International Electronic Journal of Pure and Applied Mathematics

Volume 7 No. 4 2014, 161-171
ISSN: 1314-0744
url: http://www.e.ijpam.eu
doi: http://dx.doi.org/10.12732/iejpam.v7i4.2

AUTOMATIC DECODING KEYS GENERATION FOR DIGITAL SIGNING IN A DATABASE

Dimitar Pilev

University of Chemical Technology and Metallurgy
8, Kl. Ohridski, 1756 Sofia, BULGARIA

Abstract: Provision of information security in Web-based databases is extremely pressing issue. Some of the most often used approaches for protection of data against unauthorized access are the encryption and digital signature. Within these approaches, storage of decoding keys is very important. The present article offers an algorithm for automatic generation of decoding keys, using data from the client's digital certificate. The algorithm is implemented and used for protection of a particular Web-based information system.

Key Words: database security, database encryption, digital certificate, digital signature, web-based information system

1. Introduction

Modern web-based information systems (IS) occupy larger and larger part in the public information service. Guaranteeing the security of stored information plays an essential part in the design, construction and maintenance of these systems.

There is a wide diversity of approaches, which can be used for provision of security of the web-based applications. Each one of them has its advantages and disadvantages. The three main levels of security, which each information system must have - authentication, authorization and access to data are presented in [1]. Most often, the attacks against the IS are related to breaches, using default account user names and passwords. The system administrators are continuously working hard to secure users' accounts. Other attacks are related to making non-encrypted, sensitive data available to the public and/or modifying it in an unauthorized way.

Received: January 12, 2014

International Electronic Journal of Pure and Applied Mathematics – IEJPAM, Volume 7, No. 4 (2014)

Today, there are a number of methods of ensuring security of data, stored in the IS. In general, they are connected to provision of security of the application, database and operating system. Database (DB) security is of particular importance [2], [3]. Classical attacks on DB and ways for their prevention are described in [1]. One of the most often used methods for guaranteeing security of data in the DB is the encryption [4]. It consists in using encoding techniques transforming ordinary text into an encrypted text. This way, data can only be read by people, who have the necessary decryption key. Different methods for encryption of data in DB are presented in [5]. However, the fundamental problem in all of them is the storage of decoding keys.

The present development presents an algorithm for digital signing [6] of sensitive, and particularly important data. Characteristic for this algorithm is that data signing is implemented on application level. This allows only part of the data stored in the DB to be signed, which increases the system performance. On the other hand, no additional method for storage of private keys of authorized users is required. Such keys are never stored in their native form in the DB, application or file system of the Web-server. This significantly increases the security of data.

2. Data Protection through Encryption

One of the most often used approaches for security provision of database is encryption. It can be symmetric or asymmetric. Symmetric encryption is most often used for protection against unauthorised access used for modification of data and/or making it publicly available. This method can be used to encrypt the entire DB, a single table or a table column, and the encryption and decryption are implemented using one and the same key [10], [13]. To certify the authenticity of the stored data, as well as to find out who performed their modification, it is necessary to use asymmetric encoding or digital data signing [4]. This method is particularly suitable for information systems, where many users have access and can modify different parts of data. Using encryption for protection of web-based applications can be done on one of the three levels – application, DB and operating system [5], [7].

There are a number of DB encryption forms. Usually, they are divided into transparent and user encryption. Transparent encryption covers the whole DB. It does not require change in the code of the application. User encryption is applied only on selected objects within the DB. It requires changes in the code of the application. Transparent encryption is in fact designed to protect data on disks and tapes. User encryption can be also used for protection against data abuse.

Encryption of data in the DB ensures high security of the data. However, when developing the encryption strategy, we have to select the encryption level, data to be encrypted, encryption algorithm, access to the cryptographic keys, etc. All these factors have affect the system performance.

