DATABASE SECURITY: General and Research Perspectives

Md. Rafiqul Islam, Harihodin Selamat and Mohd. Noor Md. Sap.
Faculty of Computer Science and Information System
Universiti Teknologi Malaysia,
54100 Kuala Lumpur, Malaysia. Tel: 03-2904957.

Abstract

Since database technology is a necessity for many government as well as commercial organizations it is extremely important for the various communities to be aware of the developments made in securing database systems. Database security has been the subject of active research for the past several years. Rapid progress has been made in defining what security means for such systems and in developing laboratory prototypes and even products that meet those definitions. However, much more work remains to be done in key research area. This paper provides an overview of the database security including policies and models as well as security issues in various aspects are also given.

1. Introduction

Security has always been a major concern to implement information technology. That is why in addition to military applications, security for commercial applications such as medical information systems and banking systems have received a lot of attention. Since database technology is a necessity for many government as well as commercial organizations, it is extremely important for the various communities to be aware of the developments made in securing database systems. It should be noted, however, that progress in database security is still comparatively slow if we take into account the growing dependence on database systems. In database security we are concerned with the ability of the system to enforce a security policy governing the disclosure, modification, or destruction of information. For example mandatory policies restrict access to classified information to cleared personnel. Discretionary security policies, on the other hand, define access restrictions based on the identity of users, the type of access (e.g., select, update, insert, delete), the specific object being accessed. The access controls commonly found in most database systems are examples of discretionary access controls.

Considering the importance of database security here we discuss policies and models of database security. Issues both in mandatory and discretionary policies for database security as well as in distributed database systems are presented. In this paper we also point out security issues for database in wireless computer systems. Our emphasis is on the research perspective.

2. Importance of Database Security

Database security is concerned with the ability of the system to enforce a security policy governing the disclosure, modification or destruction of information. Typically database is used as a technical tool for storing, processing and communicating information. Organization such as the US National Computer Security Center, and recently the EEC [19] have addressed in detail the issue of database security by prototyping different levels of trusted databases and developing appropriate security evaluation criteria. The following points detail some of the requirements produced for such systems.

- The database security considerations take into account all system software and hardware that touches flowing into, and out of, the database. For example, an easily penetrated operating system would render a superbly protected database management system useless.
- Data integrity is a key requirement. The database system must preserve integrity of the data stored in it. The user must be able to trust the system to give back the same that is put in the system and to permit data to be modified only by authorized users. The data should not be destroyed or altered either accidentally, as in a system crash, or maliciously, as in some unauthorized person modifying the data. At the very least, the user should know if the data were corrupted.
- Data should be available when needed. This implies system fault tolerance and redundancy in data, software and hardware. Inference and aggregation must be studied and controlled.
- Audit should be detailed enough to be useful and sufficient enough so as not severely burden system performance.
- The aim should be to maximize secrecy (prevent disclosure) and yet preserve integrity by using appropriate concurrency and integrity controls (e.g., referential integrity).
- The prototype should be of general purpose, commercial quality, relational systems. The relational system has been chosen in most cases because it is currently the model of performance in the commercial world.

2. Policies For Database Security

A *Security Policy* is a general plan or principal course of access controls. Due to different security requirements, there are different policies. Thus, access control plans or courses may be different in different database systems. A *Security Policy* of a database system specifies the authorized accesses to the databases for the user of the database systems. Since policies are primarily dealing with access controls, they are also called access control policies. We will discuss the policies below.

2.1 Discretionary Policy

In a *Discretionary Policy,* authorized access to certain collections of data for a particular user is on a *need-to-know* basis. In other words, this particular user has the discretion to access certain data due to his need of data. This approach allows the owner of an object to provide other users with access to data objects. DAC policies are typically represented using the access matrix [15]. This policy is sometimes called the *policy of least privilege,* because all users and programs operate with the smallest set of privileges necessary to perform their functions.

2.2 Mandatory policy

A *Mandatory Policy* is a set of constraints that must be satisfied to uphold some policy mandated by some law, rule or practice[4]. In a mandatory policy, the data in a database is categorized on the basis of the *degree of sensitivity* of the data. A user's access to the data is confined to the authorized category of data. It is also called *nondiscretionary policy* [1]. It is nondiscretionary because if a user is authorized to access certain categories of data, the user does not have the discretion to access other remaining categories of data.

This kind of policy is created by higher authority which means that the relationship between subjects and objects is not changeable by the object owner. But the change of classification levels can be rather arbitrary, as there is uncertainty as to who determines when information is sufficiently sanitized for a particular security classification.

2.3 Authorization policy

The authorization policy governs the disclosure and modification of information in the database (access control). The authorization policy includes the mandatory policy, the discretionary policy and the personal knowledge approach.

2.4 Personal knowledge approach

The personal knowledge approach has been developed with an emphasis to favour support of privacy before any other design goal. Technically it combines techniques of relational databases, object oriented programming and capability based information [19].

3. Models of Database Security

A model can serve as a design tool, framework for researchers, educational tool and as a comparison and evaluation tool. Here we will describe several models of database security. We start with a basic model that applies to access control, is based on the access rules. It should be noted that access control which is governed by the authorization policy has two aspects: the first one is that we wish to deny access to data to those users who do not have a right to access them, the second is the need to guarantee access to all relevant data to those database users who exercise their access privilege properly.

