SECURITY ANALYSIS OF ELECTRONIC SIGNATURE APPLICATIONS AND TEST SUITE STUDY

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF APPLIED MATHEMATICS
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

TAMER ERGUN

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF DOCTOR OF PHILOSOPHY
IN
CRYPTOGRAPHY

SEPTEMBER 2013

Approval of the thesis:

SECURITY ANALYSIS OF ELECTRONIC SIGNATURE APPLICATIONS AND TEST SUITE STUDY

submitted by **TAMER ERGUN** in partial fulfillment of the requirements for the degree of **Doctor of Philosophy in Department of Cryptography, Middle East Technical University** by,

Prof. Dr. Bülent Karasözen _____
Director, Graduate School of **Applied Mathematics**

Prof. Dr. Ferruh Özbudak _____
Head of Department, **Cryptography**

Prof. Dr. Ferruh Özbudak _____
Supervisor, **Department of Mathematics**

Examining Committee Members:

Prof. Dr. Ersan Akyıldız (Head of the examining com.) _____
Department of Mathematics, METU

Prof. Dr. Ferruh Özbudak (Supervisor) _____
Department of Mathematics, METU

Assoc.Prof. Dr. Ali Doğanaksoy _____
Department of Mathematics, METU

Assist. Prof. Dr. Ömer Küçüksakallı _____
Department of Mathematics, METU

Assist. Prof. Dr. Sedat Akleylek _____
Department of Computer Engineering, OMÜ

Date: _____

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last Name: TAMER ERGUN

Signature :

ABSTRACT

SECURITY ANALYSIS OF ELECTRONIC SIGNATURE APPLICATIONS AND TEST SUITE STUDY

ERGUN, TAMER

Ph.D., Department of Cryptography

Supervisor : Prof. Dr. Ferruh Özbudak

September 2013, 75 pages

Digital signature technology is used widely for security and trust in electronic business and communications. Nowadays it becomes commonly used especially in government agencies. From this point of view, it is crucial to implement correct applications to create and verify digital signatures. CEN (European Commitee for Standardization) has introduced the security requirements for signature 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 has to be hardly tested and the responses of the application to a wide range of wrong scenarios have to be well analyzed. In our thesis 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 set of certificates and signature files created for this aim. We also aimed to solve some security and efficiency problems derived from validation processes of revocation datas.

Keywords: Digital Signature, PKI, X509, CMS, Time Stamp

ÖZ

ELEKTRONİK İMZA UYGULAMALARI GÜVENLİK ANALİZİ VE TEST SUİT ÇALIŞMASI

ERGUN, TAMER
Doktora, Kriptografi Bölümü
Tez Yöneticisi : Prof. Dr. Ferruh Özbudak

Eylül 2013, 75 sayfa

Dijital imza teknolojisi, elektronik ticaret ve haberleşmede yaygın olarak kullanılmaktadır. Yakın zaman içerisinde, devlet kurumlarının da elektronik imzayı sıklıkla kullanmaya başlamasıyla, güvenli dijital imzalama ve doğrulama uygulamaları oluşturmanın önemi daha da artmıştır. CEN (Avrupa Standardizasyon Komitesi) imza oluşturma ve doğrulama uygulamalarının güvenlik gereksinimleri üzerine çalışmalar yapmıştır fakat imza uygulamalarının güvenilirliliğini test etmek adına dizayn edilmiş PKI modeli veya uygulanabilir bir test paketi bulunmamaktadır. Tez çalışmamızda bu eksikliği gidermek için çalıştık ve uygulamaların imza oluşturma ve doğrulama alanlarında güvenlik testlerini yapacak PKI modelini ve bu modelin gerçeklenmesiyle elde edilen E-İmza Test Süitini oluşturduk. E-İmza Test Süiti, içerisinde birçok sertifika ve imzalı dosyayı barındıran, belirtilen amaç için üretilmiş ve PKI altyapısında oluşabilecek hatalı durumların büyük çoğunu kapsayan test paketidir. Bu tezde ayrıca sertifika ve imza doğrulama konularındaki bir takım güvenlik ve verimlilik problemlerine de çözümler ürettik.

Anahtar Kelimeler: Digital Signature, PKI, X509, CMS, Time Stamp

This thesis is dedicated to my father Mehmet Ergun and my whole family who have been a great source of motivation and inspiration. Also, this thesis is dedicated to Serap Ergun and Ayşe Naz Ergun who believe me to accomplish my thesis work. Finally, this thesis is dedicated to Ferda Topcan who has supported me all the way since the beginning of my studies.

ACKNOWLEDGMENTS

Foremost, I would like to express my best regards to my thesis advisor Prof.Dr. Ferruh Özbudak for his invaluable guidance.

Besides my advisor, I would like to thank Assoc.Prof.Dr. Ali Doğanaksoy for his inspiration and trust to my thesis.

My sincere thanks also goes to my supervisors and colleagues in TUBITAK-BILGEM-KAMUSM for their great support.

I would like to express my sincere gratitude to Ferda Topcan for the continuous support of my Ph.D study and research, for her motivation, enthusiasm, and immense knowledge. Her support helped me in all the time of research and writing of this thesis.

Last but not the least, I would like to express my heart-felt gratitude to my family. They have aided and encouraged me throughout this endeavor. Special thanks to my dear wife Serap Ergun and my little lovely daughter Ayşe Naz Ergun for their patience and providing me a sweet home throughout my study.

PREFACE

This thesis work will be a guideline for all researchers related with PKI and signature application developers to test their works by means of accuracy and compatibility to international standards. Researchers either use the proposed PKI model and implement it with their country specific extensions or directly use the E-Signature Test Suite to test their signature applications both in signature creation and verification.

TABLE OF CONTENTS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

CHAPTER 1

INTRODUCTION

Digital signature technology is widely used for security and trust in electronic business and communications. For secure and reliable electronic transactions, some core concepts such as authentication, integrity, confidentiality and non-repudiation have to be satisfied. Digital signature technology satisfies these concepts except confidentiality and, if needed, can be fulfilled by generally combining with a symmetric algorithm.

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). The system binding cryptography with identity and serving a reliable enviroment is defined as PKI. PKI needs various components to satisfy a reliable environment. These components will be covered in chapter 2. PKI has a chain structure and trusting a ring depends on trusting other rings of the chain. Since commonly used cryptographic algorithms for digital signature are safe enough, one can try to break this chain structure from the weekest ring. Therefore a PKI system must strongly built and has a provable security.

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 [12] 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 hierarchy 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 components. 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 [7].

In our thesis 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.

For analyzing signature creation applications, the part of the test suite defined above is sufficient. For analyzing signature verification applications, further work was necessary. European countries use ETSI standards as signature formats and there exists 10 types of advanced signature types from CAdES-BES to CAdES-ESA [7]. Details of these signature types are defined in section 2.4. Therefore to satisfy demands on each type of signature, we created wrong scenarios for 10 signature types. Each advanced signature format in the sequence has at least greater or equal scenarios then the previous one. Additionally, since application developers may implement attached or detached types, we created all types of signatures both in attached and detached forms. In signature verification part, most of the scenarios are derived from the signature creation suit. But beside these, scenarios coming from Type-1, Type-2 and ES-A specific time stamps, defined in 2.3 or the compatibility of certificate values - references and

revocation values - references are also added.

After this brief introduction, let us give you the structure of our thesis work. In section two, we will study PKI and cover some necessary PKI concepts, in section three, the theory and the design principle of Test Suite will be covered, in section four, we will analyze the implementation details of Test Suite, in section five, we will examine the results of the Test Suite and give some efficient implementation tips and conclude the work in section six.

CHAPTER 2

PRELIMINARIES

In this chapter we will analysis public key infrastructure deeply and define each component of PKI. Before beginning to define PKI concepts, we should focus on the term PKI. In terms of cryptography, signing a message is the encryption of digest of that message with the private key of the signer. Also verifying a signed message is just to check the equivalence of the message digests obtained by applying public key of the signer to the signed data and the one obtained by digesting the message to be signed. For cryptographic point of view these definitions are enough to define digital signature. Since the main point is digital signature, cryptography does not deal with how the signer sends his public key to the verifier and identifies himself. Besides we know that public-private key pairs must have an expiration date with respect to the length of modulus and this issue is also not a primary problem for cryptography. We can give more examples as the above and this shows that the operational dimension of digital signature has more than an asymmetric cryptographic algorithm. Therefore all necessary mechanisms and policies needed to securely handle digital signature process is defined as PKI. In the rest of this chapter, we will introduce the components of PKI.

2.1 Certificate

As we mentioned before, it is important for the verifier to identify the signer of the signature. Verifier wants to trust that the public key used to verify the signature belongs to the signer that he assumes. In order to remove this ambiguity, certificate structure is created.

Certificate is a digitally signed data, binding identity of a signer with his public key.

Since certificates serve for identifying signer, everyone can create a certificate with the public key of the actual signer and the identities of himself as his name, surname and identification number. Therefore an additional step is required to ensure that this certificate belongs to the actual signer and no one can change any information of the certificate to cheat the verifier. This additional step is a trusted certification authority (CA) who issues the certificate of the actual signer and signs the certificate with his private key to ensure that no one can change any data of the certificate. Thus PKI has a new element named as certificate authority.

Certificate Authority An authority authorized legally to create and assign public-key certificates.

There are many kinds of certificates in PKI and we will define them when we need but let us introduce an important kind of certificate known as the qualified certificate.

Qualified Certificate A certificate which meets the requirements in [18], [9], [11].

Qualified certificates are important because in some countries as Turkey, a soft document signed by a qualified certificate has the same legal responsibilities with its wet signed hard copy [9], [37]. More information about certificates can be found in [3], [10], [11].

A certificate covers all the necessary informations needed to interact with the PKI services and proving its validity. Expiration date is an important information because as we mentioned before, public-private key pairs have a confidence time interval with respect to its key length. Therefore a certificate used to sign a document after expiration date is considered to be invalid. Beyond expired certificates, we can count lots of reasons that make a certificate invalid and among them, another famous reason is revocation status. As credit cards, an end entity may need to revoke his digital certificate because another end entity may have the probability to use his private key. This case implies that the revocation status of a certificate may change during its lifetime and the PKI system must support this issue and it does. In case of a revocation request, the related certificate authority revokes the requested certificate and publishes this information in two ways. One of them is CRL and the other one is OCSP.

2.2 Revocation Values

Revocation issue is another important concept in PKI and certificate authorities handle this issue in two ways. Let us begin with Certificate Revocation Lists (CRL).

Certificate Revocation List is an electronic data including all the revoked certificates with serial number and revocation date.

Therefore CRL is a dynamic list and every revoked certificate is inserted to this list by its issuer CA. In this case a question arises such that how PKI handle with the size of these lists as they are getting larger with time. One should keep in mind that, growing is in control because a revoked certificate listed in CRL is deleted when it is also expired. Since the reason of using a revocation data is to ensure that if a certificate is revoked and so invalid, there is no need to check the revocation status of a certificate when it is expired because expiration is enough to decide that the certificate is invalid.

Another issue about CRLs are, as all the evidences used in PKI, CRL is a signed data. As we mentioned before, one can construct a copy of a valid CRL and may change the status of some certificates to mislead PKI users but never signs it with the trusted part's private key. In this case the trusted part is the issuer of the certificate. As mentioned in [3], a CRL including a certificate must be signed by the issuer of that certificate.

CRL is a necessary component of PKI but has some disadvantages. A CRL has a lifetime bounded with thisUpdate, nextUpdate dates and a CRL is refreshed when we reach the nextUpdate time. This means a certificate revoked between thisUpdate and nextUpdate date is published in the newer CRL released before nextUpdate date. Therefore CRL is not proper for online transactions. Another disadvantage of CRL is its size. Since all the revoked but not expired certificates are listed, its size may become a problem. This is because PKI has an alternative way of publishing revoked certificates which is known as OCSP.

Online Certificate Status Protokol is a response data containing the revocation status of a specific certificate.

This definition differs OCSP from CRL as CRL contains all the revoked certificates where as OCSP includes the status of an individual certificate. It is clear that OCSP has an advantage

of a smaller size over CRL. Besides, OCSP servers creates the response to a request at request time and so serves the freshest status about the requested certificate. Therefore for online transactions, OCSP is recommended.

As we stated for CRL, OCSP is also a signed data and the signer of the response is not the issuer of the certificate but a specific certificate issued by the issuer of the requested certificate, known as **OCSP Certificate**. The rule for CRL signer of a certificate is similar for the OCSP such that the signer of an OCSP response for a specific certificate must be the OCSP certificate issued by the issuer of that specific certificate. More information on OCSP can be found in [5].

Using OCSP servers seems a better choice than using CRL but in some cases CRL has advantages over OCSP. First of all, using OCSP servers requires being online all the time but for off-line systems one can not use OCSP service if the sub-root authority does not belong to the off-line system. In this case CRL is useful and also same CRL can be used for the certificates within the thisUpdate and nextUpdate period but for OCSP, newer requests must send to OCSP server at each time. This results show that each serving method has advantages and disadvantages and the choice should be made with respect to the implementation policy.

It is now clear that the revocation status of a certificate is a reason to accept it as valid or invalid but this issue is very releated with the exact time. Since revocation time has an exact value, the time we investigate the revocation status of a certificate is crucial. From this point of view, PKI has a new concept known as time stamp.

2.3 Time Stamp

In section 2.2 we specified the role of revocation status of a certificate and it is clear that when we interpret a certificate is revoked, it is necessary to compare the actual date and the date that the certificate is revoked. In order to make this comparison reliable for everyone, the time we reference must be trusted. This implies that, PKI must have a component serving trusted time. This component is known as time stamp.

Time Stamp is a token containing a specific information and a date such that, that specific information exists before the date in token.

There are two reasons that make a time stamp trusted. One of them is the accuracy of the date that the time stamp server serves and the other one is the detectability of time stamp response against editions. Second one means, if someone edits a response and changes the date, this situation must be detectable. PKI handles this necessity as it does for CRL, OCSP and others. Time stamp token is a signed data issued by a trusted authority.

Time Stamp Certificate is the certificate authorized to sign time stamp tokens.

An ambiguity about time stamp is, it does not indicate the date that an event is happened but it indicates that an event is happened before the date in time stamp. Therefore to minimize the difference, one should time stamp the data as soon as possible. Time stamp is divided into 5 types with respect to its usage in signing process [7],[26].

Signature Time Stamp is a time stamp token applied over signature value of a signer in signer info field. This type of time stamp is used to identify that the signature is created before the time in time stamp.

Content Time Stamp is a time stamp token applied over content value of a signed data field. This type of time stamp is used to identify that the content is created before the time in time stamp.

CAdES-C Time Stamp is a time stamp token applied over CAdES-C signature. This type of time stamp is used to protect against CA key compromise.

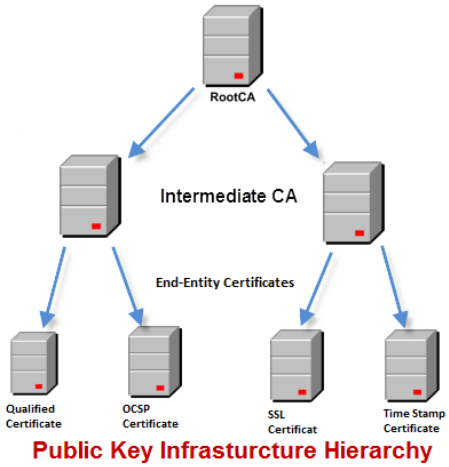
Time Stamped Certs CRLs References is a time stamp token applied over certificate and revocation references. This type of time stamp is used to protect against CA key compromise.

Archive Time Stamp is a time stamp token applied over many fields of signed data. This type of time stamp is used to protect week algorithms used in signature file and to protect against CA key compromise.

More information about time stamp can be found in [4].

Lastly with time stamp, we covered the necessary definitions about PKI and also the signature creation part of our work. In figure 2.1, an example of a hierarchical two-level chain structure is shown.

Figure 2.1: CA Hierarchy



2.4 Signature Types

We have some more definitions required for the signature verification part of our thesis work. For signature verification part we created numerous signature files in 10 signature types defined by ETSI. ETSI, the European Telecommunications Standards Institute, is a standardization organization in the telecommunications industry and also specifices European standards for e-signature. When we were creating signature files, we obeyed the e-signature standards of ETSI [19] and created all types of signatures defined by ETSI.

Signature Types:

BES namely the basic electronic signature containing signers document, some signed attributes and the signature.

EPES is a BES sign with signature policy in signed attributes.

EST is BES with signature time stamp.

ESC is EST with the references of certificates and revocation values.

ESX Type 1 is ESC with CAdES-C time stamp.

ESX Type 2 is ESC with time stamped certs CRLs references.

ESXL is ESC with the values of certificates and revocation values.

ESXL Type 1 is ESXL with CAdES-C time stamp.

ESXL Type 2 is ESXL with time stamped certs CRLs references.

ESA is ESXL with archive time stamp.

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.

In this section we dealed with necessary concepts in ETSI signature types and for more information and technical details, we can refer [7] for CMS Advanced Electronic Signatures (CAdES) and [8] for XML Advanced Electronic Signatures (XAdES).

CHAPTER 3

TEST SUITE DESIGN THEORY

In this chapter we will introduce our PKI model designed for test suite and define the error cases, nodes of PKI tree, both in signature creation and signature verification parts.

3.1 Design Philosophy

Our design philosophy depends on constructing a PKI model including all the possible error cases in terms of end-entity certificates, subroot certificates, root certificates, time stamp servers, OCSP servers and certificate revocation lists. While creating these scenarios, we paid attention on creating each error case individually. Let us consider that we have an aim to test the response of an application to a specific invalid OCSP server. In this case, we are creating the targeted invalid OCSP responder and an end-entity certificate. The characteristic of this certificate is, it has the targeted OCSP server address as OCSP address in authority information access [3] and has no CRL address. Our aim is to force the application to use the targeted OCSP server while validating the end-entity certificate. Therefore we are expecting that the application catches the invalid OCSP response and warn the user about this situation. Beside this aim, we are creating the defined problem not only for end- entity certificates but also for subroot and root certificates. As we were creating these error scenarios, we obeyed the PKI restrictions about its components at the same time.

Beyond the aim of testing the relations between signature creation applications and PKI components, we also constructed the possible error cases for signature files. In section 2.4, we defined the 10 types of advanced electronic signatures released by ETSI. We constructed the error cases for signature files in 10 signature types to test the accuracy of signature verification

applications. Since each signature type has further properties then the previous one, we have more error scenarios as from signature type BES to ESA. Big portion of error cases designed for signature files are derived from the constructed problematic PKI model but we constructed further error cases related with properties of signature types.

In the next two sections, we will classify and describe the whole error scenarios both in signature creation and verification parts.

3.2 Signature Creation Test Scenarios

In this section we will define error cases designed to construct the PKI model and to test signature creation applications. While we are designing error cases we dealed with each PKI component seperately. Throughout this section we will analyze error cases with this manner. The first part of signature creation suite is about end-entity certificates.

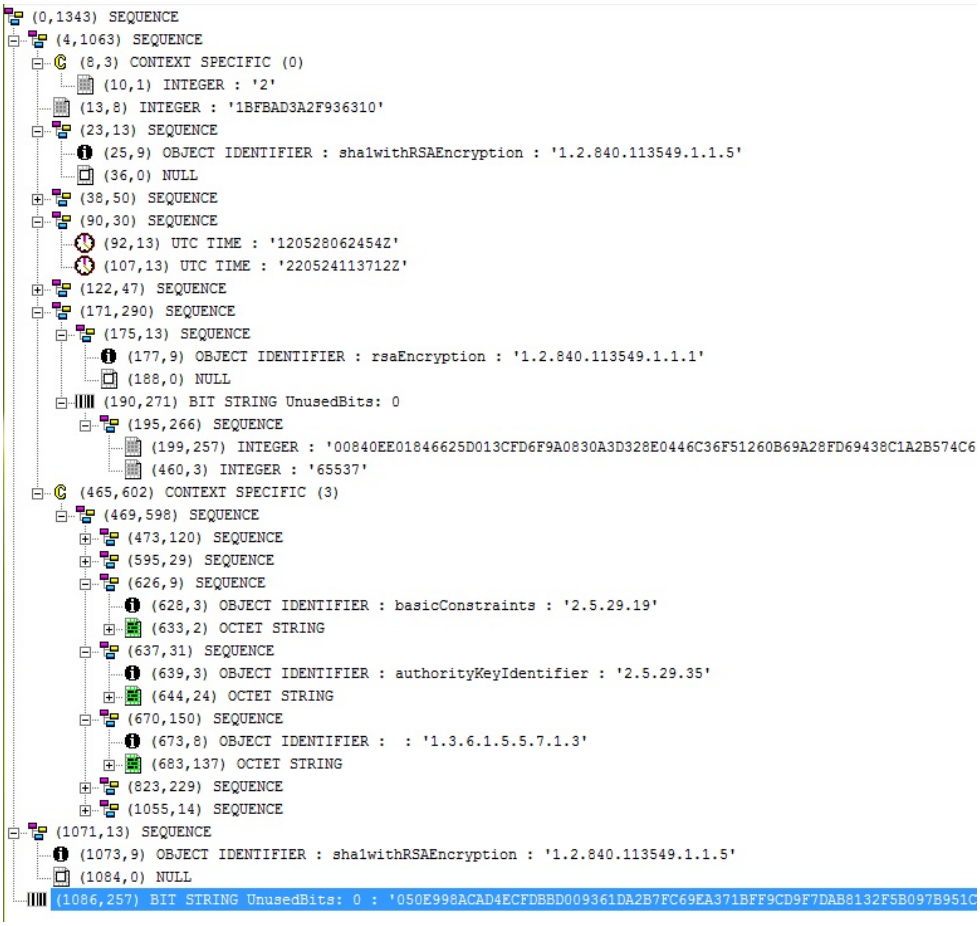
3.2.1 End-Entity Certificate Validation Scenarios

This subsection is related with the error cases designed directly for end-entity certificates. End-entiy certificates can be considered invalid for numerous cases. In case of signing with these problematic certificates, signature creation applications have to detect the problem and warn client about the status of the certificate. As we are stating the error cases for end-entity certificates, it is better to group them with respect to the source of the problem and the first error case we consider will be the signature of an end-entity certificate.

3.2.1.1 Certificate Signature Check

We mentioned before that every evidence used in PKI is signed by a trusted authority. This issue is very important because if the evidence is not signed, one can edit it and change the specific datas. Also it is clear that the signer of the data has to be trusted. From this point of view, a reliable signature creation application has to check the signature of a certificate in signing process. To test the stated capability of signature creation application, we designed a signature forged signer certificate and added this node to our PKI tree.

Figure 3.1: Certificate Signature ASN.1



Digital certificates are encoded in abstract syntax notation one (ASN.1) [21],[22],[24] , which is a standard for representing data in telecommunication. In figure 3.1 ASN.1 representation of a certificate is shown. In this figure, the highlighted BIT STRING field is the signature of the certificate. Therefore in order to create a signature forged certificate, we created a valid certificate and then edited its signature field.

3.2.1.2 Certificate Expiration Check

In cryptography every public-private key pair has a lifetime. The period of the lifetime depends on the length of the key for the encryption algorithm. For RSA, key is the modulus and for weak keys, one can reveal the private component. Also if one can get the private component, he can generate signatures with the name of the owner of the public-private key pair. Since we defined certificate as a digital data binding public key of an end-entity with his id, it must have a lifetime. A certificate consumed its lifetime is known as expired certificate and a signature application has to check expiration status of signer certificate. If it is expired then the application must treat it as invalid, cancel the signing process and warn the client about this issue. The lifetime of a certificate is between notBefore and notAfter [3] dates of the certificate. In figure 3.2, an ASN.1 representation of an expired certificate is shown. The highlighted field of the certificate is the notAfter field and in validation time it must be greater than the actual time. From this point of view, to test this capability of the signature creation application, we designed an expired certificate and added this node to PKI tree.

3.2.1.3 Qualified Certificate Checks

Qualified certificates, as defined in 2.1 have a major role in PKI. This is because, a soft document signed by a qualified certificate has the same legal responsibilities with its wet signed hard copy in some European countries such as Turkey. Therefore it is crucial for the signature creation application to detect if the signer certificate is qualified or not. Qualifing properties that make a certificate qualified is defined in [9], [11], [13], [16], [17] and [18]. In order to test signature creation applications, we designed five types of non-qualified certificates such that in each certificate, all the qualifing properties except one are satisfied. These scenarios are defined below.

Figure 3.2: Expired Certificate

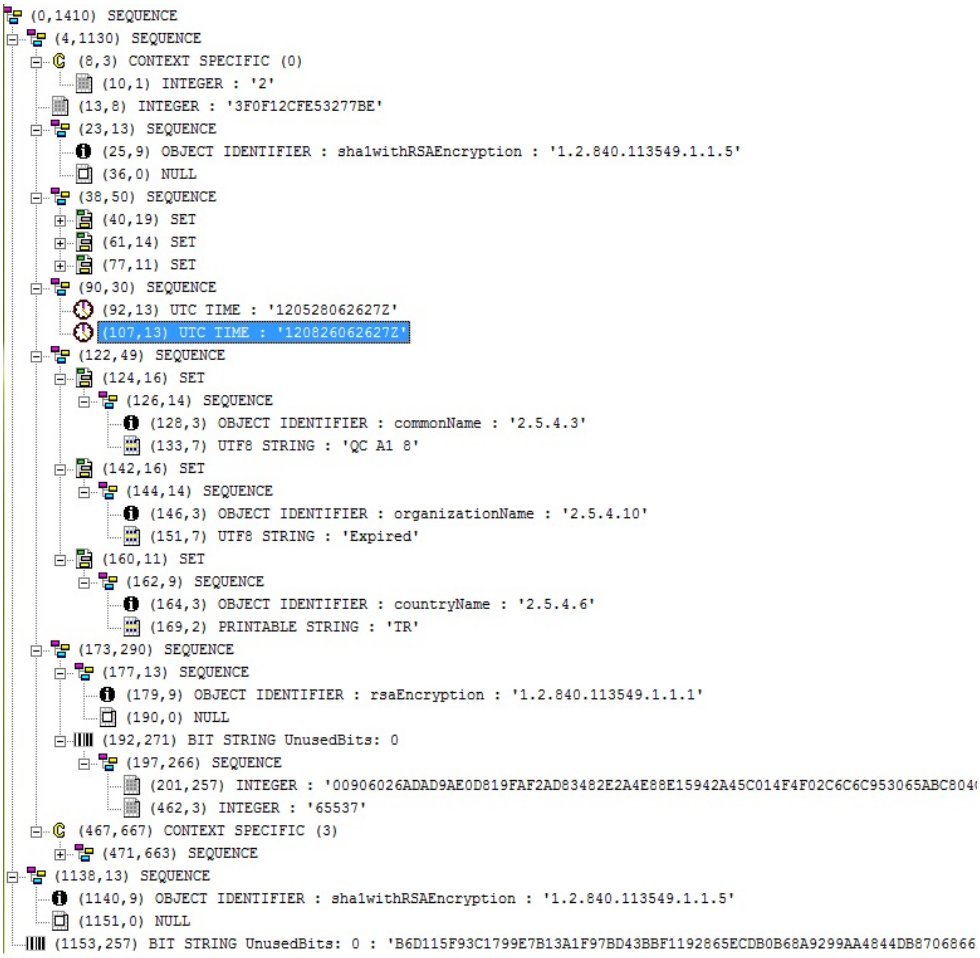
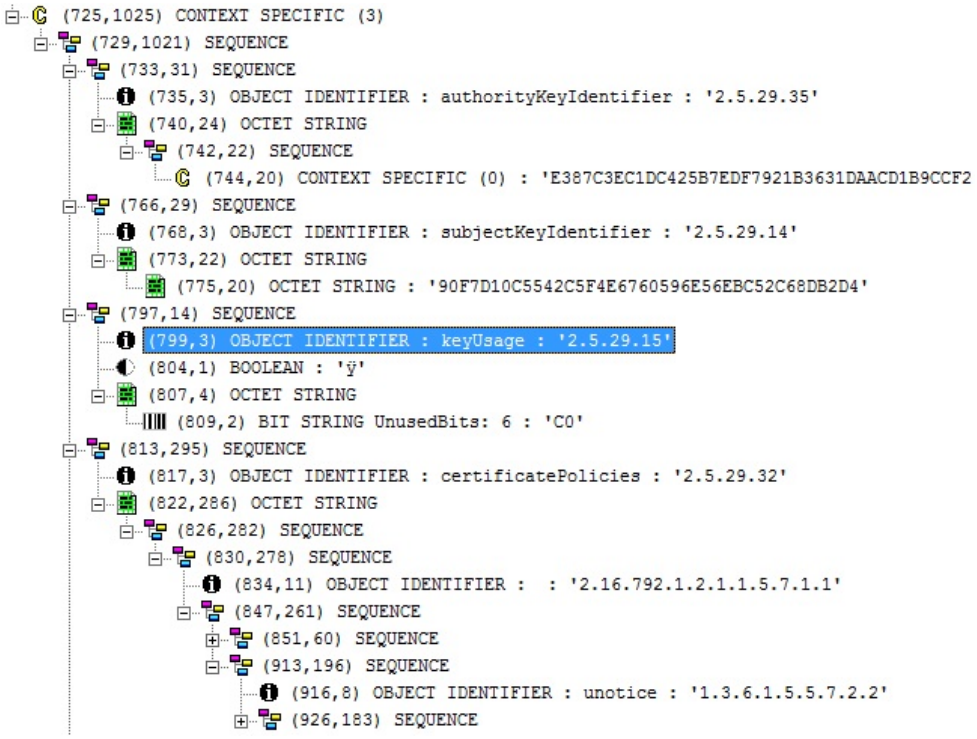


Figure 3.3: Key Usage

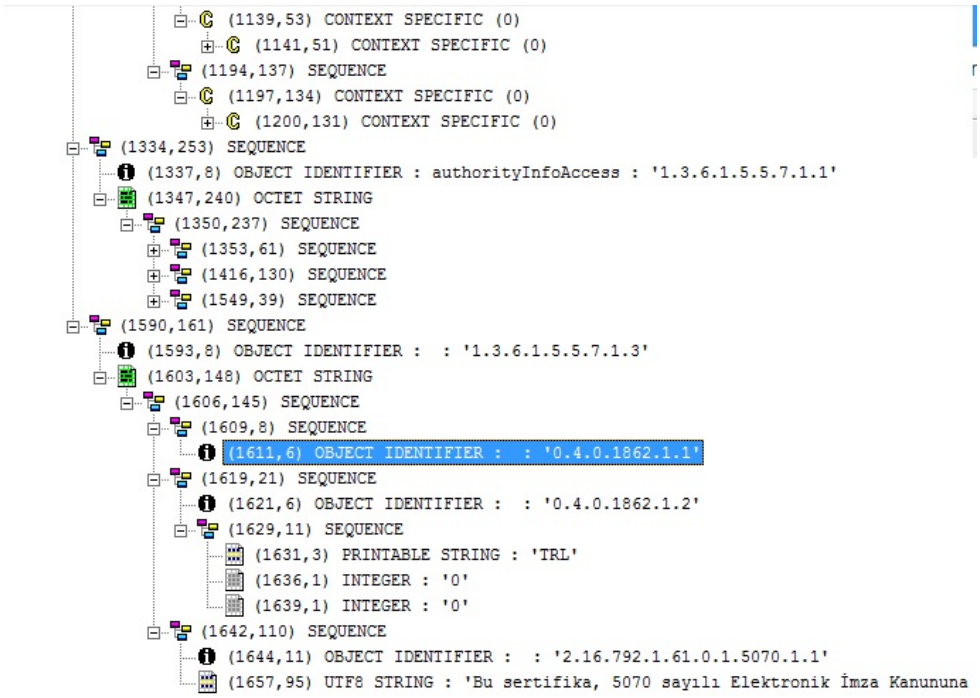


Non-Repudiation Absence: nonRepudiation is a MUST field in a qualified certificate and is included in the critical field of a certificate, known as key usage field [3]. In figure 3.3, an ASN.1 part of a qualified certificate including key usage field is presented. When this field is decoded according to the ASN.1 rules of key usage [3], one can observe that it includes nonRepudiation field. In this scenario, a qualified certificate without nonRepudiation field is designed. We aim to determine if the application checks signer certificate's nonRepudiation field to accept it as a qualified certificate.

ETSI QC Statement ID Absence: ETSI QC Statement ID (0.4.0.1862.1.1) is a MUST field of a qualified certificate [11] and is included in qualified certificate statements field with id (1.3.6.1.5.5.7.1.3) [3]. In figure 3.4, an ASN.1 part of a qualified certificate including qualified certificate statements field is presented. The highlighted field included in qc statements field is ETSI QC Statement ID (0.4.0.1862.1.1). In this scenario, a qualified certificate without ETSI QC Statement ID is designed. We aim to determine if the application checks signer certificate's ETSI QC Statement ID field to accept it as a qualified certificate.

ICTA QC Statement ID Absence: ICTA is the abbreviation of Information and Communi-

Figure 3.4: ETSI OID



cations Technologies Authority in Turkey (BTK). As stated in [9], ICTA QC Statement ID is a MUST field for a certificate to be qualified certificate. This field is encoded under qc statements field with OID (2.16.792.1.2.1.1.5070.7.1.1). In figure 3.5, the highlighted field is the ICTA OID. In this scenario, a qualified certificate without ICTA QC Statement ID is designed. We aim to determine if the application checks signer certificate's ICTA QC Statement ID field to accept it as a qualified certificate.

ICTA QC Statement Info Absence: Another MUST field necessary for being qualified certificate is ICTA QC Statement Info [9]. In this info notice, it is written that, "Bu sertifika, 5070 sayılı Elektronik İmza Kanununa göre nitelikli elektronik sertifikadır." means "This certificate is a qualified certificate according to e-signature law numbered 5070." In figure 3.6, the highlighted field is the ICTA QC Statement Info. In this scenario, a qualified certificate without ICTA QC Statement Info is designed. We aim to determine if the application checks signer certificate's ICTA QC Statement Info field to accept it as a qualified certificate.

CP User Notice Statement Absence: Certificate policies is an existing field in certificate which indicates the policy under which the certificate has been issued and the purposes for which the certificate may be used. This extension contains a sequence of one

Figure 3.5: ICTA QC Statement ID

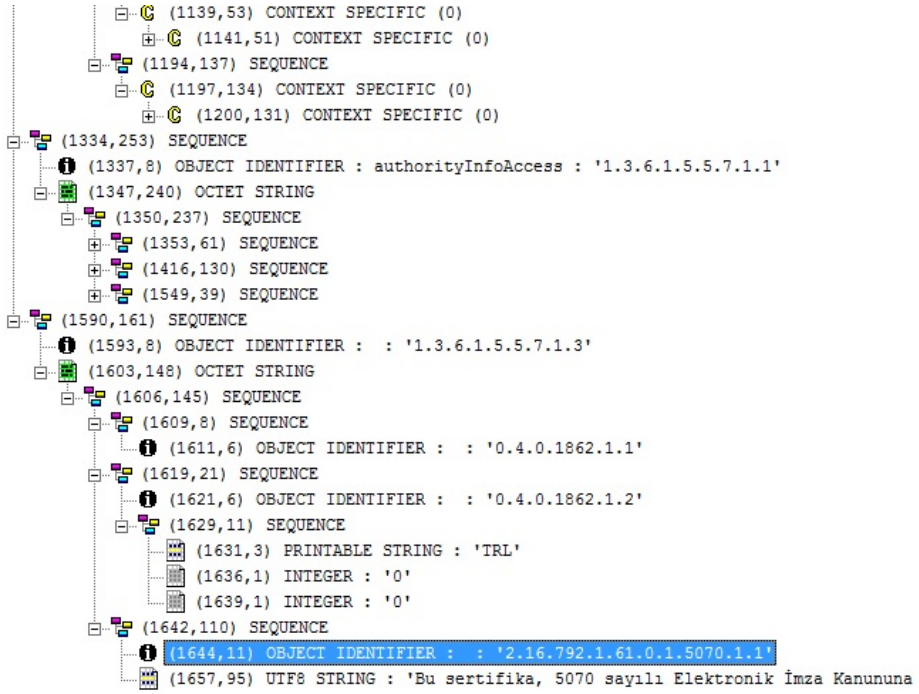
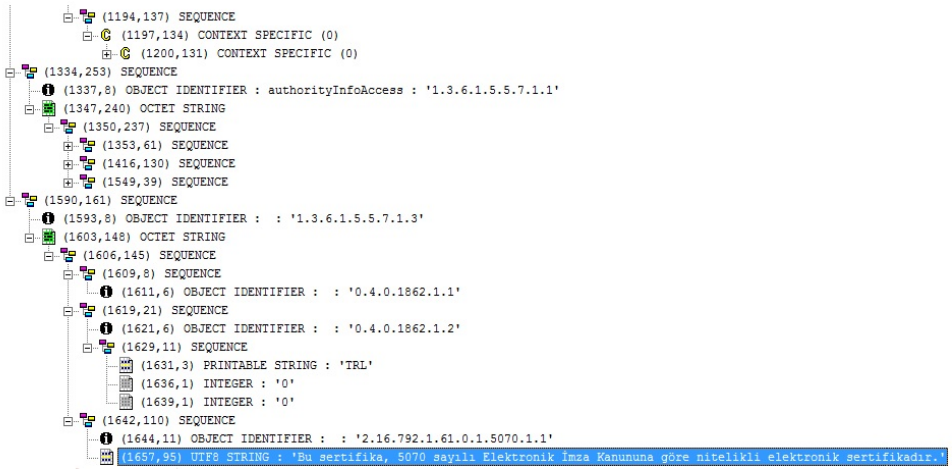


Figure 3.6: ICTA QC Statement Info



or more policy information terms, each of which consists of an object identifier (OID) and optional qualifiers. According to [9], the same statement as ICTA QC Statement Info MUST exist in certificate policies field with object id (2.16.792.1.2.1.1.5.7.1.1). This statement is called as CP User Notice Statement. In this scenario, a qualified certificate without CP User Notice Statement Info is designed. We aim to determine if the application checks signer certificate's CP User Notice Statement field to accept it as a qualified certificate.

The last error case caused by signer certificate directly is the revocation status of a signer certificate.

3.2.1.4 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. In order to prevent the signing process because of revocation, signature creation application must enable its clients to be informed about the revocation status of an issued certificate from one of its certificate authorities. PKI handles this issue in two ways. One of them is CRL and the other one is OCSP which are defined in 2.2. Clients use these services by using access points stated in certificates. CRL for a specific certificate can be obtained by using CRL Distribution Points (CDP) field of the certificate and similarly, OCSP response for a certificate can be obtained by using Authority Information Access (AIA) 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 scenarios in which there exists a revoked certificate. 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 aimed to determine if the signature creation application checks certificates' revocation info accurately in both ways.

The stated wrong scenarios in this subsection is about the error cases of an end entity certificate and in case of signing operation with one of these certificates, a signature creation application must not allow signing. In the following subsection we will concern error cases caused by not the signer certificate itself but the issuer certificates such as subroot and root

certificates.

3.2.2 Issuer Certificates Validation Scenarios

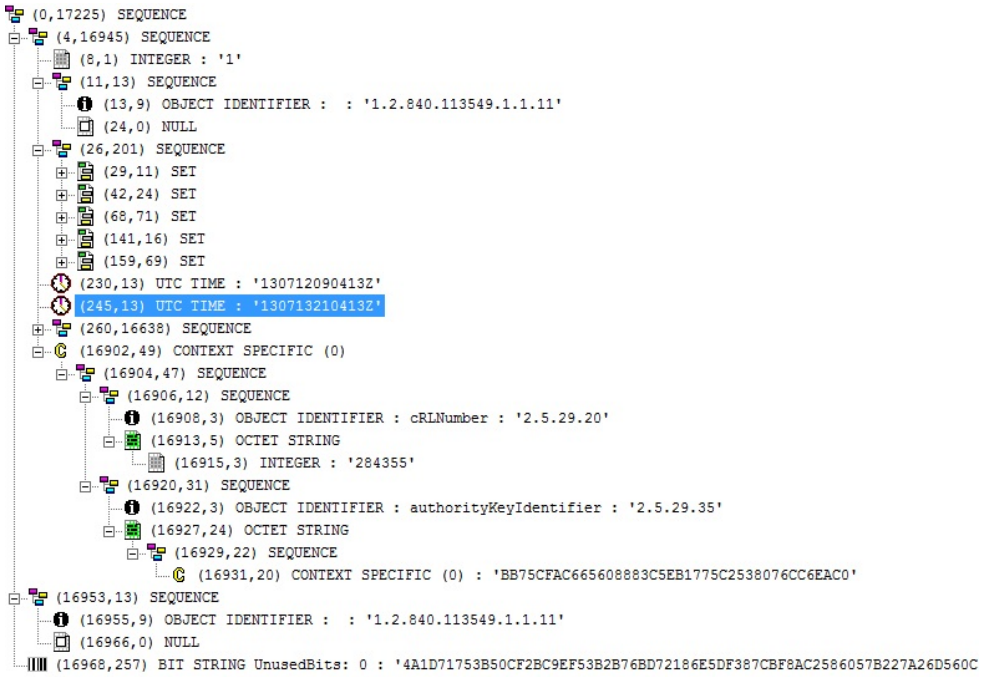
So far we defined wrong scenarios related with end entity certificates and in daily life, most of the error cases occured in PKI environment is caused by end entity certificates but when we design this suite we take care of future possible attacks on PKI. In future, when the PKI awareness is increased, attackers will try to create fake subroot and root certificates and mislead the system. Since PKI mechanism can handle with these problems, it is enough to design and add these scenarios to our PKI model and test signature creation applications with these scenarios. Most of the stated wrong scenarios designed for end entity certificates can be applied to the subCA and RootCA certificates. In the case where just end entity certificates are concerned, we have a tree path with a valid root, valid subroot and defined end entity certificates but while creating problematic subroots and roots, our PKI model became enlarged. The details of problematic issuer certificates will be covered in chapter 4.

In the following section, beyond the occurable error cases caused by end entity, subroot and root certificates, we will concern error cases related with revocation values, defined in 2.2, which are used to validate these certificates.

3.2.3 Revocation Data Validation Scenarios

Revocation data is an important concept in PKI. It is important because it is used to check whether a certificate is revoked or not and so is an evidence for certificate validation. From this point of view, it is desirable for a third party who wants to mislead the creator or verifier of a signature. Therefore a signature creation or a validation application has to be careful while processing revocation datas and ensures that these datas are acceptable and trusted. As stated before, in PKI, revocation datas are published in two ways, CRL and OCSP. In this section we will discuss criterias that make a revocation data invalid and embed these criterias into our PKI model to test signature creation applications from this point of view. Let us start with CRLs.

Figure 3.7: Expired CRL



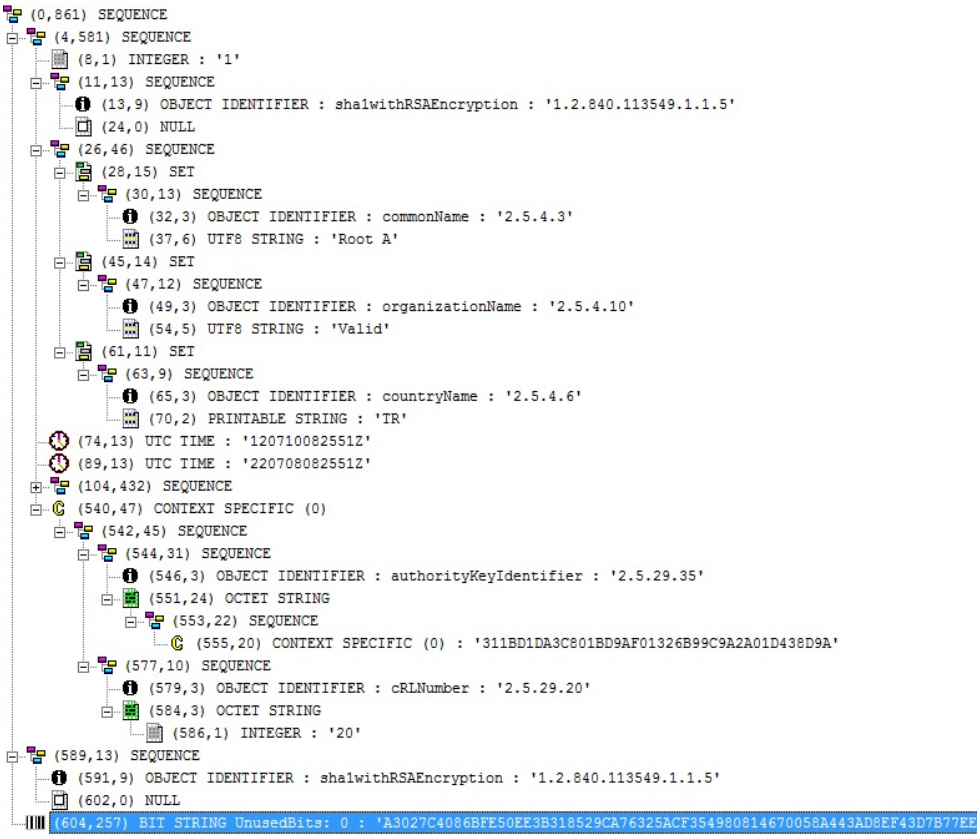
3.2.3.1 CRL Validation Checks

CRL is a kind of validation data which serves all the revoked certificates issued by a certificate authority. Error cases for CRLs can be grouped in two category as expired CRL and signature forged CRL.

Expired CRL: As certificates, CRL files have also life time. This life time is restricted between thisUpdate and nextUpdate [3] fields of CRL file. nextUpdate field indicates the date by which the freshest CRL will be issued. Therefore a CRL whose nextUpdate value is smaller than the validation time can not be used as a valid CRL and is named as expired CRL. A signature creation application must interpret such a CRL as invalid because since validation time is greater than nextUpdate, there is a fresh CRL in validation time and signer certificate may be revoked between the nextUpdate of the past CRL and the validation time. It is clear that this information is published in the freshest CRL. In figure 3.7, ASN.1 representation of an expired CRL is shown. The highlighted field is the nextUpdate field of CRL.

In this scenario an expired CRL is designed and aim is to determine if the application compares nextUpdate field with validation time.

Figure 3.8: Signature Forged CRL

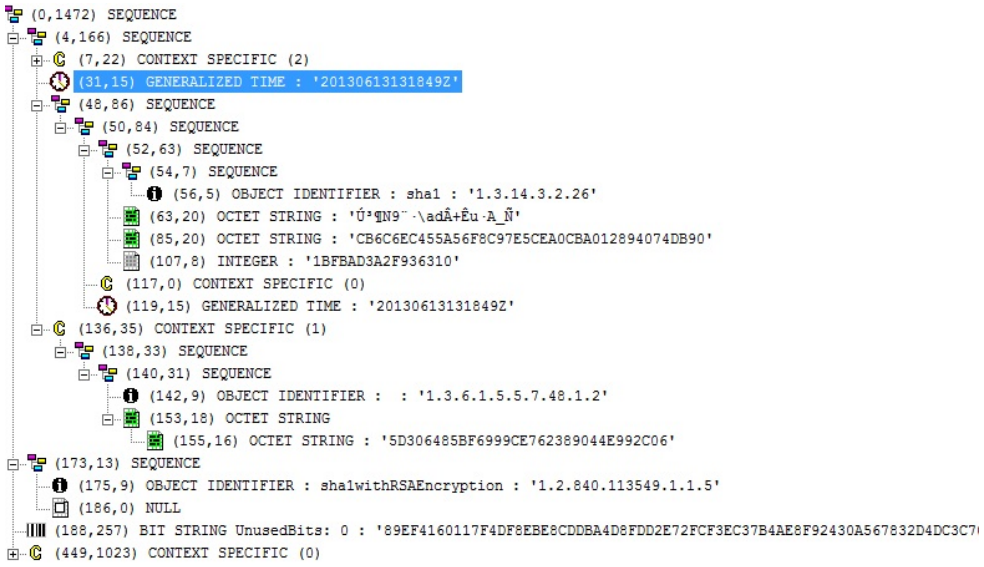


Signature Forged CRL: We stated before that all the evidences used in PKI are signed by trusted authorities and so CRL is also a signed data. Signer of the CRL, is the issuer of the certificates that the CRL gives revocation status information about. Signing CRL is a protection against unauthorized insertion and deletion. One may want to mislead the signature creation application and wants to indicate a valid certificate is revoked or a revoked certificate is valid. In this situation, signature creation application has to verify the signature of CRL against editions. In figure 3.8, ASN.1 representation of a signature forged CRL is represented. The highlighted field is the signature of the CRL.

In this scenario a signature forged CRL file is designed and aim is to determine if the application verifies the signature of the CRL file.

A client may use OCSP instead of CRL and therefore it is also important to design invalid situations for OCSP servers.

Figure 3.9: Expired OCSP Response



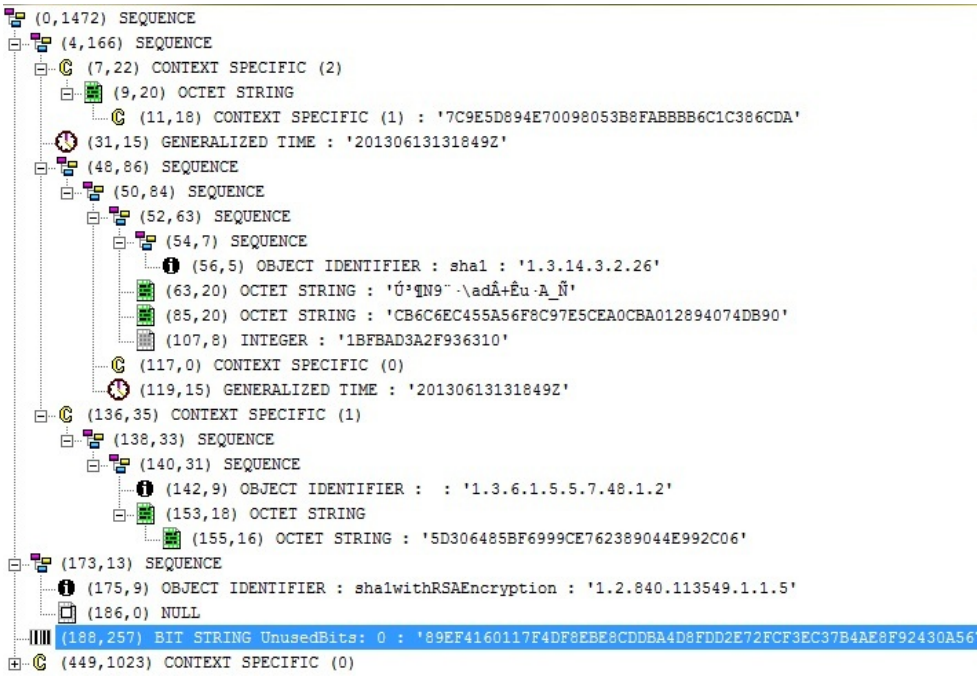
3.2.3.2 OCSP Validation Checks

OCSP is another way of serving revocation data. Similar with CRL, error cases related with OCSP responses have to be discussed and designed to test signature creation applications. OCSP has further error cases than CRL because the issuer of an OCSP response is a certificate, named OCSP certificate and scenarios coming from the status of OCSP certificate is added to CRL scenarios. In this section we will discuss OCSP server error cases splitted into 5 groups.

Expired OCSP Response: Similar with CRL, OCSP responses are expired for some reasons. OCSP responses have thisUpdate, nextUpdate and producedAt fields. 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 actual time and if not, signature creation application must interpret the OCSP response as expired. In figure 3.9, ASN.1 representation of an expired OCSP response is shown. The highlighted field is the producedAt field of the response.

In this scenario an OCSP server producing expired OCSP responses designed and aim is

Figure 3.10: Signature Forged OCSP Response



to determine if the signature creation application compares producedAt with validation time.

Signature Forged OCSP Response: A malicious user of PKI may try to lead the system from the weak side. Since OCSP response is a component of PKI, system must ensure its clients to trust OCSP responses. A way of ensuring is signing the response as PKI does for other components. Signing mechanism protects the response on editing its time field, revocation status or identity of the requested certificate and so on. From this point of view, a signature creation application has to check the signature of the response before trusting. In figure 3.10, ASN.1 representation of a signature forged OCSP response is shown. The highlighted field is the signature of the response.

In this scenario an OCSP server producing signature forged OCSP responses designed and aim is to determine if the signature creation application verifies signature of the OCSP response.

Expired OCSP Certificate: At the beginning of this section we introduced the concept OCSP certificate, which is authorized to sign the OCSP responses created on behalf of a set of certificates such that the issuer of the OCSP certificate is same with the issuer of the requested set of certificates. While validating an OCSP response, one must also

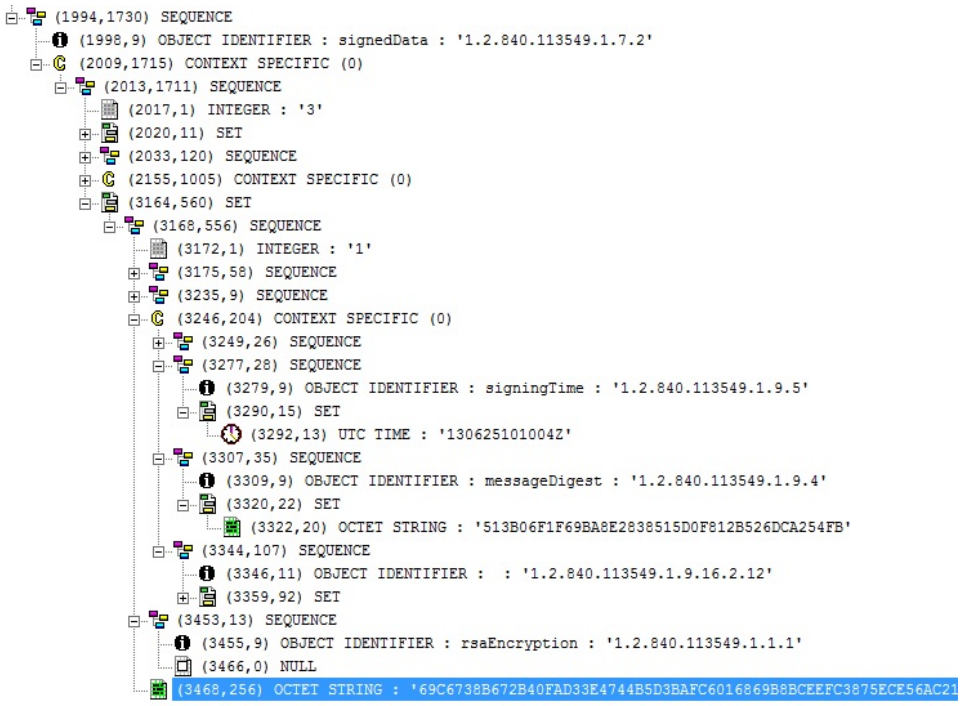
validate its OCSP certificate and checking expiration status of the OCSP certificate is the primary step. As stated before, certificates has a lifetime and a certificate exceed its lifetime is called as an expired certificate. In this scenario, an OCSP server producing valid OCSP responses signed by an expired OCSP certificate is designed and aim is to determine if the signature creation application verifies expiration status of OCSP certificate.

Signature Forged OCSP Certificate: Second verification issue with OCSP response is the signature of the OCSP certificate. One must verify the signature of the OCSP certificate to fulfill whole verifications about OCSP response. In this scenario, an OCSP server producing valid OCSP responses signed by a signature forged OCSP certificate is designed and aim is to determine if the signature creation application verifies signature of OCSP certificate.

Revoked OCSP Certificate: Last verification control about OCSP response is the revocation status of OCSP certificate. If the OCSP certificate has ocsp-no-check [5] field then there is no need to control the revocation status of the certificate but if such a field is added then it requires some routines as renewing the certificate more often against compromise of the responders key. In this scenario an OCSP server with a revoked OCSP certificate, producing responses are designed. OCSP certificate does not have ocsp-no-check field and aim is to determine if the application checks the revocation status of the OCSP certificate.

In this section, we covered revocation datas and focused on the necessary elements for verification of a validation data. Checking the revocation status of a certificate is a primary step in validation but revocation is meaningful with respect to a trusted time. This notion implies that reliable revocation checking depends on reliable time. Reliable time is known as time stamp in PKI as defined in 2.3. Since time stamp servers are part of PKI, a signature creation application must also check necessary criterias about time stamp responses. In the following section we will cover problematic cases about time stamp servers.

Figure 3.11: Signature Forged Time Stamp Token



3.2.4 Time Stamp Validation Scenarios

In digital signature technology, time stamp is used for different reasons [7]. 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.

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. In both situation, the signature of the token is forged. To realize such an attack, signature creation application must validate the signature of the time stamp token. In figure 3.11, ASN.1 representation of a signature forged time stamp token is shown. The highlighted field is the signature of the response.

In this scenario a time stamp server producing signature forged tokens designed and 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 certificate authorized to sign time stamp responses is known as time stamp certificate and is a part of verification checks for time stamp servers. In this scenario a time stamp server with an expired time stamp certificate, producing tokens are 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: In this scenario a time stamp server with a signature forged time stamp certificate is designed and aim is to determine if the application checks the signature of the time stamp certificate.

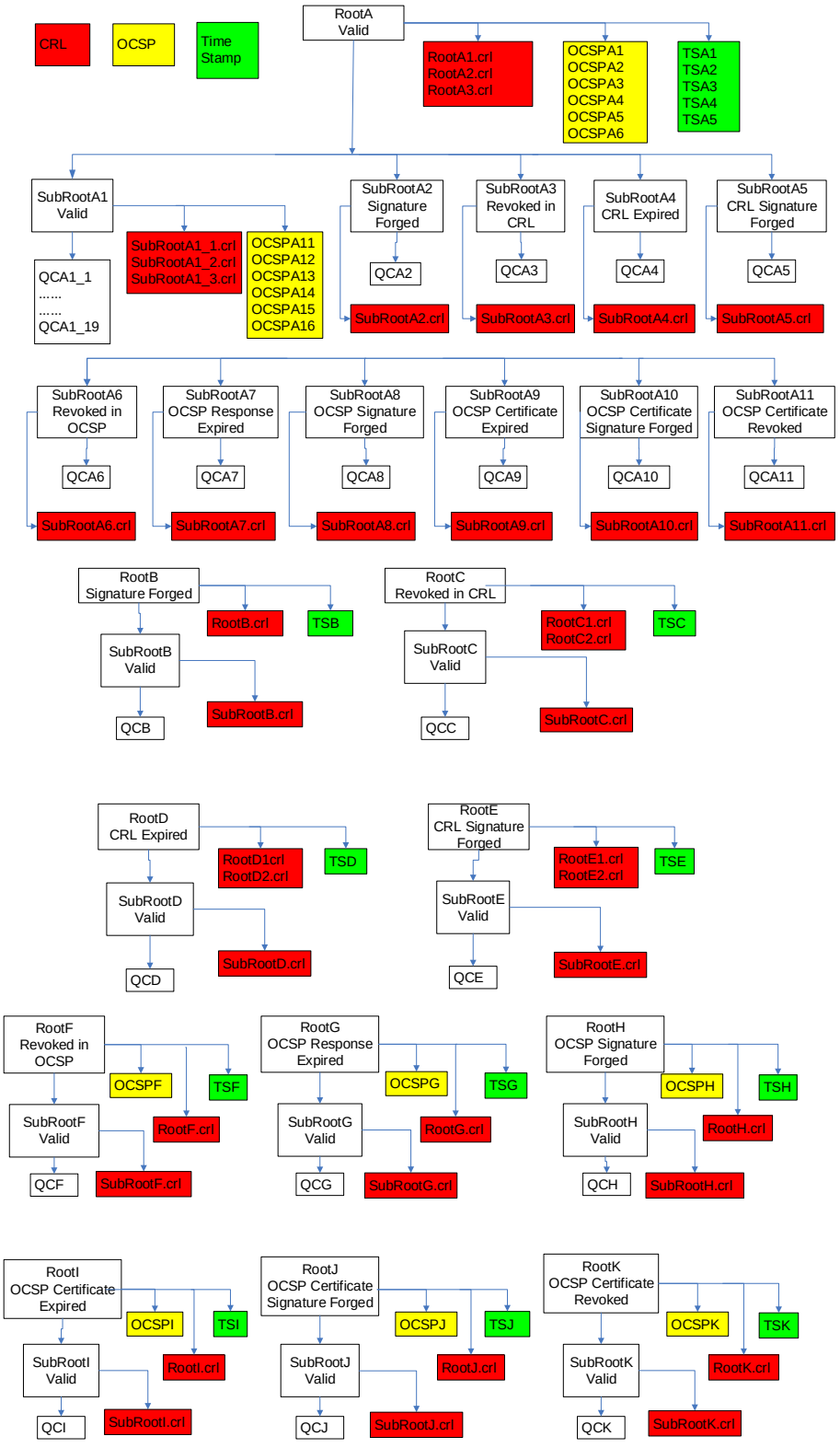
Revoked TS Certificate: As other certificates, time stamp certificate can be revoked for some reasons. In this scenario a time stamp server with a revoked time stamp certificate, producing time stamp tokens are designed and aim is to determine if the application checks the revocation status of the time stamp 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.

Time stamp validation scenarios were the last problematic scenarios designed to check signature creation applications. Begining with end entity certificates to time stamp checks generated a huge certificate hierarchy. In figure 3.12, the uniquely designed PKI model is shown. This PKI model is global and can be used for any signature application developer except the 3 qualified certificate scenarios special for Turkey. These scenarios are explained in section 3.2.1.3. In the following chapter we will discuss implementation details of figured PKI model.

While designing test suite, we also aimed to design problematic scenarios for testing signature verification applications. In the following section we will focus on signature verification scenarios and their design principles.

Figure 3.12: Test Suite Certificate Hierarchy



3.3 Signature Verification Test Scenarios

Signature verification applications have the same importance with signature creation applications and therefore must be well analyzed against possible error cases. From this point of view, similar to section 3.2, we aimed to design all possible error cases to specify the verification accuracy of signature verification applications. ETSI defined 10 types of advanced signature profiles defined in section 2.4 and to satisfy our aim we are to generate each problematic scenario, if possible, for 10 signature types. The simplest signature type is BES and the most advanced one is the ES-A (archive). From BES to ES-A, the detail in signature format is increased and so further possible error cases are added. In this section we will explain the design theory of signature verification test scenarios. While defining these scenarios, we will group them with respect to the signature types. Since a more detailed signature type is constructed over a less detailed type, it includes scenarios of the less detailed one, and so error cases defined for a specific signature type is common for all signature types greater than or equal to the specified. We are beginning to define the innermost type, BES and EPES scenarios.

3.3.1 BES-EPES Verification Scenarios

These signature types are the basic ones among all types. They have the primary properties that all the ETSI defined signature types must have. The only difference between BES and EPES is, EPES has the signature-policy-identifier [7],[25],[35] attribute. A signature policy defines the rules for creation and validation of an electronic signature, and is included as a signed attribute with every Explicit Policy-based Electronic Signature. All the advanced signature types are built on one of these basic types. In our signature verification suite, all the advanced signature types are built on BES but we also designed error cases for EPES type. In this subsection we will define error cases for BES and EPES but these cases are common and so included in all advanced types. We gathered error scenarios in 5 groups.

3.3.1.1 Certificate Validation Scenarios

In section 3.2, we defined the designed possible error cases in PKI. These scenarios were about end-entity certificates, issuer certificates, revocation datas and time stamp servers and

in signature creation process, if one of the scenarios is encountered, it is expected that the process is terminated. In this portion of BES-EPES scenarios, we created as much signature files as the specific error cases designed in section 3.2. Aim is to determine if the signature verification application catches the problematic cases derived from PKI components failure.

3.3.1.2 ESS Signing Certificate Attribute Validation Scenario

In cryptographical point of view, digital signature is the encryption of the digest of the message with the private key of signer but in PKI, message digest is not the only input for encryption. All the inputs for encryption process is known as signed attributes. Some of the signed attributes are mandatory and some of them are optional. ESS signing certificate [20],[23],[7] is a mandatory attribute with OID (1.2.840.113549.1.9.16.2.12). This attribute contains reference to signer certificate by including the digest value of the certificate, its issuer's informations and the serial number of the signer certificate. ESS signing certificate is accepted as a mandatory attribute to prevent simple substitution and reissue attacks. Let us think about a situation where an attacker makes two PKCS10 request [14] for the same key pair and ESS signing certificate is not a mandatory signed attribute. If there exists a signature file signed by the private key in the request then it is an ambiguity that which certificate is the signer certificate. In order to remove this ambiguity, ESS signing certificate is accepted as a mandatory signed attribute and since this attribute includes the digest of the signer certificate and also serial and issuer informations, there remains no ambiguity about the identification of the signer. In figure 3.13, ASN.1 representation of the ESS signing certificate attribute in a signed file is shown.

A signature verification application has to check that if this attribute is signed with the other signed attributes or not. In this scenario a signature file, where the signer certificate digest in ESS signing certificate is forged, has designed and aim is to determine if the signature verification application recognizes this faulty and warns the user about this issue.

3.3.1.3 Message Digest Validation Scenario

Message digest [6],[27],[28],[29],[30],[31] is another mandatory signed attribute with OID (1.2.840.113549.1.9.4). This field includes the digest of the message. Signing the digest

Figure 3.13: ESS Signing Certificate

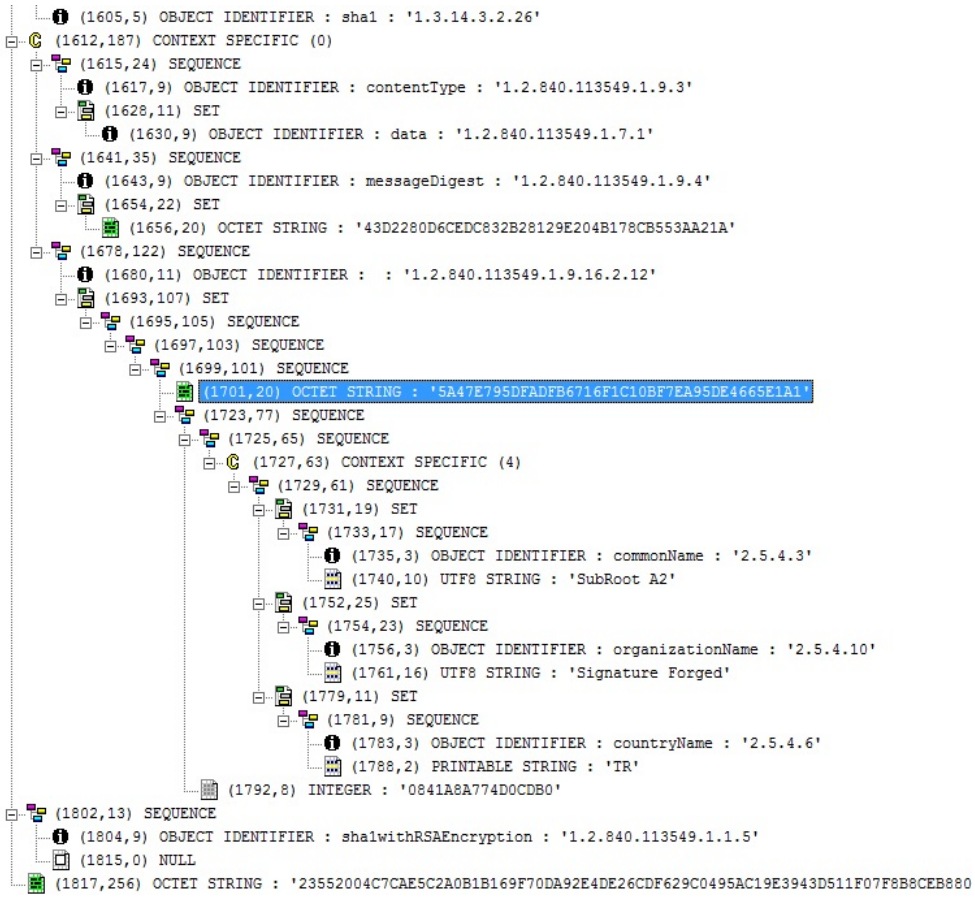
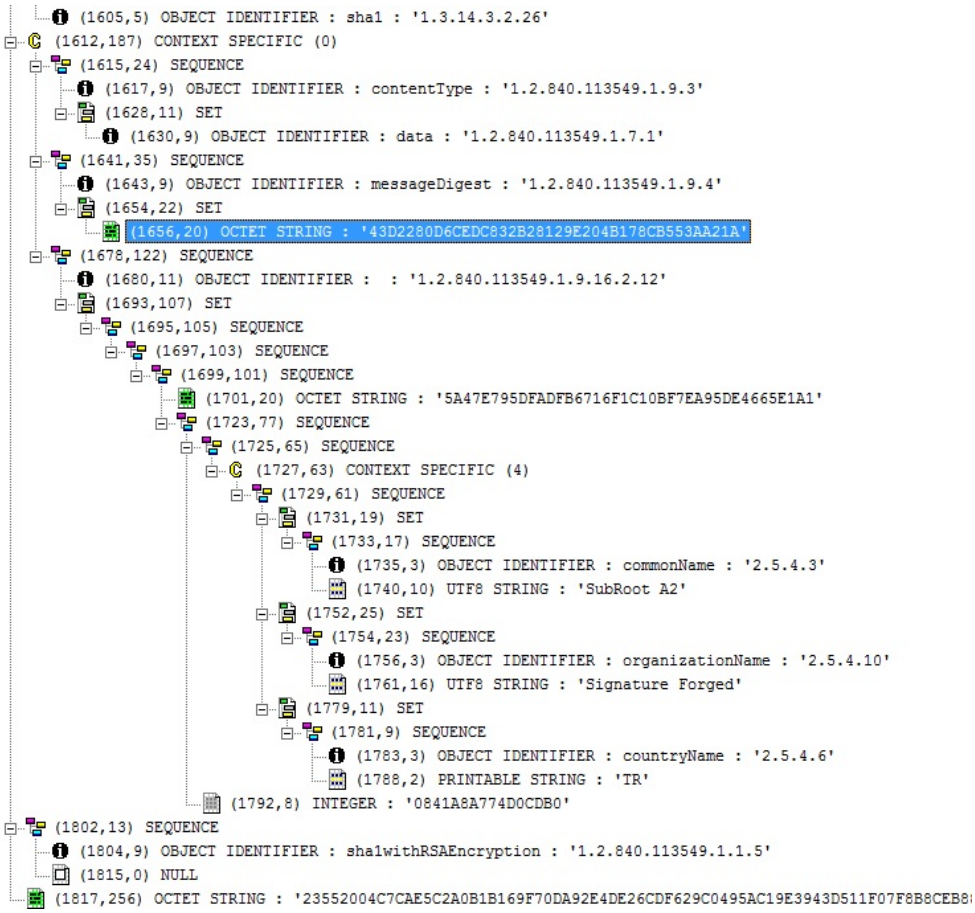


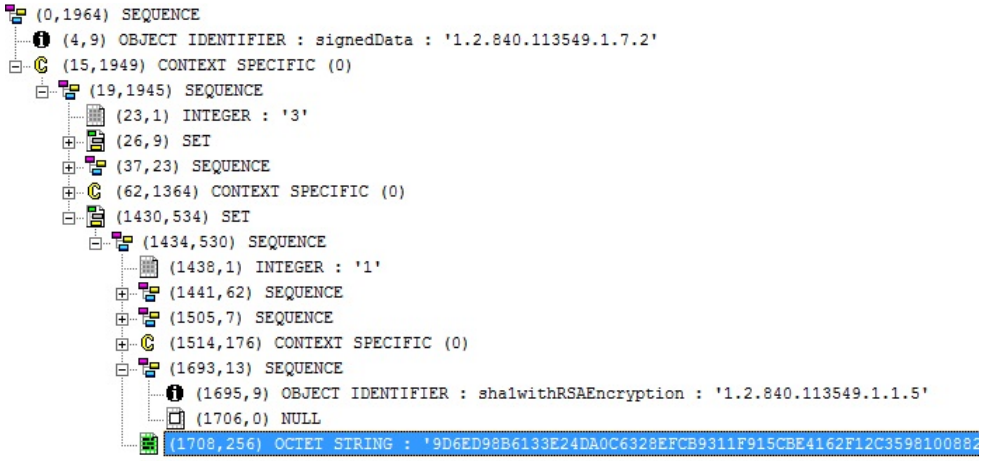
Figure 3.14: Message Digest



of the message instead of the original hava some advantages. One of them is the public encryption of the actual data is very time consuming and so inefficient. Secondly, signing message digest gives the control of recognizing the disruption of the integrity of the message. We know that integrity is an important concept satisfied with digital signature. In figure 3.14, ASN.1 representation of the message-digest attribute in a signed file is shown.