International Electronic Journal of Pure and Applied Mathematics – IEJPAM, Volume 7, No. 4 (2014)

2.1. Encryption on Operating System Level

Data is encrypted in the storage system, thus protecting it against unauthorised access in case of theft of disks and archiving tapes. This encryption level is suitable for encryption of files or whole directories within the operating system. No modification in the existing applications is required. However, in this case no difference is made in the users' privileges, which does not allow using different encrypting keys. Selective encryption of data is not allowed, either.

2.2. Encryption on Database Level

Encryption of data can be part of the DB design and thus, it can be related to the sensitivity of data and user privileges. On this level, selective encryption is possible. It can be implemented on different levels – tables, columns, rows. Depending on the encryption used and the database management system (DBMS), the encryption process may require some modifications within the application. There is a potential for a DBMS performance drop, since encrypted data indexing cannot be used.

Within both levels of encryption, data is decrypted on the DB server. In this way, the cryptographic keys must be transmitted to the DB server or stored together with the encrypted data. This provides limited protection again unauthorised access to the operating system of the web-server. Attacks can be directed against the server's operation memory and they can target discovering the necessary keys and sensitive data in their plain form.

2.3. Encryption on Application Level

The encryption/ decryption process is performed by the application. Data is sent to the DB in encrypted form. It has to be decrypted by the application. The advantage of this approach is that it separates encryption keys from encrypted data. Modifications within the application are required. Usage of certain additional functionalities of the DB (such as, stored procedures and triggers) is forbidden.

Problems, related to storage of keys for decryption of encrypted data occur within all three data encoding approaches. The keys must never be stored in their plain form in the DB, application or file system of the Web-server. Otherwise, if unauthorised user somehow manages to gain access to them, the effect from encryption of data will be null.

Most manufacturers of DBMS provide integrated encryption options, permitting application developers to use additional data security measures through selective encryption of stored data. Such options can be implemented in the form of encryption packages [10], functions, which can be integrated in SQL queries [9] or SQL extensions [11], [10].

SQL Server 2008 [10] introduced TDE (transparent data encryption), which is

International Electronic Journal of Pure and Applied Mathematics – IEJPAM, Volume 7, No. 4 (2014)

similar to the encryption on operating system level. The whole DB is protected by one key (Database Encryption Key - DEK). TDE performs all cryptographic operations on input/output level, but within the DB, eliminating thus any need of creation of user code for encryption and decryption of data by the application developers. Oracle (version 10gR2/11gR1) offers a similar technology, which considerably extends the opportunities for using cryptography in the DBMS [12]. Selective encryption can be implemented within a column, table or a set of data files, corresponding to one or several tables and indexes.

3. Database Data Authentication Algorithm

The issues related to integration and functionality of the digital signature in relational DB applications are detailed in [13]. The article does not consider one of the most important parts in the process of digital data signing, and namely, storage of keys used for the electronic signature. The present development offers efficient algorithm for storage and manipulation of keys of the authorised users, which are used for digital signing of sensitive data. The algorithm includes the following three stages:

- Storage of the private and public keys of the authorised users
- Digital signing of modification sensitive data by authorised user
- Verification of the digitally signed data

Signing is implemented on the application level, and can be done selectively – one column, several columns or the whole table. The method is particularly suitable for information systems, where many authorised users have access to modify different part of data. Certifying the authenticity of stored data is of particular importance for these systems. The private and the public key of the authorised user of the IS, required for digital signing and verification of the modified data are stored in the DB. The public key used for signature verification, is stored in its plain form, while the private key to it, which is used for the digital signing of data, is stored in the DB in an encoded form. The encoding is done using symmetrical algorithm [14]. The symmetrical key for encoding and decoding (SKED) of the private key of the user is generated by a specific algorithm. Data, extracted from the digital user certificate, is used. In this way, storage of SKED in the DB, in the application or on the disk is not required.

International Electronic Journal of Pure and Applied Mathematics – IEJPAM, Volume 7, No. 4 (2014)

3.1. Algorithm for Storage of Private and Public Key of an Authorised User

To get an authorised access to the resources of the web-based IS, the user has to possess a valid digital client certificate. Through their client certificates users can authenticate themselves within the server and the application, servicing the IS.

