An Implementation-independent Evaluation Model for Server-based Signature Solutions

Thomas Zefferer and Bernd Zwattendorfer
Institute for Applied Information Processing and Communications, Graz University of Technology, Inffeldgasse 16a, 8010 Graz, Austria

Keywords: Electronic Signatures, Server-based Signature, Security Evaluation, Evaluation Model.

Abstract: During the past years, a general trend towards server-based signature solutions can be observed. Server-based signature solutions rely on a secure central server component that is able to securely store cryptographic keys and to create electronic signatures on behalf of users. Due to their various advantages compared to client-based solutions, it must be expected that server-based signature solutions will be increasingly deployed in security-critical fields of application in future. This raises the need for appropriate means to systematically evaluate the security of such solutions. Unfortunately, existing evaluation methods (e.g. Protection Profiles according to Common Criteria) are only partly applicable for evaluating server-based signature solutions. To overcome this issue, we propose a new implementation-independent evaluation model for server-based signature solutions. The proposed evaluation model is based on an abstract architectural model for server-based signature solutions and can hence be applied to arbitrary implementations. This way, we provide a powerful instrument to assess the security of future server-based signature solutions and pave the way for their adoption in security-critical fields of application.

1 INTRODUCTION

Electronic signatures have evolved to be an important instrument in online services. Depending on the given legal and organizational framework, the technical realization and implementation of electronic signatures is often no trivial task. For instance, in the European Union (EU) the EU Signature Directive (European Union, 1999) defines strict requirements for the creation of qualified electronic signatures, which are legally equivalent to handwritten signatures.

Considering European law, two basic approaches can be followed to implement solutions for the creation of qualified electronic signatures. *Client-based signature solutions* rely on a signature-creation token that is possessed, controlled, and maintained by the user, i.e. the signatory. Contrary, *server-based signature solutions* rely on a secure central server component that is able to store cryptographic keys and to create qualified electronic signatures on behalf of users.

During the past years, client-based signature solutions relying on smart cards have been the preferred implementation option especially for security-critical application scenarios. Unfortunately, client-based signature solutions usually lack an appropriate level of usability. This often leads to unsatisfactory user acceptance for applications based on qualified electronic signatures. Server-based signature solutions have the potential to overcome usability-related limitations of client-based approaches. An example is the server-based signature solution that has been introduced in Austria in 2010 (A-Trust, 2010). A conducted usability analysis has shown that this solution clearly outperforms existing client-based solutions in terms of user acceptance (Zefferer and Krnjic, 2012). Furthermore, this solution shows that server-based solutions are able to comply with existing legal and organizational requirements.

Due to the given advantages of server-based signatures, it can be expected that there will be an increasing number of server-based signature solutions in future. Hence, new server-based signature solutions will also be deployed in security-critical fields of application. This raises the need for appropriate means to systematically evaluate the security of server-based signature solutions in order to assess their suitability for security-critical fields of applications. Approved evaluation methods for signature solutions such as Common Criteria (Common Criteria, 2013) are already available. However, most of these methods are

Zefferer T. and Zwattendorfer B..
An Implementation-independent Evaluation Model for Server-based Signature Solutions.
DOI: 10.5220/0004839603020309
In *Proceedings of the 10th International Conference on Web Information Systems and Technologies* (WEBIST-2014), pages 302-309
ISBN: 978-989-758-023-9

only partly applicable for server-based signature solutions, as they usually do not properly consider their special characteristics.

To overcome this issue, we propose a new implementation-independent evaluation model for server-based signature solutions. The proposed evaluation model is based on an abstract architectural model for server-based signature solutions and can hence be applied to arbitrary implementations that comply with this architectural model. This way, we provide a powerful instrument to assess the security of future server-based signature solutions in order to assure their suitability and applicability in security-critical fields of application.

The remainder of this paper is structured as follows. In Section 2, basics of and related work on server-based signature solutions are discussed and their growing importance is emphasized. In Section 3, a generic architectural model for server-based signature solutions is introduced. Based on this model, the methodology that has been followed to develop the evaluation model proposed in this paper is presented in Section 4. Based on the presented methodology, an evaluation model for server-based signature solutions is developed and presented in Section 5. Finally, conclusions are drawn in Section 6.

2 SERVER-BASED SIGNATURES

The idea to rely on a trusted server component for the creation of electronic signatures on behalf of users is not new. Early concepts of server-based signature solutions have for instance been proposed in (Ding et al., 2002), (Bicakci and Baykal, 2003), and (Bicakci and Baykal, 2005). In general, server-based signature solutions have various advantages compared to client-based approaches. As computationally complex cryptographic operations are implemented by a server component, users do not need to acquire and maintain additional hardware or software in their client domain that is able to accomplish this task. Furthermore, server-based signature solutions can also be conceptually beneficial in terms of security, if security-critical cryptographic operations are carried out in a protected central environment such as a highly secure data-processing center.