A signature verification application has to check that if this attribute is signed with the other signed attributes or not. In this scenario a signature file, where the message digest field is forged, has designed and aim is to determine if the signature verification application recognizes this faulty and warns the user about this issue.

Figure 3.15: Signature Value



3.3.1.4 Deprecated Digest Algorithm Scenario

Since the computer technologies and cryptanalysis techniques are advancing rapidly, cryptographic algorithms and their accepted key sizes are changing respectively. In this case, the authority of the government responsible with certificate authorities publishes the new accepted algorithms and key sizes [17],[38] with the issuing dates. After this date, a signature file created with an invalid algorithm or a key size will be accepted as invalid. From this point of view, a signature verification application has to check the digest and encryption algorithm of the signature file. In this scenario a signature file with an invalid digest algorithm, namely MD5, has designed and aim is to determine if the signature verification application recognizes this faulty and warns the user about this issue.

3.3.1.5 Invalid Signature Value Scenario

Signature value check is a primary check for signature verification applications. In this scenario a signature file, where the signature value field is forged, has designed. Aim is to determine if the signature verification application recognizes this faulty and warns the user about this issue. Signature value field in a signed file is located at the end of the signerInfo field with the OID of its signature algorithm. In figure 3.15, the signature value with OID (1.2.840.113549.1.1.5)-sha1withRSAEncryption is figured.

3.3.2 ES-T Scenarios

ES-T is simply BES or EPES with time stamp. Therefore ES-T scenarios include section 3.3.1 and have further scenarios related with time stamp. In this section we will introduce these further scenarios of signature type ES-T. The origin of these scenarios are coming from section 3.2.4.

Message Imprint Forged: Time stamp tokens includes message imprint field which is the hash of the data to be time stamped. In this scenario, we designed an ES-T type signature where the message imprint field of time stamp token is forged. Aim is to determine if the signature verification application recognizes this faulty and warns the user about this issue.

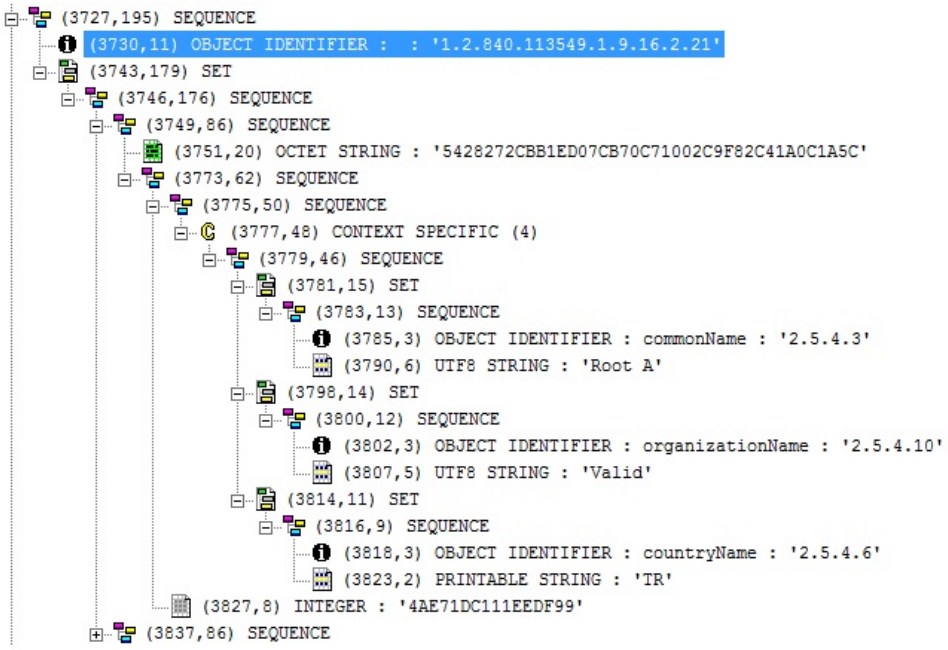
TS Token Signature Forged: In this scenario we use the time stamp server created in section 3.2 which serves signature forged time stamp tokens. We created a BES type signature and time stamped it with the defined time stamp server to make an invalid ES-T signature. Aim is to determine if the signature verification application recognizes this faulty and warns the user about this issue.

TS Certificate Expired: In this scenario we use the time stamp server created in section 3.2 which serves time stamp tokens signed by expired time stamp certificate. We created a BES type signature and time stamped it with the defined time stamp server to make an invalid ES-T signature. Aim is to determine if the signature verification application recognizes this faulty and warns the user about this issue.

TS Certificate Signature Forged: In this scenario we use the time stamp server created in section 3.2 which serves time stamp tokens signed by signature forged time stamp certificate. We created a BES type signature and time stamped it with the defined time stamp server to make an invalid ES-T signature. Aim is to determine if the signature verification application recognizes this faulty and warns the user about this issue.

TS Certificate Revoked: In this scenario we use the time stamp server created in section 3.2 which serves time stamp tokens signed by revoked time stamp certificate. We created a BES type signature and time stamped it with the defined time stamp server to make an invalid ES-T signature. Aim is to determine if the signature verification application recognizes this faulty and warns the user about this issue.

Figure 3.16: Complete Certificate References



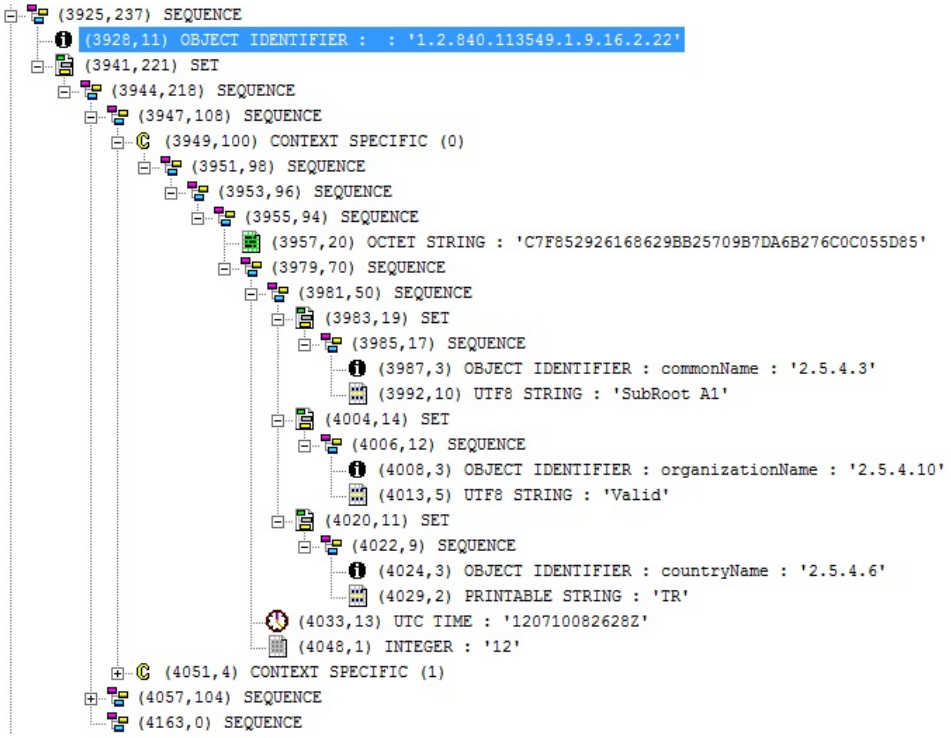
TS Root CA Scenarios: In this scenario we use each time stamp servers created in section 3.2 which serve time stamp tokens signed by a time stamp certificate issued by a problematic certificate authority. In each case, we created a BES type signature and time stamped it with one of the defined time stamp servers to make an invalid ES-T signature. Aim is to determine if the signature verification application recognizes all the faulties sourced by these servers and warns the user about this issue.

In the following section we will describe scenarios related with ES-C.

3.3.3 ES-C Scenarios

ES-C is a type of a digital signature with complete validation data references. Structurally, ES-C is ES-T with complete certificate references and complete revocation references [7]. Complete certificate references is an attribute with OID (1.2.840.113549.1.9.16.2.21) which includes all the certificate references, except signer certificate, necessary for validating the signature. Signer certificate reference is not included since it is present in signer info of the signed data. In figure 3.16, ASN.1 representation of the complete certificate references attribute in a signed file is shown.

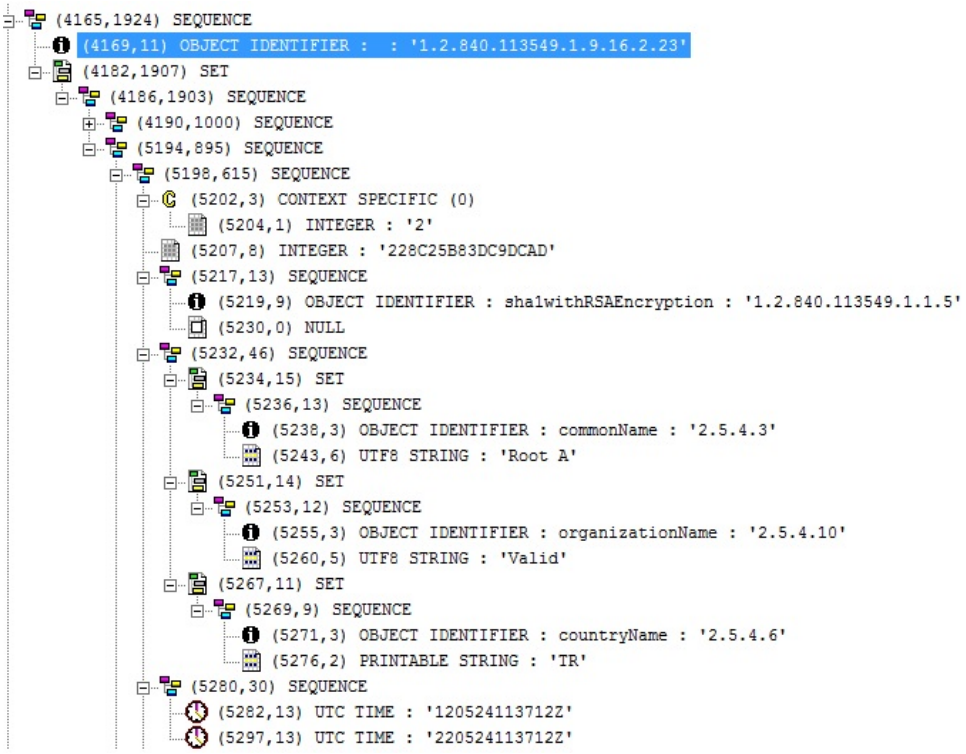
Figure 3.17: Complete Revocation References



Complete revocation references is an attribute with OID (1.2.840.113549.1.9.16.2.22) which includes all the CRL and OCSP responses references necessary for validating the certificate path of the signature. In figure 3.17, ASN.1 representation of the complete revocation references attribute in a signed file is shown.

The placement of complete certificate references (CCR) and complete revocation references (CRR) are in a way such that the first revocation reference in CRR is for validating signer certificate and the other revocation references in CRR are for validating the certificate references in CCR respectively. Since the order of CCR and CRR are important, forging or removing a certificate reference or a revocation reference makes an ES-C type signature unverifiable. In ES-C specific scenarios, we are aimed to examine the signature verification applications, as if they recognize such a forging or removing faulty of each possible certificate or revocation references in CCR and CRR.

Figure 3.18: Certificate Values



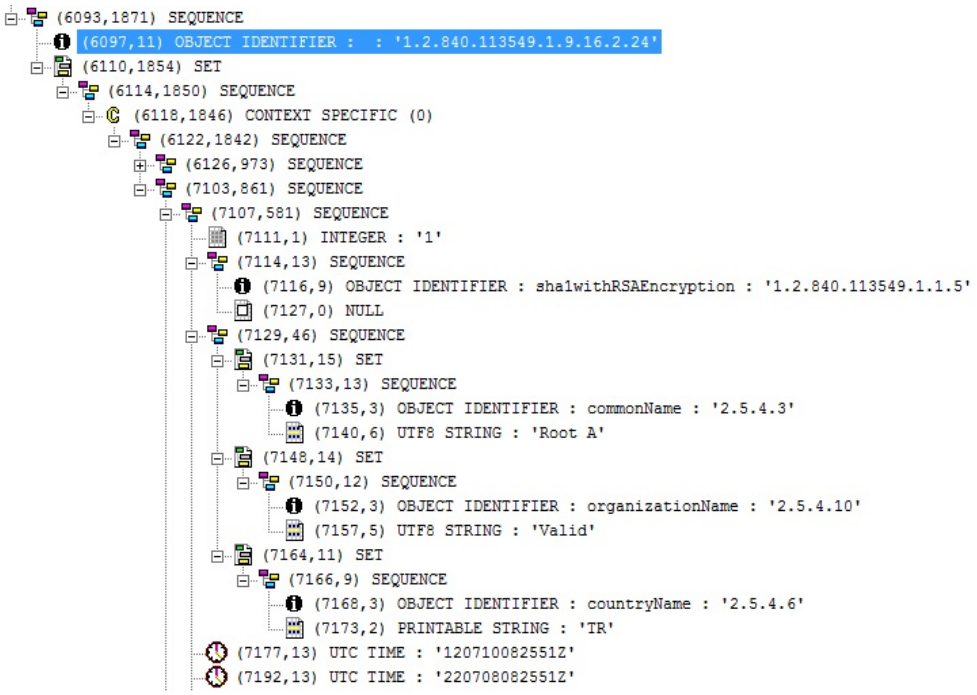
3.3.4 ES-XL Scenarios

ES-XL is a type of a digital signature with complete validation data references and values. Structurally, ES-XL is ES-C with certificate values and revocation values [7]. This additional attributes included in ES-XL provides a reprository for the certificate and revocation values and removes the problem of loss of resources for future verification of ES-C. Certificate values is an attribute with OID (1.2.840.113549.1.9.16.2.23) which includes all the certificate values, except signer certificate, necessary for validating the signature. Signer certificate value is not included since it is present in certificates field of signedData. In figure 3.18, ASN.1 representation of the certificate values attribute in a signed file is shown.

Revocation values is an attribute with OID (1.2.840.113549.1.9.16.2.24) which includes all the CRL and OCSP responses necessary for validating the certificate path of the signature. In figure 3.19, ASN.1 representation of the revocation values attribute in a signed file is shown.

In verification of ES-XL signatures, after mapping certificate references and revocation reference as mentioned in section 3.3.3, the values of the related references are searched in

Figure 3.19: Revocation Values



certificate and revocation values fields. Therefore forging or removing a certificate or a revocation value makes an ES-XL type signature unverifiable. While designing ES-XL specific scenarios, we aimed to examine the signature verification applications, as if they recognize such a forging or removing faulty of each possible certificate or revocation values in related fields.

3.3.5 X Type 1 - X Type 2 - Archive Scenarios

X Type 1, extended format with time type 1, is a signature type extending from ES-C where the whole ES-C is time stamped. It is against to protect the certificates, revocation values in case of a later compromise of a CA key, CRL key, or OCSP issuer key.

X Type 2, extended format with time type 2, is a signature type extending from ES-C where the certificate and revocation references are time stamped. It is against to protect the certificates, revocation values in case of a later compromise of a CA key, CRL key, or OCSP issuer key.

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 cryptographic algorithms and hashing algorithms.

Three of these signature types have common specific scenarios sourced by the second time stamp. All these types have scenarios derived from the previous signature types and have further scenarios with the second time stamp. We will not define these scenarios since they are same with the section 3.3.2.

As a summary of this section, we described the design principle and theory of our e-signature test suite in terms of signature creation and verification. In the following section we will give the implementation details of certificates, signature files and servers.

CHAPTER 4

IMPLEMENTATION DETAILS

In previous chapter we described the design theory of our test suite and explained the designed scenarios for evaluating the accuracy of signature creation and verification applications. In this chapter we will reveal the implementation details of each component of our unique PKI model. We will give all the informations related with PKI concepts such as OCSP servers, time stamp servers, whole certificate set and whole signature set. Let us begin with OCSP servers.

4.1 OCSP Servers

In this section we will give necessary informations about OCSP servers established for test suite. In figure 4.1, we give the names and the access informations of each OCSP server.

