One Implementation of Time-stamping Authority

STEVAN MILINKOVIĆ[1], BRANISLAV MILOJKOVIĆ[1], DRAGAN SPASIĆ[2], LJUBOMIR LAZIĆ[3]

[1]Union University, School of Computing
Knez Mihailova 6, 11000 Belgrade, SERBIA
smilinkovic@raf.edu.rs; http://www.raf.edu.rs
bmilojkovic07@raf.edu.rs; http://www.raf.edu.rs

[2]Public Enterprise of PTT Communications „Serbia“
Katićeva 14-18, 11000 Belgrade, SERBIA
dspasic@ptt.rs; http://www.ca.posta.rs

[3]State University of Novi Pazar
Vuka Karadžića bb, 36300 Novi Pazar, SERBIA
llazic@np.ac.rs; http://www.np.ac.rs

Abstract: - This paper describes the architecture and redundancy of the TSA (Time-Stamping Authority) system of the Serbian Post. The process of issuing a time-stamp and the life-cycle of the TSA server's cryptographic keys are being explained. Functionality overview of the TSA billing application and user portal is given. Finally, the time-stamp client application and test TSA server of the Serbian Post are presented.

Key-Words: - Serbian Post, Time-Stamping Authority - TSA, time-stamp, Serbian Electronic Document Act

1 Introduction

Time-stamping of a document guarantees that the electronic document exist at the time of the stamping and that it’s integrity is kept. An electronic document that has been time-stamped can not be altered without detection.

Time-stamping is a service that is provided by an institution that is called a Time-Stamping Authority (TSA). The Public Enterprise of PTT Communications "Serbia" has built an information system for issuing time-stamps and has become a TSA in the Republic of Serbia. The Serbian ministry in charge of information society has certified Serbian Post TSA in March 2012 for issuing time-stamps. The time-stamps issued by Serbian Post are intended for all participants of electronic business in the Republic of Serbia, for natural as well as legal persons (state government, local government, public enterprises, banks, insurance companies, and various organizations and institutions).

Serbian Post TSA is built in accordance with international standards and technical specifications: RFC 3161 [1], ETSI TS 102 023 [2] and ETSI TS 101 861 [3].

The most important tasks during the building of the Serbian Post TSA system were the following:

- Designing the architecture of the TSA system.
- The choice of TSA software [4].
- Acquiring the hardware and software for the TSA system.
- Developing the TSA proxy software.
- Developing the software for billing and recording of issued time-stamps.
- Developing the portal for overview of a time-stamp user profile and issued time-stamps.
- Developing the time-stamp client application.

This paper describes major components of the TSA system at Serbian Post.

2 Architecture and Redundancy

The architecture of the Post TSA system is shown in the figure 1. The order of requests for issuing a time-stamp is the following:

- User (client) requests for issuing time-stamps are equally balanced on two TSA Proxy servers. If one of the TSA Proxy servers is unavailable, all user requests are automatically redirected to the other TSA Proxy server. TSA Proxy servers are in a load balancer configuration.
- From the TSA Proxy servers, the user requests for issuing time-stamps are forwarded to two TSA servers. If one of the TSA servers is unavailable, all user requests are automatically redirected to the other TSA server. TSA servers are in a load balancer configuration.

Redundancy and high availability of the Post TSA service is achieved by installing two TSA Proxy servers and two TSA servers. The system is modular, so it is easy to add new TSA Proxy servers, in the case of failure or high load on the existing servers.

The Post TSA Proxy application is installed on TSA Proxy servers. This application is developed in accordance to Post requirements, and is integrated with the application for charging and keeping track of issued time-stamps (Billing application), which is installed on the Billing server. TSA Proxy servers perform the following tasks:

- Reject TSA requests that are not created in accordance with RFC 3161 [1] standard and ETSI TS 101 861 [3] technical specification.
- Reject TSA requests that specify an unsupported hash algorithm.
- Reject TSA requests that specify an unsupported TSA policy OID.
- Reject TSA requests that do not demand the TSA certificate inside the time-stamp response (certReq=true).
- Forward user information to the Billing application (server) for the purpose of authentication.
- Forward TSA requests of registered user to the Billing application and to the TSA servers.

TSA Proxy servers significantly reduce the load on TSA servers because they forward only valid TSA requests from registered users, thus preventing DoS attacks on TSA servers.

TSA servers are the most important servers in the TSA system because they issue the time-stamps. TSA servers are equipped with HSM (Hardware Security Module) devices for generating and storing the TSA private key, which is used for digitally signing time-stamps. The TSA servers have Thales e-Security [5] TSA software installed.

A Barracuda Networks Hardware Load Balancer (HLB) is placed in front of the TSA Proxy servers and the TSA servers. Barracuda HLB devices support two algorithms for load balancing [6]:

- Weighted Round-Robin. In this algorithm, the number of connections to a server depends on the weight coefficient that is assigned up-front to each server, so that a server with a higher weight will have more connections. The disadvantage of this algorithm is that it does not check if old connections are still active, since it is difficult to determine this for long-lasting connections.
- Weighted Least Connections. In this algorithm, the number of connections to a server depends on the weight coefficient that is assigned up-front to each server, and on the number of active connections. This algorithm is generally recommended, and the one that is used in Serbian Post TSA system.

Both HLB devices have the IPS (Intrusion Prevention System) service running, which additionally protects the TSA system from potential malicious attacks.

Internet users access the Post TSA system publicly, however, they have to register first. Registered users can authenticate in one of two ways when making a time-stamp request:

- User name and password.
- Digital certificate.

Users can configure their identification parameters via the Serbian Post Portal for viewing time-stamp user profiles and issued time-stamps.

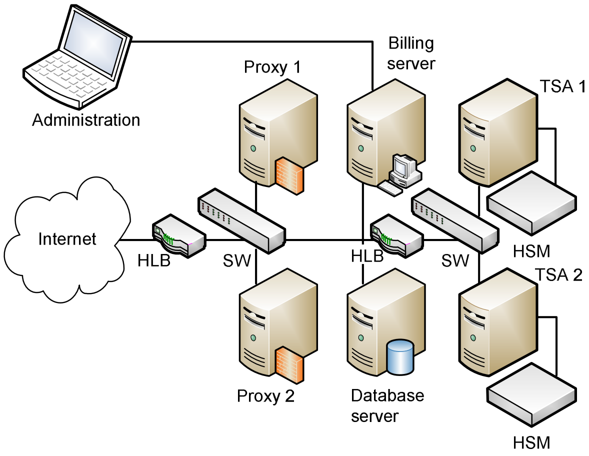

Fig. 1. Architecture of implemented system

3 Time-stamping Process

3.1 Time-stamp

The Post TSA department ensures that all time-stamps are issued securely, and that the time information is accurate. The Post TSA department issues only one type of time-stamp [7]. Each time-stamp contains an identification number of the time-stamping policy (TSA Policy OID) and a unique time-stamp serial number.

The time-stamp is digitally signed by the TSA server private key. The Serbian Post TSA system uses RSA algorithm with PKCS#1 standard and 2048 bit key length to create the digital signature. The time-stamp itself carries the TSA digital certificate which is used to verify the time-stamp signature.

The time-stamp contains UTC time, with a maximum error of ±1 seconds compared to UTC actual time. The time-stamp is valid for the same period of time as the TSA digital certificate which is used to verify the validity of the time-stamp.

3.2 Clock Synchronization with UTC

The Post TSA department ensures that time information in an issued time-stamp is not more than ±1 second different from UTC actual time, and does not issue time-stamps outside of this specified accuracy.

The Post TSA department provides automatic time synchronization of the TSA server with a source of accurate time, in accordance with the above specified precision. The TSA system uses a source of accurate time with NTP (Network Time Protocol) synchronization and internal clock. The NTP synchronization is performed with Stratum 1 and 2 servers.

If TSA server time is not synchronized with the source of accurate time, it stops issuing time-stamps until synchronization is re-established.

TSA server synchronization is done in such a way that there cannot be a larger deviation than the specified error level.

4 Key managenet Life Cycle

4.1 TSA key generation

In the Post TSA department, asymmetric keys used to digitally sign the time-stamps are always generated under a strictly controlled environment, which includes the following:

- Generating TSA server keys for digital signing is performed in a secure environment by at least two persons, i.e. TSA administrators, who have trusted roles,
- Generating TSA server keys for digital signing is done by using the HSM device, which is certified in accordance with security criteria EAL 4+ and FIPS 140-2 level 3.

4.2 TSA private key protection

The Post TSA department ensures that TSA server private keys stay secret and that their integrity is kept, which includes the following:

- TSA server private keys for digital signing are stored and used inside the HSM device, which is certified in accordance with security criteria EAL 4+ and FIPS 140-2 level 3,
- There are no copies of TSA server private keys for digital signing.

4.3 TSA public key distribution

The Post TSA department publishes TSA digital certificates used to verify the time-stamp signature, along with additional parameters and validity time on its Web page.

The Post TSA department ensures availability of this information, as well as making sure that the published information stays accurate during distribution to all involved parties.

4.4 Rekeying TSA's key

At least once in three months, the Post TSA department generates a new pair of asymmetric keys for signing time-stamps.

4.5 End of TSA key life cycle

The Post TSA department ensures that TSA server private keys are not used after their planned expiration date. Private keys are changed before the previous key’s expiration date, which is when the old key is no longer in use and is permanently destroyed.

If the planned expiration date of three months expires, the Post TSA department will not issue any time-stamps until a new private key is generated and put into use.

4.6 Life cycle management of HSM

The Post TSA department ensures the safety of the HSM device which creates and stores the TSA server keys for digital signing of time-stamps.

Before relocating, or otherwise compromising the HSM device environment, the Post TSA department will cease to use and permanently destroy the TSA server private keys.

5 System Access Management

The Post TSA department grants access only to authorized personnel:

- Protection of the internal computer network of the Post TSA system from unauthorized access by using a firewall and IPS (Intrusion Prevention System) system,
- An effective user account administration is ensured for accounts that have access to the TSA system, to maintain a proper level of protection,
- Access to data and applications is limited in accordance with the control access policy,
- Post TSA department staff is authenticated via digital certificates before granted administration privileges for critical applications,
- All TSA staff activities are logged,

- Local network components (firewall, hardware load balancer, and switch) are located in the system room, in a physically isolated environment. Their configuration is periodically checked in accordance with the requirements.

6 Time-stamp Client Application

During development of the Post TSA system, there rose a need to use a time-stamp client application to test the correctness of the TSA service. Initially, Adobe Reader [11] and TimeStampClient [12] were used as time-stamp client applications. For various limitations of these two applications, from the viewpoint of testing TSA functionality, a decision has been made to develop a custom application for testing the TSA service.

In accordance with specified technical requirements, a Time-Stamp Client application version 1.2 (hereinafter, the application) has been developed inside Serbian Post with the following functionalities:

- The application can be used on Windows machines that have Microsoft .NET Framework 4 installed. A Linux user has notified us that the application also works with Mono 2.10 on GNU/Linux systems, as long as the configuration file "Time-Stamp konfiguracija.txt" is renamed to "Time-Stamp klijent Poste v1.2.exe.config".
- The configuration file of the application allows the client's user to modify the following parameters:
- TSA server address. The address is specified as a URL (with http:// or https:// prefix). It is possible to specify any number of TSA server addresses, which are then used to make time-stamp requests. All addresses need to be separated by a semicolon character (;).
- Proxy server address, port, user name and password to access the proxy server. The address is given without the http:// prefix.
- The application's main form allows the user to select or input the following values:
- TSA server address.
- Mode of logging the user into the TSA server (user name and password, digital certificate, or anonymous access).
- Time-stamp user name and password, if this mode of log in is selected previously.
- Hash algorithm that will be used to calculate the message digest (SHA-1, SHA-256, SHA-384, SHA-512 or MD5).
- TSA Policy OID. The application checks the validity of the specified OID, and if it is invalid, it displays an error message.
- Nonce - a random number. The application allows only digits, and the first digit can not be zero. The largest number that can be entered is 9.223.372.036.854.775.807 ($2^{64}/2-1$; the field is of type "long variable", without the possibility of entering negative numbers).
- Demand TSA certificate in the time-stamp (yes/no).
- The file that will be time-stamped. If no file is selected for stamping prior to making the request when the request is made, the application will display an error message.
- Once the file has been successfully time-stamped, two new files are generated: TSA request (the request for issuing a time-stamp), and TSA response (the time-stamp in a more general sense, in which the TSA token which represents the actual time-stamp and optionally, the TAC (Time Attribute Certificate) if the TSA server generates it). If there is an error in the time-stamping process, the application notifies the user with the appropriate message.
- The application supports display of TSA requests (.tsq extension), TSA responses (.tsr), TSA tokens (.tst) and TACs (.tac) in a user-friendly format. The display of the TSA request and TSA response content is in accordance with the terminology and order specified in RFC 3161 [1] standard, and terms are used in English. The application supports display of all TSA tokens that are contained in one .tst file. Display of a TAC (Time Attribute Certificate) file is in accordance with Thales SDK (Software Development Kit) [13]. Terms are in English.
- On the form that displays the TSA response, which shows the details of an issued time-stamp, the client can do the following:
- Display the TSA certificate, if it is contained inside the time-stamp. Before displaying the TSA certificate, the application checks if the TSA certificate has been revoked.
- Display the TSA certificate chain, if it is contained inside the time-stamp.
- Check the time-stamp integrity. The algorithm for checking the time-stamp integrity is shown in the figure 2.
- On the form that displays the TSA response that shows the details about the TAC, it is possible to save the TAC to hard drive as a file.

