PROCEEDING
INA-CISC 2005

INDONESIA CRYPTOLOGY AND INFORMATION SECURITY

March 30-31, 2005

Organized by

National Crypto Agency of the Republic of Indonesia
National Crypto Institute of the Republic of Indonesia

Incorporation with

Indonesian Society on Electrical, Electronics, Communication and Information

PROCEEDING
INACISC 2005
INDONESIA CRYPTOLOGY AND INFORMATION SECURITY
March 30-31, 2005

Editors:
Andriyat Kurniawan
Sri Sutanto
Setiyo Cahyono
Ahmad Muammar W.K.
Aldi Rija Pramada

ISBN 979-99407-0-2

Printed by INFORMATIKA Bandung

Preface

The 1st Indonesia Cryptology and Information Security Conference (INACISC) was held in Jakarta, Indonesia, during March 30-31, 2005. INACISC was organized by the National Crypto Agency of the Republic of Indonesia (known as Lembaga Sandi Negara Republik Indonesia) and National Crypto Institute of the Republic of Indonesia (known as Sekolah Tinggi Sandi Negara Republik Indonesia), in cooperation with Indonesian Society on Electrical, Electronics, Communication and Information (IECI).

The conference received submissions that represent cryptography and security applications in the cryptographic and national academic community, even the other community who interested in the substance of the conference. The program committee reviewed the papers and 28 papers were accepted for presentation in the technical program of the conference. These proceedings contain the original of the accepted full papers. There are no revisions for the substance of papers, and the authors bear full responsibility for the contents of their papers.

The conference program included two invited speakers and three invited tutorials. The invited speakers are Suguru Yamaguchi from the Nara Institute of Science and Technology, Japan, talked on "Internet Security"; and Budi Rahardjo from the Bandung Institute of Technology, Indonesia, talked on "Information Security". The invited tutorials are Romi Satria Wahono from Indonesia Institute of Science (known as LIPI-Lembaga Ilmu Pengetahuan Indonesia), Indonesia, talked on "Unix Security"; Malcolm Shore from Canterbury University, Christchurch, New Zealand, talked on "Information Warfare, Computer Forensic"; and Ken Umeno from National Institute of Information and Communications Technology, Japan, talked on "Vector Stream Cipher".

With our pleasure to thank the Head of National Crypto Agency of the Republic of Indonesia for his best advise in the conference. We are also very thank to the General Chair, Program Chair and Organizing Chair, for their guidance and good collaboration.

We are very grateful to the program committee and the secretariat, whose members worked very hard over several months for the conference.

Finally, we would like to thank all the other people who provided any assistance, and the authors who submitted their papers to INACISC 2005.

March 2005

Editors

Committees

Organizer

National Crypto Agency of the Republic of Indonesia
National Crypto Institute of the Republic of Indonesia

In Corporations with
Indonesian Society on Electrical, Electronics, Communication and Information (IECI)

Conference Organization

Advisor	Nachrowi Ramli Head of National Crypto Agency, Indonesia
General Chair	Ruly Nursanto Deputy of National Crypto Agency, Indonesia
Program Chair	Ferdinand Imanuel Head of National Crypto Institute, Indonesia
Organizing Chair	Kabul Warsono Primary Secretary of National Crypto Agency, Indonesia

Technical Program Committee

1. J. M. Sunarto – National Crypto Agency, Indonesia
2. Yan Adikusuma – National Crypto Agency, Indonesia
3. Farid Wazdi – Bandung Institute of Technology, Indonesia
4. Edy Tri Baskoro – Bandung Institute of Technology, Indonesia
5. Budi Rahardjo – Bandung Institute of Technology, Indonesia
6. Sarwono Sutikno – Bandung Institute of Technology, Indonesia
7. Surjadi Slamet – University of Indonesia, Indonesia
8. Bambang Nurcahyo Prastowo – Gadjah Mada University, Indonesia
9. Moh. Mustafa Sarinanto – Indonesian Society on Electrical, Electronics, Communication and Information, Indonesia
10. Harry Prihanto – Indonesian Society on Electrical, Electronics, Communication and Information, Indonesia
11. Kiki Sugeng – University of Indonesia, Indonesia
12. Willy Susilo – Wollongong University, Australia
13. Malcolm L. Shore – Canterbury University, New Zealand
14. T. Basarudin – University of Indonesia, Indonesia