Table 4.1: OCSP Servers Access Information

OCSP Server	**Server URL**
OCSPA1	http://ocspsA1.test.kamusm.gov.tr
OCSPA2	http://ocspsA2.test.kamusm.gov.tr
OCSPA3	http://ocspsA3.test.kamusm.gov.tr
OCSPA4	http://ocspsA4.test.kamusm.gov.tr
OCSPA5	http://ocspsA5.test.kamusm.gov.tr
OCSPA6	http://ocspsA6.test.kamusm.gov.tr
OCSPA11	http://ocspsA11.test.kamusm.gov.tr
OCSPA12	http://ocspsA12.test.kamusm.gov.tr
OCSPA13	http://ocspsA13.test.kamusm.gov.tr

Continued

OCSP Server	Server URL
OCSPA14	http://ocspsA14.test.kamusm.gov.tr
OCSPA15	http://ocspsA15.test.kamusm.gov.tr
OCSPA16	http://ocspsA16.test.kamusm.gov.tr
OCSPF	http://ocspsF.test.kamusm.gov.tr
OCSPG	http://ocspsG.test.kamusm.gov.tr
OCSPH	http://ocspsH.test.kamusm.gov.tr
OCSPI	http://ocspsI.test.kamusm.gov.tr
OCSPJ	http://ocspsJ.test.kamusm.gov.tr
OCSPK	http://ocspsK.test.kamusm.gov.tr

In figure 4.2, we give the reason for establishing the OCSP server, namely the error scenario of the OCSP server and the issuer of the OCSP certificate.

Table 4.2: OCSP Servers Properties

OCSP Certificate	OCSP Property	OCSP Certificate Signer
OCSPA1.crt	Valid	RootA.crt
OCSPA2.crt	Expired OCSP Response	RootA.crt
OCSPA3.crt	Signature Forged OCSP Response	RootA.crt
OCSPA4.crt	Expired OCSP Certificate	RootA.crt
OCSPA5.crt	Signature Forged OCSP Certificate	RootA.crt
OCSPA6.crt	Revoked OCSP Certificate	RootA.crt
OCSPA11.crt	Valid	SubRootA1.crt
OCSPA12.crt	Expired OCSP Response	SubRootA1.crt
OCSPA13.crt	Signature Forged OCSP Response	SubRootA1.crt
OCSPA14.crt	Expired OCSP Certificate	SubRootA1.crt
OCSPA15.crt	Signature Forged OCSP Certificate	SubRootA1.crt
OCSPA16.crt	Revoked OCSP Certificate	SubRootA1.crt
OCSPF.crt	Valid	RootF.crt
OCSPG.crt	Expired OCSP Response	RootG.crt
OCSPH.crt	Signature Forged OCSP Response	RootH.crt
OCSPI.crt	Expired OCSP Certificate	RootI.crt
OCSPJ.crt	Signature Forged OCSP Certificate	RootJ.crt
OCSPK.crt	Revoked OCSP Certificate	RootK.crt

As defined before, OCSP servers serve the revocation status of a certificate issued by the issuer of the OCSP certificate. In order to create some scenarios, we constructed cases where a certificate is known revoked in OCSP but not revoked in CRL. From this point of view, figure 4.3 shows which certificate is revoked from OCSP.

Table 4.3: OCSP Servers vs. Revoked Certificates

OCSP Certificate	**Revoked Certificates**
OCSPA1.crt	SubRootA6.crt
OCSPA2.crt	-
OCSPA3.crt	-
OCSPA4.crt	-
OCSPA5.crt	-
OCSPA6.crt	-
OCSPA11.crt	QCA1_11.crt
OCSPA12.crt	-
OCSPA13.crt	-
OCSPA14.crt	-
OCSPA15.crt	-
OCSPA16.crt	-
OCSPF.crt	RootF.crt
OCSPG.crt	-
OCSPH.crt	-
OCSPI.crt	-
OCSPJ.crt	-
OCSPK.crt	-

4.2 Time Stamp Servers

In this section we will give necessary informations about time stamp servers established for test suite. In figure 4.4, we give the names and the access informations of each time stamp server.

Table 4.4: TS Servers Access Information

TS Server	Server URL
TSA1	http//zdsA1.test.kamusm.gov.tr
TSA2	http//zdsA2.test.kamusm.gov.tr
TSA3	http//zdsA3.test.kamusm.gov.tr
TSA4	http//zdsA4.test.kamusm.gov.tr
TSA5	http//zdsA5.test.kamusm.gov.tr
TSB	http//zdsB.test.kamusm.gov.tr
TSC	http//zdsC.test.kamusm.gov.tr
TSD	http//zdsD.test.kamusm.gov.tr
TSE	http//zdsE.test.kamusm.gov.tr
TSF	http//zdsF.test.kamusm.gov.tr
TSG	http//zdsG.test.kamusm.gov.tr
TSH	http//zdsH.test.kamusm.gov.tr
TSI	http//zdsI.test.kamusm.gov.tr
TSJ	http//zdsJ.test.kamusm.gov.tr
TSK	http//zdsK.test.kamusm.gov.tr

In figure 4.5, we give the reason for establishing the time stamp server, namely the error scenario of the time stamp server and the issuer of the time stamp certificate.

Table 4.5: TS Servers Properties

TS Certificate	TS Property	TS Certificate Signer
TSA1.crt	Valid	RootA.crt
TSA2.crt	Signature Forged TS Token	RootA.crt
TSA3.crt	Expired TS Certificate	RootA.crt
TSA4.crt	Signature Forged TS Certificate	RootA.crt
TSA5.crt	Revoked TS Certificate	RootA.crt
TSB.crt	Root Signature Forged	RootB.crt
TSC.crt	Root Revoked in CRL	RootC.crt
TSD.crt	Root CRL Expired	RootD.crt
TSE.crt	Root CRL Signature Forged	RootE.crt
TSF.crt	Root Revoked in OCSP	RootF.crt

Continued

TS Certificate	TS Property	TS Certificate Signer
TSG.crt	Root OCSP Response Expired	RootG.crt
TSH.crt	Root OCSP Response Signature Forged	RootH.crt
TSI.crt	Root OCSP Certificate Expired	RootI.crt
TSJ.crt	Root OCSP Certificate Signature Forged	RootJ.crt
TSK.crt	Root OCSP Certificate Revoked	RootK.crt

4.3 CRL

In this section we will give necessary informations about released CRLs used in test suite. In figure 4.6, we give the reason for releasing the specified CRL, namely the error scenario of the released CRL and its issuer. CRLs including OCSP suffix are indicating the sources used by the related OCSP responders.

Table 4.6: CRL Properties

CRL Name	CRL Property	CRL Signer
RootA1.crl	Valid	RootA.crt
RootA2.crl	Expired CRL	RootA.crt
RootA3.crl	Signature Forged CRL	RootA.crt
RootAOCSP.crl	Valid	RootA.crt
RootB.crl	Valid	RootB.crt
RootC1.crl	Valid	RootC.crt
RootC2.crl	Valid	RootC.crt
RootD1.crl	Expired CRL	RootD.crt
RootD2.crl	Valid	RootD.crt
RootE1.crl	Signature Forged CRL	RootE.crt
RootE2.crl	Valid	RootE.crt
RootF.crl	Valid	RootF.crt
RootFOCSP.crl	Valid	RootF.crt
RootG.crl	Valid	RootG.crt
RootH.crl	Valid	RootH.crt
RootI.crl	Valid	RootI.crt
RootJ.crl	Valid	RootJ.crt
RootK.crl	Valid	RootK.crt

Continued

CRL Name	CRL Property	CRL Signer
SubRootA1_1.crl	Valid	SubRootA1.crt
SubRootA1_2.crl	Expired CRL	SubRootA1.crt
SubRootA1_3.crl	Signature Forged CRL	SubRootA1.crt
SubRootA1OCSP.crl	Valid	SubRootA1.crt
SubRootA2,.,K.crl	Valid	SubRoot{A2,.,K}.crt

Similar to figure 4.3, figure 4.7 shows which certificate is revoked in which released CRL and further shows the lifetime of the CRL.

Table 4.7: CRL Expiration and Revoked Certificates Info

CRL Name	Expiration	Revoked Certificates
		SubRootA3.crt
RootA1.crl	10 year	OCSPA6.crt
		TSA5.crt
RootA2.crl	3 month	-
RootA3.crl	10 year	-
RootAOCSP.crl	10 year	SubRootA6.crt
RootB.crl	10 year	-
RootC1.crl	10 year	RootC.crt
RootC2.crl	10 year	-
RootD1.crl	3 month	-
RootD2.crl	10 year	-
RootE1.crl	10 year	-
RootE2.crl	10 year	-
RootF.crl	10 year	-
RootFOCSP.crl	10 year	RootF.crt
RootG.crl	10 year	-
RootH.crl	10 year	-
RootI.crl	10 year	-
RootJ.crl	10 year	-
RootK.crl	10 year	OCSPK.crt
SubRootA1_1.crl	10 year	QCA1_10.crt
		OCSPA1_6.crt

Continued

CRL Name	Expiration	Revoked Certificates
SubRootA1_2.crl	3 month	-
SubRootA1_3.crl	10 year	-
SubRootA1OCSP.crl	10 year	QCA1_11.crt
SubRootA2,.,K.crl	10 year	-

All the CRLs defined above can be accessed from the address http://depo.test.kamusm.gov.tr/CRL Name, i.e. http://depo.test.kamusm.gov.tr/RootA1.crl.

4.4 Certificate Profiles

In this section we will describe the certificate profiles for root, SubRoot, qualified, OCSP and time stamp certificates. Algorithm used to sign the certificates is RSA-with-Sha1 and the key size for RSA encryption is 2048 bits. Information about encryption and hash algorithms can be retrieved from [40], [39], [31], [32], [33],[34]. In each certificate profile, name of the certificate, distinguished name of the certificate, expiration period, available revocation check methods and revocation status of the certificate is shown. Figure 4.8 is the root certificate profile.

Table 4.8: Root Certificates Profile

Certificate	DN Field	Expiration	CRL	OCSP	Revocation Status
RootA.crt	CN: Root A O: Valid C: TR	10 year	RootA1.crl	-	Valid
RootB.crt	CN: Root B O: Signature forged C: TR	10 year	RootB.crl	-	Valid
RootC.crt	CN: Root C O: Revoked in CRL C: TR	10 year	RootC1.crl	-	Revoked
RootD.crt	CN: Root D O: Expired CRL C: TR	10 year	RootD1.crl	-	Valid
RootE.crt	CN: Root E	10 year	RootE1.crl	-	Valid

Continued

Certificate	DN Field	Expiration	CRL	OCSP	Revocation Status
	O: CRL Signature Forged C: TR				
RootF.crt	CN: Root F O: Revoked in OCSP C: TR	10 year	-	OCSPF	Revoked
RootG.crt	CN: Root G O: Expired OCSP Response C: TR	10 year	-	OCSPG	Valid
RootHcrt	CN: Root H O: OCSP Response Signature Forged C: TR	10 year	-	OCSPH	Valid
RootI.crt	CN: Root I O: Expried OCSP Certificate C: TR	10 year	-	OCSPI	Valid
RootJ.crt	CN: Root J O: OCSP Certificate Signature Forged C: TR	10 year	-	OCSPJ	Valid
RootK.crt	CN: Root K O: Revoked OCSP Certificate C: TR	10 year	-	OCSPK	Valid

In figure 4.9, SubRoot certificate profile is presented.

Table 4.9: SubRoot Certificates Profile

Certificate	DN Field	Expiration	CRL	OCSP	Revocation Status
SubRootA1.crt	CN: SubRoot A1 O: Valid	10 year	RootA1.crl	-	Valid

Continued

Certificate	DN Field	Expiration	CRL	OCSP	Revocation Status
	C: TR				
SubRootA2.crt	CN: SubRoot A2	10 year	RootA1.crl	-	Valid
	O: Signature forged				
	C: TR				
SubRootA3.crt	CN: SubRoot A3	10 year	RootA1.crl	-	Revoked
	O: Revoked in CRL				
	C: TR				
SubRootA4.crt	CN: SubRoot A4	10 year	RootA2.crl	-	Valid
	O: Expired CRL				
	C: TR				
SubRootA5.crt	CN: SubRoot A5	10 year	RootA3.crl	-	Valid
	O: CRL Signature				
	Forged				
	C: TR				
SubRootA6.crt	CN: SubRoot A6	10 year	-	OCSPA1	Revoked
	O: Revoked in OCSP				
	C: TR				
SubRootA7.crt	CN: SubRoot A7	10 year	-	OCSPA2	Valid
	O: Expired OCSP				
	Response				
	C: TR				
SubRootA8.crt	CN: SubRoot A8	10 year	-	OCSPA3	Valid
	O: OCSP Signature				
	Forged				
	C: TR				
SubRootA9.crt	CN: SubRoot A9	10 year	-	OCSPA4	Valid
	O: Expired OCSP				
	Certificate				
	C: TR				
SubRootA10.crt	CN: SubRoot A10	10 year	-	OCSPA5	Valid
	O: OCSP Certificate				
	Signature Forged				
	C: TR				
SubRootA11.crt	CN: SubRoot A11	10 year	-	OCSPA6	Valid

Continued

Certificate	DN Field	Expiration	CRL	OCSP	Revocation Status
	O: Revoked OCSP Certificate				
	C: TR				
SubRootB.crt	CN: SubRoot B	10 year	RootB.crl	-	Valid
	O: Valid				
	C: TR				
SubRootC.crt	CN: SubRoot C	10 year	RootC2.crl	-	Valid
	O: Valid				
	C: TR				
SubRootD.crt	CN: SubRoot D	10 year	RootD2.crl	-	Valid
	O: Valid				
	C: TR				
SubRootE.crt	CN: SubRoot E	10 year	RootE2.crl	-	Valid
	O: Valid				
	C: TR				
SubRoot {F, G, ,...K}.crt	CN: SubRoot {F, G,...K}	10 year	Root {F, G,...K}.crl	-	Valid
	O: Valid				
	C: TR				

In figure 4.10, OCSP certificate profile is presented.

Table 4.10: OCSP Certificates Profile

Certificate	DN Field	Expiration	CRL	Revocation Status
OCSPA1.crt	CN: OCSP A1	10 year	RootA1.crl	Valid
	O: Valid			
	C: TR			
OCSPA2.crt	CN: OCSP A2	10 year	RootA1.crl	Valid
	O: Expired OCSP Response			
	C: TR			
OCSPA3.crt	CN: OCSP A3	10 year	RootA1.crl	Valid
	O: OCSP Response			

Continued

Certificate	DN Field	Expiration	CRL	Revocation Status
	Signature Forged			
	C: TR			
OCSPA4.crt	CN: OCSP A4	3 month	RootA1.crl	Valid
	O: Expired			
	C: TR			
OCSPA5.crt	CN: OCSP A5	10 year	RootA1.crl	Valid
	O: Signature Forged			
	C: TR			
OCSPA6.crt	CN: OCSP A6	10 year	RootA1.crl	Revoked
	O: Revoked			
	C: TR			
OCSPA1_1.crt	CN: OCSP A1 1	10 year	SubRootA1_1.crl	Valid
	O: Valid			
	C: TR			
OCSPA1_2.crt	CN: OCSP A1 2	10 year	SubRootA1_1.crl	Valid
	O: Expired OCSP			
	Response			
	C: TR			
OCSPA1_3.crt	CN: OCSP A1 3	10 year	SubRootA1_1.crl	Valid
	O: OCSP Response			
	Signature Forged			
	C: TR			
OCSPA1_4.crt	CN: OCSP A1 4	3 month	SubRootA1_1.crl	Valid
	O: Expired			
	C: TR			
OCSPA1_5.crt	CN: OCSP A1 5	10 year	SubRootA1_1.crl	Valid
	O: Signature Forged			
	C: TR			
OCSPA1_6.crt	CN: OCSP A1 6	10 year	SubRootA1_1.crl	Revoked
	O: Revoked			
	C: TR			
OCSPF.crt	CN: OCSP F	10 year	RootF.crl	Valid
	O: Root Revoked in OCSP			
	C: TR			

Continued

Certificate	DN Field	Expiration	CRL	Revocation Status
OCSPG.crt	CN: OCSP G O: Expired OCSP Response C: TR	10 year	RootG.crl	Valid
OCSPH.crt	CN: OCSP H O: OCSP Response Signature Forged C: TR	10 year	RootH.crl	Valid
OCSPI.crt	CN: OCSP I O: Expired C: TR	3 month	RootI.crl	Valid
OCSPJ.crt	CN: OCSP J O: Signature Forged C: TR	10 year	RootJ.crl	Valid
OCSPK.crt	CN: OCSP K O: Revoked C: TR	10 year	RootK.crl	Revoked

In figure 4.11, time stamp certificate profile is presented.

Table 4.11: TS Certificates Profile

Certificate	DN Field	Expiration	CRL	Revocation Status
TSA1.crt	CN: TS A1 O: Valid C: TR	10 year	RootA1.crl	Valid
TSA2.crt	CN: TS A2 O: TS Signature Forged C: TR	10 year	RootA1.crl	Valid
TSA3.crt	CN: TS A3 O: Expired C: TR	3 month	RootA1.crl	Valid
TSA4.crt	CN: TS A4 O: Signature Forged	10 year	RootA1.crl	Valid

Continued

Certificate	DN Field	Expiration	CRL	Revocation Status
	C: TR			
TSA5.crt	CN: TS A5	10 year	RootA1.crl	İptal
	O: Revoked			
	C: TR			
TSB.crt	CN: TS B	10 year	RootB.crl	Valid
	O: Root Signature Forged			
	C: TR			
TSC.crt	CN: TS C	10 year	RootC2.crl	Valid
	O: Root Revoked in CRL			
	C: TR			
TSD.crt	CN: TS D	10 year	RootD2.crl	Valid
	O: Root CRL Expired			
	C: TR			
TSE.crt	CN: TS E	10 year	RootE2.crl	Valid
	O: Root CRL			
	Signature Forged			
	C: TR			
TSF.crt	CN: TS F	10 year	RootF.crl	Valid
	O: Root Revoked in OCSP			
	C: TR			
TSG.crt	CN: TS G	10 year	RootG.crl	Valid
	O: Root OCSP			
	Response Expired			
	C: TR			
TSH.crt	CN: TS H	10 year	RootH.crl	Valid
	O: Root OCSP Response			
	Signature Forged			
	C: TR			
TSI.crt	CN: TS I	10 year	RootI.crl	Valid
	O: Root OCSP			
	Certificate Expired			
	C: TR			
TSJ.crt	CN: TS J	10 year	RootJ.crl	Valid
	O: Root OCSP			

Continued

Certificate	DN Field	Expiration	CRL	Revocation Status
	Cert Signature Forged			
	C: TR			
TSK.crt	CN: TS K	10 year	RootK.crl	Valid
	O: Root OCSP			
	Certificate Revoked			
	C: TR			

In figure 4.12, qualified certificate profile is presented.

Table 4.12: QC Certificates Profile

Certificate	DN Field	Expiration	CRL	OCSP	Revocation Status
QCA1_1.crt	CN: QC A1 1	10 year	SubRootA1_1.crl	-	Valid
	O: Valid				
	C: TR				
QCA1_2.crt	CN: QC A1 2	10 year	-	OCSPA11	Valid
	O: Valid				
	C: TR				
QCA1_3.crt	CN: QC A1 3	10 year	SubRootA1_1.crl	OCSPA11	Valid
	O: Non-repuduation				
	Absent				
	C: TR				
QCA1_4.crt	CN: QC A1 4	10 year	SubRootA1_1.crl	OCSPA11	Valid
	O: CP user notice				
	Statement Absent				
	C: TR				
QCA1_5.crt	CN: QC A1 5	10 year	SubRootA1_1.crl	OCSPA11	Valid
	O: ETSI QC Statement				
	ID Absent				
	C: TR				
QCA1_6.crt	CN: QC A1 6	10 year	SubRootA1_1.crl	OCSPA11	Valid
	O: BTK QC Statement				
	ID Absent				
	C: TR				

Continued

Certificate	DN Field	Expiration	CRL	OCSP	Revocation Status
QCA1_7.crt	CN: QC A1 7 O: BTK QC Statement Info Absent C: TR	10 year	SubRootA1_1.crl	OCSPA11	Valid
QCA1_8.crt	CN: QC A1 8 O: Expired C: TR	3 month	SubRootA1_1.crl	OCSPA11	Valid
QCA1_9.crt	CN: QC A1 9 O: Signature Forged C: TR	10 year	SubRootA1_1.crl	OCSPA11	Valid
QCA1_10.crt	CN: QC A1 10 O: Revoked in CRL C: TR	10 year	SubRootA1_1.crl	-	İptal
QCA1_11.crt	CN: QC A1 11 O: Revoked in OCSP C: TR	10 year	-	OCSPA11	İptal
QCA1_12.crt	CN: QC A1 12 O: Expired CRL C: TR	10 year	SubRootA1_2.crl	-	Valid
QCA1_13.crt	CN: QCA1 13 O: CRL Signature Forged C: TR	10 year	SubRootA1_3.crl	-	Valid
QCA1_14.crt	CN: QC A1 14 O: Expired OCSP Response C: TR	10 year	-	OCSPA12	Valid
QCA1_15.crt	CN: QC A1 15 O: OCSP Response Signature Forged C: TR	10 year	-	OCSPA13	Valid
QCA1_16.crt	CN: QC A1 16 O: Expired OCSP Certificate	10 year	-	OCSPA14	Valid

Continued

Certificate	DN Field	Expiration	CRL	OCSP	Revocation Status
	C: TR				
QCA1_17.crt	CN: QC A1 17 O: OCSP Certificate Signature Forged C: TR	10 year	-	OCSPA15	Valid
QCA1_18.crt	CN: QC A1 18 O: Revoked OCSP Certificate C: TR	10 year	-	OCSPA16	Valid
QCA1_19.crt	CN: QC A1 19 O: Monetary Limit and Roles Included C: TR	10 year	SubRootA1_1.crl	-	Valid
QCA2.crt	CN: QC A2 O: SubRoot Certificate Signature Forged C: TR	10 year	SubRootA2.crl	-	Valid
QCA3.crt	CN: QC A3 O: SubRoot Certificate Revoked in CRL C: TR	10 year	SubRootA3.crl	-	Valid
QCA4.crt	CN: QC A4 O: SubRoot CRL Expired C: TR	10 year	SubRootA4.crl	-	Valid
QCA5.crt	CN: QC A5 O: SubRoot CRL Signature Forged C: TR	10 year	SubRootA5.crl	-	Valid
QCA6.crt	CN: QC A6 O: SubRoot Revoked in OCSP C: TR	10 year	SubRootA6.crl	-	Valid
QCA7.crt	CN: QC A7	10 year	SubRootA7.crl	-	Valid

Continued

Certificate	DN Field	Expiration	CRL	OCSP	Revocation Status
	O: SubRoots OCSP Response Expired C: TR				
QCA8.crt	CN: QC A8 O: SubRoots OCSP Response Signature Forged C: TR	10 year	SubRootA8.crl	-	Valid
QCA9.crt	CN: QC A9 O: SubRoots OCSP Certificate Expired C: TR	10 year	SubRootA9.crl	-	Valid
QCA10.crt	CN: QC A10 O: SubRoots OCSP Certificate Signature Forged C: TR	10 year	SubRootA10.crl	-	Valid
QCA11.crt	CN: QC A11 O: SubRoots OCSP Certificate Revoked C: TR	10 year	SubRootA11.crl	-	Valid
QCB.crt	CN: QC B O: Root Certificate Signature Forged C: TR	10 year	SubRootB.crl	-	Valid
QCC.crt	CN: QC C O: Root Certificate Revoked in CRL C: TR	10 year	SubRootC.crl	-	Valid
QCD.crt	CN: QC D O: Root CRL Expired C: TR	10 year	SubRootD.crl	-	Valid
QCE.crt	CN: QC E	10 year	SubRootE.crl	-	Valid

Continued

Certificate	DN Field	Expiration	CRL	OCSP	Revocation Status
	O: Root CRL Signature Forged C: TR				
QCF.crt	CN: QC F O: Root Certificate Revoked in OCSP C: TR	10 year	SubRootF.crl	-	Valid
QCG.crt	CN: QC G O: Root Certificate OCSP Response Expired C: TR	10 year	SubRootG.crl	-	Valid
QCH.crt	CN: QC H O: Root Certificate OCSP Response Signature Forged C: TR	10 year	SubRootH.crl	-	Valid
QCI.crt	CN: QC I O: Root Certificate's OCSP Certificate Expired C: TR	10 year	SubRootI.crl	-	Valid
QCJ.crt	CN: QC J O: Root Certificate's OCSP Certificate Signature Forged C: TR	10 year	SubRootJ.crl	-	Valid
QCK.crt	CN: QC K O: Root Certificate's OCSP Certificate Revoked C: TR	10 year	SubRootK.crl	-	Valid

4.5 Signatures

In section 3.3 we defined signature verification scenarios and in this section we will explain details of the signed files constructed for this aim. Also as described in 3.3, error scenarios are extending from signature type BES to ES-A. Therefore for avoiding duplicate definitions of properties of signed files, we grouped implementation properties of signed filed in 5 groups. BES-EPES scenarios are the core part of the signature verification section and common for all signature types.

4.5.1 BES-EPES Scenarios

In this section we will explain implementation details of BES-EPES scenarios. These scenarios are related with common attributes of all signature types. Figure 4.13 shows the common signature scenarios. Algorithm used in signing for all the signature files except A1_1_6.txt.p7s is RSA-with-Sha1 and the modulus is 2048 bits.

Table 4.13: Common Signature Scenarios

Signer Certificate	**Signature File**	**Signature Property**
QCA1_1.crt	A1_1_1.txt.p7s	Valid (Revocation check from CRL and All signed attributes are added!)
	A1_1_2.doc.p7s	Signature file whose EncapsulatedContentInfo is a file including macro
	A1_1_3.bmp.p7s	Signature file with HTML MIME type in ContentHints and has BMP file extension
	A1_1_4.txt.p7s	Signature file whose ESS-Signing-Certificate signature attribute is forged
	A1_1_5.txt.p7s	Signature file whose ”messageDigest” signature attribute is forged
	A1_1_6.txt.p7s	Signature file in which MD5 digest algorithm is used
	A1_1_7.txt.p7s	Signature file whose signature value is forged
QCA1_2.crt	A1_2.txt.p7s	Valid (Revocation check from OCSP)
QCA1_3.crt	A1_3.txt.p7s	Signature file signed by a certificate which does not have non repudiation field in key usage extension
QCA1_4.crt	A1_4.txt.p7s	Signature file signed by a certificate which does not have ICTA UserNotice in CertificatePolicies extension

Continued

Signer Certificate	Signature File	Signature Property
QCA1_5.crt	A1_5.txt.p7s	Signature file signed by a certificate which does not have ETSI OID in QualifiedCertificateStatement
QCA1_6.crt	A1_6.txt.p7s	Signature file signed by a certificate which does not have ICTA OID in QualifiedCertificateStatement
QCA1_7.crt	A1_7.txt.p7s	Signature file signed by a certificate which does not have ICTA UserNotice in QualifiedCertificateStatement
QCA1_8.crt	A1_8.txt.p7s	Signature file signed by an expired certificate
QCA1_9.crt	A1_9.txt.p7s	Signature file signed by a signature forged certificate
QCA1_10.crt	A1_10_1.txt.p7s	Signature file signed by a revoked certificate whose revokeness is published in CRL
QCA1_11.crt	A1_11_1.txt.p7s	Signature file signed by a revoked certificate whose revokeness is published in OCSP
QCA1_12.crt	A1_12.txt.p7s	Signature file signed by a certificate whose validation data, CRL, is expired
QCA1_13.crt	A1_13.txt.p7s	Signature file signed by a certificate whose validation data, CRL, is signature forged
QCA1_14.crt	A1_14.txt.p7s	Signature file signed by a certificate whose validation data, OCSP response, is expired
QCA1_15.crt	A1_15.txt.p7s	Signature file signed by a certificate whose validation data, OCSP response, is signature forged
QCA1_16.crt	A1_16.txt.p7s	Signature file signed by a certificate whose validation data, OCSP response is signed by an expired OCSP certificate
QCA1_17.crt	A1_17.txt.p7s	Signature file signed by a certificate whose validation data, OCSP response is signed by a signature forged OCSP certificate
QCA1_18.crt	A1_18_1.txt.p7s	Signature file signed by a certificate whose validation data, OCSP response is signed by a revoked OCSP certificate
QCA2.crt	A2.txt.p7s	Signature file signed by a certificate whose sub root certificate is signature forged
QCA3.crt	A3_1.txt.p7s	Signature file signed by a certificate whose sub root certificate is revoked in CRL

Continued

Signer Certificate	Signature File	Signature Property
QCA4.crt	A4.txt.p7s	Signature file signed by a certificate whose sub root certificate' revocation data, CRL, is expired
QCA5.crt	A5.txt.p7s	Signature file signed by a certificate whose sub root certificate' revocation data, CRL, is signature forged
QCA6.crt	A6_1.txt.p7s	Signature file signed by a certificate whose sub root certificate is revoked in OCSP
QCA7.crt	A7.txt.p7s	Signature file signed by a certificate whose sub root certificate' revocation data, OCSP response, is expired
QCA8.crt	A8.txt.p7s	Signature file signed by a certificate whose sub root certificate' revocation data, OCSP response, is signature forged
QCA9.crt	A9.txt.p7s	Signature file signed by a certificate whose sub root certificate' revocation data, OCSP response, is signed by an expired OCSP certificate
QCA10.crt	A10.txt.p7s	Signature file signed by a certificate whose sub root certificate' revocation data, OCSP response, is signed by a signature forged OCSP certificate
QCA11.crt	A11_1.txt.p7s	Signature file signed by a certificate whose sub root certificate' revocation data, OCSP response, is signed by a revoked OCSP certificate
QCB.crt	B.txt.p7s	Signature file signed by a certificate whose root certificate is signature forged
QCC.crt	C_1.txt.p7s	Signature file signed by a certificate whose root certificate is revoked in CRL
QCD.crt	D.txt.p7s	Signature file signed by a certificate whose root certificate' revocation data, CRL, is expired
QCE.crt	E.txt.p7s	Signature file signed by a certificate whose root certificate' revocation data, CRL, is signature forged
QCF.crt	F_1.txt.p7s	Signature file signed by a certificate whose root certificate is revoked in OCSP
QCG.crt	G.txt.p7s	Signature file signed by a certificate whose root certificate' revocation data, OCSP response, is expired

Continued

Signer Certificate	Signature File	Signature Property
QCH.crt	H.txt.p7s	Signature file signed by a certificate whose root certificate' revocation data, OCSP response, is signature forged
QCI.crt	I.txt.p7s	Signature file signed by a certificate whose root certificate' revocation data, OCSP response, is signed by an expired OCSP certificate
QCJ.crt	J.txt.p7s	Signature file signed by a certificate whose root certificate' revocation data, OCSP response, is signed by a signature forged OCSP certificate
QCK.crt	K_1.txt.p7s	Signature file signed by a certificate whose root certificate' revocation data, OCSP response, is signed by a revoked OCSP certificate

4.5.2 ES-T Specific Scenarios

In this section we will give implementation details of ES-T signature package excluding the BES-EPES scenarios which are defined above. Figure 4.14 shows the ES-T specific signature scenarios.

Table 4.14: ES-T Specific Signature Scenarios

Signer Certificate	Signature File	Signature Property
QCA1_1.crt	A1_1_8.txt.p7s	Signature file with "signatureTimeStamp" whose "messageImprint" field in "TSTInfo" is forged
	A1_1_9.txt.p7s	Signature file with "signatureTimeStamp" taken from time stamp server TSA2 such that token's signature is forged.
	A1_1_10.txt.p7s	Signature file with "signatureTimeStamp" taken from time stamp server TSA3 such that time stamp certificate is expired.
	A1_1_11.txt.p7s	Signature file with "signatureTimeStamp" taken from time stamp server TSA4 such that time stamp certificate is signature forged.
		Continued

Signer Certificate	Signature File	Signature Property
	A1_1_12_1.txt.p7s	Signature file with ”signatureTimeStamp” taken from time stamp server TSA5 such that time stamp certificate is revoked before the signing time.
	A1_1_12_2.txt.p7s	Signature file with ”signatureTimeStamp” taken from time stamp server TSA5 such that time stamp certificate is revoked after the signing time.
	A1_1_13.txt.p7s	Signature file with ”signatureTimeStamp” taken from time stamp server TSB such that time stamp certificate’s issuer certificate’s signature forged.
	A1_1_14_1.txt.p7s	Signature file with ”signatureTimeStamp” taken from time stamp server TSC such that time stamp certificate’s issuer certificate is revoked before the signing time
	A1_1_14_2.txt.p7s	Signature file with ”signatureTimeStamp” taken from time stamp server TSC such that time stamp certificate’s issuer certificate is revoked after the signing time
	A1_1_15.txt.p7s	Signature file with ”signatureTimeStamp” taken from time stamp server TSD such that time stamp certificate’s issuer certificate’s revocation data, CRL, is expired
	A1_1_16.txt.p7s	Signature file with ”signatureTimeStamp” taken from time stamp server TSE such that time stamp certificate’s issuer certificate’s revocation data, CRL, is signature forged
	A1_1_17_1.txt.p7s	Signature file with ”signatureTimeStamp” taken from time stamp server TSF such that time stamp certificate’s issuer certificate is revoked in OCSP before signing time
	A1_1_17_2.txt.p7s	Signature file with ”signatureTimeStamp” taken from time stamp server TSF such that time stamp certificate’s issuer certificate is revoked in OCSP after signing time

Continued

Signer Certificate	Signature File	Signature Property
	A1_1_18.txt.p7s	Signature file with ”signatureTimeStamp” taken from time stamp server TSG such that time stamp certificate’s issuer certificate’s revocation data, OCSP response, is expired
	A1_1_19.txt.p7s	Signature file with ”signatureTimeStamp” taken from time stamp server TSH such that time stamp certificate’s issuer certificate’s revocation data, OCSP response, is signature forged
	A1_1_20.txt.p7s	Signature file with ”signatureTimeStamp” taken from time stamp server TSI such that time stamp certificate’s issuer certificate’s revocation data, OCSP response, is signed by an expired OCSP certificate
	A1_1_21.txt.p7s	Signature file with ”signatureTimeStamp” taken from time stamp server TSJ such that time stamp certificate’s issuer certificate’s revocation data, OCSP response, is signed by a aignature forged OCSP certificate
	A1_1_22_1.txt.p7s	Signature file with ”signatureTimeStamp” taken from time stamp server TSK such that time stamp certificate’s issuer certificate’s revocation data, OCSP response, is signed by a revoked OCSP certificate before signing time
	A1_1_22_2.txt.p7s	Signature file with ”signatureTimeStamp” taken from time stamp server TSK such that time stamp certificate’s issuer certificate’s revocation data, OCSP response, is signed by a revoked OCSP certificate after signing time

4.5.3 ES-C Specific Scenarios

In this section we will give implementation details of ES-C signature package excluding the ES-T scenarios which are defined above. Figure 4.15 shows the ES-C specific signature scenarios.

Table 4.15: ES-C Specific Signature Scenarios

Signer Certificate	Signature File	Signature Property
QCA1_1.crt	A1_1_23_1.txt.p7s	Signature file without root certificate reference in ”Complete-certificate-references” attribute
	A1_1_23_2.txt.p7s	Signature file with edited root certificate reference in ”Complete-certificate-references” attribute
	A1_1_24_1.txt.p7s	Signature file without sub root certificate reference in ”Complete-certificate-references” attribute
	A1_1_24_2.txt.p7s	Signature file with edited sub root certificate reference in ”Complete-certificate-references” attribute
	A1_1_25_1.txt.p7s	Signature file without sub root revocation data reference in ”Complete-revocation-references” attribute
	A1_1_25_2.txt.p7s	Signature file with edited sub root revocation data reference in ”Complete-revocation-references” attribute
	A1_1_26_1.txt.p7s	Signature file without qualified certificate revocation data reference in ”Complete-revocation-references” attribute
	A1_1_26_2.txt.p7s	Signature file with edited qualified certificate revocation data reference in ”Complete-revocation-references” attribute

4.5.4 ES-XL Scenarios

In this section we will give implementation details of ES-XL signature package excluding the ES-C scenarios which are defined above. Figure 4.16 shows the ES-XL specific signature scenarios.

Table 4.16: ES-XL Specific Signature Scenarios

Signer Certificate	Signature File	Signature Property
QCA1_1.crt	A1_1_27_1.txt.p7s	Signature file without root certificate value in ”Certificate-values” attribute
	A1_1_27_2.txt.p7s	Signature file with edited root certificate value in ”Certificate-values” attribute

Continued

Signer Certificate	Signature File	Signature Property
	A1_1_28_1.txt.p7s	Signature file without sub root certificate value in ”Certificate-values” attribute
	A1_1_28_2.txt.p7s	Signature file with edited sub root certificate value in ”Certificate-values” attribute
	A1_1_29_1.txt.p7s	Signature file without sub root revocation value in ”Revocation-values” attribute
	A1_1_29_2.txt.p7s	Signature file with edited sub root revocation value in ”Revocation-values” attribute
	A1_1_30_1.txt.p7s	Signature file without qualified certificate revocation value in ”Revocation-values” attribute
	A1_1_30_2.txt.p7s	Signature file with edited qualified certificate revocation value in ”Revocation-values” attribute

4.5.5 X Type 1 - X Type 2 - XL Type 1 - XL Type 2 - Archive Scenarios

In this section we will give implementation details of ES-X1, ES-X2, ES-XL1, ES-XL2 and ES-A signature packages. ES-X1, ES-X2 type signatures are established over ES-C by taking a second time stamp where as ES-XL1, ES-XL2 and ES-A type signatures are established over ES-XL by taking a second time stamp. Since ES-C and ES-XL type signatures' error scenarios are concerned so far, we will concern second time stamp scenarios. Figure 4.17 shows the second time stamp scenarios.

Table 4.17: X Type 1 - X Type 2 - XL Type 1 - XL Type 2 - Archive Signature Scenarios

Signer Certificate	Signature File	Signature Property
QCA1_2.crt	A1_2_6.txt.p7s	Signature file with ”signatureTimeStamp” whose ”messageImprint” field in ”TSTInfo” is forged
	A1_2_7.txt.p7s	Signature file with ”signatureTimeStamp” taken from time stamp server TSA2 such that token's signature is forged.
	A1_2_8.txt.p7s	Signature file with ”signatureTimeStamp” taken from time stamp server TSA3 such that time stamp certificate is expired.

Continued

Signer Certificate	Signature File	Signature Property
	A1_2_9.txt.p7s	Signature file with ”signatureTimeStamp” taken from time stamp server TSA4 such that time stamp certificate is signature forged.
	A1_2_10_1.txt.p7s	Signature file with ”signatureTimeStamp” taken from time stamp server TSA5 such that time stamp certificate is revoked before the signing time.
	A1_2_10_2.txt.p7s	Signature file with ”signatureTimeStamp” taken from time stamp server TSA5 such that time stamp certificate is revoked after the signing time.
	A1_2_11.txt.p7s	Signature file with ”signatureTimeStamp” taken from time stamp server TSB such that time stamp certificate’s issuer certificate’s signature forged.
	A1_2_12_1.txt.p7s	Signature file with ”signatureTimeStamp” taken from time stamp server TSC such that time stamp certificate’s issuer certificate is revoked before the signing time
	A1_2_12_2.txt.p7s	Signature file with ”signatureTimeStamp” taken from time stamp server TSC such that time stamp certificate’s issuer certificate is revoked after the signing time
	A1_2_13.txt.p7s	Signature file with ”signatureTimeStamp” taken from time stamp server TSD such that time stamp certificate’s issuer certificate’s revocation data, CRL, is expired
	A1_2_14.txt.p7s	Signature file with ”signatureTimeStamp” taken from time stamp server TSE such that time stamp certificate’s issuer certificate’s revocation data, CRL, is signature forged
	A1_2_15_1.txt.p7s	Signature file with ”signatureTimeStamp” taken from time stamp server TSF such that time stamp certificate’s issuer certificate is revoked in OCSP before signing time

Continued

Signer Certificate	Signature File	Signature Property
	A1_2_15_2.txt.p7s	Signature file with ”signatureTimeStamp” taken from time stamp server TSF such that time stamp certificate’s issuer certificate is revoked in OCSP after signing time
	A1_2_16.txt.p7s	Signature file with ”signatureTimeStamp” taken from time stamp server TSG such that time stamp certificate’s issuer certificate’s revocation data, OCSP response, is expired
	A1_2_17.txt.p7s	Signature file with ”signatureTimeStamp” taken from time stamp server TSH such that time stamp certificate’s issuer certificate’s revocation data, OCSP response, is signature forged
	A1_2_18.txt.p7s	Signature file with ”signatureTimeStamp” taken from time stamp server TSI such that time stamp certificate’s issuer certificate’s revocation data, OCSP response, is signed by an expired OCSP certificate
	A1_2_19.txt.p7s	Signature file with ”signatureTimeStamp” taken from time stamp server TSJ such that time stamp certificate’s issuer certificate’s revocation data, OCSP response, is signed by a aignature forged OCSP certificate
	A1_2_20_1.txt.p7s	Signature file with ”signatureTimeStamp” taken from time stamp server TSK such that time stamp certificate’s issuer certificate’s revocation data, OCSP response, is signed by a revoked OCSP certificate before signing time
	A1_2_20_2.txt.p7s	Signature file with ”signatureTimeStamp” taken from time stamp server TSK such that time stamp certificate’s issuer certificate’s revocation data, OCSP response, is signed by a revoked OCSP certificate after signing time

CHAPTER 5

CONCLUSION

In this thesis work, we focused on to solve an important lack of digital signature process, security and reliability of signature creation and verification applications. ETSI [12] and CEN [1], [2] 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. This is an important necessity because in so many countries, a soft document signed by a qualified certificate has the same legal responsibilities with its wet signed hard copy. Therefore signature creation and verification applications must be well analyzed and a well defined PKI model must be referenced. From this point of view, we studied on designing a unique PKI model and implemented this model to produce a E-Signature Test Suite. The output of this work is also used by TÜBİTAK to test signature applications in Turkey used by government agencies. This task is given to TÜBİTAK-BİLGEM-KAMUSM by the prime ministry [15],[36].

In chapter 3 and 4, we explained the design theory and implementation details of E-Signature Test Suite. Signature creation part of the suite composed a huge public key infrastructure statistically charted in figure 5.1.

Table 5.1: PKI Hierarchy Statistics

PKI Element Type	**Total**
Root Certificates	11
SubRoot Certificates	21
Qualified End-Entity Certificates	39
Time Stamp Servers	15
OCSP Servers	18
CRLs	39

Also the results of signature verification part is shown in figure 5.2.

Table 5.2: Signature Suite Statistics

Signature Type	**Attached**	**Detached**
CAdES-BES	46	46
CAdES-EPES	46	46
CAdES-EST	70	70
CAdES-ESC	78	78
CAdES-X TYPE 1	93	93
CAdES-X TYPE 2	93	93
CAdES-ESXL	95	95
CAdES-XL TYPE 1	110	110
CAdES-XL TYPE 2	110	110
CAdES-ESA	110	110
	851	851
Total	1702	

Subsequent to implementation of E-Signature Test Suite, we focused on to test the suite with the e-signature libraries of TÜBİTAK. First step was to implement a signature creation and a verification application using TÜBİTAK libraries and we produced the software ETP (E-Signature Test Platform). This software is also produced to create the signature verification scenarios which are theoretically explained in section 3.3 and implementation details are covered in section 4.5.

In tests we realized that we made some mistakes especially in root scenarios. These scenarios are explained in figure 4.13 from signature file ”B.txt.p7s” to ”K_1.txt.p7s”. The mistake was not to split the revocation datas belonging to the root and sub root authorities. In older version, we were using the faulty revocation datas for roots and also for sub roots. In this case, the expected error warning was catched for sub root authority instead of root authority. We solved this problem by creating a valid revocation data for sub root and the targeted invalid revocation data for root authority.

After we test our suite with TÜBİTAK digital signature libraries, we began to test it with other digital signature libraries developed by government agencies or the software developing companies. During these tests we fixed minor bugs about our suite but also realized another issue. Some libraries, for some scenarios related with revocation datas, cathes not the targeted error but another one. The bug is the scenario was including two faulty cases for OCSP response. One of them was the signature of response was forged and the other one was the porducedAt date of the response was expired. Libraries which validates the signature first was catching the targeted error case and the others were catching the

expiration faulty. PKI was not concerning about the order of the validation but solving this issue also increases the efficiency of the signature applications. From this point of view we focused on to solve the problem of ordering the validation of revocation datas.

5.1 Efficiency Recommendations

In section 3.2.3 we mentioned about the scenarios related with validation data, CRL and OCSP. There is no limitation in which order the sub components of revocation datas will be validated but a meaningful approach on the validation order increases the efficiency of signature creation and validation applications.

Our approach on this issue is to begin with the outermost layer of the revocation data and then to continue with the inner layers. Because if there is a problem with the signature of a response then there is no need to cope with the necessary validation items for signer certificate of the response. As stated in section 3.2.3.1 and 3.2.3.2, there are two outer checks with validation datas. One of them is signature check and the other one is the expiration status check. It is clear that checking expiration status costs less than signature check. There are also three inner checks for validation datas related with the signer certificate of the response. First one is the signature check, second one is expiration check and the last one is revocation status check. Revocation check must be the last check since it requires connection with the revocation data serving machines but the other ones are handled without connection. Also the orders of the other two items are same with the outer layer. This validation order should also be applied to time stamp token since the structure of a time stamp token is similar to OCSP response and is also a validation data. The figure 5.3 shows the recommended order of validation.

Table 5.3: Efficient Order of Verification

Order	**Layer**	**Verification Type**	**CRL**	**OCSP Response**	**TS Token**
1	Outer	Validation Data	Expiration Status	Expiration Status	-
2			Signature Control	Signature Control	Signature Control
3	Inner	Signer Certificate	-	Expiration Status	Expiration Status
4			-	Signature Control	Signature Control
5			-	Revocation Status	Revocation Status

REFERENCES

[1] CEN, *CWA 14170 - Security requirements for signature creation applications*, version, May 2004

[2] CEN, *CWA 14171 - General guidelines for electronic signature verification*, version, May 2004.

[3] IETF, *RFC5280 - Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile*, May 2008.

[4] IETF, *RFC3161 - Internet X.509 Public Key Infrastructure Time-Stamp Protocol (TSP)*, August 2001.

[5] IETF, *RFC2560 - X.509 Internet Public Key Infrastructure Online Certificate Status Protocol - OCSP*, June 1999.

[6] IETF, *RFC5652 - Cryptographic Message Syntax*, September 2009.

[7] Electronic Signatures and Infrastructures (ESI), *ETSI TS 101 733 - CMS Advanced Electronic Signatures (CAdES)*, V1.8.1, November 2009.

[8] Electronic Signatures and Infrastructures (ESI), *ETSI TS 101 903 - XML Advanced Electronic Signatures (XAdES)*, V1.4.1, June 2009.

[9] Official Gazette, *E-signature law No. 5070, 25355*, January 2004.

[10] ITU-T, *Rec. X.509 - Information technology - Open systems*, November 2008.

[11] Official Journal, *Directive 1999/93/EC of The European Parliement and of The Council*, December 1999.

[12] Electronic Signatures and Infrastructures (ESI), *ETSI TS 102 853 - Signature verification procedures and policies*, V1.1.1, July 2012.

[13] Electronic Signatures and Infrastructures (ESI), *ETSI TS 101 862 - Qualified Certificate profile*, V1.3.1, March 2004.

[14] IETF, *RFC2986 - PKCS 10 Certification Request Syntax Specification*, V1.7, November 2000.

[15] Official Gazette, *Kamu Sertifikasyon Hizmetlerine İlişkin Usul ve Esaslar, 2006/13*, 19 April 2006.

[16] Official Gazette, *Elektronik İmza Kanununun Uygulanmasına İlişkin Usul ve Esaslar Hakkında Yönetmelik, 25692*, 06 January 2005.

[17] Official Gazette, *Elektronik İmza ile İlgili Süreçlere ve Teknik Kriterlere İlişkin Tebliğ, 25692*, 06 January 2005.

[18] Committee Decision, *Nitelikli Elektronik Sertifika, SİL ve OCSP İstek/Cevap Mesajları Profilleri Rehberine İlişkin Kurul Kararı, 2007/DK-77/207*, 18 April 2007.

[19] Committee Decision, *Güvenli Elektronik İmza Oluşturma ve Doğrulama Uygulamaları ile Güvenli Elektronik İmza Formatlarına Dair Usul ve Esaslar Hakkında Kurul Kararı, 2006/DK-77/353*, 01 June 2006.

[20] IETF, *RFC2634 - Enhanced Security Services for S/MIME*, 1999.

[21] ITU-T, *Information technology - Abstract Syntax Notation One (ASN.1): Specification of basic notation*, 1997.

[22] ITU-T, *Specification of Abstract Syntax Notation One (ASN.1)*, 1988.

[23] IETF, *RFC5035 - ESS Update: Adding CertID Algorithm Agility*, 2007.

[24] ITU-T, *Information technology ASN.1 encoding rules: Specification of Basic Encoding Rules (BER), Canonical Encoding Rules (CER) and Distinguished Encoding Rules (DER)*, 2002.

[25] IETF, *Electronic Signature Policies*, 2000.

[26] Electronic Signatures and Infrastructures (ESI), *ETSI TS 101 861 - Time stamping profile*, V1.4.1, July 2011.

[27] ISO, *ISO/IEC 10118-1 Information technology - Security techniques - Hash-functions - Part 1: General*, 2000.

[28] ISO, *ISO/IEC 10118-2 Information technology - Security techniques - Hash-functions - Part 2: Hash-functions using an n-bit block cipher*, 2000.

[29] ISO, *ISO/IEC 10118-3 Information technology - Security techniques - Hash-functions - Part 3: Dedicated hash-functions*, 2004.

[30] ISO, *ISO/IEC 10118-4 Information technology - Security techniques - Hash-functions - Part 4: Hash-functions using modular arithmetic*, 1998.

[31] ANSI, *Public Key Cryptography Using Irreversible Algorithms - Part 2: The Secure Hash Algorithm (SHA-1)*, 1997.

[32] IETF, *Public-Key Cryptography Standards (PKCS) 1: RSA Cryptography Specifications Version 2.1*, 2003.

[33] IEEE, *Standard Specifications For Public Key Cryptography*, 2000.

[34] FIPS, *Secure Hash Standard (SHS)*, 2002.

[35] Electronic Signatures and Infrastructures (ESI), *ETSI ES 201 733 - Electronic Signature Formats*, V1.1.3, May 2005.

[36] Doç. Dr. T. K. Bensghir, F. Topcan, *E-imza: Türkiye'de Kamu Kurumlarında Uygulanması*,TODAİE Yayın No:356, Ankara, Turkey, December 2010.

[37] Electronic Signatures and Infrastructures (ESI), *ETSI TR 102 438 - Application of Electronic Signature Standards in Europe*, V1.1.1, March 2006.

[38] Electronic Signatures and Infrastructures (ESI), *ETSI TS 102 176-1 - Algorithms and Parameters for Secure Electronic Signatures; Part 1: Hash functions and asymmetric algorithms*, V2.0.0, November 2007.

[39] Alfred J. Menezes, Paul C. van Oorschot, Scott A. Vanstone, *Handbook of Applied Cryptography*, CRC Press, August 2001.

[40] Bruce Schneier, *Applied Cryptography*, John Wiley, 1996.

[41] IETF, *Key words for use in RFCs to Indicate Requirement Levels*, March 1997.

VITA

PERSONAL INFORMATION

Surname, Name	Ergun, Tamer
Nationality	T.C.
Date and Place of Birth:	19 March 1981, Samsun
Martial Status	Married
Phone	+90 535 276 94 22
Mail	tamer.ergun@tubitak.gov.tr

PROFESSIONAL EXPERIENCE

TÜBİTAK 2005-present

BİLGEM, Full-Time

Designing and implementing E-Signature Test Suit, a huge set of certificates and signatures, designed to analyze the accuracy of digital signature applications working mechanism

Designing and implementing ETP(E-Signature Test Software) to create E-Signature Test Suit and to test BİLGEM digital signature libraries both in signature creation and verification and KamuSM PKI components

Designing and implementing ÇEVRİM software, a tool for converting forged signature files to valid files

Designing and implementing other tools (PKCS10 Parser, Digest Calculator) necessary for PKI tasks

Worked on automated cryptanalysis of symmetric crypto systems and implemented tools

Lectured on cryptographic topics

Lecturing on PKI and digital signature topics periodically (every month) in TODAİE (Türkiye ve Orta Doğu Amme İdaresi Enstitüsü)

Worked on other cryptographic projects

TÜBİTAK 1999-2000

BİLTEN, Part-Time

Worked on Test Net project and have role on selecting mathematical problems and arranging with respect to the difficulty level for high school students

PROFESSIONAL INFO

Subject Fields	Mathematics, Cryptology
Skills	Java, LaTeX, C, C++, Python,SQL
Development Environments	Eclipse SDK, Oracle jDeveloper, Kile
	Microsoft Visual Studio 2010
Tools and Libraries	EJBCA (Open Source PKI Certificate Authority)
	TÜBİTAK-BİLGEM Digital Signature Libraries
	Magma, Maple, Matlab
Operating Systems	Unix/Linux (Pardus, RedHat, Ubuntu), Windows 8/7
Languages	English (fluent)
	German (beginner)

EDUCATION

MSc. in Cryptography 2003-2005

Middle East Technical University - Institute of Applied Mathematics

BSc. in Mathematics 1999-2003

Middle East Technical University

High School 1992-1999

Ankara Atatürk High School

CONFERENCES

EFPE 2013 - Lecturer

European Forum On Electronic Signature EFPE 2013, Miedzyzdroje , Poland, 05-07 June 2013

EUROCRYPT 2008 - Attendee

27th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Istanbul, Turkey, April 13-17, 2008

ECRYPT 2007 - Student

Summer School, Emerging Topics in Cryptographic Design and Cryptanalysis, Samos, Greece, April 30- May 4, 2007