Data and Communication Security

Dr. Sadeq ALHAMOUZ
Amman Arab University for Graduate Studies
Amman - 11935/Jordan
Sadeq@aau.edu.jo

ABSTRACT

The regional initiative was presented by the United Nations Economic and Social Commission for Western Asia in preparation for the world summit, Dec 2003. The Initiative by itself and away from regional trouble and differences between both the Arab countries and other countries in the regions is a good and noble Initiative. However with such differences and lack of trust the security issue should be the first issue tackled and resolved. In this paper it is aimed to look at present tools and techniques available, and then suggest alternatives when possible.

Keywords: Regional Information Society, Security, Communication, RDBMS's.

1. INTRODUCTION

Regional information Society [1] would encourage the business community, governmental and private organizations to depend more and more on carrying out their day to day data transmission and communication using the information technology infra structure if the privacy and integrity of their data is protected and secured.

When using the old and new communication services, people are worried about the privacy and integrity of their business data. Information society means that government will employ E-government principles and provide all its services on the Internet. At the same time businesses will conduct all or most of their business on open net, and people will conduct their daily activities using the net. Once this is achieved different regional Governments will be able to conduct their related governmental activities over the net. In the absence of the regional problems this would be an ideal situation but unfortunately the region is not ideal, since it contains problems and differences, and by making the region an open information society without providing adequate information security at individual business and country level, would result in an unsecured region.

Security is the primary concern of government and companies that want to do business on the World Wide Web. By design information society is a free and open system, but it should also be a society where government and businesses can maintain privacy and remain secure with their information and properties. If one looks at a simple purchase activity of a web customer and traces the data flow, one can see that there are two security issues to be addressed:

- Secure data transmission: when a customer submits her/his confidential information (e.g. credit card numbers) through her/his web browser, the information should remain confidential on its way to the web server, the application server, and the backend DB server.

- Secure data storage and access: when the confidential customer data arrives at the DB server, the data should be stored in such a way that only authorized users can access them.

The secure transmission of data is well studied and well supported in today's e-business market. Almost all web browsers and web server support SSL (Secure Socket Layer) [2] or TLS (Transport Layer Security) [3].

However, once the data arrives at the backend, there is no sufficient support in storage and processing in a secure way. For example, a RDBMS might not ever provide an encryption mechanism to securely store data. Although the general problem of secure data storage is well studied, the importance of secure data storage in an RDBMS has not been fully understood, and no serious work has been carried out on how to encrypt DB data.
All of the above methodologies are not enough to convince regional government to join a Regional Information Society.

It is believed that it is vital to integrate cryptographic support into RDBS's. Cryptographic support is another important dimension of security. Complementary to access control, both should be used to guide the storage and access of confidential data in a system.

When one considers integrating cryptographic support into RDBMS, there are two general approaches:

- Loose coupling: In this case a database server can consult a third-party crypto service and there are only minor changes on server side.
- Tight coupling: In this case a complete set of basic crypto primitives are built into the database server as a set of new SQL statements, together with the necessary control execution content to ensure that those new SQL statements can be executed securely.

In this paper it is aimed to focus on two problems: how to ensure the access control mechanism to make user management more secure and close some major security holes of current RDBS's, and how to support encryption.

This paper can be summarized as follows:

1. It introduces the important concept of security.
2. Based on the new concept of security, it is aimed to show how to enhance the security of current user management mechanism deployed in all E-Government, E-business systems.
3. Propose several encryption methods.

2. THE PROBLEM

Although access control has been deployed as a security mechanism for a long time, security of a DB was considered an additional problem to be addressed when the need arose, and after threats to the secrecy and integrity of data had accrued [4]. The approach of adding security support as an optional feature is not very satisfactory, since it would always penalize the system performance, and more importantly, it is likely to open new security holes.

Database security is a wide research area [4, 5] and includes topics such as statistical database security [6], intrusion detection [7], and most recently privacy preserving data mining [8]. This section focuses on the topics of user management, access control and encryption. Briefly review how they are supported, analyze their security, and point out the potential problems.

3. DATA PRIVACY

Privacy on the Internet is an issue that is of significant interest. There are two fundamental issues:

- Privacy of data during transmission;
- Privacy of stored data.

The first issue, privacy during network transmission, has been studied widely in the Internet area and addressed by the Secure Socket Layer protocol (SSL) [2] and Transport Layer Security (TSL) protocol [3]. The second issue, privacy of stored data in relational databases is less studied and of greater relevance to database as a service model. If database as a service is to be successful, and customer data is to reside on the site of the database service provider, then the service provider needs to find a way to preserve the privacy of the user data. There needs to be security measure in place so that even if the data is stolen, the thief cannot make sense of it.

Encryption is the perfect technique to solve this problem. Prior work [9] [10] does not address the critical issue of performance. But in this work, for the first time, we have addressed and evaluated the most critical issue for the success of encryption in databases, performance. To achieve that, we have analyzed different solution alternatives.

