Implementation of Universal Authentication Framework in the Electronic Archive Management

JIRI SOBOTKA, ONDREJ RASO, VACLAV ZEMAN
Brno University of Technology
The Faculty of Electrical Engineering and Communication
Department of Telecommunication
Purkynova 118, 612 00 Brno, CZECH REPUBLIC
Email: sobotkaj@feec.vutbr.cz, ondra.raso@phd.feec.vutbr.cz, zeman@feec.vutbr.cz

Abstract: The paper describes requirements for electronic archive secure access and management. In each section the credible archive, access options and universal authentication framework are described. For the user authentication, a portable hardware device stored certificates are used. The aim of this article is to propose a universal authentication framework as a secure way of an archive management.

Key-Words: Authentication, Authentication framework, Electronic archive, Archive management.

1 Introduction

Digital electronic systems have many advantages compared to the classical paper based ones. Complete listing of them is out of the subject of this article. All of them are summarized in the term “paperless office”[1]. The electronic documents, as well as paper documents, have to be archived. Electronic archives again in comparison with the classical have many advantages. One of the best one is easy copy making and thus backup performing. Also the archive access is much easier and available. Last but not least the price of electronic archive is much lower. On the contrary, electronic archives have also some disadvantages; the most important is ensuring the credibility of the archive. A credible electronic archive is considered an archive, containing legally indisputable electronic documents [1]. Credible electronic archive must be able to satisfy following requirements [2]:

1.1. Document content integrity – the document has to be provably protected from any alteration. That can be accomplished by cryptographic algorithms, for example hash.

1.2. Document content authenticity – origin of the document must be indisputable. Authenticity of the document can be accomplished by digital signature, which is however dependable on its certificate validity. Therefore electronic signature with additional authentication information is used more.

1.3. Document permanent readability – the document must be readable in the time, when technologies used to its creation will be obsolete. To extend the readability, migration, emulation and encapsulation methods are used.

1.4. Time identification – document date of creation and date of the insertion to the archive must be identical with information in metadata or in the document itself. That is accomplished by time stamp which works with similar principle as electronic signature.

1.5. User rights and credibility – the document must be accessible only to authorized persons. Document confidentiality in electronic archive is achieved by encryption by some cryptographic algorithm.

Successful accomplishment of those criteria is more or less dependable on access control and security. Simplest and therefore most used way of archive access control is by password. The user enters his identification data while accessing the archive, and then proves knowledge of some confidential information – enters the password. By this process the user is authenticated.

The technology of multifactor authentication is only solving the user identity verification problem. Other obstacles to secure archive management, as access control, user rights, and content access and author rights have to be solved by other technologies. For the purposes of integration in other systems and for user friendly applications, a universal access system is required. This system is proposed in following article with implementation of universal authentication framework [4].
2 Universal Authentication Framework and ACP protocol

ACP protocol works with the principle of communication between Initiator and Addressee [4]. Initiator is the part initiating the communication, the archive access applicant. Addressee is in this system authority permitting the archive access. If the addressee does not know the initiator, therefore is not able to verify his identity, he can ask another authority to authorize this applicant. In this case new transaction is opened, where the addressee is the initiator and another authority is the addressee. All parts involved in the process of user authentication have to work with universal authentication framework. After the enrollment in to the archive, the user can use his authentication data created in different AAA system. The ACP protocol format is purposely similar to EAP protocol and is displayed in the Figure 1.