The following algorithm (see Figure 1) is suggested for extraction and storage of the private and public keys of the authorised users in the DB:

1. A client certificate is generated, signed by the web-server.
2. The client's certificate is provided to the user.
3. A copy of the issued and signed by the web-server client's certificate is provided to the administrator of the web-based IS.
4. The private and public keys are extracted from the client's certificate.
5. Additional information from the client's certificate is extracted and SKED is generated.
6. The client's private key is encoded with SKED.
7. The public and the SKED encoded private key of the authorised user are stored in the DB.
8. All keys and the copy of the client's certificate provided to the administrator are destroyed.

Characteristic for the proposed algorithm is that after the storage of the required data in the DB, all materials, which can in any way be used to find out the private key of a certain user are destroyed. Only the person, who has the respective access rights and the client's certificate will be able to sign data in the database using the respective private key.

3.2. Algorithm for Digital Signing of Data Sensitive to Modifications by an Authorised User

In order to guarantee the authenticity of the data, as well as to find out the user, who stored this data, it is necessary to use additional facilities for validation and authentication.

Specific for the suggested algorithm is the way, in which the public and private keys (used for signing and validation of data in the DB) are stored. The algorithm for digital signing of data, sensitive to modifications, is as follow:

International Electronic Journal of Pure and Applied Mathematics – IEJPAM, Volume 7, No. 4 (2014)

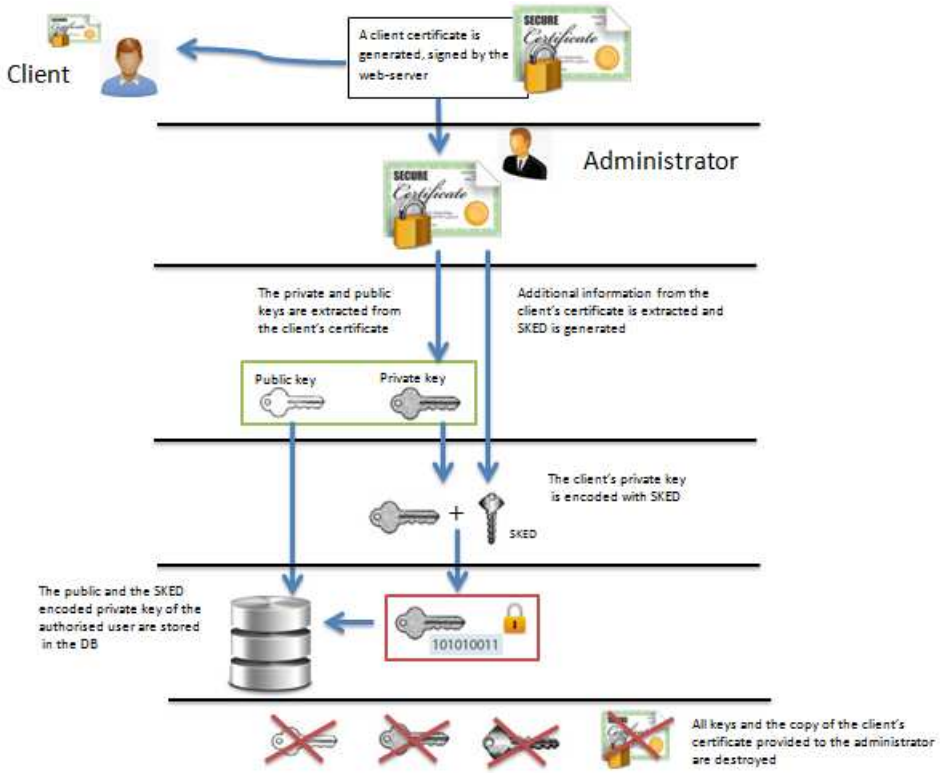

Figure 1: Algorithm for extraction and storage of the private and public keys of the authorised user in the DB

1. Logging of the user into the system (user name and password required). If the registration is successful the system proceeds to step 2, if not – to step 10.