Table of contents

Database security using SeaView model on Microsoft Access

Darmawan Satyananda†, *UM, and* Intan Rahayu*, *LSN*

† Dept. of Mathematics, Fac. of Mathematics and Natural Sciences
State University of Malang, Indonesia
Tel/fax. (0341) 552182, email : dsatyananda@telkom.net
* LSN, Jl. Harsono RM No.70, Jakarta 12550
Tel. 7806829, fax. 7806829 email : intan.rahayu@gmail.com

Abstract– **Microsoft Access is desktop or small size DBMS. Basically, it provides security facilities to secure data inside, but in simpler form than other DBMS. In the need to implement more robust secure system, Access cannot handle by itself. Implementation can be done with the help of other program or components.**

One model of security model is SeaView Security model. This model is application of Bell-LaPadula model and BIBA Model to relational database In this model, there are classification of object (database and its elements) and subject (user or system that access the object). Access to database is based on this classification. Subject can read data if its clearance dominates the access class of object. Typically access class or clearance is divided into Top Secret [TS], Secret [S], Classified [C], Unclassified [U] where TS is the highest level and U is the lowest ($TS \geq S \geq C \geq U$).

In this paper, the SeaView is implemented in Microsoft Access database. One program is made to support security and data encryption. Program is made using Microsoft Visual Basic 6.0, whereas its security using AES-Rijndael algorithm and its hash value using MD5 algorithm.

Keywords– Database Security, Microsoft Access, SeaView Security Model

I. INTRODUCTION

Database security [5] comprises a set of measures, policies and mechanisms to provide secrecy, integrity and availability of data. The threats to database security can be defined as a hostile agent that, either casually or by using a specialized technique, can disclose or modify the information managed by system. Violation to database security can be grouped into three categories:

1. improper release of information caused by reading of data from intentional or accidental access by improper user.
2. improper modification of data, this is violation to data integrity through improper data handling or modifications.
3. denial of service, could prevent users from accessing data or using resource.

Not all relational database implements proper security. One of them is Microsoft Access. Security in Microsoft Access is user level security with the user right such as create table, read table, add record, etc. Access does not implements high-level security because it is mostly used as small-to-medium size database in small environment (mostly for personal use). Access also has encryption system. But the purpose of this encryption is to protect objects in database (such as form, report, query) in order to user cannot change the design, not for data in table.

To provide such security, a user-made program must be used. The function of this program is as an interface to database, to control the access right of existing user, and to provide the right data for specified user.

II. BASIC THEORY

2.1. Security on Microsoft Access

Basically, Microsoft Access as desktop database software has provided security facilities, in the simpler form than other DBMS such as Microsoft SQL Server. There are two layers of security in Access: (1) Access Control that control user that have right to use the database and what the user can do to the database, and (2) file encryption to protect data such as unauthorized user can read the data use plain text editor, for example.

2.1.1. Access Control

Access use Jet Database Engine to save and retrieve data, execute query and other functions. Security in Access also based on security in Jet. Access control in Jet is implemented in a workgroup. Information about this workgroup is saved in a file (SYSTEM.MDW as a default file name). This file contains information valid database users and groups, such as logon name, password, system ID, and privilege of user or group to all object in the database Jet use this file to validate user when Access is started or a database is opened.

By default, each workgroup has two groups: Admins and Users. Admins is a group of all users that have the right

to administer the database. At least one user exists in Admins, that is Admin. Users is a group to all user of database. Each user may have different level of right. Security in Access is an option (we can choose whether to use it or not), by default all user is assumed as Admin with no password, that give impression that no user or group exist.

Permission for user can be divided up into three groups: permission for data access, permission for database design, and Administer. Data permissions are Read, Update, Insert, and Delete. Permissions for database are Read Design and Modify Design. Administer permission Is used to change the permission of other user and to change the ownership of an object in database.

Permissions assigned directly to a user account are called "explicit" permissions, whereas if a user inherits them because he or she is a member of a group then they are called "implicit" permissions. In the case that there are different permission, permission used is the less secure permission.

By using this way, the security of database is less maximum. The data in database is still can be read using some tricks. One of them is to import secure database from a blank database.

Security on table in Access is implemented on table as a whole, not for a specified data (rows or columns in a table). To implement a security for element of table, a user-made program should be used. This program functioned as an interface to access the database. Other functions are to examine the query submitted by user to know whether the user access the protected data or not, or to see the right of user to access row or column in a table. Modification of query is allowed to ensure that the qualified data is given to user.

2.1.2. **File Encryption**

Since an MS Access database can be opened and read with a plain text editor even with the advantages of password protection, another level of security should be added for truly sensitive data. Fortunately, another helpful tool is provided within the MS Access program. This second tool is for data encryption. As the name suggests, using this tool will encrypt all data within an MS Access database.

The type of encryption used by MS Access is an RC4-encryption algorithm with a 32-bit key from RSA Data Security Incorporated. The probable reason that Microsoft chose such a short key was to comply with United States export laws, which allow a key length of less than 40 bits. Because of this less-than-ideal key length, however, MS Access encryption has been cracked numerous times. With that said, it is still sufficiently secure against all but a determined and experienced individual. Nonetheless, following are some recommendations and cautionary characteristics of using MS Access encryption:

- Since the Jet database engine will automatically decrypt data, whether a file is opened directly in MS Access or whether it's opened in a program, encryption should always be used in conjunction with a file-level password. This combination will, for most uses, alleviate the weaknesses associated with each method used separately.
- To protect extremely sensitive data such as social security and credit card numbers, a more robust encryption method could be used to encrypt individual data fields before submitting the data to Jet. This technique can even be used along with file-level passwords and MS Access-native encryption to provide an additional layer of security. One especially useful benefit of this technique is that these doubly-encrypted fields cannot even be read by opening the file in MS Access, even with the file-level password. In order to add this level of security, programming is required.

2.2. **Security Model**

The objective of security model is to produce a high-level, software-independent, conceptual model, starting from requirement specifications that describe the protection needs of the system. Security model can be broadly classified in two categories: discretionary and non-discretionary (or mandatory) models.

Discretionary security models govern the access of users to information on the basis of the users' identity and of rules that specify, for each user and object in the system, the type of access the user is allowed for the object.

Mandatory security models govern the access to the information by the individual on basis of the classifications of subjects and objects in system. Object are the passive entities storing information, such as data files, records, file in records, etc. Subjects are active entities that access the object. Generally, subjects are active processes operating on behalf of users.

2.3. **Sea-View Security Model**

The Sea View (Secure dAta VIEW) model [6] is a security model for the protection of relational database system. This model is application of Bell-LaPadula model and Biba model to relational database. This system is best implemented in high security environments.

The model governs access to the data stored in the database on the basic of mandatory security model, and formulated in two layers: the MAC (Mandatory Access Control) model and the TCB (Trusted Computing Base) model.

2.3.1. **The MAC model**

MAC corresponds to monitor that enforces security policies of the Bell-LaPadula model. In this model, user must have authorizations for classified information.

Concepts of Bell-LaPadula and Biba models is used here, including access class, Subject, Object, Access Models.

2.3.2. The TCB model

TCB model defines multilevel relations, supports discretionary access control for multilevel relations and views, and formalize discretionary security policies.

Multilevel relations

Multilevel relation extends the concepts of relation to include classification labels. Each attribute A in schema R is associated with a classification attribute C, and each attribute value in a tuple is associated with a corresponding security classification. A tuple classification attribute TC is added to the relation attributes to provide a classification for each tuple as a whole. Hence, a multilevel relation schema R with n attributes would be represented as $R(A_1,C_1,A_2,C_2, ..., A_n, C_n,TC)$ where each C_i represents the classification attribute associated with attribute Ai.

In Sea View model, each object in database (record, tuple, and the others) has typical access class or security level such as Top Secret [TS], Secret [S], Classified [C]; Unclassified [U] where TS is the highest level and U is the lowest ($TS \geq S \geq C \geq U$). Each user which is accessing that data has **clearance** which represent right of user to that data. The access class is also known as **security label**. Since there are different policies or classifications then different user might see different view from the same data, depends on their clearance.

Properties that must be satisfied on classifications of multilevel relations:

1. Multilevel Entity Integrity:
   - No tuple $r \in R$ can have null values for any primary key attribute.
   - Attributes forming the primary key (PK) must have the same access class within any tuple, and
   - access class(PK) must be dominated by access class of every other attribute. Access class L1 is said dominate access class L2 (written as $L1 \geq L2$) if level L1 is greater or equal with L2.
2. Multilevel Referential Integrity:
   - If a foreign key is visible at a given access, then a tuple containing the referenced PK must also be visible at that access class, and all references must be downward in access classes.