<table>
<thead>
<tr>
<th>Code</th>
<th>Identifier</th>
<th>Length</th>
<th>AVP₁</th>
<th>...</th>
<th>AVPₙ</th>
</tr>
</thead>
</table>

Fig.1: ACP protocol messages format

Description of the message field of the ACP protocol [4]:

a) Code: This field distinguishes message of ACP protocol and EAP protocol. The differentiation is necessary in the environment where both protocols are used simultaneously. Binary format of the field is \(x₇ \cdot x₆ \cdot x₅ \cdot x₄ \cdot x₃ \cdot x₂ \cdot x₁ \cdot x₀\), where:
   a. \(x₇ = 1\): indicates message of ACP protocol,
   b. \(x₆ \cdot x₅ \cdot x₄ \cdot x₃ = 0000\): meantime no purpose,
   c. \(x₂ \cdot x₁ \cdot x₀\): defines message type (see next)

b) Identifier (3 B): This field is the transaction identifier in the line. Each transaction has its unique identifier, established by the Initiator. The identifier of the transaction is transferred only in first message of the transaction to lower the network burden.

c) Length (3 B): This field presents total length of the message in bytes. Maximum length of the message is \(2^{24} - 1\) bit = 2MB.

d) AVP (Attribute-Value Pair): variable with its value in the ACP protocol format. The AVP field has format presented in the Figure 2:

<table>
<thead>
<tr>
<th>Type</th>
<th>Length</th>
<th>Value</th>
</tr>
</thead>
</table>

Fig. 2: AVP field format

AVP fields description:
a) Type (1B): this field identifies the AVP field
b) Length (2 B): this field describes length of Value field in bytes.
c) Value (0 to \(2^{16}\) -1B): this field describes value of the corresponding AVP type. Size of this field is set for transport of the pictures (in kB), usage of modern cryptographic algorithms (signature with 2048 bites) and transport of whole EAP messages.

The body of each message can transport more fields of AVP type. The AVP types for each message are optional. Detailed protocol implementation is described in chapter 4.

3 Hardware storage devices and certificates

In the proposed system, the user will be authenticated by two factor authentication [3]. In two factor authentication, the user is proving his identity by two means – factors: by proving knowledge of some information (password), and by proving a possession of some item (token). By adding second factor of authentication by token, represented by certificate, strengthened user access is ensured. One of the crucial part of the secure authentication system for archive access are certificates. The certificate is a data file in exactly defined form, signed by certificate authority. The certificate has two parts. First part contains information like certificate version, serial number, signature algorithm, certificate issuer, validity, public key and add-ons [5]. Second part represents information about signature algorithm and certificate authority signature, confirming the stored data. The certificate data structure is liable to particular standards. New certificates are issued under the standard X. 509 version 3 (X.509v3) [6] [7]. This standard is working with ASN.1 (Abstract Syntax Notation One) notation [8]. ASN.1 describes the data structures in user readable format. That means the way of data transfer from binary form used by computers to communicate.

The other way, from binary form into text form is described by BER standard (Basic Encoding Rules) [9], and its specification DER (Distinguished Encoding Rules) [9]. Base62 [10] is determining rules for transfer between data forms (ASCII and Binary alphabet, etc.). PEM standard (Privacy Enhanced Mail) [11] is used together with certificates. PEM is older recom-
mendation of improved email security. Some of the certificates are issued by PKCS standard (Public Key Cryptography Standards), which is specified by RSA laboratories. The most used are standards PKCS #7 (Cryptographic Message Syntax Standard) [14] and PKCS #12 (Personal Information Exchange) The data structure of the certificate is displayed in the Fig. 3. Horizontal direction represents conversion between data formats and vertical direction represents composition of data formats. The certificates are used in infrastructure based on asymmetric cryptography, according to the PKI standard. The basic principle of asymmetric cryptography is in use of two different keys: private key and public key, while is not possible to use one key to calculate the other key. Most used asymmetric algorithms are RSA (Rivest, Shamir, Adleman) and D-H (Diffe-Hellman). The RSA algorithm is used for cryptographic operations like encrypting and electronic signature. With usage of certificates, the RSA is used for key exchange between involved parties. The certificate serves as storage medium for user authentication data. Those data are sent to the addressee during authentication.