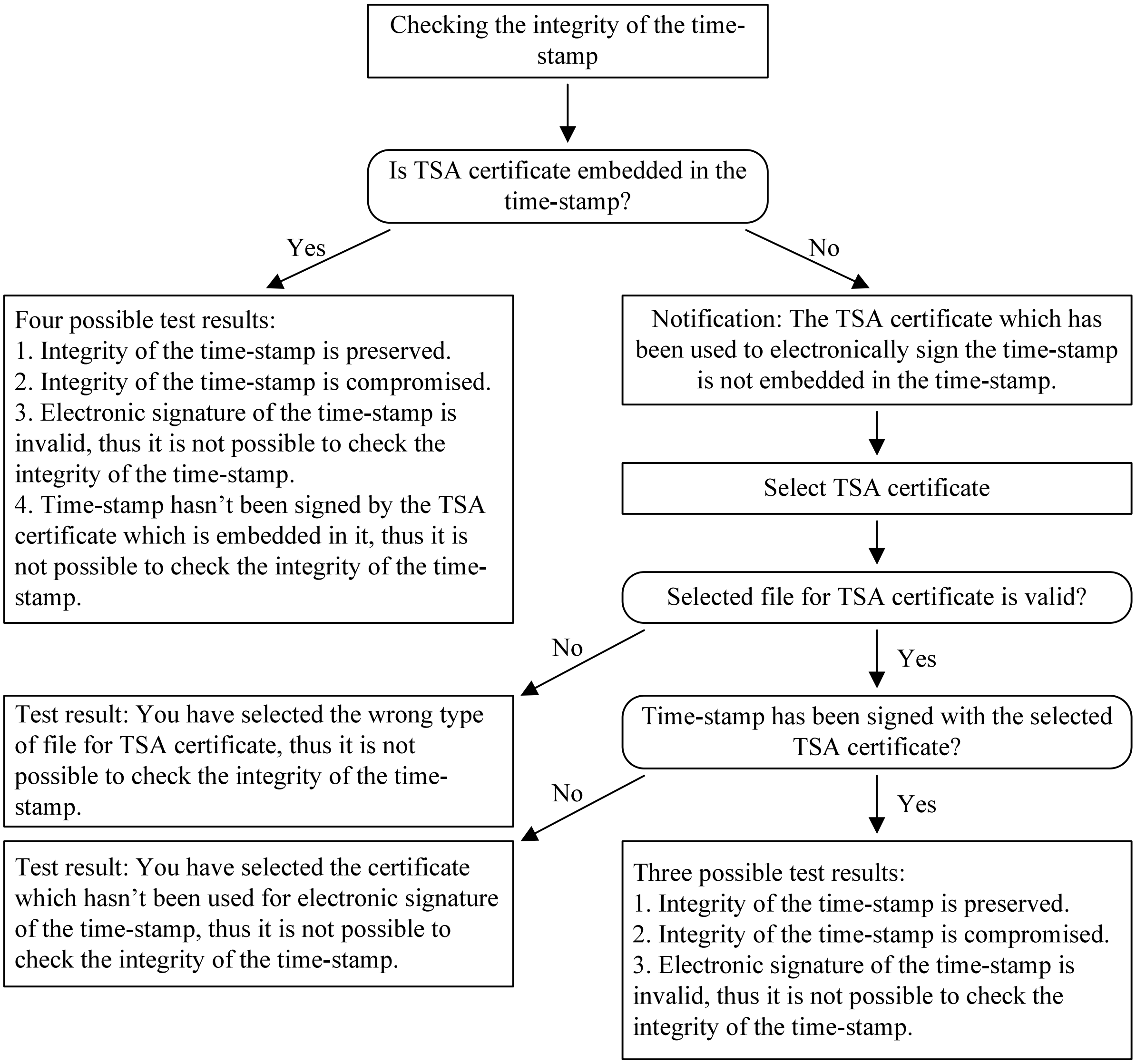
Checking the integrity of the time-stamp
Is TSA certificate embedded in the time-stamp?
Yes
No
Four possible test results:
1. Integrity of the time-stamp is preserved.
2. Integrity of the time-stamp is compromised.
3. Electronic signature of the time-stamp is invalid, thus it is not possible to check the integrity of the time-stamp.
4. Time-stamp hasn't been signed by the TSA certificate which is embedded in it, thus it is not possible to check the integrity of the time-stamp.
Notification: The TSA certificate which has been used to electronically sign the time-stamp is not embedded in the time-stamp.
Select TSA certificate
Selected file for TSA certificate is valid?
No
Yes
Test result: You have selected the wrong type of file for TSA certificate, thus it is not possible to check the integrity of the time-stamp.
Time-stamp has been signed with the selected TSA certificate?
No
Yes
Test result: You have selected the certificate which hasn't been used for electronic signature of the time-stamp, thus it is not possible to check the integrity of the time-stamp.
Three possible test results:
1. Integrity of the time-stamp is preserved.
2. Integrity of the time-stamp is compromised.
3. Electronic signature of the time-stamp is invalid, thus it is not possible to check the integrity of the time-stamp.

Fig. 2. Checking the integrity of a time-stamp in the Serbian Post Time-Stamp Client application

The Serbian Post Time-Stamp Client application is used by the following users:

- TSA administrators, for the purpose of testing the TSA servers, that operate in accordance with RFC 3161 [1] standard.
- Users of the TSA service: for verifying that communication with the TSA servers is operational.
- Programmers: as a starting point for developing time-stamp clients inside their own applications.
- Interested potential TSA service users and students for educational purposes.

Suggestions for improvement of the current application version (i.e. version 1.2) are stored, and when enough useful operations are collected, a new version of the application (ver. 1.3) will be developed. So far, there haven't been error reports for the current version of the application.

8 Conclusion

A prerequisite for introducing electronic business is ensuring the safety of electronic data and documents. Electronic data and documents are easy to alter, so it is very important to keep their integrity and to support a possibility of proving that the data and documents existed at a given time on a given date. Time-stamping provides a guarantee that a particular piece of data, or a document has existed at the time of stamping, and that its integrity is kept.

Time-stamping is usually applied as an added value to a digital signature. Reasons to add a time-stamp to a digital signature are the following:

- Time-stamping makes it impossible to fake the time and date of signing the document.
- Time-stamping enables successful verification of a digital signature even after the expiry of the certificate that was used to create the signature.
- Time-stamping enables successful verification of a digital signature even after revocation of the digital certificate used to create the signature.

In the Republic of Serbia, there is a legal framework which regulates the issuing of time-stamps. The legal regulation in Republic of Serbia allows issuing time-stamps by using the simple scheme (producing independent tokens), but does not allow issuing time-stamps by using the linked scheme (producing linked tokens).

Public Enterprise of PTT Communications "Serbia" (Serbian Post) is an accredited time-stamp issuer in the Republic of Serbia. The time-stamps issued by Serbian Post are intended for all participants of electronic business in the Republic of Serbia, for natural as well as legal persons (government, public enterprises, banks, insurance companies and other organizations and institutions).

References:

[1] C. Adams, P. Cain, D. Pinkas, R. Zuccherato: RFC 3161: Internet X.509 Public Key Infrastructure Time-Stamp Protocol (TSP), *Internet Engineering Task Force (IETF)*, August 2001.

[2] European Telecommunications Standards Institute: Electronic Signatures and Infrastructures (ESI); Policy requirements for time-stamping authorities, ETSI TS 102 023 V1.2.2 (2008-10).

[3] European Telecommunications Standards Institute: Electronic Signatures and Infrastructures (ESI); Time stamping profile, ETSI TS 101 861 V1.4.1 (2011-07).

[4] S. Milinković, B. Milojković, D. Spasić, Lj. Lazić: Evaluation of Some Time-Stamping Authority Software, *Proc. the 6th International Conference on Methodologies, Technologies and Tools enabling e-Government*, Belgrade, Serbia, July 2012, pp. 89-99.

[5] Thales e-Security: Thales Time Stamp Server Administrator Guide, March 2012.

[6] Barracuda Networks: Barracuda Load Balancer Administrator's Guide Version 3.6, 2011.

[7] Official Journal of the Public Enterprise of PTT Communications "Serbia": Serbian Post Time-Stamp Policy, No. 782, 2012.

[8] D. Spasić, I. Lazarević, S. Milinković, B. Milojković: Serbian Post Certification Authority Software for Billing and Recording of Issued Time-Stamps, *Proc. 30th Symposium on Novel Technologies in Postal and Telecommunication Traffic*, Belgrade, Serbia, December 2012, pp. 149-158.

[9] Official Journal of the Republic of Serbia: Electronic Document Act, No. 51, 2009.

[10] Official Journal of the Republic of Serbia: Time-Stamp Issuance Regulation, #112, 2009.

[11] Adobe Systems Incorporated: Adobe Reader, [Online] http://www.adobe.com.

[12] J. Imrich: TimeStampClient, [available online] http://timestampclient.sourceforge.net.

[13] Thales e-Security: Thales Time Stamp Server SDK Reference Guide, August 2009.

[14] R. Miškinis, D. Smirnov, E. Urba, A. Burokas, B. Malyško, P. Laud, F. Zuliani: Digital Time Stamping System Based on Open Source Technologies, *IEEE Trans. Ultrasonics, Ferroelectrics, and Frequency Control*, Vol. 57, No. 3, March 2010, pp. 721-727.

[15] D. Mills, J. Martin, Ed., J. Burbank, W. Kasch: RFC 5905: Network Time Protocol Version 4: Protocol and Algorithms Specification, Internet Engineering Task Force (IETF), June 2010.

[16] B. Haberman, Ed., D. Mills: RFC 5906: Network Time Protocol Version 4: Autokey Specification, Internet Engineering Task Force (IETF), June 2010.

[17] D. L. Mills: *Computer network time synchronization: The Network Time Protocol on Earth and in Space*, 2nd edition, CRC Press, Boca Raton, FL, USA, 2011.

[18] Ascertia: TSA Crusher, [avalilable online] www.ascertia.com/Products/TSA-Crusher.aspx.

[19] ISO/IEC 18014-2: Information technology - Security techniques - Time-stamping services - Part 2: Mechanisms producing independent tokens, 2009.

[20] ISO/IEC 18014-3: Information technology - Security techniques - Time-stamping services - Part 3: Mechanisms producing linked tokens, 2009.

Acknowledgment:

Results presented in this paper are part of the research that is supported by Ministry of Education and Science of the Republic of Serbia, Grants No. III-45003 and TR-35026.