A subject (user) can read certain object if its clearance level dominates the security level of the object. Consider the standard table of relational model in Figure 1, which is named SOD. This table contains for each starship its name, its objective and its destination. The key is Starship (of course, there won't be any tuple with the same value).

Starship	Objective	Destination
Enterprise	Exploration	Talos
Voyager	Spying	Mars

Fig.1. SOD Relation

The data above does not have access class, so user can see all data if they submit a query. The issue in multilevel security is how access class is assigned to tuple in relations. Access class can be assigned to tuple in relation, to attribute in relation, or to individual data element of tuple in relation. In this case access class is also named classification labels.

Multilevel relation of SOD relation is shown figure 2. We use security level specified above (TS, S, C, U).

Starship	CS	Objective	CO	Destination	CD	TC
Enterprise	U	Exploration	U	Talos	U	U
Voyager	U	Spying	S	Mars	S	S

Fig 2. SOD Relation with access class

As mentioned above, a subject can read objects if its clearance level dominates the security level of the object. On the other hand, on an object with a higher access class, that user will only see null as an attribute value (the real value is not shown). For example, relation in figure 3 is a relation that will be seen by an user with U access class.

Starship	CS	Objective	CO	Destination	CD	TC
Enterprise	U	Exploration	U	Talos	U	U
Voyager	U	null	U	null	U	U

Fig 3. SOD Relation for user with U access class

On MLS, the same entity can be represented by many tuples with different security level. Providing of information into user with lower clearance level, which is the information is different with the information provided to user with higher clearance level is called **cover story**. Cover story provides mechanism to protect information that only may be known by user with higher clearance level.

Suppose user with U clearance level changes the second tuple of a SOD relation in figure 3 into {Voyager, Exploration, Talos}. This changes is not allowed in relational database, since the Starship key will have the same attribute value. But this is allowed in multilevel database because there may be a lost information. Figure 4 shows the result of the change. To differ the tuples with the same key, then tuple access class (classification level) may take role. The situation where there more than one tuple with same value for the key (or attributes) but have different access class (for tuple or attribute) is called **polyinstantiation**.

Starship	CS	Objective	CO	Destination	CD	TC
Enterprise	U	Exploration	U	Talos	U	U
Voyager	U	Spying	S	Mars	S	S
Voyager	U	Exploration	U	Talos	U	U

Fig 4. Example of polyinstantiation

One thing that should be noticed is a key that formed from several attribute. For this case, all attribute of the key must have the same access class that ensure all value will be displayed.

Some cases on polyinstatiation:

1. Subject clearance level is dominated by (or incompatible) with access class of data:

- A polyinstatiated tuple arises if subject inserts a tuple that has the same value of primary key with one of tuples in relation.
- A polyinstatiated tuple is happened if subject updates the tuples that displayed by null (actual data is hidden since different access class)

2. Subject clearance level dominates the access class of the data:
   - A polyinstantiated tuple arises whenever a subject inserts a tuple that has the same PK as an existing tuple at a lower class
   - A polyinstantiated attribute arises whenever a subject updates an attribute which has a value classified at a lower level

In general, subject with access class (clearance level) c can only change element with access class c, not for a higher or lower access class. Access class of elemen that inserted or modified become equal with the access class of the subject. As an example, notice the last tuple of relation if figure 3. Access class of Objective and Destination attribute is U, since insertion is performed by U clearance level subject.

Insertion of new tuple is accepted if the tuple has different access class although have the same key, whereas modification is accepted if access class of subject and object (tuple) is the same. If the access class of subject and object is different, then the tuple is inserted into relation as a new tuple. On figure 3, suppose there is a S class subject changes the value of Destination attribute into Mars, then there is a new tuple on SOD relation, that is {Voyager, Exploration, Mars}, with S access class S. Tuple classification will be the same as the subject's security clearance.

Covert channel is an indirect flow of information from a higher level user to a lower level user. Example of covert channel is suppose a lower level user wishes to insert a tuple that already exists in the database at a higher level of security. If this insertion is rejected by the system, then the lower level user will know that there exists a tuple at a higher level. Another example is transaction of lower level user. If the results from a lower security level transaction are delayed when there is a higher level security transaction, then the lower security level user can determine there are transactions at higher level, and may even be able to infer information from the length of the delay.