4 Secure archive access

Realization of proposed security algorithms is performed in two different steps. First, the secure connection between communicating parties is established. Then login form is presented to the user. To improve the security, a hardware cryptographic token is used. The tokens, as is mentioned above, store important information used for user identification and connection establishment. For mutual cooperation between services providing security mechanisms, the correspondent infrastructure has to be developed. The infrastructure then serves as medium for all services involved. Proposed infrastructure is based on system of management and distribution of public keys (PKI – Public Key Infrastructure) as depicted in the Figure 4. Each block represents installed and set up applications from Open Source Software environment. Fig. 4 consists of four quadrants, separated by horizontal and vertical axes. Horizontal axe represents layers according to the ISO/OSI reference model. Vertical axe represents the model client-server. The client part in our case is the user computer. In this model, two communicating parties are used (client and server), but in practical live, more users are connected to the server.

Fig. 3: Data structure of the certificate

Fig. 4: Block scheme of security architecture

Most of the used applications are operation on application layer. An exemption is support for hardware cryptographic modules. The support is provided by OpenSC project, which creates so called interlayer for connection of the modules and end applications. Established secure connection between client and server is using application protocol HTTPS (Hypertext Transfer Protocol Secure), but its foundation is TLS/SSL (Transport Layer Security/Secure Socket Layer). This protocol is operating on lower layers of ISO/OSI reference model.

Used application environment:

The applications used for creation of secure infrastructure are usually fully or partly from Open Source project. The list of used application on client and server side is following:

a) Client
- Web browser – any kind of web browser with a support digital certificates and hardware cryptographic modules (tokens, chip cards, etc).
- OpenSC – operating system extension of hardware cryptographic modules

b) Server
- Alfresco – a system for content management employed as an electronic archive
- Apache Tomcat – creates web interface for Alfresco
- OpenSSL – represents certificate authority, which is issuing certificates for secure HTTPS connection in mutual authentication
- **OpenLDAP** – is an address structure and access protocol used by Alfresco for user account storage.

For cooperation with other applications, each of used application requires specific setup in the scope of secure infrastructure. Following description is divided in client part and server part. Server part is about installation and application setup.

### a) Client part
For ordinary use of electronic archive, the client needs to install a standard web browser with certificate support. Improved security is achieved by mutual and two factor authentication. The protected assets are the user certificate and his private key. These items have to be stored in secure password protected data storage. Secure data storage can be in software and hardware form. To improve the level of the security, the hardware cryptographic tokens are used as a tool of two factor authentication. The tokens are used as a device for secure storage of the user certificates. Relevant private keys are for the purposes of mutual authentication. The support of the tokens in the operating system (MS Windows or GNU/Linux) is realized by OpenSC project. After installation of necessary parts from OpenSC project, the tokens can by utilized. The OpenSC project is primarily developed for GNU/Linux operating system. Simultaneously it is imported into other operating systems. OpenSC is providing complex support in the GNU/Linux system.

### b) Server part
The server part is a complex connection of mentioned applications. First software device, the Alfresco system is installed together with Apache Tomcat. If installing the Alfresco separately, at first the Apache Tomcat has to be installed. Then OpenSSL and OpenLDAT projects are installed. The Alfresco represents a system for content management and runs on services of other applications. Access to the Alfresco by web interface is provided by Apache Tomcat. For establishment of secure connection, the Apache Tomcat is using certificates issued by the certification authority. The certification authority is created by OpenSSL project. During establishment of secure connection, the mutual two factor authentication is used. The authentication is working with certificates as private keys (server key and client key). Goal of the mutual authentication is to authenticate all objects involved in communication, in our case server and client (user). For the reasons of two factor authentication, hardware cryptographic tokens are used. The client certificates and corresponding private keys are stored within these tokens. An OpenLDAP address structure and access protocol is used to store files from user accounts (user name, password, email address, real name and surname, etc.). The files are synchronized between Alfresco and OpenLDAP.

