A Unified Network Security and Fine-Grained Database Access Control Model

Amit Kuraria M.E. (computer science) IVth semester Shri Ram Institute of Technology,Jabalpur(M.P.),INDIA

Prof.Vikram Jain, Shri ram Institute of technology ,Jabalpur(M.P),INDIA
(Mtech. VIT,vellore.Thesis Co-guide)
Prof. Sweta Modi, Shri ram Institute of technology ,Jabalpur(M.P) ,INDIA
(Mtech ,IIT Roorkee,Thesis Guide)

Abstract—With the development of Internet and Intranet, Web and distributed databases have been used more and more widely. It is important to properly handle network and Web database security issues including authentication, denial of service, and fine-grained access control. When database access control and the network security are addressed separately, the security systems are not optimized sufficiently as a whole. This paper proposes a method of integrating network security with criterion based access control to handle network security and the fine-grained Web database access control simultaneously. To improve efficiency, the model adopts two step access controls. The first preliminary access control is combined with the firewall function, and the second fine-grained access decisions are determined by the user's digital credentials as well as other factors such as his/her IP address.

Keywords: *Network Security, fine-grained access control, Web and distributed database, criterion-based access control*

1. Introduction

Web applications become wide-spread and more and more companies take the advantage of them to increase their revenues. Web and distributed databases play the key role in most of these Web applications and thus it is critical to protect them from unauthorized access and malicious attacks. Web and distributed database security has aroused many researchers interests. Because of the high accessibility, Web and distributed databases tend to be more vulnerable and expose to various attacks from wide variety of sources. To address this issue, a more efficient and flexible security mechanism is required to systematically authenticate users, control network traffic, and provide efficient fine-grained access control. Web and distributed databases need a strong authentication system. In the Internet environment, the possibilities of impersonation increase. The identity of a remote user must be verified based on his/her IP address, password, and credentials to combat the repudiation attack. Since denying the requests at early stage can significantly increase the efficiency of the network, the required firewall system should not only filter the network traffic but also provide the preliminary access control. Because a remote user's permissions depends on his/her credentials as well as other factors including his/her IP address (location), the security mechanism should further refine the user's permissions based on all of these factors. The security mechanism should also provide the fine-grained access control based on these refined permissions. Currently, network security and database security are often addressed separately and therefore the security system is not optimized properly as a whole. Besides, the computational cost of fine-grained access control is high if the access control system directly implements the organization’s security policy. This is because different organizations usually have different security policy and there are big semantic gaps between security policy and implementation. To improve the efficiency of the fine-grained access control, a criterion-based multilevel database access control approach has been proposed. The approach transforms security policy into security criterion expressions and security criterion subsets upon which the fine-grained access control is achieved. Although this approach is not specifically designed to address the security requirements of Web and distributed database, it can actually be applied to these situations and combined with network security mechanisms.

This paper presents how to apply the criterion-based access control to Web and distributed database, and explores the method of developing a unified network and database security system. The rest of the paper is organized as follows. Section 2 briefly overviews the previous works of network security and fine-grained database access control. In section 3, the criterion based access control is reviewed. Many important concepts of this model are discussed. Section 4 presents a unified network and fine-grained Web and distributed database security model. Section 5 concludes the paper.

2. Overview of Network Security and Fine-Grained Database Access Control

Computer network security has been intensively studied for several decades. The first step of network security is to authenticate users. The authentication can be done based on one or more factors such as what you know (e.g. password), what you have (e.g. smart card), and what you are (e.g. fingerprint). After authentication, a firewall enforces access policies such as what services are allowed to be accessed by the network users. The types of firewall include packet filter, application gateway, circuit-level gateway, and proxy server. The third technique for the network security is the intrusion detection system which monitors the network and detects and stops the unexpected traffics and abnormal behaviors. To protect the information in transition, it can be encrypted by public or private key encryption.

Multilevel database security has also attracted a lot of attention. C. Pfleeger and S. Pfleeger presented many important multilevel database security solutions, including partitioning model, encryption, integrity and sensitivity locks, trusted front-end, and view etc. Partitioning model divides the database into separate databases according to the security level. Each of the divided database stays at a specific security level. However, this solution is against the basic database principle of elimination redundancy. Encryption model uses a unique key to encrypt data of each security level. The problem with this solution is the high overhead when processing a query because data need to be decrypted first. In the proposal of integrity and sensitivity lock, each data item has a lock which is the combination of a unique identifier and unforgettable, unique, and concealed label. The lock is used to handle access control. The disadvantages of this solution include inefficiency, high storage cost, and Trojan horse attack. Trusted front-end solution adds a trusted front-end between users and DBMS. A disadvantage of this method is the complexity of the front-end system and the separation of the database. The view solution uses a view to represent and filter a user's subset of database. The drawback of this method is the complexity of creating and maintaining the views. There are also many other solutions. L. Null etc. proposed a method of combining trusted filter and an inference engine T. Didriksen presented a rule based database access control. He partitioned a database into fragments and extended SQL to specify data fragmentation and access control. He also adopted "meta" table to hold the security policy.

In our criterion-based method, the security policy is transformed into security criterion expressions without partitioning the database or introducing the "meta" table. Meanwhile, users' security attributes are specified by the security criterion subsets. The fine-grained access control is achieved by evaluating security criterion expressions with user's security criterion subset. This method is further explained in the next section.

3. The Criterion-Based Access Control

A. *Basic Concepts*

The criterion-based access control approach was first proposed to integrate with role-based access control model to deal with multilayer security of multimedia applications .The approach also works well independently for fine-grained database access control. In this approach, security criteria, security criterion expressions, and security criterion subsets are introduced. Security criterion expressions and security criterion subsets serve as locks and keys, respectively. Each object or sub object is embedded into a lock and each user (subject) is assigned a set of keys. The user's keys are used to actuate the locks and the state of the locks determines whether the user has access to an object or sub object. A security criterion is a criterion used to both specify the user's security attributes and define the object's (and the sub object's) security attributes. Each security criterion is represented by a symbol s_i . Security criteria are abstracted from authorization rules. An authorization rule specifies who is authorized to do what. For example, an authorization rule may specify that a junior bank teller do not have access to customers' mortgage information. From this authorization rule, we introduce a security criterion s_3 to indicate both users of junior bank teller and objects (sub objects) of mortgage information. From the whole set of authorization rules, a set of security criteria s_1 , s_2 ,..., s_n in an application domain can be abstracted. The collection of all security criteria, their complement counterparts (s_j), constant false F and true T form a set which is called the security criterion set, and is denoted as SCS, that is, SCS={F, T, s_1 , s_2 ,..., s_n , s_1 , s_2 ,..., s_n}. A user may have more than one security attributes. So, several security criteria are often required to specify the user's security attributes. The set composed of these security criteria is called a security criterion subset (SCSS). When a user is associated with a security criterion subset (SCSS), he/she is enhanced to be a secure user (SU). For example, a secure user's security criterion subset is { s_1 , s_2 , s_3 }, where s_1 , s_2 , s_3 represent "employee," "not a manager", and "a junior bank teller," respectively. The special null security criterion subset indicates that there are no authorization rules confining the user accessing any part of the objects. To precisely reflect authorization rules, objects (and the sub objects), as well as their security attributes, are defined by security criterion expressions. A security criterion expression (SCE) is a Boolean expression in terms of security criteria. A Boolean expression is considered to be a security criterion expression if it reflects one or more authorization rules. Following are legal security criterion expressions:

(1) A constant true, T, or false, F

(2) A Boolean expression derived directly from an authorization rule

(3) Logical "OR" of (1) and (2)

The constant true, T, or false, F, represents special cases. When an (sub) object needs unconditional protection, its corresponding security criterion expression should be the constant true, T (reserving T for completeness in theory). On the other hand, when the security criterion expression is the constant false, F, the related (sub) object is accessible in any circumstances. To support fine-grained multilevel access, in an object, each part (i.e. sub object) with different security attributes and thus of different security levels has an embedded security criterion expression to specify its security attributes. A sub object with an embedded security criterion expression is a secure sub object. In a relational database, the objects that need to be protected include tables, views, logs and so forth. Because views and logs can be considered as special tables, to simplify the discussion, we confine the objects to tables. A sub object can be a cell, a row or a column in a table. If the whole database is regarded as an object, the table can also be treated as a sub object.

Example 1 presents a secure object: In a bank system, some information is sensitive and inaccessible to some employees according to the security policy. The sensitivity needs to be specified. To specify the security attributes and the security level of the data in a table, a special row and a column are added to hold corresponding security criterion expressions of each row and column. Non-sensitive information corresponds with a special security criterion expression, constant F, and sensitive information corresponds with more complex security criterion expressions abstracted from authorization rules. Table 1 shows the secure object (table of customer records). Its first row and last column are used to contain corresponding security criterion expressions. In the first row, a security criterion expression in certain column (e.g. Column 6) specifies the security attributes and security level of the data in that column (e.g. Mortgage). In the same way, the security criterion expression in certain row (e.g. the fourth row) and last column specifies the security attributes and security level of the data in that row (e.g. William Wilson). The security attributes and security level of the cell (4, 6) (e.g. 40,000) is the logical "OR" of these two security criterion expressions (e.g. ($s_1 \wedge s_2$) $\vee$ ($s_1 \wedge s_3$)).

B. Security criterion abstraction and secure object and secure user generations

A systematic method has been developed to abstract security criteria from authorization rules, to transform authorization rules into security criterion expressions, and to generate security criterion subset based on the authorization rules . To save space, only the summary of the major steps is presented here. For details, please reference .

Step 1. Transform conditions required to specify users' security attributes into security criteria from authorization rules.

The basic idea is that only those conditions required to specify the users' security attributes are abstracted. One security criterion is abstracted to represent one specific condition. For example, two security criteria s_1 , s_3 are abstracted to specify "employee" and "junior bank teller" from the authorization rule "junior bank teller do not have access to the mortgage information."

Step 2. Creating secure object

Step 2.1 Using security criterion expressions to represent the authorization rule(s). Each authorization rule can be expressed by a security criterion expression. For example, the security criterion expression corresponding to the above authorization rule is $s_1 \wedge s_3$, which means "employees of junior bank teller don't have access to the Mortgage information." If more than one authorization rules are relevant to an (sub) object, the logical "OR" of the security criterion expressions with respect to different authorization rules is the final security criterion expression for that (sub) object.

Step 2.2 upgrade the object The table (object) is extended to insert a row and a column to which the relevant security criterion expressions are added (see table 1).

TABLE I. A SECURE OBJECT OF CUSTOMER TABLE

F	F		F	F	$s_1 \wedge s_3$	$(s_1 \wedge s_2) \vee (s_1 \wedge s_3)$, $s_1 \wedge s_3$, $s_1 \wedge s_2$
Name	Address	…	Check account	Investment	Mortgage	F
Shawn Smith	1234 Ontario Street, Nile, IL	…	102.68	15,000	81,000	F
William Wilson	568 35th street, Nile IL	…	1,218	40,000	123,000	$s_1 \wedge s_2$
…	…	…	…	…	…	…
Jim Johnson	5969 College Ave. Nile, IL	…	250.25	32,000	54,000	F
Hannah Howard	7785 Beach St. Nile, IL	…	89.35	0	68,000	$s_1 \wedge s_2$
Wendy White	332 Fisher St. Nile, IL	…	3,022	100,000	228,000	F

Step 3. Secure user generation The security criterion subset associated with a user is generated according to the user's security attributes. In most cases, the security criterion subset can be generated from the authorization rules directly (see step 1). However, the security subset may have one or more concealed security criteria which must be identified by analyzing the relationship among the authorization rules .The security criterion subset for the above authorization rule should be { s_1 , s_2 , s_3 } rather than { s_1 , s_3 } , where s_2 represents "not a manager.".

C. Achieving fine-grained access control

In the Criterion-Based Access Control model, an (sub) object's security attributes and security level are implied by indicating users who do not have access rather than explicitly defining them. The system becomes simpler because one mechanism is used to define both the user's security attributes and the (sub) object's security attributes.

The security criterion expressions embedded in a secure object can be regarded as locks, while the security criteria in the security criterion subset can be considered as keys. When a secure user accesses a secure object, he/she uses the available keys to actuate the locks. Whether the secure user is allowed to access the secure sub object (the cell, column, or row) depends on the state of the corresponding locks.

A security criterion expression is evaluated in the following two steps. First, substitute all the security criteria in the security criterion expression with true, T, or false, F, according to the following rules: all the security criteria in a security criterion expression that also appear in the secure user's security criterion subset

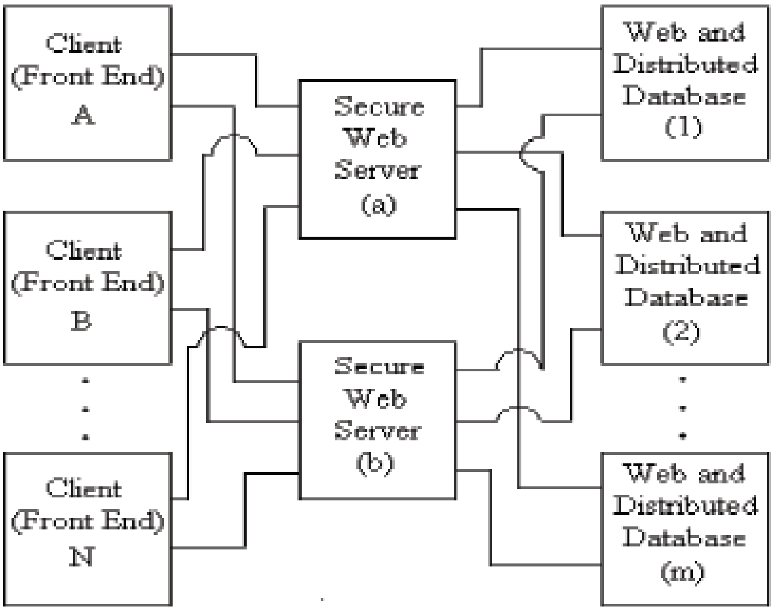


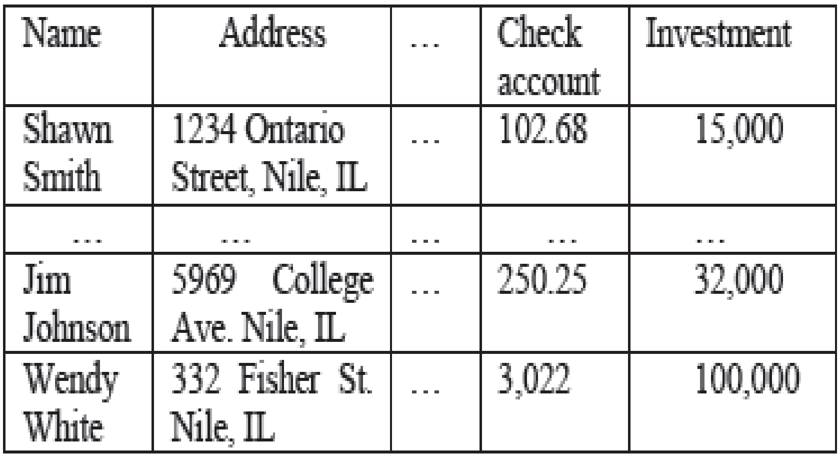
Figure 1. Logical structure of the unified network and database security system

have the value true, T, while other security criteria have the value false, F. Second, the security criterion expression is evaluated according to the normal evaluation procedure in Boolean algebra. The evaluation value T of a security criterion expression implies that users with security attributes specified by these security criteria are not allowed to access the corresponding secure sub object, according to the sub object's security criterion expression transformed from the authorization rules. On the contrary, a false evaluation value, F, of the security criterion expression implies that the security criterion expression (actually the authorization rules) does not prevent these secure users from accessing this sub object.

The fine-grained access is achieved as following. If the evaluation value of a security criterion expression in a column (row) is true, T, the column (row) is not accessible to the user. Therefore, the column (row) is filtered and not be sent to the user. To improve the efficiency of the system, we usually first evaluate three special security criterion expressions stored in the cell of first row and the last column . If the evaluation value of the first one is false F, the whole table is accessible, and the rest of the security criterion expressions need not be evaluated. This is because the first security criterion expression is the logical "OR" of all the security criterion expressions in different columns and rows. When the evaluation value of this security criterion expression is false F, the evaluation value for every of its term must also be false F. On the other hand, if the evaluation value of the first one is true T, there must be at least one security criterion expression in a column or a row whose evaluation value is true T. therefore, the rest of the security criterion expressions should be evaluated. We can evaluate the second security criterion expression to see if any column is inaccessible. If the evaluation value of this one is true T, we need to evaluate the security criterion expressions in different columns one by one to find out all the columns that are inaccessible. In the same way, we can find out the inaccessible rows.

Example 2: Suppose a junior bank teller requests to access the table 1. As discussed above, the user's security criterion subset is { s_1 , s_2 , s_3 }.The rows and columns with security criterion expressions of constant false F are by default accessible and need not be evaluated. When the security criterion subset is used to evaluate the other security criterion expressions, the evaluation values are true T. Therefore, the corresponding column (mortgage) and rows (records of William Wilson and Hannah Howard) are inaccessible. By filtering these inaccessible sub objects, the user gets the following table 2.

TABLE II ACCESSIBLE COLUMNS AND ROWS FOR A BANK TELLER



Name	Address	…	Check account	Investment
Shawn Smith	1234 Ontario Street, Nile, IL	…	102.68	15,000
…	…	…	…	…
Jim Johnson	5969 College Ave. Nile, IL	…	250.25	32,000
Wendy White	332 Fisher St. Nile, IL	…	3,022	100,000

4. A Unified Network and Fine-Grained Web and Distributed Database Security Model

The proposed model includes three tires: client tire, secure Web server tire, and Web and distributed database tire. The client tire is the front end of the system on which client software such as browser and c applications are run. Remote users scattered in different locations send their requests from the client tire. The Web and distributed database tire is on the other end. All of the databases have been upgraded by inserting a row and a column and embedding security criterion expressions derived from the authorization rules (security policy). The secure Web server tire sits between the client tire and the Web and distributed database tire. The secure Web server tire provides the function of

Web server and security services. Figure 1 shows the logical structure of the model. The redundant connections between the tires increase the ability of fault tolerance and resisting the denial-of-service attack.

Discussing the Web server function is out of the scope of this paper. The following discussion focuses on the security service function of the secure Web server. When the user's request is received, the first job a secure Web server does is to authenticate the user. The authentication is performed based on multiple factors including user name, password, and digital credential. Once the user is authenticated successfully, the user's request, IP address, and digital credential are forward to the preliminary access control system. The preliminary access control can be achieved with an upgraded firewall system, which filters the traffic based on the user's request, IP address, and digital credential. For example, although a bank teller (possesses the credential of bank teller) has access to the customer saving information in the local databases, he/she does not have the permission to access the similar information located in the remote databases.

This indicates that user's permissions to a database vary when he/she is in different places. The preliminary access control is a valid way to improve the system efficiency because the user's disallowed requests are terminated at the early stage. If the user's request is allowed by the preliminary access control, the user's request is forwarded to the databases. Meanwhile, a component in the secure Web server generates the security criterion subset for the user based on his/ her IP address and digital credential (as well as other factors required by the security policy).

The component is actually a well-designed program which implements the function of efficient abstracting the security criteria from the authorization rules. When the result of the request comes back to the secure Web server, based on the security criterion subset and the embedded security criterion expressions, the fine-grained access control is achieved by the secure Web server as described in section 3.

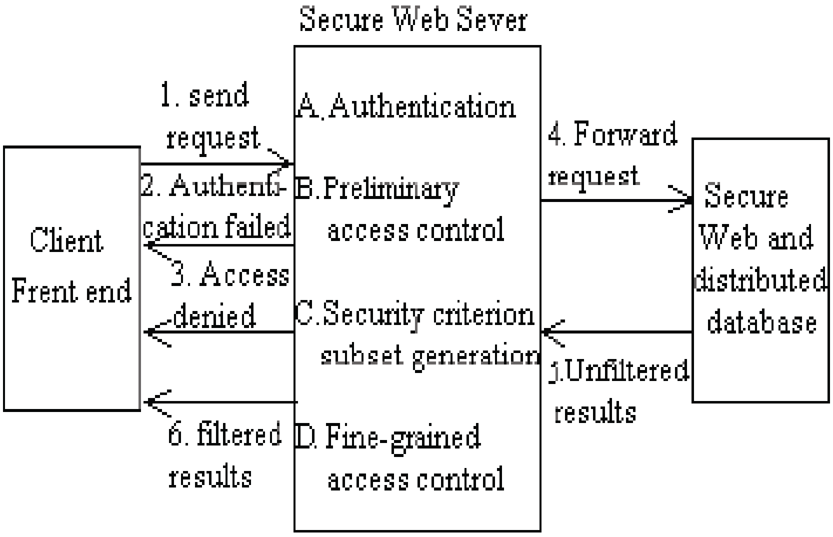


Figure 2. Working procedure of the proposed security system

Figure 2 summarizes the work of the proposed system.

Note: 1. Abstracting security criteria in secure Web server prevents users from modifying their security criterion subsets.

2. The authorization rules (security policy) may be different for different Web and distributed databases. It is convenient to implement flexible fine-grained access control for each database based on different authorization rules.

5. Conclusions

Addressing network security and database security simultaneously leads to efficient unified security system. The information used for authentication can be reused for the preliminary access control and fine-grained access control. The termination of the users' requests at the early stage avoids to unnecessarily process the requests further. The proposed model can be applied to many areas such as finance, health care, government, and military.

References

[1] Ponemon Institute LLC, "Database Security 2007: Threats and Priorities within IT Database Infrastructure", June 4, 2007,http://www.appsecinc.com/techdocs/whitepapers/2007-Ponemon- Database-Security-Study-Sponsored-by-Application-Security-Inc.pdf

[2] L.M. Null and J. Wong "A unified approach for multilevel database security based on inference engines" ACM SIGCSE Bulletin, Volume 21 , Issue 1 (February 1989), Pages: 108 – 111

[3] D. E. Denning, Cryptographic Checksums for Multilevel Database Security, 1984 IEEE Symposium on Security and Privacy P. 52..

[4] R. Schell and M. Heckman, "Views for Multilevel Database Security" 1986 IEEE Symposium on Security and Privacy p. 156

[5] E. Fernández-Medina and M. Piattini "A Methodology for Multilevel Database Design" http://ftp.informatik.rwthaachen.de/Publications /CEUR-WS/Vol 74/files/FORUM_09.pdf

[6] E. Bertino, B. Catania and E. Ferrari, "A nested transaction model for multilevel secure database management", ACM Transactions on Information and System Security (TISSEC), Volume 4 , Issue 4 (November 2001), Pages: 321 – 370

[7] W. Rjaib, **"**An introduction to multilevel secure relational database management systems", Proceedings of the 2004 conference of the Centre for Advanced Studies on Collaborative research, Markham, Ontario, Canada, Pages: 232 – 241.

[8] W. Itani, A. Kayssi and A. Chehab, **"**An enterprise policy-based security protocol for protecting relational database network objects" Proceedings of the 2006 international conference on Wireless communications and mobile computing, Vancouver, British Columbia, Canada, SESSION: T1-B: computer and network security symposium, Pages: 343 – 348.

[9] T. Didriksen, "Rule based database access control—a practical approach" Proceedings of the second ACM workshop on Role-based access control, Fairfax, Virginia, United States, Pages: 143 – 151.

[10] Charles P. Pfleeger and Shari Lawrence Pfleeger. *Security in Computing*, Fourth Edition. Prentice Hall, 2007.

[11] Pan, L., Zhang, C.N., and Yang, C. (Spring 2006 issue) A Role-Based Multilevel Security Access Control Model. The Journal of Computer Information Systems. Volume XLVI, Number 3, Pages: 1-10.

[12] Pan, L., and Zhang, C. N. (will be published in 2007) A Web-Based Multilayer Access Control Model for Multimedia Applications in MPEG-7. International Journal of Network Security. Vol.4, No.2, Pages: 155–165.

[13] Pan, L., and Zhang, C. N. (September 2006) Achieving Multilayer Access Control for Role-Based, Mandatory, and Discretionary Access control Models. Accepted by Information Systems Security.

[14] Pan, L., and Zhang, C. N. (2006) A Criterion-Based Role-Based Multilayer Access Control model for multimedia Applications. IEEE International Symposium on Multimedia (ISM2006). San Diego, Pages: 145-152..

[15] Pan, L., and Zhang, C. N, "A Criterion-Based Multilayer Access Control Approach for Multimedia Applications and the Implementation Considerations". Accepted by ACM Transactions on Multimedia Computing, Communications and Applications (ACM TOMCCAP), March, 2008.

[16] Pan, L, Using Criterion-Based Access Control for Multilevel Database Security. Proceedings of The International Symposium on Electronic Commerce and Security, Guangzhou, China, August 3-5, 2008, 518-522.

[17] Brands S (1999) A technical overview of digital credentials, http://www.xs4all.nl/#brands/

[18] M.R. Neilforoshan Network security architecture, Journal of Computing Sciences in Colleges, Volume 19 , Issue 4 (April 2004), Pages: 307–313

[19] D. Davis, R. Swick, Network security via private-key certificates, ACM SIGOPS Operating Systems Review Volume 24 , Issue 4 (October 1990) Pages: 64 – 67.

[20] K. Sadasivam, B. Samudrala, T. A. Yang, Design of network security projects using honeypots, Journal of Computing Sciences in Colleges, Volume 20 , Issue 4 (April 2005), Pages: 282 – 293