3.1 Access Control Model

Access control is very important in information security systems, because of the increasing complexity of various sorts of information, the large number of users, and the widely used communication networks. It protects privacy, integrity and availability of information in computer systems. It determines the accesses to information resources stored in a system by verifying the access rights of accessors. In 1972, Graham and Denning [5] initiated the abstract protection model for computer systems. In their model, a protection system includes the following three basic elements: *a set of subjects, a set of objects* and *an access control matrix.* The subject represents any type of individual, such as users, processors, or utility programs. The object indicates a group of information storage, such as disks, magnetic tapes, files, or storage segments. In the access control matrix, the accessible objects are lined in rows and the objects are lined in columns. Each element in the access control matrix represents the access right for a subject with respect to its corresponding object.

The access control for a computer system is achieved by employing an access control matrix [5], as shown in Fig. 1. The access control matrix defines the access rights with respect to the accessors(users) and the resources (files). For any user U_i and file F_j the (i,j)th entry a_{ij} of the access control matrix denotes the access right of U_i to F_j. For instance, from Figure 1, user U_4 has the 'Read/Write/Delete' right to files F_1 and U_1 has the 'Read' right to files F_3 and F_4.

Files Users	F_1	F_2	F_3	F_4	F_5
U_1	0	0	1	1	0
U_2	1	2	1	0	0
U_3	2	1	3	1	1
U_4	2	0	1	0	2

0: No Access
1: Read
2: Read/Write
3: Read/Write/Delete

Fig. 1. Access control Matrix.

3.1.1 Single Key-lock Access Control

The concept of the Single-Key-Lock (SKL) model was first introduced by Wu and Hwang [6] in 1984. In the SKL method, each file F_j is assigned a lock L_j and each user U_i is computed a key K_i. And then the access right a_{ij} for U_i to F_j can be constructed by a defined mathematical function *f* such that

$$a_{ij} = f(K_i, L_j) \qquad \text{(i)}$$

The key K_i is possessed by the user while the lock L_j kept by the system. When U_i wants to request the access right a_{ij} to F_j, the system can easily verify the access request by applying the predefined function *f*. Theoretically, the SKL scheme is more secure and convenient than those needed to maintain a centralized access control matrix in the system.

Several relevant methods appeared in the literature after Wu and Hwang's work. Chang proposed two of them based respectively on the Chinese remainder theorem and Euler's theorem in number theory [7, 8]. In 1989 Laih et al. used Newton's interpolating polynomial to design another method [9]. While Chang and Jiang presented a binary version of Wu and Hwang methods [10]. Based on the Chang and Jan's Key-Lock pair access control model Jan et al. proposed a dynamic access control scheme for frequently inserted or deleted users and files [11]. These were classified as SKL schemes.

3.1.2 Access Control with Binary Keys

A new access control scheme for implementing the access control matrix is proposed by Chang et al. [12]. The proposed scheme is different from the methods which are based on the concept of Key-Lock pairs. In the proposed scheme, each user is assigned a binary key, which is derived from the access rights with respect to the files. The binary key is possessed by the user, and can be used to derive the access right to the files.

In the proposed scheme access right a_{ij} is rewritten in binary forms b_{ij} *as* $(b^1_{ij}.b^2_{ij}..b^c_{ij})$, where b^x_{ij} means the *x*th bit in the bit string b_{ij} and $c = \lfloor 1 + \log w \rfloor$ here *w* is the maximal value of a_{ij}s in the access control matrix. For each user U_i, the following key vectors are defined:

$$K_{i1} = (b^1_{i1}.b^1_{i2}\ldots b^1_{in}),$$
$$K_{i2} = (b^2_{i1}.b^2_{i2}\ldots b^2_{in}),$$
$$\qquad\qquad\qquad\qquad \text{(ii)}$$
$$K_{ic} = (b^c_{i1}.b^c_{i2}\ldots b^c_{in}).$$

and the access right a_{ij} of U_i to F_j is written as

$$a_{ij} = (K^j_{i1}K^j_{i2} \ldots K^j_{ic}). \qquad \text{(iii)}$$

There is no need for explicit locks in the system. Access right checking and file updating is performed by using bit-shift operations.

In [15] Mohammed and Dilts have pointed out that the access matrix model supports name-dependent access, but it must be extended using predicates to manage content-dependent access. A predicate is a condition which must be true before access can be granted. Extensions to the model allow additional authorization requirements and request validation and procedures to be performed during or after the request validation.

3.2 Multilevel Security Model

The need for multilevel security arises when a computer system contains information with a variety of classifications and has some users who are not cleared for the highest classification of data contained in the system. The classification of the information to be protected reflects the potential damage that could result from unauthorized disclosure of the information. The clearance assigned to a user reflects the user's trustworthiness not to disclose sensitive information to individuals not so trusted. Multilevel security model is developed by Bell and LaPadula [20]. This model introduces the concepts of *level* and *category*. Each subject is assigned a *clearance level* and each object a *classification level.* A subject generally represents a process executing on behalf of a user and having the same clearance level as the user. The objects can be area of storage, program variables, files, I/O devices, users, or anything else that can hold information. A *security level* represents by a pair (A, C), where A denotes classification level and C a set of categories. For the military environment there are four classification levels :

0 - *Unclassified*
1 - *Confidential*
2 - *Secret*
3 - *Top Secret*

Each subject and each object also has a set of categories such as *Atomic* and *Nuclear.* One security level is said to *dominate* another if and only if :

1. its classification or clearance level ≥ the other, and

2. its category set contains the other.