### Implementation of hardware cryptographic tokens
As mentioned above, the certificates (server and client) are used as medium for authentication. Client certificates and corresponding private keys have to be safely protected. For that reason, the hardware cryptographic modules were chosen. The tokens iKey 3000 from SafeNet Inc. Company were chosen for their compatibility with different operating systems. Also usability on multiple platforms was considered. The advantage of these tokens is support of RSA standard of PKCS #15 (*Public-Key Cryptography Standards: Cryptographic Token Information Format Standard*). This standard allows usage of alternative cryptographic tools because of its platform independence. The tokens are just a one part of the group of hardware cryptographic modules. Other part of that group are smart cards. The hardware cryptographic modules utilization can be divided into two basic parts:

#### a) Operations performed by hardware cryptographic modules manager
#### b) User operations

If those two parts are connected together, as shown in the Figure 5, the hardware cryptographic modules life cycle is established. The operations of life cycle can be performed practically indefinitely until the tokens are mechanically damaged.

![Fig. 5: Life cycle of hardware cryptographic token operations](image-url)
The Fig. 5 is describing main operations performed by the hardware cryptographic tokens. Detailed description is in following text.

a) Operations performed by hardware cryptographic modules manager

The hardware cryptographic modules are firstly formatted and initialized by its manager. The reason for this procedure is that the modules can be initialized by manufacturer for other cryptographic tools or can contain old data. While reusing the modules, usually no further formatting is necessary. Only actualization of certain data can by sufficient. The final process of enrollment of new token is dependent on security rules of the organization. Next step is creation of a user profile. Then all necessary data of the user (client certificate and private key) are stored in the memory of the cryptographic module. In last step, the module is delivered together with preset access PIN to the user.

a) User operations

Before first use of the hardware cryptographic module, the user performs system and application setup (drivers and application support installation, application setup for authentication process based on use of modules). Then the only user interaction with module during authentication is to insert token access PIN. By this, the client certificate is presented to the server. If the content of the module is obsolete (the certificate is expired), or the data are not reliable (the key has been compromised), the user returns the module to the system manager. The manager repeats the enrolment process with actualized credentials.

Description of secure communication

The connection establishment progression, included Alfresco log in, is presented in Figure 6. The picture is divided into two mutually communicating parts – the client and the server.

The main steps necessary for connection establishment are displayed. In the Figure 6, different forms of authentication are distinguished by color. The blocks displaying mutual authentication are red frames. Two factor authentication is highlighted by blue text. Blocks with green background depicts authentication by TLS/SSL. And blocks with grey background are used for authentication by Alfresco login form. The Figure 6 is presenting sequential authentication scheme. The detailed description of whole process is in following text.

Fig. 6: Secure connection establishment with authentication methods distinguishing
transfers the necessary data of the user account to the Alfresco. Alfresco compares correspondence of user name and password with inserted data. If both values are the same, the user has granted access and can start working in the archive. The user account data are also synchronized.

5 Security evaluation of the proposal

Implementation of the proposal is done in two levels. First, the safe HTTPS connection is established by mutual and two factor authentication. The tools for mutual authentication are certificates with corresponding private keys. Two factor authentication is performed by hardware cryptographic tokens, which are used as a secure storage of client certificates and private keys. The login process is performed by login form. The user inserts his credentials – login name and password. These data are stored in OpenLDAP and Alfresco for the purposes of authentication. Alfresco is comparing credentials stored in OpenLDAP with credentials entered in login form. Used authentication scheme increases the security and decreases the severity of potential direct attack at Alfresco. Thus the attacks at the login form, which is the most frequent target, are adverted. The implementation is built on the foundations of Open Source Software. An advantage is independence on used platform and used operational system at the client side. The necessary equipment in client computer is standard web browser with certificates support. The client certificate and private key are stored in safe data storage. The token support in operational systems (MS Windows XP and GNU/Linux) is provided by OpenSC.

6 Conclusion

Presented proposal is significantly increasing security of archive access and thus increasing the security of the archived information. Implementation of the universal authentication framework ensures ease of system deployment, because the users can use their existing electronic credentials. Also implementation of Open Source software provides low starting costs. Only one significant investment is purchase of hardware cryptographic modules iKey 3000.

7 Acknowledgements

This work has been supported by the grant OPVK no CZ.1.07/2.2.00/28.0062, joint activities of BUT and TUO while creating the content of accredited technical courses in ICT.

References: