Annales UMCS Informatica AI 3 (2005) 355-364

Annales UMCS
Informatica
Lublin-Polonia
Sectio AI

http://www.annales.umcs.lublin.pl/

Implementation of certification subprotocol for electronic auction

Bogdan Księżopolski*, Adam Dziurda

Faculty of Mathematics, Physics and Computer Science, M. Curie-Skłodowska University, pl. M. Curie-Skłodowskiej 1, 20-031 Lublin, Poland

Abstract

One of the main components of public key infrastructure (PKI) is the center of certification. The function which it can fulfill, depends on the specific protocol. In the article the center of certification is the trustworthy third part, which includes the center of authorization (CA) and time stamping authority (TSA).

The center of certification is the realization of certification subprotocol, the element of main electronic auction protocol [1]. In the article, we present the implementation and efficiency characteristics of the center of certification which is based on the free licence (GNU licence) library openSSL [2] and the patch to it openTSA [3].

1. Introduction

Nowadays, the institutions of public services change the classical form of information exchange to the electronic one. The benefits of using the electronic way are significant, thanks to which one can increase their availability, efficiency and reliability decreasing simultaneously the expenses of functioning. Currently, the research is conducted on different parts of public institutions [4], the projects are created which support that initiative (eTen, IDAII). Some of them are now partially implemented, as an example one can point to ZUS.

The important problem, about which one cannot forgot, seems to be the protection of information which the systems of public institutions have access to. Security of these systems is a very complicated issue [5], including many elements. Among them there are problems of security network architecture, protection mechanisms of network infrastructure, cryptographic protocols, defined staff duties etc. Nowadays different architecture models are created [6] which guarantee the first defined security level.

In the article we present implementation and efficiency tests of certification subprotocol, the element which realizes electronic auction [1]. The main element of the mentioned subprotocol is the center of certification. This center includes

*Corresponding author: *e-mail address*: bogdan@kft.umcs.lublin.pl

the authorization center (CA) and time stamping authority (TSA). The additional assumption to the project was using only "open source" software.

2. Cryptographic protocol

The mentioned e-auction consists of four subprotocols: *certification, notification of auction, notification of offer as well as choice of offer*. In protocol *N* bidders (O_1, ... ,O_N), third trustworthy person that is *MAA* (main auction agency) as well as firm, which wants to announce the auction take part.

The first step of protocol is verification by MAA of the participants taking part in e-auction that is the bidders O_N as well as the firm F which wants to announce the auction (the *subprotocol of certification*). The next step is notification of the auction to MAA by the verified firm F. MAA publishes the conditions of notified auction, giving all requirements notified by F (*the subprotocol of notification of auction*). In the next step, a person wanting to take part in the auction, after earlier verification, sends his offer to MAA (*the subprotocol of notification of offer*). The last subprotocol is executed after time elapsing for notification of offers, then firm F as well as bidders O_N, send their parts of secret (needed to read offers) to MAA. After their decoding, they will be sent to firm F, where the winning offer will be chosen. In the same subprotocol, the firm F sends information about the winning offer to MAA, and then it will be published to (be generally known) public message (*the subprotocol of choice of offer*).

The communication between participants of protocol is safe. We achieve it thanks to using the public key cryptography, where every participant of protocol possesses his private key (SK) as well as the public key (PK). Those practical keys are not solid; their validity ends with the validity of registration number, which is achieved in the subprotocol of certification.

Offers sent by O_N bidders are coded by the public key of given auction. We can read them on account that we possess the private key, which in subprotocol of notification of auction, becomes divided into parts with the help of the suitable safe threshold scheme of division of secret. In protocol we also use the random numbers generator (KG) to create the identification number of participants of auction as well as the numbers of auctions.

Auction ends after definite time, to determine this moment, the time stamp (T) is used.

3. Certification subprotocol

The article presents implementation of certification subprotocol which to a large extept is responsible for registration and assigning the privileges for the parties taking part in e-auction.

Step 1

The possibility of participation in e-auction has to be preceded by obtaining suitable authorization. A person who applies the for certificate, that is a firm wanting to announce auction or bidder, should possess appropriate documents D_c as well as the private key $SK_{C_{CA}}$ obtained from one of the centres of authorization (CA) chosen earlier. He signs the above mentioned digitally documents by using $SK_{C_{CA}}$ and then codes using the public key PK_{GAP} and later sends it to MAA.

Step 2

MAA decodes documents and verifies them. After positive verification, generates them by means of the generator of random numbers (KG), unique registration number for a given person NR_C. The registration number is valid for a definite time, so that the time stamp of registration number T_{NR_C} is generated. MAA generates also the private key (SK_C) and the public key (PK_C) for a given subject, which will be used in the next subprotocol. Validity of these keys ends when the time shown by T_{NR_C}.

Step 3

It signs the data generated digitally and codes by means of the public key PK_C and then sends them back to C. The C part decrypts the received data and saves them in the appropriate place in the system.

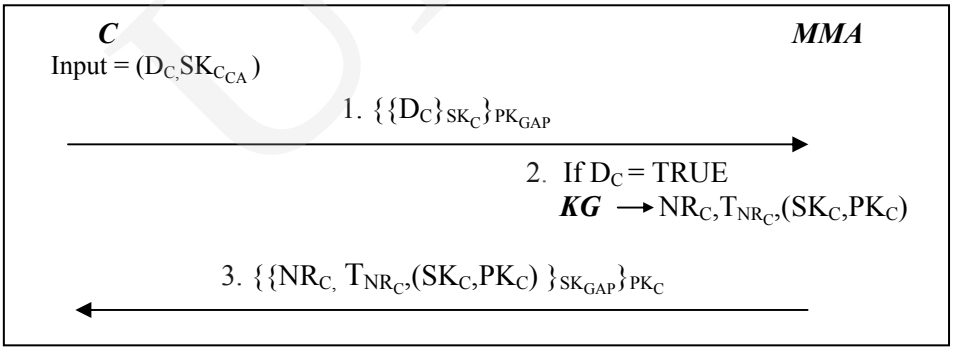

Fig. 1. The graph of certification subprotocol

4. Implementation

The presented graph (Fig.1) describes cryptographic modules used in the subprotocol and communication steps needed for realization of main protocol. The crucial element of the subprotocol is Main Auction Agency (MAA) which is the center of certification and includes the center of authorization (CA) and time stamping authority (TSA). The center of certification will be the main element implemented in the described subprotocol.

The second implemented element is client party thanks to it the parties wanting to take part in e-auction must obtain an appropriate privilege.

The mentioned elements, the center of certification and the client party, were designed as separate modules. The access to them is possible by the user interface (Fig.2) from which the users can execute any actions defined in the subprotocol.

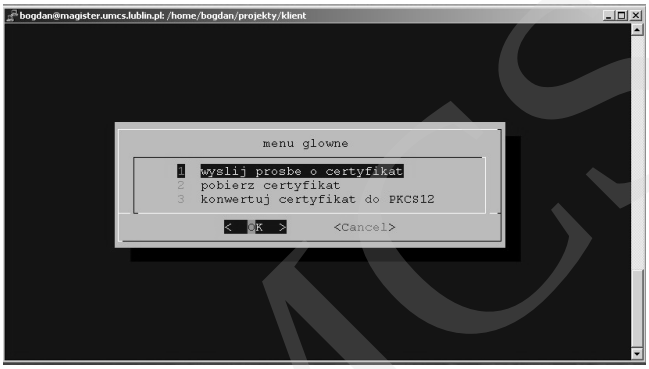

Fig. 2. The user interface to client party module

4.1. Environment

In the implemented process of described subprotocol the assumption was to use programming environment and software which are based on open licence. Cryptographic mechanisms used in the certification subprotocol were mainly realized by openSSL [2] library. The additional action which must be taken to implement time stamping authority was patching the openSSL library by openTSA [3] packet.

The created software was meant for the system Linux, in that environment it was implemented into two mentioned modules, the center of certification and the client side. Software was implemented in the linux operating system layer, called bash. The data are exchanged between the parts of certification subprotocol by means of electronic mail, in the project we used Postfix [7] mail system.

1.2. Setting up the center of certification

The main element of certification subprotocol is the center of certification (MAA). Setting up the mentioned center is connected with fulfilling many security requirements [8], which depend on specific electronic process and can be adjusted to different security levels. The security levels are described by appropriate norms and international standards (ISO,ETSI) [9,10].