That means given classes (A, C) and (A′, C′), (A, C) $\leq$ (A′, C′) if and only if A $\leq$ A′ and C $\subseteq$ C′. For example, transmissions from (2, {Atomic}) to (2, {Atomic, Nuclear}) or to (3, Atomic}) are permitted, but those from (2, {Atomic}) to (1, {Atomic}) or to (3, {Nuclear}) are not [14, 27].

3.3 Lattice Model of Information Flow

The lattice model is an extension of the Bell and LaPadula model [20] which is introduced by Denning [13]. This model was introduced to describe policies and channels of information flow, but not what it means for information to flow from one object to another. An *information flow system* is modeled by a lattice-structured flow policy, states, and state transitions. An information flow policy is defined by a lattice (SC, $\leq$), where SC is a finite set of security classes, and $\leq$ is a binary relation partially ordering the classes of SC. The security classes correspond to disjoint classes of information; they are intended to encompass, but are not limited to the familiar concepts of "security classification" and "security categories" [14]. For security classes A and B, the relation $A \leq B$ means class A information is lower than or equal to class B information. As we know information is permitted to flow within a class or upward, but not downward or to unrelated classes.

A flow policy (SC, $\leq$) is a *lattice* if it is a partially ordered set (*poset*) and there exist least upper and greatest lower bound operators, denoted $\oplus$ and $\otimes$ respectively, on SC. That (SC, $\leq$) is a *poset* implies the relation $\leq$ is reflexive, transitive, and antisymmetric; that is, for all A, B, and C in SC:

1. Reflexive: $A \leq A$

By extension, corresponding to any subset $S = \{A1, \ldots An\}$ of SC, there is a unique element $\otimes S = A_1 \otimes A_2 \otimes \ldots \otimes A_n$ which is the greatest lower bound for the subset. The lowest security class, *Low*, is thus $Low = \otimes SC$. *Low* is an identity element on $\oplus$; that is, $A \oplus Low = A$ for all $A \in SC$. Similarly, *High* is an identity element on $\otimes$.

The lattice shown in Fig. 2 represents a system containing personal data of three types: medical (m), financial (f), and criminal (c) [13]. The classes shown are all the possible subsets of {m, f, c}; they represent combination of data types. Information flows (as shown by the arrows) only into classes at least as inclusive. A flow violation would occur, for example, on an attempt to move information produced from combining medical and financial data into the class designated medical only.

2. Transitive: $A \leq B \text{ and } B \leq C \text{ implies } A \leq C$
3. Antisymmetric: $A \leq B \text{ and } B \leq A \text{ implies } A = B$.

That $\oplus$ is a *least upper bound* operator on SC implies for each pair of classes A and B in SC, there exits a unique class $C = A \oplus B$ in SC such that:

1. $A \leq C$ *and* $B \leq C$, and
2. $A \leq D$ and $B \leq D$ implies $C \leq D$ for all D in SC.

By extension, corresponding o any nonempty subset of classes $S = \{A_1, \ldots, A_n\}$ of SC, there is a unique element $\oplus S = A_1 \oplus A_2 \oplus \ldots \oplus A_n$ which is the least upper bound for the subset. The higher security class, *High*, is thus $High = \oplus SC$.

That $\otimes$ is a *greatest lower bound* operator on SC implies for each pair of classes A and B in SC, there exists a unique class $E = A \otimes B$ such that:

1. $E \leq A$ and $E \leq B$, and
2. $D \leq A$ and $D \leq B$ implies $D \leq E$ for all D in SC.

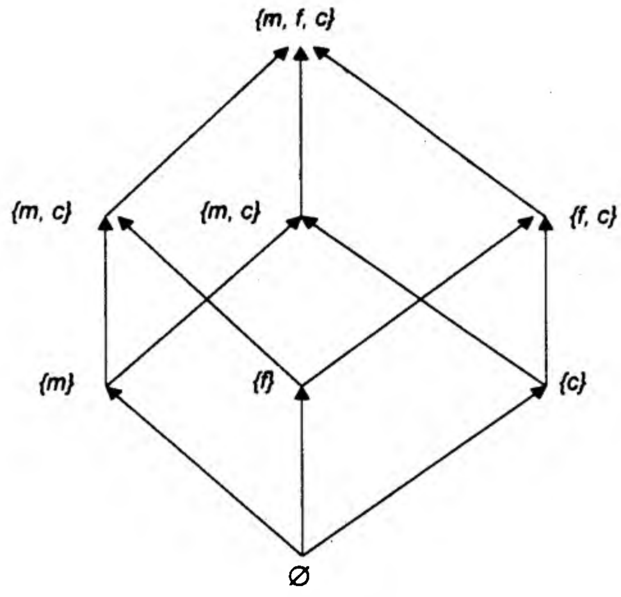

Fig. 2. Lattice of subsets {m, f, c}

3.4 User-Role Based Security Model

In an effect to focus on application-dependent security constraints, a user-role based security (URBS) model has been proposed by Ting [17] to permit the expression and enforcement of application-dependent security constraints with the framework of a database system. URBS is intended to provide a security environment sensitive to, and semantically rich for use by users in performing their tasks within the boundary of security requirements. To access data, users must specify the application and their role within that application. Such user-role based access controls can be represented in a set of well formed transactions (WFTs) which control the data access operations and determine the data objects which are legitimate for the unspecified role. Ting stress that URBS definition and analysis should be an integral part of the database design process, and not an activity that is deferred until the system is implemented, when end-users begin to request access to the database.

Lochovsky and Woo [18] also see the need to consider finer-grained classifications of users, based on the roles they play in the organization. However, they discuss user roles only in terms of an object-oriented DBMS. Using the concept that user can have certain roles, access to the database can be specified using these roles. Roles and data can then be represented as objects which allow the DBMS to control and manage operations performed on the data, a well as the types of access to the data itself. Mohammed and Dilts [15] extend this idea in a relational database system and the extended model is augmented with the capability to respond to dynamic events.

4. Issues in Database Security

There are so many issues in database security. In this section we will discuss some important issues in mandatory and discretionary security policies. Multilevel security issues in distributed database management systems as well as security issues in wireless environment are also described.

4.1 Issues in Mandatory (Multilevel) Policy

A multilevel database system supports data having different classifications or access classes and users having different clearances. In the most general case, the ability to individually classify atomic facts in a database is required. In the relational model, this means that data are classified at the level of individual data elements. Special cases of multilevel relations may be classified at the attribute level (i.e., all the data associated with a particular attribute have the same classification); at the row level (i.e., every tuple has a single classification); or at the relation level (i.e., all the data in the relation have the same classification) [16]. Multilevel security affects the data model because not all data are visible to all users. One effect involves polyinstantiation, which we describe shortly.

4.1.1 Polyinstantiation

Multilevel security has an unexpected but unavoidable effect, which is known as polyinstantiation [20]. *Polyinstantiation* is the simultaneous existence of multiple data objects with the same name but different access classes. It is a phenomenon of multilevel data. As such, it exists as a property of information and is not merely the result of any specific technology. Thus, we cannot simply choose not to support it in our systems. Rather, we must investigate how best to reflect it in our developing technologies.

Polyinstantiation has two fundamental forms: polyinstantiated entities and polyinstantiated attributes of entities. These are described below.

1) *Entity Polyinstantiation*

This form of polyinstantiation occurs if there are two distinct real-world entities having the same "unique" identifier. For example, in a computer system a low user assigns the name "secret-operation" to a file, unaware of the fact that a high file named "secret-operation" already exists. As a result there will be two distinct entities with the same name. To preclude the possibility of an insecure information flow, the low user can not be informed of the name conflict.

Several means have been proposed for preventing this situation from occurring. One such means is to partition the global namespace into mutually exclusive low and high namespaces [16]. Another is to prevent people with low clearances from assigning entity identifiers. There will always be situations in which neither of these solutions is appropriate; in such cases, polyinstantiation must be allowed to occur in order to preserve information flow security.

2) *Attribute Polyinstantiation*

When there are two distinct values to an attribute of a real-world entity for users with different clearances, this form of polyinstantiation is occurred. For example, a flight could have a mission attribute with "space-exploration" value known to low users and with the value "spying" known to users with high clearances.

Attribute polyinstantiation could be avoided by insisting that any entity have only a single value for any attribute. To do this, if a low entity has a high value for some attribute, this fact must be made known to low users [16]. Attribute polyinstantiation cannot be avoided, however, if the very existence of the high attribute value is not known to people with low clearances. For example, if the fact that a low flight has a "spy-equipment" attribute is itself a high piece of information, people with low clearances cannot be security prevented from associating a different low attribute called "spy-equipment" with the entity.

4.1.2 Inference and Aggregation

Ensuring that the database only gives users facts, either directly or indirectly, which they are cleared to see is not enough to guarantee confidentiality. This is because a user might be able to combine together some facts, which they are cleared to see, to infer other facts which are classified higher than their clearance. Suppose a user cleared to confidential obtains the facts "*Biggles* is the pilot of flight 556" and "IF *x* is the pilot of flight *y* THEN *x* is a pilot" from the example database. The user is cleared to see both these facts so the database hands them over. However, the user can make a logical inference from these facts and deduce that Biggles is a pilot. But this fact may be classified secret in the database, so the user should not have it. In general the inference problem arises whenever a set of facts can be used to deduce further facts of higher classification.

A similar problem arises if the database allows collections of facts to be given an overall classification. Such a feature would allow a user to hold many facts, each of which they are cleared to see, so long as they cannot be formed into any collection too highly classified for them to hold. Users may obtain more facts from the database, as long as they are cleared to see each of them and to hold any collections which can be made from them [26]. However on combining the newly acquired facts with those already held it may be possible to form a collection which the user is not cleared to hold. Thus the aggression problem arises when collections, or aggregates, of facts are given overall classifications.

No practical, general purpose, solution to these two problems is possible, because the database management system (DBMS) would have to keep track of what facts each user holds. This information would have to be consulted every time facts are returned to the user to ensure that no highly classified facts may be inferred or aggregates formed. The time and space required to search and maintain such information makes it effectively impossible to solve the problems. Another drawback is that the DBMS would be unable to determine when the users discard information, because of the endless possibilities for encoding information in a computer, thus it could never apply the controls accurately. So it is not possible to allow collections of facts to be given an overall classification and care must be taken to classify facts so that the inference problem does not arise.