2. Verification of the client's certificate by the server. If the check is successful the system proceeds to step 3, if not - to step 10.

3. Data from the client's certificate is extracted and SKED is generated.

4. The encoded private key of the logged in the system user is extracted from the DB.

5. The private key of the user is decoded using the SKED generated in Section 3.

6. The hash value of the data entered by the user is extracted for storage in the database.

7. The hash value of the data entered by the user is encoded.

International Electronic Journal of Pure and Applied Mathematics – IEJPAM, Volume 7, No. 4 (2014)

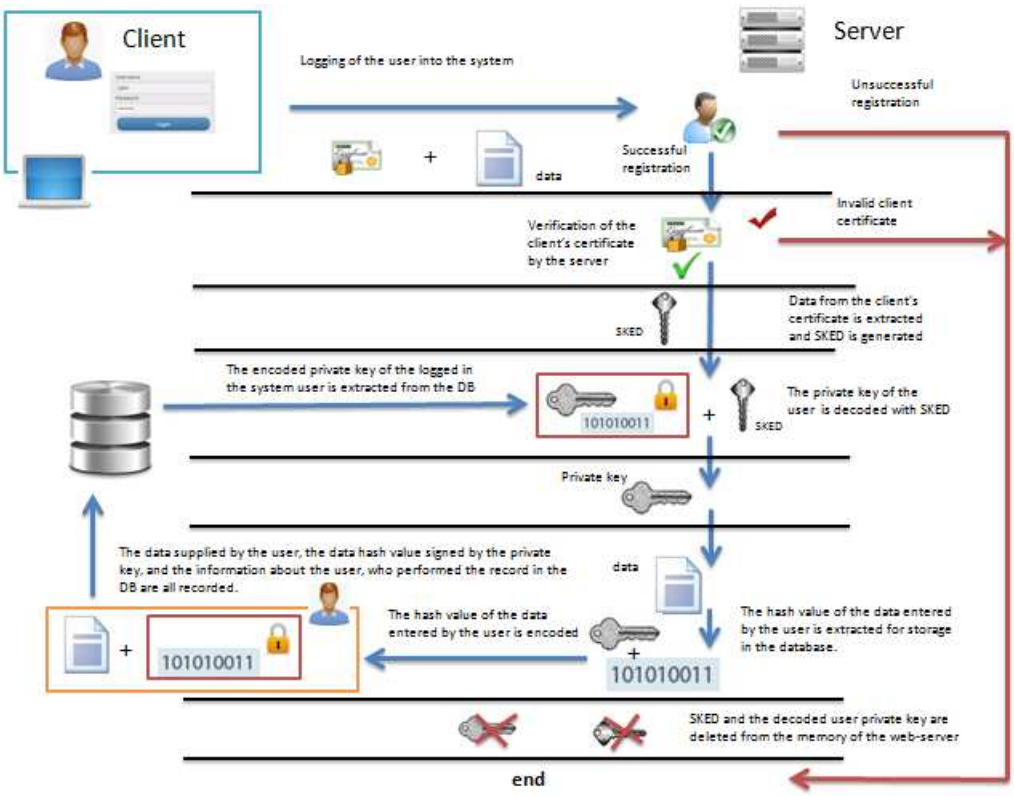

Figure 2: Algorithm for digital signing of data sensitive to modifications by authorised users

8. The data supplied by the user, the data hash value signed by the private key, the date and the information about the user, who performed the record in the DB are all recorded.

9. SKED and the decoded user private key are deleted from the memory of the web-server.

10. End of the algorithm.

The above suggested algorithm does not require any additional procedures for storage and manipulation of users' private keys, used for the digital signing of data in the DB. The keys, used for decoding of the private keys are never stored in the DB, or in the application of the IS, or in the web-server's file system. In this way, even in case of unauthorised access to the DB, application or web-server, the private keys of the authorised users cannot be extracted. To access the user's private key (used for data signing) the digital certificate will be needed, through which the user logged

International Electronic Journal of Pure and Applied Mathematics – IEJPAM, Volume 7, No. 4 (2014)

into the system. It is only this way that the SKED required for decoding of user's private key can be generated from the information stored in the user certificate.

3.3. Algorithm for Verification of Digitally Signed Data in the Database

The verification of digitally signed and stored data is performed by the familiar algorithm for verification of digitally signed data [6]:

1. Query to the DB for extraction of specific data.
2. Extraction of the data, signed hash value and information about the user, who signed the data from the DB.
3. Extraction of the public key of the user, who signed the data.
4. Decoding of the hash value of the signed data using the public key extracted in step 3.
5. The hash value of data extracted from the DB is calculated.
6. Comparison of the hash values, obtained in step 4 and 5. If they match, the system proceeds to step 7, if not - to step 8.
7. Return of the query results to the user. Proceeding to step 9.
8. Generation of error message. Notification of the system administrator about the error occurred.
9. End of the algorithm.

4. Implementation of the Algorithm for Data Protection in the DB

The suggested algorithm is implemented to guarantee the security of database within web-based information system backing the academic activities in universities [15],[16]. The system provides information about the marks from carried out examinations and additional statistical information about the results of the students by groups, subjects, streams, etc. Using the cluster analysis suggested in [17], this data can be further used for improving the quality of education. Updating and publishing of data in the IS is done by the lecturers of different subjects. This way, the quick, easy and efficient update of the information within the system is guaranteed. Guaranteeing the authenticity of data stored in the system – the students' marks, is of particularly importance. This system is characterized by the large number of

International Electronic Journal of Pure and Applied Mathematics – IEJPAM, Volume 7, No. 4 (2014)