The authorization center and time stamping authority as a center of certification fulfill the role of departments. These departments must be defined by formal and legal sense, to this end before taking an initializing action; one can specify formal details for them. In the described case, among other things, we discussed following details:

```
countryName                = PL
localityName               = Lublin
organizationName           = UMCS
organizationalUnitName     = KFT
commonName                 = GAP Certificate Autority
emailAddress               = cc@kft.umcs.lublin.pl
```

The presented data are basic information, obviously there is possibility to define more details for departments [2,3].

Descrbed departments are electronic departments, that is why apart from formal information, the technical parameters must be defined. It is necessary to define location in the system for the department confidential data, determine details of used cryptographic modules, define characteristics of the department. The selection of the mentioned parameters depends on a specific project. The example information for certification subprotocol is presented below:

```
dir                = CA                # Where everything is kept
certs              = $dir/certs        # Where the issued certs are kept
certificate        = $dir/cert.pem     # The CA certificate
private_key        = $dir/priv/privkey.pem # The private key

default_days       = 365               # how long to certify for
default_crl_days   = 30                # how long before next CRL
default_md         = md5               # which md to use.
default_bits       = 1024              # defaults bits of keys
default_keyfile    = privkey.pem       # the name of keyfile

crypto_device      = builtin           # OpenSSL engine to use for signing
signer_key         = $dir/tsakey.pem   # The TSA private key
clock_precision_digits = 0             # number of digits after dot.
```

The detailed description of all possible parameters which can characterize the center of certification is very extensive, the complete specification is described in the openSSL library documentation [3,4].

When the certification center characterizing parameters is defined, one can generate a pair of cryptographic keys, public and private and appropriate certificate for center of certification. To this end, we choose the RSA cryptosystem [11] which is used in the whole project. The mentioned action is executed by means of command shown below:

```
openssl req -config cc.config -new -x509 -out cc.cert[1]
```

[1] *"req"* – command of req group from openSSL library
"-config cc.config" – the center of certification configuration file
"-new x509" – generation of pair of cryptographic keys in x.509 standard
"-out cc.cert" – the name of the certification center certificate

4.3. Step 1

When a center of certification is created, realization of individual stages of described certification subprotocol can be started. In the first step by means of appropriate interface the client gives the documents needed to obtain the rights of getting certificate (D_C). The kinds of needed documents depend on requirements for a given electronic auction; it could be, for example, digitally signed certification of advance pay or certification of firm financial fluency. Among these data, there are also the documents which will be used for generating a new certificate.

In the next stage, the client downloads the public key of certification center (MMA) with appropriate certificate. In the described subprotocol the mentioned data are downloaded by means of the following command:

wget hostname/~cc/cert.pem -O cacert.pem[2]

In the next stage in step one, in order to realise the subprotocol assumptions the cryptographic mechanisms are used. Documents D_C are digitally signed by the private key SK_{C_CA} obtained before in the appropriate certification center:

openssl smime -sign -in dokumenty -out dokumenty_podpis –signer c_ca_cert.pem -inkey c_ca_privkey.pem[3]

In the next stage, the digitally signed data are encrypted by the public key of certification center PK_{CC}:

openssl smime -encrypt -in dokumenty_podpis -out szyfrogram cacert.pem[4]

In the last stage of that step, the data described above are sent to the certification center for verification. The information will be sent to the certification center by means of electronic mail. Thus the mail server called "postfix" [7] was used and the access to it was obtained by means of mail

---

[2] *"wget"* – the application which is used to downloading the data from remote network servers [14]
„hostname/~cc/cert.pem" – the name of network server with file localisation on remote server
"-0" – the depth of downloading data
"cacert.pem" – the name of downloaded data from remote server
[3] *"smime"* – the smime group call from openssl library
"-sign" – the data singing directive
"-in dokumenty" – input data defining
"-out dokumenty_podpis" – output data defining
"-signer c_ca_cert.pem" – localisation of the appropriate certificate for authorization center (singing)
"-inkey c_ca_privkey.pem" – localisation of the private key for the singing data side
[4] *"-encrypt"* – data encrypting directive

application called "pine"[12]. Thanks to using the mail application called "pine" the data are sent automatically and it is possible because the access to the pine can be gained by means of linux system layer called "bash".