4.1.3 Covert Channel

Another subtle way of obtaining facts is through covert channels. This is where information is stored indirectly in the database rather than directly as facts [26]. In particular, the existence of a fact, or its classification, may be used to convey information about something. For example, a user cleared to secret could choose whether or not to insert "*Mickley Mouse* is the pilot of flight 989" into the database. This fact must be classified at least secret to ensure that secret information is not lost simply by classifying it too low, and so would not be given directly to users with low clearances. However, such users could ask "select pilot where flight 989". They will either receive the answer "you are not cleared to know" or "there is no such pilot", depending on whether the secret user has inserted the fact or not.

Thus the database will reveal to any user the existence of facts which were inserted by secret users, although the actual values in the facts will not be revealed. This encodes a binary 1 or 0 and is the basis of a covert channel from high users to low users. Under normal circumstances this channel would be so slow that its use by a human user can be dismissed as a threat. However, the software that the human users run may contain code, a

Trojan horse, which uses the channel to encode secret information it obtains and leak it to a collaborator working at the unclassified level, without the user knowing. Thus, when driven by software, the confidentiality. It may be noted that the covert channel exists because the existence of a fact in the database can convey sensitive information, from the user who inserted the fact to any user who can detect its existence.

4.1.4 Trojan Horse

A Trojan horse refers to a transaction which lies hidden in another transaction. When the latter transaction is being executed, the hidden transaction then attacks the database and breaches its security. In many database systems the access privileges of a transaction are determined by the access privileges of the user of the transaction. A user with a top-secret clearance cannot reveal data with a top-secret classification to other users with a lower security clearance, say, the secret clearance [3]. However, if the user's transaction has a Trojan horse *i.e.*, a hidden transaction, the Trojan horse may divert the top-secret data to the user of only a secret clearance. In this case, the Trojan horse has breached the security of top-secret data and poses a threat to the user of top-secret data.

4.2 Issues in Discretionary Policy

Discretionary security policies for most file systems are fairly simple and straightforward. These policies can be easily modeled using any of the discussed access control models. But the models leaves many questions unanswered [16], however. Particularly troubling are some of the requirements to support for such thing as group authorization and explicit denial of authorization.

4.2.1 Groups

Discretionary security policies are concerned not only with subjects may obtain access to which objects, but also with the granting, revoking, and denying of authorizations to and from users and groups. Given the set of authorizations for users and groups, some rule must be applied for deriving authorizations for subjects [16]. In the general case, a user may belong to more than one group. In assigning privileges to subjects acting on behalf of a user, one can choose to:

- have the subject operate with the union of the privileges of all the groups to which the user belongs as well as all his or her individual privileges;
- have the subject operate with the privileges of only one group at a time as well as all his or her individual privileges;
- allow the subject to choose whether to operate with its user's privileges or with the privileges of one of the groups to which its user belongs;
- implement some other policy.

It may be mentioned that even if a subject *S* is constrained to be associated with at most one group to which its associated user *U* belongs, a user is still not constrained to operate with the authorizations of only one group at a time. For example, if user *U* belongs to a group G_1 that is authorized for a relation or view *R*, and *U* also belongs to another group G_2 that is not authorized for *R*, then *U* can still gain access to *R* by employing a subject whose associated group is G_1. Thus, this choice of policy constraints subjects rather than users, and can be thought of as a form of least privilege.

4.2.2 Roles

Some applications may require that discretionary access controls be specified on the basis of user roles. Many systems have some built-in roles (e.g., system administrator, database administrator, system security officer). However, different users are likely to have different requirements and definitions for such roles. In addition, many applications require that arbitrary user job access control requirements be formalized in terms of roles (e.g., the secure military message system) [16]. Thus, a generic capability for application-defined roles is desirable. The relationship between a user's role authorizations and his or her user and group authorizations probably depends on the application. Whether a user acting in a certain role is to be prohibited from granting some of his or her role privileges to a user acting in another role is also probably application-dependent.

4.2.3 Explicit Denial of Authorization

In the higher evaluation classes of the DoD (Department of Defense) Trusted Computer System Evaluation Criteria [16], users must be able to specify which users and groups are authorized for specific modes of access to named objects, as well as which users and groups are explicitly *denied* authorization for particular named objects. It may be noted that explicit denial of authorization is not the same as *lack* of authorization. For example, the set of users and groups authorized for an object might be implemented as an access control list (ACL) and the set of users and groups explicitly denied authorization as an *exclusionary* access control list (XACL).

4.3 Security Issues in Distributed System

Here we will present an overview of multilevel security issues in distributed database management systems. The rapid growing of the networking and information-processing industries has led to the development of distributed database management system prototypes and commercial distributed database management systems. In such a system, the database is stored in several computers which are interconnected by some communication media. The aim of a distributed database management system (DDBMS) is to process and communicate data in an efficient and cost-effective manner. It has been recognized that such distributed systems are vital for the efficient processing required in military as well as commercial applications. For many of these applications, it is especially important that the DDBMS should operate in a secure manner. For example, the DDBMS should allow users who are cleared at different levels access to the database consisting of data at a variety of sensitivity levels without compromising security.

B. Thuraisingham et al. have published a series of article [21-23] on multilevel security for distributed database management systems. In the first article [21] preliminary investigation of multilevel security issues for a DDBMS are described based on the front-end/back-end architecture. In the architecture, a front-end machine is connected to one or more back-end database systems. All requests to the database systems are via the front-end machine. That is, the front end machine controls the execution of all transactions. As a result the back-end machines cannot execute local applications. In the second article [22] security issues for DDBMS are described based on another architecture. This architecture is depicted in Fig. 3.

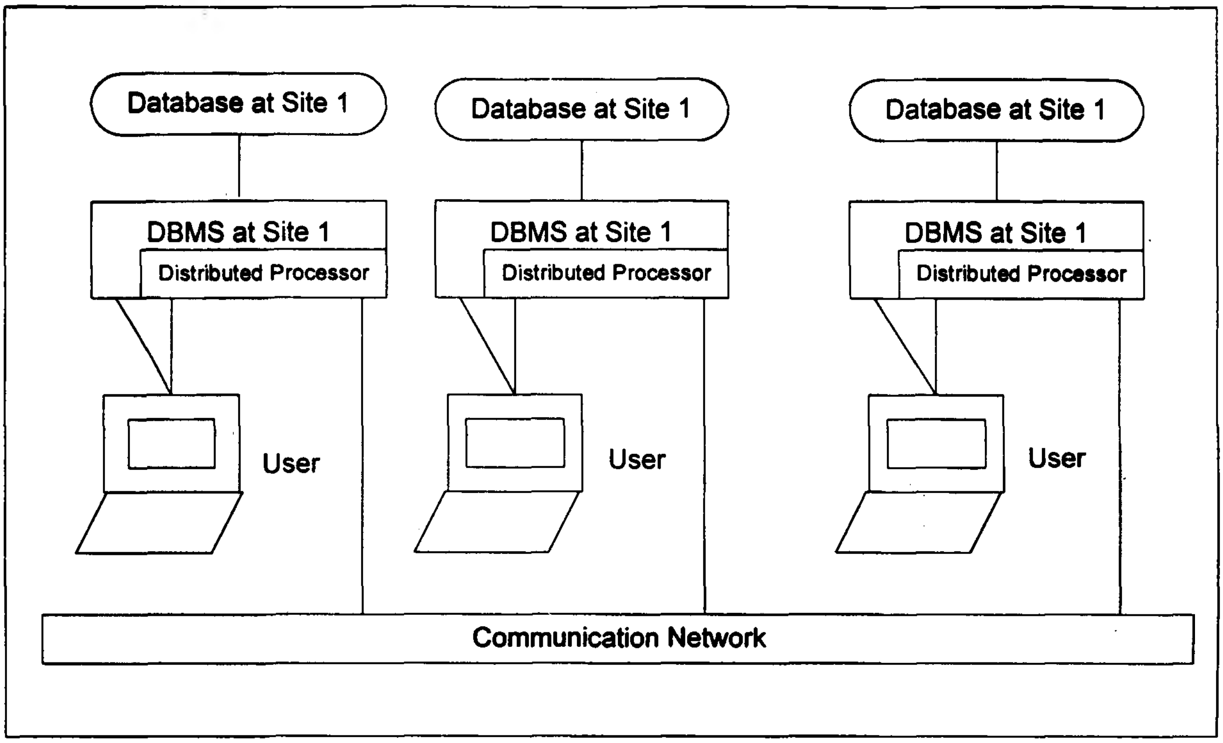

Fig. 3. DDBMS Architecture

In that architecture, the nodes are connected via a communication subsystem. This subsystem could be any network such as local area network or a long haul network . Each node has its own local DBMS which is capable of handling the local applications. In addition, each node is also involved in at least one control application. In other words, there is no centralized control in that architecture. In the architecture for an MLS/DDBMS, each node has an MLS/DBMS which is augmented by a module called the secure distributed processor (SDP). The SDP consists of components for query processing, transaction management, metadata management, and constraint processing. Each of these issues are discussed in that paper. In the third article [23] issues for DDBMS are described by considering a limited heterogeneous environment.

4.4 Issues in Wireless Systems

Wireless computer systems hold out the promise that cable-free communications between workstations, file servers, and remote devices such as printers will be an effective and efficient alternative to more traditional wire-based systems. Generally wireless systems can be divided into two main groups: those employing radio transmission techniques and those employing infrared transmission. Both of these techniques send data signals by generating waves that are part of the electromagnetic spectrum. The electromagnetic spectrum is comprised of a tremendous range of radiation frequencies.

The drawbacks associated with data transmission via radio waves are well known: The radio portion of the electromagnetic spectrum is already overcrowded, licensing is required in some bands, radio signal are susceptible to radio frequency (RF) and other noise/interference, and speed and/or bandwidth limitations hamper the effectiveness of radio as a means of transmitting data. One approach that wireless systems use to overcome these problems is the use of spread spectrum transmission (SST) technology. SST evolved from the military's attempts to thwart hostile forces interception and or jamming of radio transmissions by using frequency-hopping techniques, whereby a transmitter would send signals on one frequency, shift to another frequency, and send more signals, shift to yet another and so on.

Because of their characteristics, wireless systems have special security concerns. Much of what is applicable to security for other wireless communications is also germane to wireless computer systems, with the additional complicating factor that systems like local area networks (LANs), by design, exist to facilitate file-sharing, provide access to common data, allow communication with outside networks, and even share programs and operating systems. In short, all of the risks associated with traditional cable-based systems are also faced by wireless systems, with the additional threat imposed by the open transmission medium. The only tangible difference is that physical access to the cable is not needed in a wireless system. The chief threat systems as opposed to cable-based system is unauthorized eavesdropping on the networks, in view of the fact that the transmission media is the open air, thus putting the signal "there for taking". Actually, eavesdropping can come in two forms: the first is actually "capturing" or tuning in or radio or infrared signals carrying data, and the second type is the interception and deciphering of the electromagnetic emissions (in the form of radio waves) that all electronic devices give off. Wireless systems in and of themselves do nothing to prevent the latter. All wireless systems generate electromagnetic emissions. There are various measures one can employ against this threat, some involving shielding of the equipment or the room/building in which the devices are located. Another countermeasure is to locate as many devices as close together as possible, thus generating a near-indecipherable hash of mixed signals. The other form of eavesdropping is the capture or monitoring of the radio/light signals carrying live information as the wireless systems is operated normally. In this case, the inherent characteristics of the various technologies employed by wireless systems play a major role in determining their susceptibility to this type of threat.

SST technology is considered relatively secure to the characteristics of the signals. The broad range of frequencies over which SST-based computer systems transmit simultaneously, and the unique spreading codes employed to generate the signal; greatly reduce the possibility of listening in and deciphering the information being sent and received. Wireless systems that do not use SST technology operate on a very narrow range of frequencies. Because of this, theoretically it should be easier to target and intercept these signals.

When solving security problems introduced by radio channels it is important not to do so at the cost of increased vulnerability on the network side. For example, a solution that requires adding an on-line database of secret information (e.g., secret keys), which must be heavily protected against computer system intrusion, is inferior in that respect in comparison to a solution that only requires adding a key revocation database (which does not contain any secrets and therefore need not be protected as heavily), and an offline certification service (which is much easier to protect than an on-line database).

The next step may be more comprehensive user privacy solution that ensures end-to-end encryption so that users privacy is maintained not only on the radio channels against third parties, but also throughout the network. This requires a fine act of balance between the user's right for privacy and law enforcement agencies' ability to trace criminals. There are a few pending proposals on how to do it, but this issue is still unresolved [28].

By employing the basic steps necessary to secure information and the systems that store and provide it- such steps as per-screening of potential employees, logon and password authentication access control, good backups, physical security, compartmentalization of critical information and distribution on a need-to-know basis, and encryption for most applications wireless systems will be sufficiently secure. Here we summarized the important security issues in wireless system as a block diagram in Fig. 4.

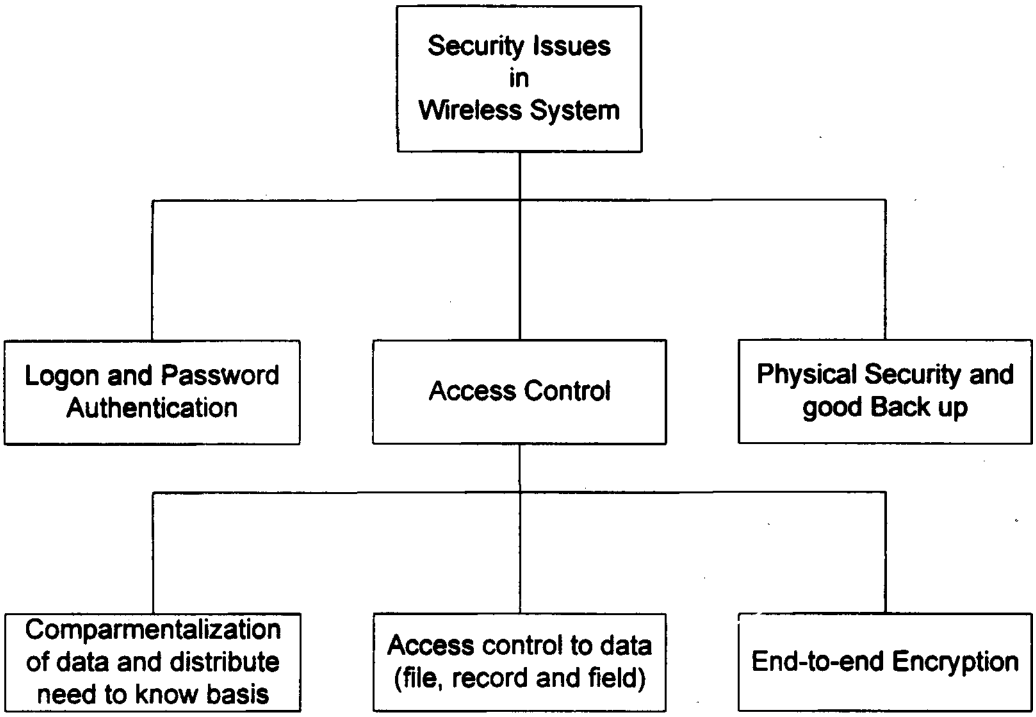

Fig. 4. Block diagram of Issues in Wireless System.

Therefore, if we talk about database security in wireless system, one of the important issues is database data security. Data must be secure within database in a site as well as when it will be transmitted to other side. In these cases data encryption is a most important issue. So, it is required to design a cipher algorithm that will be appropriate to encrypt database data.

5. Conclusion

We have described the database security including policies and models. Here the security issues in discretionary policy and mandatory policy as well as multilevel security issues in distributed database management's systems are discussed. The security issues for wireless systems are pointed out and summarized in a block diagram. Here we also point out that encryption of data is one of the most important security issues for wireless system and necessity of designing database oriented cipher algorithm. This paper acts as a general as well as research perspectives.

References

1. E. B. Fernandez, R.C. Summers, C. Wood; *Database Security and Integrity*, Addison-Wesley, 1991.
2. C. J. Date; *An Introduction to database Systems, Volume II*, Addison-Wesley, 1995.
3. David K. Hsiao; *Database Security Course Module*, In Database Security: Status and Prospect, Editor C. E Landwehr, North-Holland, 1988.
4. D. J. Thomson; *Role-Based Application Design and Enforcement*, In Database Security: Status and Prospect, Editors S. Jajodia and C.E. Landwehr, North-Holland, 1991.
5. G. S. Graham and P. L. Denning; *Protection-principle and practice*, Proc. Spring Jount Conf., 40, AFIPS Press, Montvale, NJ, 1972, p.417-429.
6. M. L. Wu and T. Y. Hwang; *Access control with single-key-lock*, IEEE Transaction on Software Engineering , SE-10(2), 1984, p.185-191.
7. C. C. Chang; *On the design of a key-lock-pair mechanism in information protection systems*, BIT, 26, 4, 1986, p.410-417.
8. C. C. Chang; *An information protection scheme based upon number theory*, The Computer journal, 30(3), 1987, p.249-253.
9. C. S. Laih, L. Harn and J. Y. Lee; *On the design of a single-key-lock mechanism based on Newton's interpolating polynomial*, IEEE Transaction on Software Engineering, 15, 9, 1989, p. 1135-1137.
10. C. K. Chang and T. M. Jiang; *A binary single-key-lock system for access control*, IEEE Transaction on Computers, 38, 10, 1989, p. 1462-1466.
11. J. K. Jan, C. C. Chang and S. J. Wang; *A dynamic key-lock-pair access control Scheme*, Computers & Security, 10, 1991, p. 129-139.
12. C. C. Chang, J. J. Shen and T. C. Wu; *Access control with binary keys*, Computers & Security, 13, 1994, p. 681-686.
13. D. E. Denning; *A lattice model of secure information flow*, Comm. ACM 19, 5, 1976 p. 236-243.
14. D. E. Denning; *Cryptography and Data Security*, Addison-Wesley, 1983.
15. I. Mohammed and D.M. Dilts; *Design for dynamic user-role-based security*, Computers & Security, 13, 1994, p. 661-671.
16. T. F. Lunt; *Security in Database Systems: a research perspective*, Computers & Security, 11, 1992, p. 41-56.

17. T. C. Ting; *A user-role based data security approach*, In Database Security: Status and Prospect, Editor, C. E. Landwehr, North-Holland, 1988.

18. F. H. Lochovsky and C. C. Woo; *Role-based security in database management systems*, In Database Security: Status and Prospect, Editor, C. E. Landwehr, North-Holland, 1988.

19. G. Pangalos; *A tutorial on secure database systems*, Information and Software Technology 36, 12, 1994, p. 717-724.

20. T. F. Lunt, D. E. Denning, R. R. Schell, M. Heckman and W. Schckley; *The SeaView security model*, IEEE transactions on software engineering, Vol. 16, No. 6, June 1990, p. 593-607.

21. McHugh and B. Thuraisingham; *Multilevel security issues in distributed database management systems*, Computers & Security, 7(4), 1988, p. 387-396.

22. B. Thuraisingham; *Multilevel security issues in distributed database management systems II*, Computers & Security, 10, 1991, p. 727-747.

23. B. Thuraisingham and H. H. Rubinovitz; *Multilevel security issues in distributed database management systems-III*, Computers & Security, 11, 1992, p. 661-674.

24. D. L. Lathrop; *Security aspects of Wireless local area Networks*, Computers & Security, 11, 1992, p. 421-426.

25. S. K. Cunningham; *Database security: Features and considerations*, Computer Security Journal, Vol. 9, No. 2, 1993, p. 13-25.

26. S. Wiseman; *Control of Confidentiality in Databases*, Computers & Security, 9, 1990, p. 529-537.

27. Md. Rafiqul Islam, Harihodin bin Selamat and Mohd. Noor Md. Sap; *Polyinstantiation and Integrity in Multilevel Security: A Survey*, Journal Teknologi Maklumat, 7, 1, 1995, p1-10.

28. Y. Yacobi; *Security for Wireless Systems: Guest Editor's Note*, IEEE Personal Communications, August 1995, p. 2.

29. A. Aziz, W. Diffie; *Privacy and Authentication for Wireless Local Area Networks: A Secure communications protocol to prevent unauthorized access*, IEEE Personal Communications, First Quarter 1994, P. 25-31.

30. Y. Frankel, A. Herzberg, P. A.. Karger, H. Krawczyk, C. A. Kunzinger and M. Yung; *Security Issues in a CDPD Wireless Network: Enchanced Security protocols for the CDPD network*, IEEE Personal Communications, August 1995, p. 16-27.