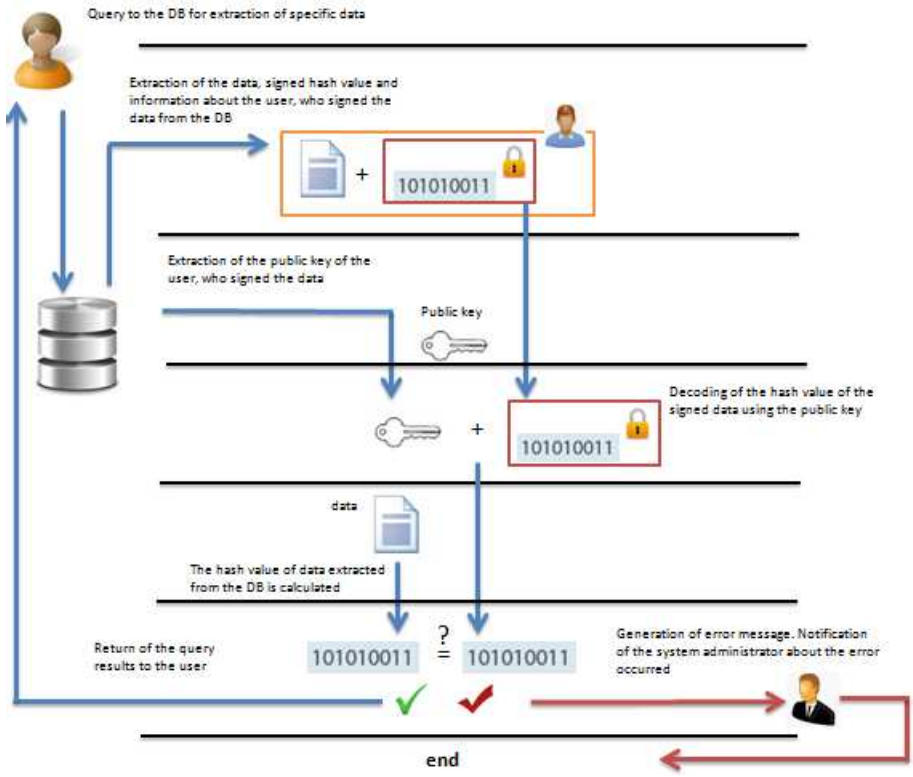

Figure 3: Algorithm for verification of digitally signed data in the database

authorised lecturers, who can enter marks only of the students they teach, in the respective subjects. The digital signing of the marks of the students guarantees their authenticity and the authenticity of the lecturer who entered them. No additional procedure for storage of lecturers' private keys is required.

5. Conclusion

The proposed algorithm is for authentication of data in the DB using digital signing. This algorithm is characterized by storing in the DB of the private key of the authorised user in an encoded form. Symmetric encoding is used. The symmetric key for encoding and decoding (SKED) of the private key of the user is generated automatically. To this end, data from the user's digital certificate is used. In this way, SKED is never stored in the DB, or in the application, or in the file system of the Web-server. This considerably increases the security of data. The algorithm can be used in all systems where many different authorised users have access to different parts of the data.

International Electronic Journal of Pure and Applied Mathematics – IEJPAM, Volume 7, No. 4 (2014)

References

[1] Database security - attacks and control methods, Emil burtescu1, PhD, Associate Professor, Department of Accounting and Management Informatics, University of Pitesti, Pitesti, Romania

[2] R. Agrawal and J. Kiernan. Watermarking relational databases. In 28th Int'lConference on Very Large Databases, Hong Kong, China, August 2002.

[3] R. Agrawal, J. Kiernan, R. Srikant, and Y. Xu. Hippocratic databases. InProc. of the 28th Int'l Conference on Very Large Databases, Hong Kong,China, August 2002.

[4] Laboratories, R. PKCS #1 v2.1: RSA Cryptography Standard, Available from:ftp://ftp.rsasecurity.com/pub/pkcs/pkcs-1/pkcs-1v2-1.pdf

[5] Database Encryption, Luc Bouganim, Yanli GUO, INRIA Rocquencourt Le Chesnay, France

[6] Jalal Feghhi, Jalil Feghhi, Peter Williams, Digital Certificates: Applied Internet Security,1999

[7] Database Encryption - How to Balance Security with Performance Ulf T. Mattsson Protegrity Corp.

[8] Database Encryption in Oracle9i, An Oracle Technical White Paper, February 2001

[9] IBM corporation, IBM Database Encryption Expert: Securing data in DB2, 2007.

[10] Sung Hsueh, Database Encryption in SQL Server 2008 Enterprise Edition, SQL Server 9 Technical Article, 2008, available from: http://msdn.microsoft.com/en-us/library/cc278098.aspx.

[11] Sybase Inc, Sybase Adaptive Server Enterprise Encryption Option: Protecting Sensitive Data, 2008.

[12] racle Corporation, Oracle Advanced Security Transparent Data Encryption Best Practices, White Paper, July 2012.

[13] Gradkell systems INC, Digital Signatures in Relational Database Applications, 2001

[14] Kessler, G.C. An Overview of Cryptography, 2014, Available from: http://www.garykessler.net/library/crypto.html

[15] A Georgieva, D. Pilev, Web - Based Information System with Temporal Database, 40 Years Department of Industrial Automation Anniversary Scientific Conference with International Participation, Sofia,18-19 April 2011, ISBN 978-954-465-043-8

[16] Dimirar Pilev, Aneta Georgieva, Effective Time Temporal Database Model, International Journal on Information Technologies and Security, 2012. N2: p. 33-46, ISSN 1313-8251

[17] P. Halachev, Model for Diferentiating Students Through Cluster Analysis, Technical University of Gabrovo,UNITECH 2013 International Scientific Conference Proceedings Gabrovo, Bulgaria, 22-23 November 2013, Volume II, ISSN 1313-230X, p. 397-401