4.4. Step 2

In the second step of the mentioned subprotocol, the centre of certification received encrypted data which were sent by the part applying for appropriate certificate. In the next action, the center of certification is decrypted by the received data by using its private key SK_{CC}:

openssl smime -decrypt -in szyfrogram -out dokumenty_podpis -recip cc_cert.pem -inkey cc_privkey.pem[5]

When the data were decrypted, then the digital signature of sender is verified which checks the authenticity of the received data:

openssl smime -verify -in dokumenty_podpis -certfile ca_cert.pem -CAfile cacert.pem -out dokumenty[6]

As a result of described operations, the certification centre received documents which were sent by the party applying for certificate. If the documents satisfy the appropriate specific process requirements, then automatically the pair of keys; public, private and suitable certificate are generated:

openssl req -config c.config -new -x509 -out c.cert

In the next stage, the certification centre generates the appropriate registration number NR_C which is referred for the party whose keys were previously generated:

openssl rand -out nr_c.rand -base64 1000[7]

The created registration number NRC is digitally signed, it is made in two stages. Firstly, the request for signing of the appropriate data are created; it is referred to the time stamping authority (TSA). In the described case it is made

[5] *"-decrypt"* – data decrypting directive
"-recipe cc_cert.pem" – localisation of the appropriate certificate for certification center (decrypting)
[6] *"-verify"* – verifying of received digital signature
"-CAfile" – localisation of the CA certificate which generates certificate for singing side
[7] *"rand"* – the rand group call from openssl library
"-base64" – coding the received registration number in base64 standard
"1000" – the length of received characters sequence

easier because the certification centre is simultaneously the time stamping authority. It is made by means of the following command:

openssl ts -query -data nr_c.rand –out nr_c.tsq [8]

If we have the appropriate request, then the certification centre can generate time stamp for the data:

openssl ts -reply -queryfile nr_c.tsq –signer cc_cer.pem -inkey cc_privkey.pem –out nr_c.tsr[9]

4.5. Step 3

During step 2, the verification process of parties taking part in the subprotocol was done and the appropriate data were generated. In step 3, the appropriate data are sent to the party applying for the certificate. Before it takes place, the data will be digitally signed by the certification center and encrypted by the appropriate public key. The commands by means of which the data are digitally signed and encrypted were discussed in step 1.

The digitally signed and encrypted data are sent to the party which applies for a certificate by postfix mail server. The access to the mail server was also obtained by pine mail application.

The parties applied for certificate and received the data from the certification center are decrypted, digitally signed and verified. The commands used for that operation were described in step 2.

In the last stage, the keys, the certificate and registration number are saved in an appropriate place in the system.

5. Tests of efficiency

One of the aims of certification subprotocol implementation is to test efficiency of main mechanisms used in the described subprotocol. As main mechanisms were recognised the actions were taken as in step 2. Step 2 includes: generation of pair of keys; public, private and appropriate certificate, generation of a registration number for C party, time stamping of generated registration number, digital signing and encryption of all generated data.

---

[8] "ts" – the ts group call from openssl library
"-query" – request creating for time stamping directive
"-data nr_c.rand" – input data for time stamping authority
"-out nr_c.tsq" – the name for creating request for time stamping

[9] "-reply" – time stamping directive
"-queryfile nr_c.tsq" – localisation of the data including request for time stamping
"-out nr_c.tsr" – the name for creating time stamp

The basic tests were taken on two different servers; the first has Intel Pentium II processor with 233MHz frequency and 64Mb of RAM memory; the second has the Intel Xeon processor with 2000MHz frequency and 512Mb of RAM memory. The main parameter whose value is important in the described mechanisms is the length of generated keys. In the tests two possibilities were researched; the keys with 1024 bits and twice larger 2048 bits. The load of described mechanism was simulated for 50 successive calls. The results are presented in Figure 2.