Despite their conceptual advantages, server-based signature solutions have previously led a niche existence especially in security-critical application scenarios. The focus on client-based approaches in Europe was mainly caused by the EU Signature Directive (European Union, 1999), which defines the legal basis for electronic signatures in EU Member States. The EU Signature Directive defines that so-called *advanced electronic signatures* and *qualified electronic signatures*, which are legally equivalent to handwritten signatures, must be *'created using means that the signatory can maintain under his sole control'* (European Union, 1999). For several years, there has been common consent that this passage implies the use of user-owned signature-creation devices such as smart cards, even though this has never been explicitly postulated by this directive.

First discussions on alternative interpretations of the EU Signature Directive have been started in a *Public Statement on Server Based Signature Services* published by the *Forum of European Supervisory Authorities for Electronic Signatures (FESA)*[1]. This document explicitly supports the idea that server-based signature solutions comply with the requirements defined by the EU Signature Directive (Forum of European Supervisory Authorities for Electronic Signatures, 2005). In 2010, a server-based signature solution that supports the creation of qualified electronic signatures has been introduced and deployed in Austria (A-Trust, 2010). This solution has been based on a concept discussed in (Orthacker et al., 2010) and shows that server-based signature solutions are indeed feasible and applicable in security-critical application scenarios.

The growing importance of server-based signature solutions is also considered by legislative bodies and standardization committees. Documents published e.g. by the European Committee for Standardization (CEN) (European Committee for Standardization, 2013) and the European Commission (European Union, 2012) already take the possibility of server-based signature solution into account. The recent development of productive server-based signature solutions and the ongoing work on related standards and legal frameworks in Europe show that server-based signature solutions can be expected to further gain importance in the near future.

In order to provide appropriate means and methods to systematically evaluate the reliability and security of such solutions, we propose a new implementation-independent evaluation model for the security assessment of server-based signature solutions. To achieve implementation independence, the proposed evaluation model bases on a generic architectural model for server-based signature solutions, which is introduced in the following section.

[1] http://www.fesa.eu

3 ARCHITECTURAL MODEL

The evaluation model proposed in this paper focuses on solutions that require users to authorize signature-creation processes by means of a two-factor authentication. These solutions have gained importance during the past years, as they are able to provide the same level of security as client-based signature-creation methods relying on two factors. Server-based signature solutions relying on two-factor based user authentication typically make use of a secret password to cover the authentication factor *knowledge* and of one-time passwords (OTP) being sent to a device owned by the legitimate user[2]. By proving reception of the OTP, the user proves control over the device, which in turn implements the authentication factor *possession*.

An abstract model for a server-based signature solution that relies on the delivery of OTPs to implement two-factor based user authentication is shown in Figure 1. This model comprises two core components. The *System under Evaluation (SuE)* comprises all components of the server-based signature solution that are subject to the evaluation model proposed in this paper. The server-based signature solution itself makes use of and interacts with several external components. These components are subsumed under the abstract component *Environment*.

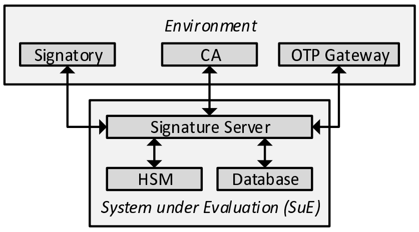

Figure 1: System model.

The SuE basically consists of three components. The *Signature Server* is the central component, which implements most functionality and interacts with external components, i.e. with the Environment, through well-defined interfaces. The Signature Server uses a *Database* to store required data such as user-related and user-identifying information. Additionally, the Signature Server uses a *Hardware Security Module (HSM)* to securely store security-critical data (e.g. signing keys) and to carry out cryptographic operations in a secure environment.

The Environment basically comprises three entities. The *Signatory* is the end user, who aims to create an electronic signature using the SuE. Users usually need to register once at the SuE prior to using its signature-creation functionality. During registration, a cryptographic key pair is generated by the SuE (more precisely by the HSM). While the private key remains at the SuE, the public key is sent to the *Certification Authority (CA)*, which issues a signing certificate for the Signatory. This certificate links the Signatory's identity to the generated key pair. After successful completion of the registration process, the Signatory can use the SuE to create electronic signatures.

To start a signature-creation process, the Signatory sends the *Data to be Signed (DTBS)* to the Signature Server. Before creating the signature, the Signature Server requests the Signatory to enter a secret password. Additionally, the Signature Server generates an OTP and transmits the OTP to the *OTP Gateway*. The OTP Gateway delivers the OTP to the Signatory. By entering the received OTP at the Signature Server, the Signatory completes the two factor-based user authentication process. After successful user authentication, the Signature Server loads required Signatory-related data from the Database and computes the electronic signature on the received DTBS in the HSM. Optionally[3], the Signature Server also displays the DTBS to the Signatory and requests an additional confirmation from the Signatory to sign these data. The computed signature is finally returned to the Signatory.

