Hackers try to get access to the private information, which needs to be highly secured. There are several organizations like banks, security agencies, electronic health records, and intelligence applications that need their data to be secured to the highest extent. Various security techniques such as hashing algorithms, public and private key algorithms have already been implemented for these databases. In this paper, the authors aim to present a framework to implement the concept of negative database on generic databases (EAV model) for enhancing security. This framework consists of various set of algorithms which manipulate the input data and store it in the database with some counterfeit data. This populated database is referred as negative database. Negative database provides an extra layer of security for these databases and will make the databases more secure.

A. Simple Entity Attribute Value (EAV) model

The EAV data model, as the name implies, consist of three parts: entities, attributes and values. This idea of EAV model originates from the concept of association lists as they are called LISP. An association of these list stores information in key-value pairs roughly equivalent to the attribute value part of the entity-attribute value tuple [6]. This idea was used in various health care database developed in 1970's, after which it is considered in many standard applications like openEHR [10], TMR (The Medical Record), and HELP CDR (Clinical data repository). The EAV model's simplest form is three column table; the entity column which describes the data item, for example a product. The attribute column that describes attributes for entities, and the value table contains the values for those attributes.

Application systems where we need to store variety of data, such as healthcare, e-commerce, and banks, this method is very useful. For such specific systems, the best way to implement a database is an EAV model, optimized for some specific needs.

Table I, illustrates an example of person's health records, we see the simple table stores records for person's disease and the second table shows the EAV representation of disease records table.

978-1-4673-7948-9/15/$31.00 ©2015 IEEE