There are two dimensions to encryption support in databases. One is the granularity of data to be encrypted or decrypted. The field, the row and the page, typically 4KB, are the alternatives. The field may appear to be the best choice, because it would minimize the number of bytes encrypted. However, as we have discovered, practical methods of embedding encryption within relational databases entail a significant start up cost for an encryption operation. Row or the page level encryption amortizes this cost over larger data. The second dimension is software versus hardware level implementation of encryption algorithms.

3.1 Software level encryption

In this work two encryption algorithms are considered: a) RSA [11] and b) Blowfish [12]. An experiment was conducted using both these algorithms and found that the performance of the Blowfish algorithm we implemented in Java is better than the RSA implementation available to us. The result of this test carried out concludes that the Blowfish algorithm is fast, compact, and simple, compared to other well-known encryption algorithms such as DES [13]. Detailed description of the algorithm is given in [13]. Blowfish is a 64-bit block cipher, which means that data is encrypted and decrypted in 64-bit chunks. This has implication on short data. Even 8-bit data, when encrypted by the algorithm will result in 64 bits.

Blowfish implementation was registered into the database as a user defined function (UDF) (also known as foreign function). Once it was registered, it could be used to encrypt the data in one or more fields whenever data was inserted into the chosen fields, the values are encrypted before being stored. On read access, the stored data is decrypted before being operated upon.

In this approach the originator of the encrypted data supplies the *key*, and the database provides the encryption function. Only those users who are given the key can decrypt the data using the decryption algorithm. Since the key is owned by the creator, and not stored at the site of the database service provider, unauthorized person who may get hold of disk files can not get hold of the key. In fact, even employees of the database service provider do not have access to the encryption key. The full security provided by the encryption algorithm is inherited by the data in the database. Note generic functions named encrypt and decrypts were used in the query. In fact, one could implement the two functions with any encryption algorithm. Also note that users of our database service can easily specify and use encryption algorithms of their choice, using the facilities provided by the database.

3.2 Hardware level encryption

Specialized encryption hardware, the IBM S/390 Cryptographic Coprocessor, is available under IBM OS/390 environment with Integrated Cryptographic Service Facility (ICSF) libraries. IBM DB2 for OS/390 provides a facility called ”editproc” (or edit routine), which can be associated with a database table. An edit routine is invoked for a whole row of the database table, whenever the row is accessed by the DBMS.

3.3 Encryption scheme alternatives

Not all possible combinations of different encryption approaches are considered, namely; software and hardware level encryption, and different data granularity.

In hardware encryption, we did not consider field level encryption. The main reason is the expansion in the original data size due to the nature of block cipher encryption algorithms. This behavior is described in the software level encryption section. This problem is not severe when the input data is typically 80-120 bytes row as generally the size of a row is relatively larger than a size of a field.

4. PROPOSED SOLUTION

From the above investigations it is clear that there is a great need for a new approach to encryption that is both highly secured and efficient to avoid the high penalty over the encrypted data. Also it needs to be easy to use and avoid any key exchange (i.e. Private or public Keys).

This encryption algorithm will need to be developed using an intelligent algorithm (i.e. Genetic Algorithm, Genetic Programming, and or Fuzzy Logic).

For test purposes a simple algorithm was developed using Genetic algorithm for key generation, which reduces the generation time and increases the efficiency of the original algorithms by 30% - 40%. The algorithm also included the facility of no key exchange as some key elements was embedded within the encrypted data in the database, it also contained the key length needed for decryption when the users retrieve the data from the database.

This algorithm is based on Rijndael Block cipher algorithm using a multi key ciphering and it offers the user the ability to choose the key length and the data to be ciphered. This algorithm is simple to use and is easy to include in any new database development, it gives complete security to the data with the total randomness provided by the Genetic Algorithm.

5. ILLUSTRATIVE EXAMPLE

A random initial population of ciphering keys is generated as shown in figure (1)

41	62	64	75
6c	6c	61	68
20	41	62	64
61	6c	69	52

Figure 1: Ciphering key generated randomly

The crossover point is chosen randomly (in this example it is assumed to be equal to 3). So each byte of the ciphering key is separated into two parts the first being three bits and the second is the remaining five bits as shown in figure (2).

Ciphering key	Ciphering key	Left Key	Right Key
41	01000001	010	00001
62	01100010	011	00010
64	01100100	011	00100
75	01110101	011	10101
6c	01101100	011	01100
6c	01101100	011	01100
61	01100001	011	00001
68	01101000	011	01000
20	00100000	001	00000
41	01000001	010	00001
62	01100010	011	00010
64	01100100	011	00100
61	01100001	011	00001
6c	01101100	011	01100
69	01101001	011	01001
52	01010010	010	10010

Figure 2: Crossover point is 3, generated randomly

Parent 1	Parent 2	Binary of Child1	Value of Child1
13	21	01100001	61
10	13	01101000	68
00	20	01000000	40
20	11	00101100	2c
21	01	01000010	42
12	01	01100010	62
02	12	01100001	61
10	11	01101100	6c
33	22	01000010	42
23	31	01101100	6c
30	22	01100010	62
31	00	01100001	61
33	00	01000001	41
20	10	00101100	2c
01	12	01100001	61
31	00	01100001	61

Figure 3: Child 1 result of mutating elements.