2.3.3. Architecture of the Sea View Cryptographic System

The architecture of the Sea View Cryptographic System can be seen in the figure 5.

Sea View user is a mandatory policy that is charged to DBMS user. A trusted Front End (TFE) also called trusted filter is responsible for enforcing security function and multilevel protection, acting as TCB. There is a validation process of user identity by comparing the hash value of password with hash value that saved in database.

All data will be encrypted with they key that generated from a Random Key Generator which is saved in separated key storage. This key is also used to encrypt the database file. This key is used again on a process to open encrypted database and to open all data in relation (table). This key is different for each user access class and the value is alternating on each transaction.

All activities will be kept in activity log table on database, to ease in monitoring access and flow control.

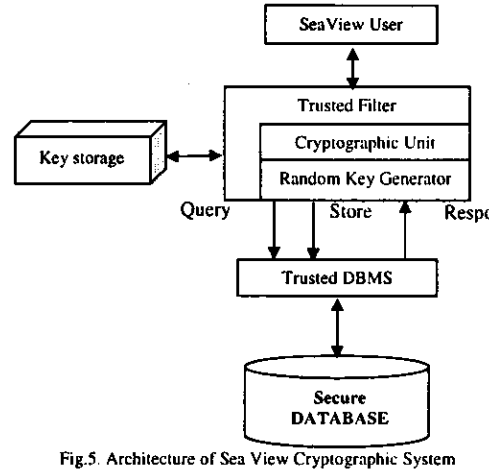

Fig.5. Architecture of Sea View Cryptographic System

III. EXPERIMENTAL RESULTS

The problem taken to implement Sea View Security Model is data of State Officer riches. This is an empirical studies, data presented here is based on observation on mass media.

As we know, state officers must send an asset report to KPK (Komisi Pemberantasan Korupsi) in the beginning of their duties, in order to check their asset in the beginning and whether an indication of illegal riches increasing. In short form, KPK will publicize this report. But the actual data is much more complicated than the publicized data. Public only see the short form of data, but the officer in charge can see the complete data. In this situation we can see the classification of data in database and .the classification of subject accessing the database.

Tables or relations needed for this implementation can be viewed in figure 6.

Details of attributes for each entity:

1. INVESTIGATOR {Investigator Code, Name, Pangkat, Golongan, Institution, Right }
   (Note: Right is access control or subject's privilege to object, TC)

2. OFFICER {Officer Code, Name, Birth Place, Birth Date, Pangkat, Golongan, Jabatan, Starting date, Institution Name, Institution Address, Salary, Tunjangan, Lain-lain, TC}
3. FAMILY {Officer code, Family code, Family Name, Relationship, Birth place, Birth date, Address, Pekerjaan, Working address, Salary per month, penghasilan lain-lain, TC}
4. RICHES {Officer Code, Riches Code, Riches name, Location, Kind of riches, Weight, Value, Date of perolehan, Source of perolehan, Letter number, Explanation, TC}
5. INVESTIGATION {Investigation code, Investigator code, Officer code, Date of investigation, TC }
6. RESULT {Result of investigation, TC }

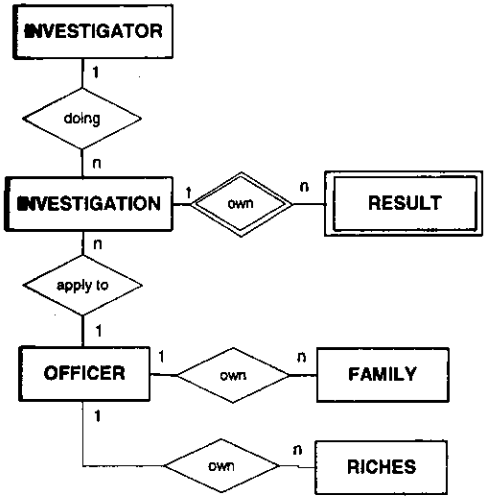

Fig.6. E-R for Investigation database

Each tuple and element is tables will have access control. The access control is divided into SR (Sangat Rahasia or Top Secret), R (Rahasia or Secret), T (Terbatas or Classified), U (Umum or Unclassified). Access control for each tuple is saved in TC field.

The decision to allow user with certain clearance to read the data with the certain access class is or what data should be given to user is done by program made for this case. The program here is made using Visual Basic 6.0. To take the hash value, the program use MD5 algorithm and to encrypt data or file, the program use AES-Rijndael algorithm. Encryption is done twice, the first for each element in table, and the second for the database file (.mdb). To generate random number, this program use built-in random key generator from Visual Basic.

To secure data of each user (state officer that being investigated), each class of investigator has their own keys. The data of officer is encrypted using this key. The encryption is done in each time the investigator logout from this software. The key that used is then saved in a file, so the other investigator can open the data. In each encryption, the key is always changed, that generated randomly by program.

Each investigation data is saved in INVESTIGATION table; INVESTIGATION is used as activity log table.

For each query submitted by user, the program examines the query to check whether there are data that should not be read or exposed. The program returns null (blank) for data being read that may not be exposed.

All operation (insertion, modification, deletion, view) is done based on access control of subject and object, and principle of multilevel relation (refer to the explanation of Multilevel Relation above).

This research has shown characteristics of SeaView security model [5] as follow:

1. Database security.
   This program provides security for database, using MD5 and AES-Rijndael encryption to protect from viewing by unauthorized user, using external program such as text viewer.
2. Discretionary policy
   This policy restrict access of subjects to objects on the basis of granting and revoking of previleges. There are two levels for assigning previleges :
   a. The account level : At this level, DBA specifies the particular privileges that each account holds independently of the relations in the database.
   b. The relation (or table) level : At this level, we can control the privilege to access each individual relation or view in the database.
3. Mandatory policy
   This policy restrict access of subjects to objects on the basis of security labels. Each objects have classification (access class) and each subjects have clearance. From the explanation above, classification type is SR, R, T,U.
4. Indirect access of information (flow control)
   Flow of sensitive information into less protected object or unauthorized subject must be prevented. Unauthorized subject will be rejected and subject that doesn't have proper clearance will be given null information.
5. Access control
   This characteristic ensures that all direct access to objects exclusively occurs to the modes and rules specified by the security policy. For each access, program will check clearance that subject may have and compare with access control of object.

IV. CONCLUSIONS

Conclusions that may be taken are:

1. It is possible to implement SeaView Security Model in Microsoft Access, but it should be done by program that control the security details
2. The structure of data is still use structure built in Access. This program does not use built in Access security and encryption.
3. It is impossible to open database file in other editor since it is encrypted. Reading and submitting query must use this program.
4. Perhaps it is easier to implement this model in DBMS that supports security, such as Oracle. This experiment only to model the security in desktop or small size database with a minimal security support.

Some suggestions that may be implemented:

1. The experiment can be tried in other database model, such as Client/Sever database and distributed database.
2. The security model is not limited to SeaView, another model is possible.
3. To enhance performance, it is better to have proprietary random key generator for any database models.

REFERENCES

[1] C.J Date, "An introduction to Database System", 7th ed, Addison Wesley Longmann, Inc. Massachusetts, 2000
[2] Ramez Elmasri, Shamkant B. Navathe, "Fundamental of Database System", 3th ed, Addison Wesley, Massachusetts, 2000
[3] Charles P. Pfleeger, Security in Computing 2nd ed, Prentice Hall Inc., New Jersey, 1997
[4] Alfred J. Menezes, Paul C. van Oorschot, Scott A. Vanstone, Handbook of Applied Cryptography, CRC Press, 1996
[5] Silvana Castano, Mariagrazia F, Giancarlo M, Pierangela S, Database Security, ACM Press Addison Wesley, Wokingham, 1994
[6] Denning D.E, Secure Distributed Data View: The Sea View Formal Security Model, Technical Report A003 SRI International, 1987
[7] Frequently Asked Questions About Microsoft Access Security for Microsoft Access versions 2.0 through 2000, http://support.microsoft.com/default.aspx?scid=/support/access/content/secfaq.asp
[8] Marshall D. Abrams, Sushil Jajodia, Harold J. Podell (editor), Information Security, An Integrated Collection of Essays. IEEE Computer Society Press.
[9] Michael Gertz. Security Models. http://sirius.cs.ucdavis.edu/teaching/289F/chap2.pdf
[10] Pan Pantziarka, Understanding MS Access Security, http://www.itp-journals.com/t1526.pdf