Test Suite Study for Security Analysis of Digital Signature Applications

Tamer Ergun, Ferruh Özbudak, Ferda Topcan

***Abstract*—Digital signature technology is used widely for security and trust in electronic business and communications. Nowadays it becomes commonly used especially in government agencies. At this point of view, it is crucial to implement correct applications to create and verify digital signatures.**

CEN [1] has introduced the security requirements for signature creation [1] and signature verification [2] applications but neither proposed a PKI model nor implemented a test suite to evaluate the accuracy of signature applications. This is a real necessity, because a signature application have to be hardly tested and the responses of the application to a wide range of wrong scenarios have to be well analyzed.

With this work, we aimed to design a unique PKI model and state whole problematic scenarios both in signature creation and verification and address the lack of such a suite by designing *E-Signature Test Suite*. *E-Signature Test Suite* is a detailed tool, formed by a huge set of certificates and signatures, designed to analyze the accuracy of digital signature applications working mechanism both in signature creation and verification.

***Index Terms*—CAdES[2], Digital Signature, PKI[3], Time Stamp, X509 Certificate.**

I. Introduction

DIGITAL signature is a world wide used technology for security and trust in electronic business and communication. This technology is based on cryptography and more particularly public key cryptography, but the management of the system, and stating a clear identification of the parties are also very important. Such necessities bring out a famous concept, known as public key infrastructure (PKI).

An end-entity needs signature creation and verification applications to get a role and use PKI services. Even though all the PKI components work well, we have to be sure of the security of the application. This shows that, secure PKI is a necessary step of reliable transaction but beside this step, signature creation and verification applications must be also reliable. These applications must be capable of detecting each security ring of the PKI chain and must be strongly analyzed. For this aim, European Committee for Standardization (CEN) has introduced guidelines CWA 14170 [1] and CWA 14171 [2] and defined some security criterias for signature creation and verification applications but these guidelines do not define each occurable wrong scenarios in PKI explicitly and open to interpretation. European Telecommunications Standards Institute (ETSI) has also released a guideline ETSI TS 102 853 [8] for security criterias in signature verification but, similar to CWAs, this guideline does not define each wrong scenarios explicitly and even does not offer a PKI model for testing signature applications. Therefore it is an open problem to design a unique PKI model which includes all the occurable problematic scenarios for each PKI component. Besides, it is important to implement this model and create the whole certificate set which is capable to interact with all the problematic components of designed PKI model. Last step is creating the signature set which includes error scenarios coming from the certificate set mentioned above and further structural error cases in each signature types defined by ETSI [5].

In this work, we aimed to address the lack of such a PKI model and a detailed tool for analyzing signature applications. We focused on analyzing applications seperately by means of signature creation and verification. First we stated the occurable wrong scenarios in a PKI hierarchy and designed our PKI model for our test suite. After this step we created root, subroot and end-entity certificates. While creating these certificates, we paid attention to create one error case from root to end-entity certificate to analyze each error case individually. Beside the certificates, we created the occurable wrong scenarios in online certificate status protocol (OCSP) servers and certificate revocation lists (CRLs) to analyze if the application validates the required properties of revocation datas. Since time stamp is an important invariant of PKI, we also designed and created the necessary error cases for time stamp servers. After all, a big PKI model with not all but most of the wrong scenarios established.

In this paper we will present our unique PKI model and the error scenarios designed to create *E-Signature Test Suite*. The structure of the paper is as follows. In Sec. 2, some definitions and basic PKI concepts are reviewed. In Sec. 3, we present our test scenarios and scetch the hierarchical certificate chain structure of Test Suite. Finally, In Sec. 4, we present the statistical results of our work and the future work plan.

II. Preliminaries

In this section we collect necessary definitions and cover some PKI concepts.

T. Ergun is with the TÜBİTAK-KamuSM, Bilim, Sanayi ve Teknoloji Bakanlığı Ek Bina, GMK Bulvarı 128/B Kat:3, Tandoğan, Çankaya, Ankara, Turkey.

F. Özbudak is with the Department of Mathematics and Institute of Applied Mathematics, Middle East Technical University, Dumlupınar Bulvarı No. 1, 06800 Ankara, Turkey.

F. Topcan is with the TÜBİTAK-KamuSM, Bilim, Sanayi ve Teknoloji Bakanlığı Ek Bina, GMK Bulvarı 128/B Kat:3, Tandoğan, Çankaya, Ankara, Turkey.

Manuscript revised August 22, 2013.

[1]European Commitee for Standardization

[2]CMS Advanced Electronic Signatures

[3]Public Key Infrastructure

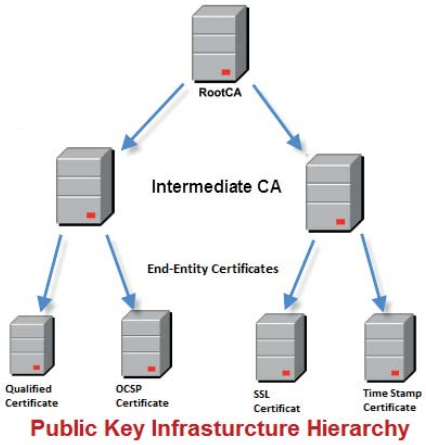

Fig. 1: CA Hierarchy

Public-Key Key of a user's key pair which is publicly known.
Private-key Key of a user's key pair which is known only by that user, and responsible for the use of it.
Certificate Digitally signed data, binding identity with public key.
Qualified Certificate A certificate which meets the requirements in [4], [17], [15].
CA Certificate An authority authorized legally to create and assign public-key certificates.

More definitions on certificates can be found in [11] and [12].

PKI hierarchy is an other important issue. Along the path from end entity certificate to a RootCA, every component of PKI is signed by an authorized signer. In this signature chain, signer is known as issuer and the signed component is known as subject. Therefore the word 'hierarchy' indicates the total issuer-subject relation of the whole system. In figure 1, a hierarchical two-level chain structure is shown. In order to trust an end entity certificate, one should verify its respective intermediate CA (issuer) and lastly the RootCA. All necessary verification items will be covered in section III. Beside the PKI hierarchy, we should care of two important concepts about certificate revocation check:

CRL is the abbreviation of Certificate Revocation List and includes all revoked certificates with serial number and revocation date.
OCSP is a response data signed by an OCSP certificate. This response data contains the revocation status of the requested certificate.
OCSP Certificate is the certificate authorized to sign OCSP responses.

Both in CRL and OCSP response, the revocation date of a revoked certificate is crucial. Since revocation date is so crucial, date used for revocation check must be trusted. As stated before, trusted concept means accurate data signed by a trusted CA. Therefore time stamp is used to indicate the accurate date.

Time Stamp is a token containing a specific information and a date such that, that specific data exists before the date in token. Time stamp token is also a signed data issued by a time stamp certificate.
Time Stamp Certificate is the certificate authorized to sign time stamp tokens.

Lastly we should cover electronic signature formats. ETSI defined 10 signature formats [5]. In signature verification part of *E-Signature Test Suite*, we implemented scenarios, derived from our unique PKI model, in these 10 formats. Now we briefly define frequently used ones.

CAdES-BES namely the basic electronic signature containing signer's document, some signed attributes and the digital signature.
CAdES-T CAdES-BES with time stamp.
CAdES-XL CAdES-T with complete certificate and revocation references and values.

Beside of ETSI signature types, signature files can be splitted into two groups as attached and detached signatures.
Attached is the type of signature where tha actual content (message) is stored in the signature files.
Detached is the type of signature where the actual content (message) is not stored in the signature files. In this case both the signature file and the content have to be stored for validation.

III. Test Suite

In this section we shall study the details of our unique PKI model designed for test suite and define the error cases, nodes of PKI tree, both in signature creation and signature verification parts.

A. Signature Creation

This part of the suite defines the error cases designed to construct the PKI model and to analyze the signature creation accuracy of signature applications. The aim is to create verifiable signature by means of applying certificate validation prior to signing process. While designing, we stated the wrong scenarios and grouped them with respect to their natures. There exists five types of wrong scenarios:

1) Certificate Self Check: Certificate self check section includes the scenarios related with the structure of end entity certificates with respect to the directives coming from [4], [6], [13], [14], [15], [17] in Turkey. Self check scenarios can be grouped into three groups as qualified certificate checks, certificate expiration check and certificate signature check.

Qualified Certificate Checks:

- **Non-Repudiation Absent:** Non-Repudiation is a MUST field in a qualified certificate. In this scenario, a qualified certificate without Non-Repudiation field is designed. We aim to determine if the application checks certificate's Non-Repudiation field to accept it as a qualified certificate.
- **ETSI[4] QC Statement ID Absent:** In this scenario, a qualified certificate without ETSI QC Statement ID

[4]The European Telecommunications Standards Institute

(0.4.0.1862.1.1) is designed. We aim to determine if the application checks certificate's ETSI QC Statement ID field to accept it as a qualified certificate.

- **ICTA[5] QC Statement ID Absent:** In this scenario, a qualified certificate without ICTA QC Statement ID (2.16.792.1.2.1.1.5.7.1.1) is designed. We aim to determine if the application checks certificate's ICTA QC Statement ID field to accept it as a qualified certificate.
- **ICTA QC Statement Info Absent:** In this scenario, a qualified certificate without ICTA QC Statement Info ("Bu sertifika, 5070 sayılı Elektronik İmza Kanununa göre nitelikli elektronik sertifikadır.[6]") is designed. We aim to determine if the application checks certificate's ICTA QC Statement Info field to accept it as a qualified certificate.
- **CP User Notice Statement Absent:** ICTA QC Statement Info has to be also stated in Certificate Policies [11] field of a qualified certificate. In this scenario, a qualified certificate without CP User Notice Statement is designed. We aim to determine if the application checks certificate's CP User Notice Statement field to accept it as a qualified certificate.

Last three items are specific for applications in Turkey but the others are global.

Certificate Expiration Check: Every public-private key pair has a cryptographic life time. Since certificate is defined as a digital data binding public key of an end-entity with his id, every certificate has a life time. In this scenario, an expired certificate is designed. We aim to determine if the application checks certificate's expiration date and do not permit to sign the document.

Certificate Signature Check: In the PKI hierarchy we stated that every evidence is signed by a trusted authority. Hence, an end entity certificate is also a signed data and is signed by its issuer, which is the intermediate CA (SubRootCA) illustrated in figure 1. The intermediate CA is also signed by RootCA and RootCA is self-signed, means RootCA signs itself. In this scenario, a signature forged end entity certificate is designed. We aim to determine if the application checks certificate's signature and if not valid, do not permit to sign the document.

2) Certificate Revocation Check: In daily life, like credit cards, a certificate may be revoked for some reasons. After the exact date of revocation, it is expected that no signature can be constructed with this certificate. Revocation status of a certificate is published in two ways. One of them is CRL and the other one is OCSP. Clients use these services by using access points stated in certificates. CRL for a specific certificate can be obtained by using CDP[7] field of the certificate. Similarly, OCSP response for a certificate can be obtained by using AIA[8] field of the certificate. In revocation check scenarios, revocation check form CRL and OCSP have to be checked seperately to determine that an application is able to deal with these services clearly. For this reason we designed two certificates such that both of them are revoked. Certificates differ from each other as one of them has just CDP field but not OCSP address and the other one has the reverse. We aim to determine if the application checks certificates' revocation info accurately in both ways.

3) Revocation Info Check: In section III-A2 the issue was to evaluate, the accuracy of getting revocation status of a certificate and the clearness of interaction with the CRL and OCSP services. In this part we examine the validity of a CRL file and an OCSP response and state in which situations they become invalid.

CRL Validation Checks:

- **Expired CRL:** As certificates, CRL files have also life time. This life time is restricted between thisUpdate and nextUpdate [11] fields of CRL file. In revocation control of a certificate from CRL, nextUpdate has to be greater than validation time. If not, in mean time, this CRL can not be used to validate a certificate since a fresh CRL has to be released before nextUpdate date of older CRL. In this scenario an expired CRL is designed. Aim is to determine if the application compares nextUpdate date of a CRL used for a specific end entity certificate with validation time.
- **Signature Forged CRL:** CRL is also a signed data and the signer CA of the CRL, used to be informed about revocation status of a specific certificate, has to be same with the signer (issuer) CA of that specific certificate [4]. If the signature of a CRL is forged than CRL file becomes invalid. In this scenario a signature forged CRL file is designed. Aim is to determine if the application verifies the signature of the CRL file.

OCSP Response Validation Checks:

- **Expired OCSP Response:** Similar with CRL, OCSP responses are expired for some reasons. OCSP responses have thisUpdate, nextUpdate and producedAt fields [9]. nextUpdate field indicates the time at or before which newer information will be available about the status of the certificate. This field is optional and if not set it means that newer revocation information is available all the time. producedAt indicates the time at which the OCSP responder signs the response. Therefore if nextUpdate is null then it means producedAt is the time after which newer information will be available about the certificate. At validation time, producedAt field has to be greater than validation time and if not, signature creation application must interpret the OCSP response as expired. In this scenario an OCSP server producing expired OCSP responses with null nextUpdate field is designed. Aim is to determine if the application compares producedAt with validation time.
- **Signature Forged OCSP Response:** OCSP response is also a signed data and is signed by an OCSP

[5]Information and Communications Technologies Authority

[6]This certificate is a quailified certificate according to e-signature law numbered 5070

[7]Certificate Distribution Point

[8]Authority Information Access

certificate. Issuer of the OCSP certificate has to be same with the issuer of the certificate to be informed about its revocation status [9]. If the signature can not be validated, the OCSP response becomes invalid. In this scenario an OCSP server producing signature forged responses is designed. Aim is to determine if the application verifies the signature of the OCSP response.

- **Expired OCSP Certificate:** An OCSP certificate has also a life time. In this scenario an OCSP server with an expired OCSP certificate, producing responses is designed. A signature application has to verify both the OCSP response and the OCSP certificate. Aim is to determine if the application checks the expiration status of the OCSP certificate.
- **Signature Forged OCSP Certificate:** An OCSP certificate is a signed data, signed by its issuer. In this scenario an OCSP server with a signature forged OCSP certificate is designed. Aim is to determine if the aplication checks the signature of the OCSP certificate.
- **Revoked OCSP Certificate:** As other certificates, an OCSP certificate can be revoked for some reasons. In our PKI model, revocation status of OCSP certificates are published via CRL. In this scenario an OCSP server with a revoked OCSP certificate, producing responses is designed. Aim is to determine if the application checks the revocation status of the OCSP certificate.

4) Issuer Check: So far we defined wrong scenarios related with end entity certificates. Beyond end entity certificates, most of the stated wrong scenarios can be applied to the subCA and RootCA certificates. We aimed to determine if the applications accurately handle necessary controls for subCA and RootCA authorities. In the case where just end entity certificates are concerned, we have a tree path with a valid root, valid subroot and defined problematic end entity certificates but while creating problematic scenarios for also subroots and roots, our PKI model became enlarged.

5) Time Stamp Check: In digital signature technology, time stamp is used for different reasons [5]. Time stamp token (response) is a signed data, signed by a trusted time stamp certificate. A signature application MUST check all necessary controls and verify the token. From this point of view we created negative scenarios that make a time stamp token invalid and embed the sources of these scenarios as new time stamp servers to our PKI model.

Time Stamp Validation Checks:

- **Signature Forged TS Token:** As all evidences used in PKI technology, time stamp token is a signed data. It is signed by a time stamp certificate with a trusted issuer. A malicious user of PKI may try to edit the token and change the time or the digest of the event. For resisting such a situation, signature creation applications must be capable of validating the signature of the token. In this scenario a time stamp server producing signature forged tokens is designed. Aim is to determine if the application verifies the signature of the time stamp token and informs the user about this issue.
- **Expired TS Certificate:** A time stamp certificate has also a life time. In this scenario a time stamp server with an expired time stamp certificate, producing tokens is designed. Signature application has to verify both the time stamp token and the time stamp certificate. Aim is to determine if the application checks the expiration status of the time stamp certificate.
- **Signature Forged TS Certificate:** Time stamp certificate is a signed data, signed by its trusted issuer. In this scenario a time stamp server with a signature forged time stamp certificate is designed. Aim is to determine if the application checks the signature of the time stamp certificate.
- **Revoked TS Certificate:** As other certificates, a time stamp certificate can be revoked for some reasons. In our PKI model, revocation status of TS certificates are published via CRL. In this scenario a time stamp server with a revoked time stamp certificate, producing time stamp tokens is designed. Aim is to determine if the application checks the revocation status of the TS certificate.
- **TS Root CA Scenarios:** All the scenarios with time stamp servers so far were about tokens and the time stamp certificates but a signature creation application must also validate the issuer of the time stamp certificate. From this point of view we designed 10 more time stamp certificates whose issuers are problematic because of different reasons. Each root CA has an individual mistake and aim is to determine if the application makes the issuer controls beyond the self check of time stamp token.

B. Signature Verification

This part of the suite is designed to analyze the signature verification accuracy of signature applications. Verification part of the suite includes signature files created in 10 signature types defined in [5]. These types are:

- BES (Basic Electronic Signature),
- EPES (Explicit Policy Based Electronic Signature),
- ES-T (Electronic Signature with Time),
- ES-C (Electronic Signature with Complete Validation Data References),
- ES-X-Type 1 (Extended Electronic Signature with Time Type 1),
- ES-X-Type 2 (Extended Electronic Signature with Time Type 2),
- ES-X-Long (Extended Long Electronic Signature),
- ES-X-Long-Type 1 (Extended Long Electronic Signature with Time Type 1),
- ES-X-Long-Type 2 (Extended Long Electronic Signature with Time Type 2),
- ES-A (Archival Electronic Signature).

Each of the signature files has one error case derived from our unique PKI model defined in section III-A and signature

type specific issues extending from type BES to type ES-A. From this point of view, each set of signatures for a particular type is equally or more detailed then the previous one and so has further analysis criterias. We can group our signature verification specific wrong scenarios in 5 groups.

1) Signature File Format Check: In these scenarios wrong formatted, do not compatible with ETSI [5], signature files are designed and aim is to determine if the signature verification application checks the format of the signature file.

2) Signature Verification Check: It means cryptographic validation of raw signature data files. In this scenario, we forged the signature of either signature file, time stamp token, CRL or OCSP response. Aim is to determine if the signature verification application verifies the signature of all signed datas used in the signed file.

3) Certificate Path Verification Check: Certificate path consists of a root certificate, an intermediate certificate and an end entity certificate. End entity certificates are either qualified certifices, OCSP certificates or time stamping certificates. For all certificates in the path one of the following errors may occur:

- **Signature Forged Certificate:** Aim is to determine if the signature verification application checks the signature of each certificate forming the path from the qualified end entity certificate to the trusted root certificate in the signature file.
- **Revoked Certificate:** Aim is to determine if the signature verification application checks the revocation status of each certificate forming the path from the qualified end entity certificate to the trusted root certificate in the signature file.
- **Expired Certificate:** Aim is to determine if the signature verification application checks the expiration status of each certificate forming the path from the qualified end entity certificate to the trusted root certificate in the signature file.
- **Certificate usage constraints does not match:** This scenario is for qualified end entity certificates. Aim is to determine if the signature verification application checks the qualified certificate properties of each signer certificate in the signature file.

4) Validation Data Access Check: Validation datas are certificates and revocation informations (CRL or OCSP responses) to validate the certificate path due to the signature time. Validation data must be collected within a short period after the signing time and stored on a local store or attached to the signature file. For signature types ES-C and above, validation datas, its references or both are attached to the signature. For all validation datas one of the following errors may occur:

- **Missing Validation Data:** In this scenario we omitted a revocation-certificate reference [5] or a revocation-certificate value from the signature. Aim is to determine if the signature verification application realizes the missing data.
- **Corrupted Validation Data:** In this scenario we corrupted a revocation-certificate reference [5] or a revocation-certificate value from the signature. Aim is to determine if the signature verification application realizes the corrupted data.
- **Expired Validation Data:** In this scenario we attached an expired validation data derived from our PKI model, defined in section III-A3. Aim is to determine if the signature verification application realizes the expired data.

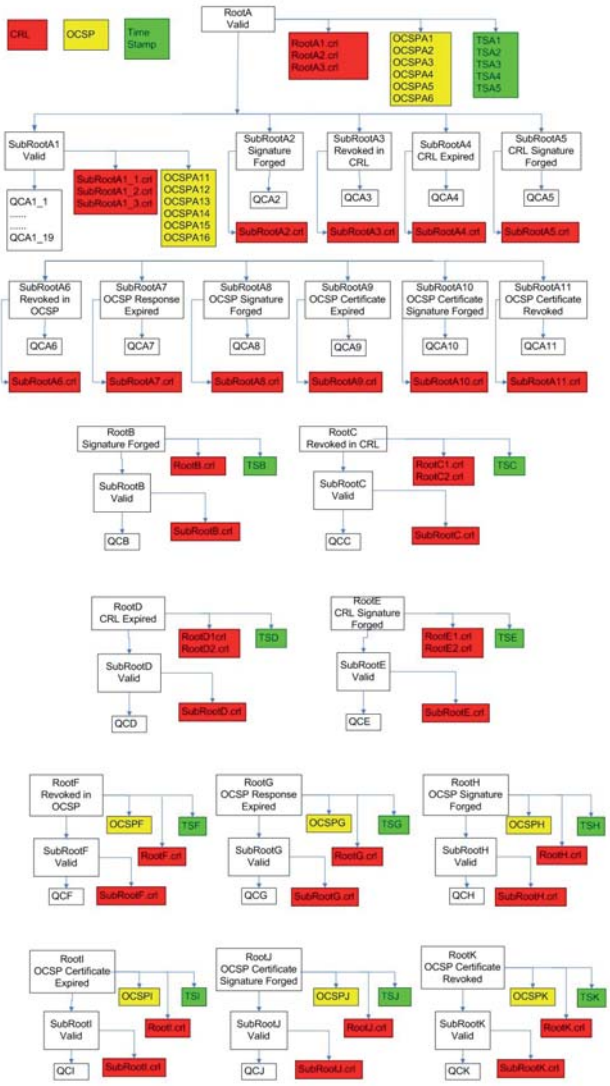

Fig. 2: E-Signature Test Suite Hierarchy

5) Archive Signature Check: ES-A, archival form, is a signature type extending from ES-XL, ES-XL Type 1 or ES-XL Type 2 where the whole signature is time stamped, known as archive time stamp. It is against to protect breaking of encryption algorithms and hashing algorithms. These tests are designed to check e-signature archiving verification process of signature verification applications.

IV. Conclusion

In this work we aimed to solve an important lack of digital signature process, security and reliability of signature creation

PKI Element Type	Total Amount
Root Certificates	11
Sub Root Certificates	21
Qualified End Entity Certificates	39
Time Stamp Servers	15
OCSP Servers	18
Released CRLs	39

Fig. 3: Signature Creation Suite

and verification applications. CEN [1], [2] and ETSI [8] has introduced some security concepts on both signature applications but these concepts are general and neither propose a PKI model nor an implementable design theory on these subjects. In order to fill this gap, we studied on designing a unique PKI model and implemented this model to produce certificate and signature set of *E-Signature Test Suite*. Our PKI model is shown in figure 2. As we stated before, each path from an end entity certificate to its root certificate contains just one faulty. Thus we succeed to evaluate the accuracy of signature applications to individual error scenarios independently.

Since we aimed to design each occurable error scenarios in a PKI model, signature creation part of the suite, defined in section III-A composed a huge public key infrastructure statistically charted in figure 3. We have established 18 OCSP Responders and 15 Time Stamp services. These services are available online and accessible during the signature creation tests. SubRoot certificates, root certificates and CRL's are also published online and accessible through the web.

In the signature verification part, we created wrong scenarios in 10 signature types defined in III-B. While creating these signatures, we used TUBITAK[9]-BILGEM[10] digital signature libraries. Types of signatures and the total amount of signature files created are shown in figure 4. Our design philosophy for creating signature files is same as signature creation hierarchy structure such that each signed file has just one error for the same reason. As seen in figure 4, from type BES to ES-A, the number of signature files created are increasing. This is because the details of the signature files are extending as signature type tends to ES-A.

In Turkey, digital signature applications used by government agencies are controlled by TUBITAK-BILGEM-KAMUSM[11]. This task is given to TÜBİTAK-BİLGEM-KAMUSM by the prime ministry [3], [16]. The output of this work is used to evaluate government agencies applications since December, 2012. During these period, we realized different kinds of mistakes that the signature applications do and correct them.

Also in this period we realized that some applications have efficiency loss. Even if they apply the validation rules successfully, the path they follow is time consuming. The order of validation for a particular issue, say OCSP response validation, affects the time spent considerably. As future work, we plan to design an efficient verification path for some

Signature Type	Attached	Detached
BES	46	46
EPES	46	46
ES-T	70	70
ES-C	78	78
ES-X TYPE 1	93	93
ES-X TYPE 2	93	93
ES-XL	95	95
ES-XL TYPE 1	110	110
ES-XL TYPE 2	110	110
ES-A	110	110
	851	851
Total	1702	

Fig. 4: Signature Validation Suite

evidences used in electronic signatures and assist signature application developers in terms of efficiency.

Acknowledgment

This research project would have been possible with the support of TUBITAK-BILGEM-KAMUSM.

References

[1] CEN, *CWA 14170 - Security requirements for signature creation applications,* May 2004.
[2] CEN, *CWA 14171 - General guidelines for electronic signature verification,* May 2004.
[3] Doç. Dr. T. K. Bensghir, F. Topcan, *E-imza: Türkiye'de Kamu Kurumlarında Uygulanması,* TODAİE Yayın No:356, Ankara, Turkey, December 2010.
[4] Committee Decision, *Nitelikli Elektronik Sertifika, SİL ve OCSP İstek/Cevap Mesajları Profilleri Rehberine İlişkin Kurul Kararı, 2007/DK-77/207,* 18 April 2007.
[5] ESI, *ETSI TS 101 733 - Electronic Signatures and Infrastructures (ESI); CMS Advanced Electronic Signatures (CAdES),* November 2009.
[6] ESI, *ETSI TS 101 862 - Electronic Signatures and Infrastructures (ESI); Qualified Certificate profile,* March 2004.
[7] ESI, *ETSI TS 101 903 - XML Advanced Electronic Signatures (XAdES),* June 2009.
[8] ESI, *ETSI TS 102 853 - Signature verification procedures and policies,* July 2012.
[9] IETF, *RFC2560 - X.509 Internet Public Key Infrastructure Online Certificate Status Protocol - OCSP,* June 1999.
[10] IETF, *RFC3161 - Internet X.509 Public Key Infrastructure Time-Stamp Protocol (TSP),* August 2001.
[11] IETF, *RFC5280 - Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile,* May 2008.
[12] ITU-T, *Rec. X.509 - Information technology - Open systems,* November 2008.
[13] Official Gazette, *Elektronik İmza ile İlgili Süreçlere ve Teknik Kriterlere İlişkin Tebliğ, 25692,* 06 January 2005.
[14] Official Gazette, *Elektronik İmza Kanununun Uygulanmasına İlişkin Usul ve Esaslar Hakkında Yönetmelik, 25692,* 06 January 2005.
[15] Official Gazette, *E-signature law No. 5070, 25355,* January 2004.
[16] Official Gazette, *Kamu Sertifikasyon Hizmetlerine İlişkin Usul ve Esaslar, 2006/13,* 19 April 2006.
[17] Official Journal, *Directive 1999/93/EC of The European Parliement and of The Council,* December 1999.

[9]The Scientific and Technological Research Council of Turkey
[10]Informatics and Information Security Research Center
[11]Government Certification Authority