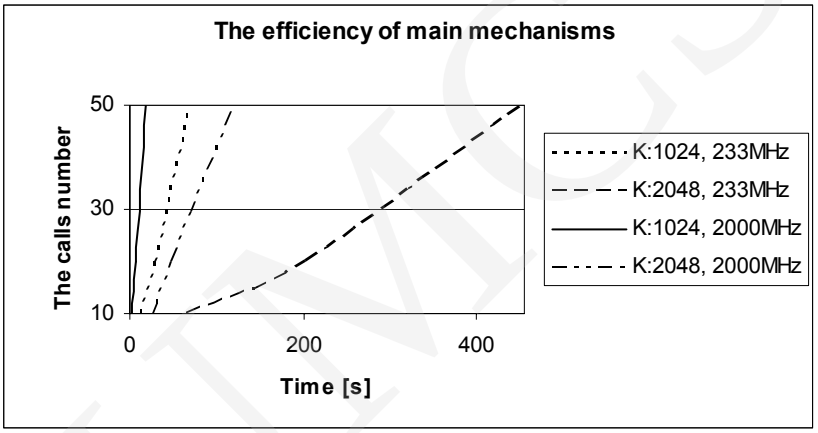

Fig. 2. The efficiency of main mechanisms

The processing time of the main mechanisms for 1024bits keys is steady for both servers. In the case of server with a processor of 233MHz frequency; 50 successive calls of described mechanisms take approximately 80 seconds. In the case of the second stronger server it takes only approximately 29 seconds. When the keys are increased to 2048 bits, then the weakest server loses efficiency and 50 successive calls take more than 400 seconds. The stronger server is still steady; the calls are processed in approximately 100 seconds.

During the analyses of prepared tests one can conclude that the main mechanism is steady for the server with a processor of 2000MHz frequency. When the keys are doubled, mechanisms are still steady. In the case of the weakest server, the mechanisms are steady for 1024 bit keys as well. However, when one increases the length of keys, the weakest server is not steady and cannot be used in practice.

6. Conclusion

In the article the implementation of certification subprotocol and one of the elements of protocol which realizes the electronic auction were described. In the implementation process only open source software was used which was one of the assumptions in the project. The crucial elements in the system are commands

using cryptographic modules. OpenSSL and openTSA library were used which include all functions needed for certification subprotocol implementation. The basic tests were processed and showed the efficiency of the main mechanisms used in the subprotocol. The researched results can be used as comparative data for other applications using cryptographic modules.

We are planning implementation of all electronic auction protocols with the assumptions similar to those in the described subprotocol in the future. Another plan is to implement the part of protocol by using the java language, thanks to which the created software will be available for different platforms. Therefore, it seems to be important to make efficiency tests for that software and compare them with the results described in the article.

References

[1] Księżopolski B., Kotulski Z., *Cryptographic protocol for electronic auctions with extended requirements*, Annales UMCS Informatica, 2 (2004) 391.
[2] Official webpage for OpenSSL Project, http://www.openssl.org/
[3] Official webpage for OpenTSA Project, http://www.opentsa.org/
[4] Czarnecki P., *Seminarium eGovernment w Polsce – teraźniejszość i przyszłość*, in Polish.
[5] ISO/IEC 17799, Information Technology – Code of practice for information security management, (2000).
[6] Hulsebosch B., Massar O., Godvolk E.J., Pont P.M., Postuma P., Mahabier A., T4D3: *Security Architectures Virtual Heaven*, (2001).
[7] Official webpage of Postfix Project, http://www.postfix.org/
[8] Księżopolski B., Kotulski Z., *Bezpieczeństwo e-urzędu – Centrum Certyfikacji* w: Kwiecień A., Gaj P., *Współczesne Problemy Systemów Czasu Rzeczywistego*, Wydawnictwa Naukowo-Techniczne, Warszawa, ISBN 83-204-3023-2, (2004) 349.
[9] ETSI TS 102 023: *Policy requirements for time-stamping authorities*, (2003).
[10] ETSI TS 102 042: *Policy requirements for certification authorities issuing public key certificates*, (2002).
[11] ISO/IEC 19790: *Security techniques – Security requirements for cryptographic modules*, (2003).
[12] Rivest R.L., Shamir A., Adelman L.M., *A method for obtaining digital signatures and public-key cryptosystems*, Communications of the ACM, 21(2) (1978) 120.
[13] Official webpage for PINE project, http://www.washington.edu/pine/
[14] Official webpage for WGET project, http://www.gnu.org/software/wget/wget.html