Figure (3): shows first child that is a result of mutating of two parents (selected randomly) whereas figure4 shows the other child. The first key of the child 1 is the brother of the first key of the child 2 as they both have the same parents.

Parent 1	Parent 2	Binary of Child2	Value of Child 2
21	13	00001011	0b
13	10	01000011	43
20	00	00000010	02
11	20	01100001	61
01	21	00010010	12
01	12	00010011	13
12	02	00001011	0b
11	10	01100011	63
22	33	00010010	12
31	23	01100011	63
22	30	00010011	13
00	31	00001011	0b
00	33	00001010	0a
10	20	01100001	61
12	01	00001011	0b
00	31	00001011	0b

Figure 4: Child 1 result of mating elements.

0a	00001010	00	33

Figure 5: Child 1, 0a comes from parent 00 and 33 elements

Figure (5) shows that child 1 is 0a comes from mating 00 element, which is 41 which is represented in binary format as in figure (6), with 33 element which is 52 and its binary representation is shown in figure (7).

The new child (0a) is constructed as follows, the first part of it (i.e. the first 5 bits) (00001) comes from ciphering key 00, which is 41 and represented in binary in figure (6). Whereas the second part of it (i.e. the remaining 3 bits) (010) comes from the ciphering key 33, which is 52 and represented as in figure (7)

41	01000001	010	00001

Figure 6: First parent with crossover point 3

52	01010010	010	10010

Figure 7: Second parent with crossover point 3

Figure (8) shows the brother of the element (0a) which is 41 and they have the same parents as 41 comes from mating 41 and 52 at crossover point 3.

Figure (8) show that the new child (41) is constructed as follows, the first part of it (i.e. the first 3 bits) (010) comes from ciphering key 33, which is 52 and represented in binary in figure (7). Whereas the second part of it (i.e. the remaining 5 bits) (00001) comes from the ciphering key 00, which is 52 and represented as in figure (6)

33	00		01000001	41

Figure 8: child 2, comes from parents 33 and 00

The result for first round is shown in figure (9) that consists of original parent and the two children.

41	62	64	75
6c	6c	61	68
20	41	62	64
61	6c	69	52

72	72	6c	72
61	52	48	69
60	75	60	69
61	72	4c	60

0b	43	02	61
12	13	0b	63
12	63	13	0b
0a	61	0b	0b

Figure 9: new population contains parent and two children

5. CONCLUSIONS

In this paper the data security concern was raised and a number of solutions is introduced to overcome this concern. The encryption option as was shown in this paper is the best available option to government and private sectors. This option of course need some special experience and needs to be unique as per individual institute other wise the proposed problems will continue to affect these institutions. The paper also proposed and showed that an intelligent encryption algorithm can be used which can overcome most of the outlined problems.

6. REFERENCES

[1] United Nations, Economic and Social Commission for Western Asia**, Towards a Regional Information Society. Western Asia Preparatory Conference for the world Summit on the Information Society,** February 4-6,2004, Beirut, Lebanon.

[2] A. Freier, P. Karlton, and P. Kocher. The SSL Protocal Version 3.0, **Internet-Draft**. November 1996.

[3] T. Dierks and C. Allen. The TLS Protocal Version 1.0, **Internet-Draft**. November 1997.

[4] S. Castano, M. Fugini, G. Martella, and P. Samarti. **Database Security.** Addison-Wessley, 1995.

[5] D. E. Denning. **Cryptography and Data Security.** Addison-Wesley Publishing Company, Inc., 1982.

[6] N. R. Adam and J. C. Wortmann. Security control methods for statistical databases: a comparative study. **ACM Computing Surveys,** 21(4):515-556, 1989.

[7] T. F. Lunt. A survey of intrusion detection techniques. **Computer & Security,** 12(4), 1993.

[8] R. Agrawal and R. Srikant. Privacy-preserving data mining. In Proceedings of the 2000 **ACM SIGMOD International Conference on Management of Data**, Dallas, Texas, 2000.

[9] J. He and M. wang. Encryption in relational database management systems. In Proc. **Fourteenth Annual IFIP WG 11.3 Working Conference on database Security (DBSec'00),** Schoorl, The Netherlands, 2000.

[10] G. Davida, D. Wells, and J. Kam. A database encryption system with subkeys. **ACM Transactions on database Systems**, 6(2), 1981.

[11] R. L. Rivest, A. Shamir, and L. M. Adleman. A method for obtaining digital signatures and public key cryptosystems. **Communications of the ACM**, 21(2):120-126, 1978.

[12] B. Schneier. Description of a new vaiable-length key, block cipher (Blowfish), fast software encryption. In **Cambridge Security Workshop Proceedings**, pages 191-204, 1994.

[13] B. Shneier. **Applied Cryptography**. John Wiley & Sons, Inc. 1996