Jurnal Teknologi Maklumat & Multimedia 1(2004): 73-89

A New IV-Based Database Encryption Scheme Using TS Block Cipher

ZAILANI MOHAMED SIDEK, NORBIK BASHAH IDRIS &
HARIHODIN SELAMAT

ABSTRACT

Current database security research classify four types of controls for the protection of data in databases: access controls, information flow controls, inference controls, and cryptographic controls. This paper covers the fourth type of controls, cryptographic controls in database security that provides security of data stored in commercial RDBMS like Oracle. The proposed database encryption scheme is based on TS Block and Stream Ciphers, and is capable of protecting data at the data element, row, and column levels using both block and stream encryptions. The design of the scheme's key generation and management system allows the controls of users' access to encrypted data in a multilevel fashion thus provide multilevel security. The scheme solves the problem of mandatory and discretionary access controls in a given organization. The security of the scheme is based on the fact that no cryptographic keys are stored in the database system. All encryption and decryption keys are stored securely in smartcards thus providing minimum cryptographic information to users. The design of the encryption scheme is based on the provably strong ciphers with 128-bit keys which is currently infeasible to be broken even by exhaustive key search. Implementation of the scheme has been conducted successfully in Oracle RDBMS and complements the Oracle encryption security available.

ABSTRAK

Penyelidikan masa kini dalam bidang keselamatan pangkalan data mengelaskan empat jenis kawalan bagi mengawal keselamatan data iaitu kawalan capaian, kawalan aliran maklumat, kawalan inferens dan kawalan kriptografi. Kertas ini menerangkan kawalan jenis keempat iaitu kawalan kriptografi yang dapat menyediakan keselamatan data bagi suatu sistem pengurusan pangkalan data hubungan komersial seperti sistem Oracle. Skema pengenkriptan pangkalan data yang di cadangkan ini adalah berdasarkan sistem-sistem sipher "TS Block" dan "TS Stream". Skema ini dapat memberi keselamatan data pada aras unsur data, baris dan lajur data

menggunakan kaedah pengenkriptan blok dan "stream". Reka bentuk sistem penjana dan pengurusan kunci kriptografi dapat mengawal pengguna mencapai data yang telah di enkripkan secara berbilang aras. Ini dapat memberi ciri keselamatan data berbilang aras (multilevel security). Skema ini mendapat ciri keselamatannya berdasarkan atas fakta bahawa tiada kunci kriptografi di simpan dalam sistem pangkalan data tersebut. Oleh itu maklumat kriptografi yang minimum diberi kepada pengguna iaitu hanya kunci kriptografi mereka disimpan secara selamat dalam kad-kad pintar. Reka bentuk skema pengenkriptan ini dibuat menggunakan sistem sipher yang terbukti kuatnya dan menggunakan kunci sepanjang 128 bit. Pada masa ini, kunci sepanjang ini tidak mungkin dapat di cari penyelesaiannya walaupun menggunakan kaedah pencarian kunci secara menyeluruh. Pelaksanaan skema ini dalam sistem pangkalan data hubungan Oracle telah dibuat dengan berjaya dan ia dapat melengkapkan lagi sistem keselamatan pengenkriptan Oracle yang tersedia ada.

INTRODUCTION

We are in the age of information warfare. It is not wrong to say that the IRAQ war is the first information warfare (Azmi 2003) the world have ever seen and demonstrated by the United States of America. This fact is further supported by writers and publishers in the international scene and locally, even by our Minister of Defense Datuk Seri Najib Tun Razak who said that knowledge and expertise in information warfare was crucial for the future development of the Malaysian armed forces (Sharmini 2003). With the current Internet access capability, security of information is now everybody's concern. Governments concern for national safety, corporations concern with the security of their businesses on the web, and the society would like to secure their personal privacy. At the root of this problem is the security of databases where much of the information is stored digitally and may be accessed 24 x 7 days all year round.

There has been a lot of research work done in the field of database security. Pernul and Luef (1992) produced a good bibliographical listing of database security but did not include cryptographic controls and security in statistical databases. They categorized the work in database security into 14 sections covering research issues in databases, evaluation criteria and standards, to other works like security monitoring and reference monitors. So what is database security? Database security is concerned with ensuring the secrecy, integrity, and availability of data stored in a database (Denning 1988). It comprises a set of measures, policies, and mechanisms to provide confidentiality, integrity and availability of data and to combat possible attacks on the system (threats) from insiders and outsiders, both malicious and accidental (Castano et al. 1995, Dastjerdi et al. 1996). To provide

database protection, Denning and Denning (1979) lists four types of controls: access controls, information flow controls, inference controls, and cryptographic controls.

Access control in a database is the ability to explicitly enable or restrict in some way subjects (users, roles, processes) who can access objects (data, stored procedures, functions, programs) and do something with a computer resource (e.g. use, change, or view). In an access control system, there is a set of access policies and rules that all subjects must follow in accessing objects, and a set of control procedures (security mechanisms) that check the access requests against the rules before allowing, denying, or modifying the requests, including filtering of unauthorized data. Bertino et al. (1995) provides three main directions in access control research: discretionary access control in relational database management systems (RDBMSs), mandatory access control in RDBMS, and the development of adequate authorization models for advanced DBMSs. In discretionary access control (DAC), subjects access objects on the basis of the subjects' identity and authorization rules. 'Discretionary' means that the possibility exists for subjects to grant and revoke access rights on some objects. On the other hand, mandatory access controls (MAC) restrict the access of subjects to objects on the basis of security labels. Security labeling for subjects and objects are made of security classes or access classes. An access class consists of two components: a security/classification levels and a set of categories. Examples of security/classification levels are Top Secret (TS), Secret (S), Confidential (C), and Unclassified (U), where TS > S > C > U (> is for dominates). An object's (data or information) security class reflects the sensitivity of the information contained in the object. A subject's (user) security level, also referred to as user clearance, reflects the user's trustworthiness not to disclose sensitive information to users not cleared to see it. Categories for subjects and objects tend to reflect their application areas or departments, such as Production-Personnel-Engineering-Administration for industrial environment, and Logistics-Procurement-Recruitment-Intelligence for defense department. Categories, in addition to security levels, are used to provide finer grained security classifications and form the basis for enforcing need-to-know restrictions.

In a multilevel RDBMS, multilevel secure RDBMS (MLS/RDBMS) can be achieved with either mandatory or discretionary access control policies. However, MAC is often referred to as multilevel access control and the US Department of Defense (DoD) mandatory security (or multilevel security) policies restrict access to classified information to cleared personnel. Therefore, MAC is always used in the context of MLS/RDBMS in which a multilevel RDBMS supports data with different security levels (classifications) and users with different security clearances (Lunt 1992). Multilevel security is needed in this type of database which contains information in various classifications and the fact that not all users have clearance for the highest classification of

data contained in it. This means that a MLS/RDBMS have some multilevel relations that will appear differently to users with different security clearances, because not all data are authorized to all users. Table 1 below illustrates an example of a multilevel relation.

TABLE 1. An example of a multilevel relation

Name	C_{Name}	Department	C_{Dept}	Budget	C_{Budget}	Project	$C_{Project}$	TC
Ahmad	TS	InfoSystem	TS	RM35m	TS	EIS	TS	TS
Malek	S	SoftwareEng	U	RM20m	S	DSS	S	S
Ismail	U	System	U	RM2m	U	Web dev	U	U

Information flow controls govern the transmission of information among accessible objects in the system or ultimately from one user to another. An information flow occurs when some data in X is read and written into Y. Flow control policies require admissible flows to be listed or regulated. If information is transmitted between two objects but the transfer request, on the basis of admitted flows, is unauthorized then it is said that an information flow violation has occurred. An information flow policy specifies the channels along which information is allowed to move. Inference controls are controls that prohibit users making conclusions from indirect access to data. They are relevant especially in statistical databases when users can deduce specific information from querying for summary information. There are three main inference channels in a system where users can make such deductions: indirect access, correlated data, and missing data. Statistical attacks can be governed by applying inference controls such as query controls, random samples, and data perturbation. The fourth type of controls for database protection is cryptographic controls. Data at-rest (data stored in databases) or data in-transit (data in the communication transmissions) are protected by transforming them into an unintelligible state where only the intended user with the correct secret key is able to understand (decipher) the message/data. In applying cryptographic controls, the encryption technique will utilize a secret key (K), e.g. "A32917E4C64EAA618BED08E-6A8875640" to transform a sensitive message/data (M) commonly referred to as the plaintext, e.g. "*When to exchange*", into its scrambled form (M^k) or commonly referred to as the ciphertext, e.g. "1DC0220EF27C45B4D51070AD6747753C". In this paper, we consider only the fourth type of controls, i.e. cryptographic controls, for the security of databases. This paper is also aimed at providing solutions to the problem of security in commercial relational databases taking the example of Oracle RDBMS.

RELATED WORK

A database encryption scheme enhances the security in databases by providing a framework for the application of cryptographic transformations to data solving the problems of MAC and DAC. Data can be transformed cryptographically either at the data element, column, or row levels in a database. Gudes et al. (1976) gave a formal description for the application of cryptographic transformation in databases. Davida et al. (1981) first proposed a record oriented encryption scheme but allows the encryption and decryption of fields within the record using subkeys. The system is based on the Chinese Remainder Theorem. Davida and Yeh (1982) showed some database operations using cryptographic relational algebra. Further enhancement to the scheme was done by Omar and Wells (1983), Cooper et al. (1984), Hwang and Yang (1997), and Lin et al. (1992). Denning (1983) used the Data Encryption Standard for the encryption and authentication of fields within a record. Hardjono and Seberry (1989) proposed a multilevel encryption scheme which allows a hierarchical organization of keys used to encrypt and decrypt data stored in databases. The scheme uses the same method as the RSA cryptosystem and was implemented using UNIFY DBMS (Hardjono *et al.* 1990). He and Wang (2001) conducted a study on how to integrate modern cryptography technology into a RDBMS to solve some major security problems. They focus their effort on how to enhance the access control mechanism to make user management more secure and close some major security holes of current RDBMSs, and how to support database encryption.

ORACLE SECURITY

Oracle database security entails permitting or denying user actions on the database and the objects within it. Oracle uses schemas and security domains to control access to data and to restrict the use of various database resources. A schema is a named collection of objects, such as tables, views, clusters, procedures, and packages. A user's security domain include user's authentication method, the default and temporary tablespaces, tablespaces accessible to the user, the user's resource limit profile, and the privileges, roles, and security policies that provide the user with appropriate access to schema objects needed to perform database operations. Oracle provides comprehensive discretionary access control. Oracle Advanced Security is a bundle of network security features such as data encryption and integrity for all network protocols into the Oracle database, supports third-party authentication, authorization, and single sign-on services, and integrates a Public Key Infrastructure (PKI). Oracle Virtual Private Database (VPD) provides row-level access control and supports the Oracle Label Security. The latter implements label-based access control policies mediating access to rows in database tables based on a label contained in the row, a label associated with each

database session, and Oracle Label Security privileges assigned to the session.

Database encryption in Oracle is provided by DBMS_OBFUSCATION_TOOLKIT. The toolkit is a built-in PL/SQL package to encrypt and decrypt stored data. It supports bulk data encryption using the algorithms Data Encryption Standard (DES) and triple DES (3DES) in both two and three key modes. Although the toolkit supports the MD5 cryptographic hash algorithm for data integrity and a random number generator for generating secure encryption keys, database users are required to device their own key management system.

OVERVIEW OF OUR APPROACH

Our approach is novel and simple in the sense it does not utilize any complex mathematical equations in deriving its cryptographic keys for multilevel access controls. We designed and implemented a new database encryption scheme based on the block cipher built by Tuan Sabri (2000), which is referred to as TS Block Cipher throughout this paper. Our database encryption scheme is highly suitable and easily implemented in any commercial relational DBMS such as Oracle™ and SQL Server™. The scheme guarantees the integrity of data in the database, and provides data confidentiality accessible only by authorized database users with appropriate cryptographic keys. This multilevel database encryption scheme is based on symmetric cryptography which means that the key used to decrypt is the same as the one used to encrypt it. All users' cryptographic keys are securely stored in smartcards external to the database management systems and are read-in during the encryption and decryption process. In this process, users' cryptographic keys are utilized to generate on-the-fly "session keys" which are finally used to encrypt or decrypt the database data. The session keys, though are actually used in the encryption-decryption process, are just temporary keys and are not stored anywhere in the system. They are created, used, and destroyed. Before we describe fully the entire database encryption scheme, we introduce the basic concept involved in using TS Block Cipher as the main building block in the whole scheme. TS Block Cipher is used extensively in the scheme's cryptographic key derivatives and achieved the multilevel security requirements in the system. Unlike TS Block Cipher, TS Stream Ciphers are used only in the encryption and decryption of data in the database.

FOUNDATIONS TO THE CRYPTOGRAPHIC TECHNIQUE USED

This section describes the basic building blocks in the design of the proposed database encryption scheme. The scheme is based on the cryptographic work done by Tuan Sabri in his Ph.D. research namely the works on the new design of block and stream cipher encryption algorithms for data security. In

this section, we also describe the important concept of TS Block Cipher before explaining the cryptographic primitives of the scheme and some enhancements made for database application purposes.

TS BLOCK CIPHER

A block cipher transforms a string of input bits of fixed length (the input block) into a string of output bits of fixed length (the output block). With the current state of development in computer technology, modern cipher systems should be able to withstand exhaustive keysearch at least up to 2^{64}. What this means is that the cryptographic strength of a cipher system should be able to withstand the risk of finding its correct key through searching all of its possible keys and the minimum combination of this possible keys must be 2^{64}. A block cipher is totally broken if a key can be found. TS Block Cipher was designed with considerations that surpassed many of the high security requirements including a 128-bit symmetric block cipher (128 bits block size), and the key length of 128-bits in which exhaustive keysearch of 2^{128} is currently infeasible. The TS Block Cipher also passed the statistical test in the design of a good cipher system. The reader is encouraged to refer to Tuan Sabri (2000) Ph.D. work for more detail explanation.

The primary use of TS Block Cipher in the proposed database encryption scheme is in its key generation, management, and distribution (KGMD) subsystem. Based on the high security features in the basic design of TS Block Cipher, and the fact it accepts a variable called the Initialization Vector (IV) that "influence" the output of the other input variable, renders the TS Block Cipher suitable for KGMD. This is shown graphically in Figure 1. The figure illustrates the capability of TS Block Cipher in generating a subordinate's master key (MK) from his/her superior MK with the use of the Superior's IV. The Superior's IV can be any value or property taken from the Superior, for example username, user_id, etc. The initialization vector is the parameter value that is used to set the initial state of the cipher system. Therefore, the use of the same IV with the same input, e.g. the Superior's MK, will produce the same subordinate's MK all the time.

The Superior MK with his/her appropriate IV can produce his/her subordinate's MK, and the subordinate's MK with his/her appropriate IV can in turn produce his/her sub-subordinate's MK, and so on. This provides the solution to the problem of access controls in a hierarchy achieving the multilevel security requirement in the scheme.

CRYPTOGRAPHIC PRIMITIVES

Figure 2 illustrates the cryptographic primitives for the proposed database encryption scheme. We notice there are three main sections: Input Process, Encryption-Decryption Process, and Key Generation Process. In the example

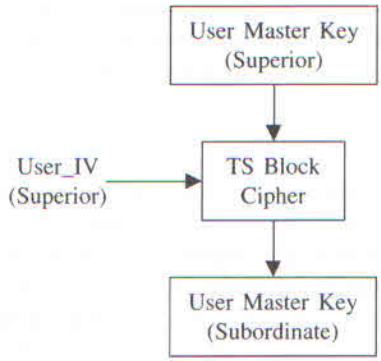

FIGURE 1. Multilevel key generation

shown, the input to the system is "Uni Tek Malaysia" comprising of 16 ASCII characters including 2 blank characters. The input is limited to 16 in this example because that is the maximum the Block Cipher can process in a block. For input data longer than 16 characters, the process will be looped 16 characters a time, and blanks will be padded in the last loop to make it 16 characters also. For this input to be encrypted, cryptographic keys need to be generated. In the Key Generation Process section, we notice there are two types of random key generator: a 128-bit key generator, and a 256-bit key generator. These random key generators will generate the 128-bit keys (32 hex characters) for use in the Encryption-Decryption (E-D) process by the TS Block Cipher, and the 256-bit keys (64 hex characters) by the TS Stream Ciphers.

However, in the proposed database encryption scheme, session keys are in fact being used in the E-D process. This is not so obviously shown in the figure. By looking closely in the Key Generation Process section, there is a sub-section called Pre- / Session Key Generator.

By pressing the button "BC Hex Encryption", a 32-hex character key is produced. This key is commonly referred as a "Pre-Session Key" in the case of TS Stream Cipher and a "Session Key" when TS Block Cipher is used. The production of Pre-Session and Session keys require two types of input. The first input type is the hashed value from the Input Process, vis-à-vis, "Uni Tek Malaysia" using the hash function MD5. This hash value is reflective of the Data IV (will be explained in the section on *Description of Our Database Encryption Scheme*). The second type of input is the user's cryptographic key. Both of these types of input will go into the TS Block Cipher algorithm to produce the session keys. It is worth mentioning that two session keys (in fact are Pre-Session keys) are concatenated to produce a 64-hex key (256-bits) required by TS Stream Cipher.

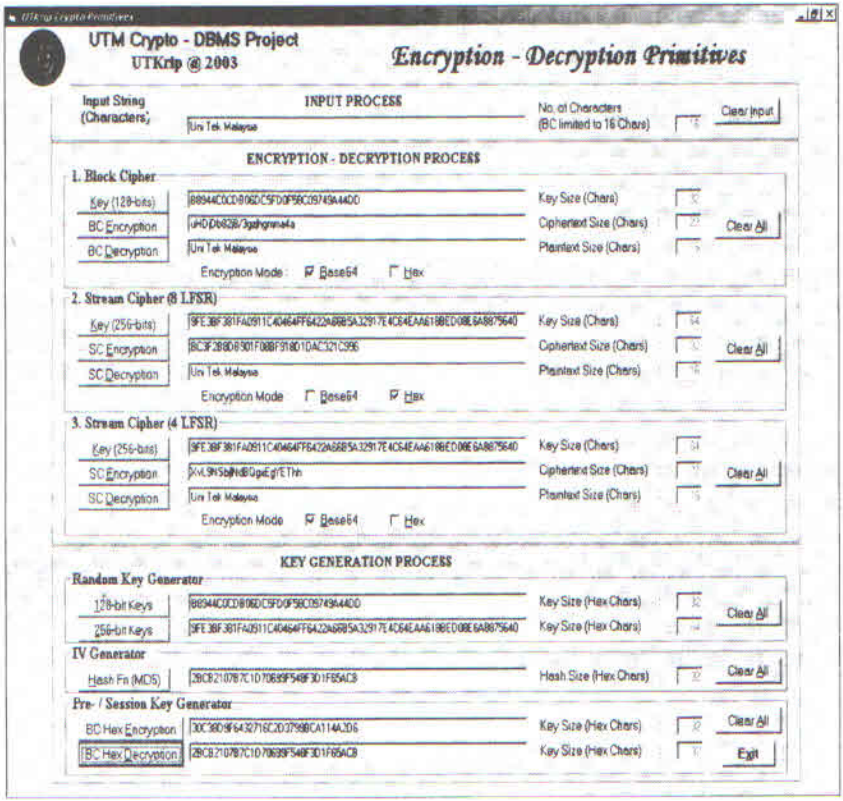

FIGURE 2. The scheme's encryption-decryption primitives

With the proper production of session keys using appropriate cryptographic keys, we discuss next the Encryption-Decryption process. The basic encryption and decryption capability of the proposed database encryption scheme has three algorithmic options for users to choose. These cryptographic algorithms are proprietary algorithms and are based on symmetric cryptography. In each of the algorithms, users can further choose the encryption mode. The input string, "Uni Tek Malaysia" is what we refer to as the plaintext or the raw form of the data to be encrypted. Once the data has been encrypted, it will be transformed into a scrambled format which we term as the ciphertext which is in an unintelligible state. The encryption mode here then is the option of choosing the state of the ciphertext output to be either in Base64 or Hexadecimal (Base16) representations. Base64 has two advantages over Hexadecimal representation, that is, it saves storage space and it is more transportable between different computer systems.

In the first encryption algorithm option users may choose the TS Block Cipher. Encryption using this cipher will provide a high degree of accuracy since block ciphers, by their very nature of design has low noise and high degree of error propagation. It is suitable for encrypting monetary and numeric fields which require high degree of accuracy. It is not suitable for encrypting voluminous amount of data because it processes 16 bytes per block of data at a time and is much slower than stream ciphers. The second and third options of encryption involved the use of TS Stream Cipher algorithm. The original TS Stream Cipher was designed using eight Linear Feedback Shift Registers (LFSR) for the secure transmission of encrypted messages. It has been proven in Tuan Sabri's Ph.D. research work that this design is highly secure. However, for the purpose of bulk data encryption and fast response time such as in database applications, a slight modification to this design was made. The alternative solution is to have only four LFSR instead of eight with no compromise in security. The security of the 4LFSR is explained in greater detail in the section on *Security of the Proposed Database Encryption Scheme*. An empirical study on the speed and performance of various encryption algorithms including TS Stream and Block ciphers will be treated in a future work. Based on our limited observation in the implementation of the scheme showed there is a great difference in throughput between the 8LFSR and the 4LFSR. The latter is much faster compared to the former.

TS STREAM CIPHER ENHANCEMENT

The original TS Stream Cipher was designed for used in the secure transmission of information and digital communications. However, in the application of security in databases where there will be voluminous amount of data to be encrypted, we need to minimize or possibly zerorized encryption patterns in the data stored in databases. This will increase the security in which cryptographic attacks based on pattern matching will be safeguarded. To achieve this security objective, the TS Stream Cipher was modified so that bulk encryption of similar data in databases will not produce obvious patterns that may expose it to cryptanalytic attacks. Figure 3 shows the encryption of similar data input having four different output ciphertext thus showing no pattern matching or correlation between the input and the output. This is true for any of the encryption mode and algorithm chosen.

DESCRIPTION OF OUR DATABASE ENCRYPTION SCHEME

The proposed database encryption scheme is composed of two sub-schemes: the first is the encryption scheme based on TS Stream Cipher algorithm; and the second is the encryption scheme based on TS Block Cipher algorithm. The

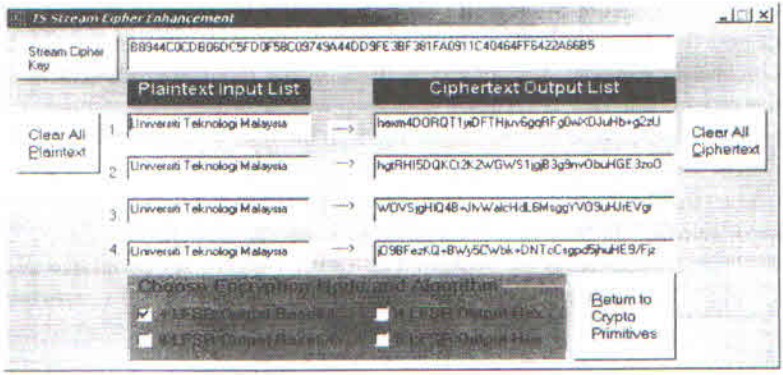

FIGURE 3. TS Stream Cipher enhancement

main difference between the two is that the former involves the generation of pre- and session keys while the later requires only the generation of session keys. As mentioned in the section on *Overview of Approach*, the encryption scheme is based on symmetric cryptography and our solutions allow the encryption and decryption of data at all levels, that is at the individual data element level, at the column/attribute or domain level, and also at the row or tuple level using both the stream and block ciphers. With this kind of flexibility, the challenge in managing the encryption and decryption processes is to ensure no double encryption of data happens. How the encryption system keep tracks of fields that have been encrypted so as to ensure no re-encryption of data will be covered in future work.

TS STREAM CIPHER ENCRYPTION SUB-SCHEME

The TS Stream Cipher sub-scheme allows the encryption and decryption of data in the database using TS Stream Cipher algorithm. Figure 4 illustrates the inner workings of the encryption engine for TS Stream Cipher sub-scheme. The encryption engine can be viewed at three levels. Level I consist of the encryption (light shaded) and decryption (dark shaded) regions, that is, the generation of the required pre- and session keys for the encryption and decryption processes, respectively. Refer to the section on *Pre-and Session Keys in the Proposed Database Encryption Scheme* for detailed explanation. Level II or the blue region is the stage where data from the database are fed into the encryption algorithm to be encrypted or decrypted, depending on the process stage. During encryption, plaintext data (not encrypted or raw data) are read from the database and with the appropriate session key generated beforehand, the data will be transformed into an unreadable format by the stream cipher before finally stored securely in the DBMS. When an authorized

user with the appropriate security clearance would like to read this encrypted data and the data class allows him to read it, the data will be fed into the deciphering program. The program then, using the same session key as produced during the encryption process, will transform the data into its original plaintext state before presenting it to the user by the database system. Level III shows the underlying database engine that allows only authorized access to the database data by cleared users based on the discretionary access control rules of the RDBMS. The ability to generate appropriate session keys from User Master Keys as explained in Level I above are reflective of the Mandatory Access Control policy of the scheme.

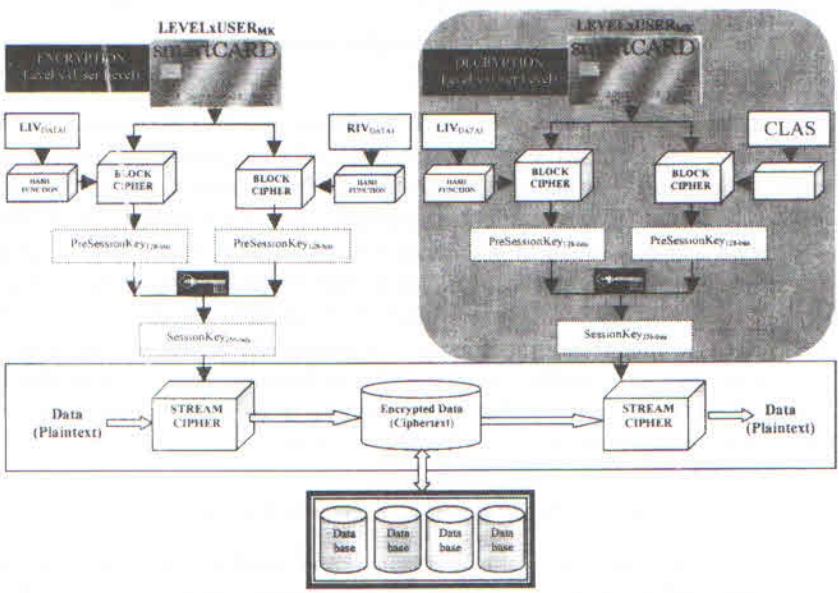

FIGURE 4. The TS Stream Cipher Database Encryption sub-scheme

TS BLOCK CIPHER ENCRYPTION SUB-SCHEME

Similar to the explanation in the previous section, the TS Block Cipher encryption sub-scheme requires only the generation of session keys. This is clearly shown in Figure 5. The encryption and decryption capability of this scheme is similar to the TS Stream Cipher encryption sub-scheme described above. Though the sub-scheme encryption process is slower compared to the TS Stream Cipher, it is more suitable for numeric or monetary fields that require high accuracy. This is true between Block and Stream Ciphers by their nature of design.

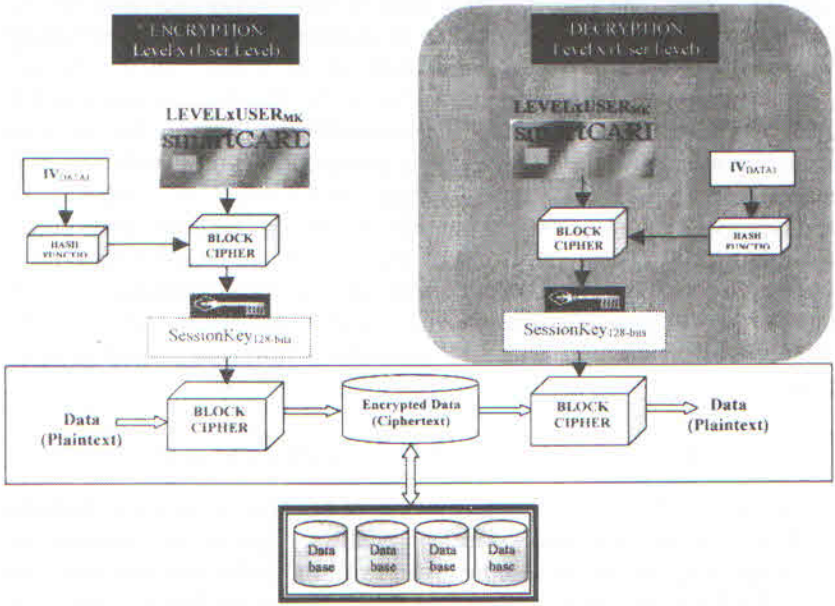

FIGURE 5. The TS Block Cipher Database Encryption sub-scheme

PRE-AND SESSION KEYS IN THE PROPOSED DATABASE ENCRYPTION SCHEME

Based on the characteristics and requirements of strong cipher design, encryption using TS Stream Cipher requires 256-bit keys and 128-bit keys for the TS Block Cipher. The database encryption scheme is based on the TS Block Cipher for its cryptographic keys which are 128-bit long. Therefore, TS Stream Cipher encryption requires two Pre-Session Keys which are concatenated to produce the 256-bit Session Key. This is achieved by dividing the data IV into two or by getting two related information regarding the data to be encrypted. These two parts of the Data IV are what we termed as the Left IV (LIV) and the Right IV (RIV).

KEY GENERATION, MANAGEMENT AND DISTRIBUTION

The driving subsystem for the database encryption scheme is its Key Generation, Management, and Distribution (KGMD) system. Figure 6 shows the KGMD system. The system produces two types of key: Master Key (MK) and Shared Key (SK). MKs are used for the MAC mechanism in a multilevel access mode while the SKs are for special access privileges and may consists of DAC and MAC implementations at various user levels, that is, at Level 0 through Level n in the users' organization hierarchy. It is discretionary in the

sense the owner may authorize other users to view his/her encrypted data by giving them a copy of the key. As can be seen from Figure 6, lower ranking MKs can be generated by the higher ranking MK holders. In the MK key generation process, users IVs are hashed using the MD5 hashing function. For SK key generation, the data class is taken as the IV. Data can be classified according to user requirements and may consist of security levels, compartments, and groups or combination of these. As have been mentioned earlier, all cryptographic keys are stored securely in medium like smartcards. Therefore, the KGMD system can be independent from the database encryption scheme for the generation, management, storage, and distribution of both types of keys. The organization may elect to place the full trust to a security manager for organization wide key generation, management, and distributions.

SECURITY OF THE PROPOSED DATABASE ENCRYPTION SCHEME

The scheme is based on TS cryptosystem. It has been proven that both the TS Block and Stream Ciphers were designed with high security meeting the set of operating conditions specified. The TS Block Cipher was designed with a 128-bit block size that passed the four statistical tests for Block Cipher local randomness developed by the Information Security Group of the Royal Holloway University of London. The TS Stream Cipher was designed with large linear complexity, large number of keys, resistant to known attacks, and passed all the statistical tests for stream ciphers.

The design of the database encryption scheme is such that all cryptographic keys are kept outside the system. There is minimum crypto information given to users that is they have their encryption and decryption keys in their smartcards. Therefore, the security of these keys is very much dependent on the secure storage of them in smartcards or other medium used. There is, however, the issue of user security protecting their own keys or collaborating with others. But this is a separate issue and should be dealt with accordingly. In this scheme, accessing data in the database is through the data IVs with appropriate cryptographic keys. This is similar to accessing the database via plaintext form; however, future work will verify the effectiveness of processing data using data IVs. In terms of security, all data IVs are stored in their hashed form.

As mentioned above, there is a separate issue of user security. In the event user keys are compromised, the risks and threats to the security of the database depends much on the status and location of the user(s) involved.

In principle, the higher the user is in the key hierarchy, the greater the risk and the amount of re-encryption needed is more. A new set of keys can be generated and given to the user(s) involved but the encrypted data need to be decrypted and re-encrypted with the new set of keys. This can be done

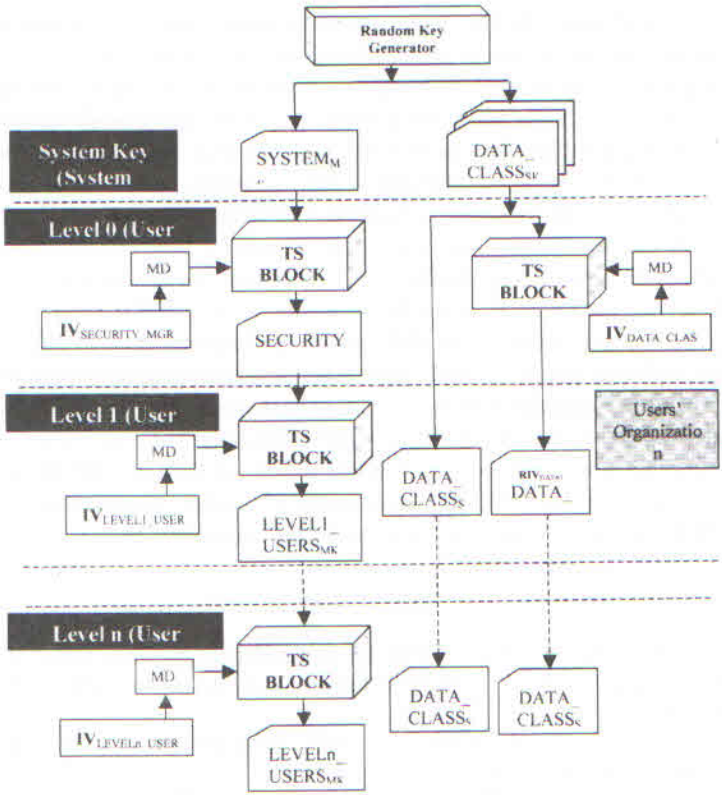

FIGURE 6. Key generation, management & distribution

offline or with a disruption in the system which is not desirable. Key recovery is easily done basically by re-generating through the KGMD system to replace non-functioning or destroyed keys. Key recovery for all user level keys can be derived from the system keys. Some system shared keys are backup keys for users. If the security procedure requires periodic key changes, new sets of keys can be generated and distributed manually by running the KGMD system. Data in the database can be re-encrypted with these new keys in the background using the backup or copy version in offline mode.

CONLUSION AND FUTURE WORK

In this paper we have presented a new database encryption scheme that is secure and suitable for implementation in a commercial RDBMS like Oracle. Our scheme allows the encryption of data at all levels including the data

elements, rows, and columns. The issuance of users' master keys and shared keys allows the enforcement of multilevel access controls and other access control policies required by organizations including discretionary and special cases. Since the scheme has limited implementation, more work need to be done to evaluate the security mechanism, and the final database performance with the scheme in place. Pernul (1995) provides some basis for the evaluation and comparison of information security. Dietrich et al. (1992) introduced the practitioner's approach to performance benchmarks and measurements for centralized databases. A benchmarking methodology for Multilevel Secure DBMS is given by Schlipper et al. (1992).

Currently, the scheme is suitable only for closed networks because of the problem with key transfer between the client and the database server. Future work will be the implementation of a public key system that will allow secure exchange of our symmetric keys for open networks and Internet accesses. With this capability, the scheme can be expanded into a well distributed database systems which the key management is capable of providing distinct key identification for each database server for controlled access.

REFERENCES

Azmi, H. 2003. Peranan Malaysia Cegah AS Kuasai Ruang Angkasa. *Berita Harian.* (dalam talian). http://202.184.94.19/bin/main.exe?f=doc&state=gf1211.9.3 (14 June 2003).

Bertino, E., Jajodia, S. and Samarati, P. 1995. Database Security: Research and Practice. *Information Systems* 20(7): 537-556.

Castano, S., Fugini, M. G., Martella, G. and Samarati, P. 1995. *Database Security.* Wokingham. UK: Addison-Wesley.

Cooper, R. H., Hyslop, W. and Patterson, W. 1984. An Application of the Chinese Remainder Theorem to Multiple-Key Encryption in Data Base Systems. *Computer Security: A Global Challenge.* In Finch, J. H. and Dougall, E. G. (Eds.) North Holland: Elsevier Science Publishers.

Davida, G. I. and Yeh, Y. 1982. Cryptographic Relational Algebra. *Proceedings of the 1982 IEEE Symposium on Security and Privacy.* pp 111-116.

Davida, G. I., Wells, D. L. and Kam, J. B. 1981. A Database Encryption System with Subkeys. *ACM Transactions on Database Systems* 6(2): 312-328.

Dastjerdi, A. B., Pieprzyk, J. and Naini, R. S. 1996. Security in Database: A Survey Study. *Communications of the* ACM 44(2): 38-44.

Denning, D. E. 1983. Field Encryption and Authentication. In Chaum, D. (Eds.) *Advances in Cryptology: Proceedings of CRYPTO 83.* pp 231-247.

Denning, D. E. 1988. Database Security. *Annual Review of Computer Science* 3:1-22.

Denning, D. E. and Denning, P. J. 1979. Data Security. *ACM Computing Surveys* 11(3): 227-249.

Dietrich, S. W., Brown, M., Cortes-Rello, E. and Wunderlin, S. 1992. A Practitioner's Introduction to Database Performance Benchmarks and Measurements. *The Computer Journal* 35(4): 322-331.

Gudes, E., Koch, H. S. and Stahl, F. A. 1976. The Application of Cryptography for Database Security, *Proceedings of the National Computer Conference* pp 97-107.

Hardjono, T. and Seberry, J. 1989. A Multilevel Encryption Scheme for Database Security. *Proceedings of the 12th Australian Computer Science Conference, Wollongong*. pp 209-218.

Hardjono, T., Jennings, M. and Lokan, C. J. 1990. Implementation of a Multilevel Encryption Scheme for Database Security. In Srinivasan, B. and Zeleznikow, J. (Eds.) *Databases in the 1990s: Proceedings of the Australian Database Research Conference*. pp 175-184.

He, J. and Wang, M. 2001. Cryptography and Relational Database Management Systems. *Proceedings of the IEEE International Symposium on Database Engineering & Applications*. pp 273-284.

Hwang, M. S. and Yang, W. P. 1997. Multilevel Secure Database Encryption with Subkeys. *Data & Knowledge Engineering* 22: 117-131.

Lin, C. H., Chang, C. C. and Lee, R. C. 1992. A Record-Oriented Cryptosystem for Database Sharing. *The Computer Journal* 35(6): 658-660.

Lunt, T. F. 1992. Security in Database Systems: A Research Perspective. *Computers & Security* 11: 41-56.

Omar, K. A. and Wells, D. L. 1983. Modified Architecture for the Sub-Keys Model. *Proceedings of the IEEE Symposium on Security and Privacy*. pp 79-86.

Pernul, G. 1995. Information Systems Security: Scope, State-of-the-Art and Evaluation of Techniques. *International Journal of Information Management* 15(3): 239-255.

Pernul, G. and Luef, G. 1992. Bibliography on Database Security. *ACM SIGMOD Record* 21(1): 105-121.

Schlipper, L. M., Filsinger, J. and Doshi, V. M. 1992. A Multilevel Secure Database Management System Benchmark. *Proceedings of the 15th National Computer Security Conference, Baltimore, Maryland.* Vol 2 pp 399-408.

Sharmini, P. 2003. Najib: Avoid War in Aceh at All Costs. *New Straits Times*. (online) http://202.184.94.19/bin/main.exe?f=doc&state=gf1211.3.1 (13 May 2003).

Tuan Sabri Tuan Mat. 2000. Design of New Block and Stream Cipher Encryption Algorithms for Data Security. Ph.D. Thesis. Skudai: Universiti Teknologi Malaysia.

Zailani Mohamed Sidek (Zailani@utmkl.utm.my)
Norbik Bashah Idris (norbik@case.utm.my)
Harihodin Selamat (harihodin@itp.utm.my)
Fakulti Sains Komputer dan Sistem Maklumat
c/o Center for Advanced Software Engineering (CASE)
Universiti Teknologi Malaysia
Jalan Semarak
54100 Kuala Lumpur
zailani@utmkl.utm.my