Based on this implementation-independent architectural model we propose an evaluation model for server-based signature solutions. The methodology followed to develop the proposed evaluation model is discussed in the next section.

4 METHODOLOGY

To rely on an approved method and to follow a systematic approach, we have based our evaluation model for server-based signature solutions on the concepts of Common Criteria (Common Criteria, 2013) and on the Protection Profile for Secure Signature Creation Devices (SSCD) Type 3 (CEN/ISSS, 2001). Development of security recommendations along the concepts of Common Criteria and Protection Profiles

[2] Other approaches to realize two-factor based user authentication are basically feasible. However, OTP-based approaches are currently wide-spread and already used in productive signature solutions such as (A-Trust, 2010). Hence, the evaluation model proposed in this paper focuses on OTP-based solutions only.

[3] In most cases, the applied legal framework determines whether DTBS must be displayed to the user during the signature-creation process or not.

is common practice. For instance, the Council of Europe has applied this methodology in the context of a risk analysis for e-voting solutions (European Council, 2004).

As Common Criteria in general and the Protection Profile for Secure Signature Creation Devices (SSCD) Type 3 in particular have not been developed with server-based signature solutions in mind, we have adapted underlying concepts where necessary to tailor our model to the special characteristics of server-based signature solutions. The resulting methodology that has been followed to develop the proposed evaluation model is shown in detail in Figure 2.

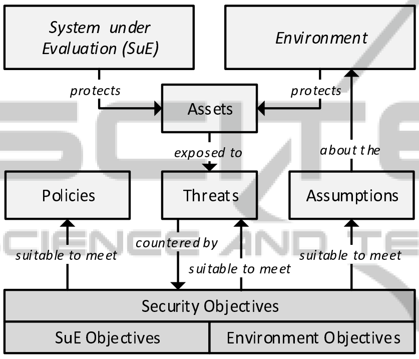

Figure 2: Methodology.

The top of the figure shows the two core components of server-based signature solutions according to the generic architecture introduced in Section 3: the *System under Evaluation (SuE)* and the *Environment*. Both components protect a set of *Assets*. Assets are values that need to be protected, e.g. secret cryptographic keys. Assets are exposed to *Threats*, which potentially compromise the assets' security. The set of potential threats is delimited by a set of *Assumptions* on the environment, in which the SuE is operated. Furthermore, potential threats are delimited by a set of *Policies*. From the made assumptions, defined policies, and identified threats, a set of *Security Objectives* is derived. The derived security objectives counter all identified threats and are also suitable to meet all made assumptions and defined policies.

According to this methodology, the proposed evaluation model for server-based signature solutions has been developed as follows. Based on the generic architecture shown in Figure 1, we have first defined a set of assets that need to be protected by server-based signature solutions. From these assets, potential threats have been identified that threaten to compromise the assets' security. Identification of potential threats has been based on a set of predefined assumptions on the environment and on a predefined set of policies. Assumptions, policies, and threats have then been used to derive security objectives for server-based signature solutions.

Concrete implementations of server-based signature solutions can be assessed with the proposed evaluation model by verifying if the given implementation meets all security objectives and is hence protected against all possible threats. Thus, the developed evaluation model is roughly comparable with a Protection Profile according to Common Criteria, whereas the identified security objectives of our model correspond to security requirements of typical Protection Profiles.

5 EVALUATION MODEL

In this section, we develop an evaluation model for server-based signature solutions following the methodology introduced in Section 4. To facilitate an understanding of analogies and relevant differences between the Protection Profile we rely on and the proposed evaluation model, we intentionally re-use notations that have been defined and introduced in (CEN/ISSS, 2001). In the following subsections, we develop the proposed evaluation model stepwise by following the methodology described above.

5.1 Assets

Assets define values that need to be protected. From the implementation-independent architectural model defined in Section 3, the following assets can be derived for server-based signature solutions:

A.1 *SCD:* SCD (signature-creation data) are private cryptographic keys, which are uniquely assigned to the Signatory (i.e. the user). These keys are used by the Signatory to create electronic signatures. SCD must always be kept confidential and protected from unauthorized access.

A.2 *SVD:* SVD (signature-verification data) are public keys, which are required to verify signatures created with the corresponding SCD. The integrity of SVD must be preserved.

A.3 *DTBS and DTBS display:* DTBS (data to be signed) represent the input to a signature-creation process, i.e. the data being signed by applying a cryptographic signature-creation function. The integrity of these data must be preserved before and during the signing process as well as during displaying these data to the Signatory in the course of the signature-creation process for confirmation purposes.

A.4 *VAD:* VAD (verification authentication data) need to be provided by the Signatory to authorize a signature-creation process. According to the used architectural model, VAD are represented by secret passwords. The authenticity and confidentiality of these data must be preserved.

A.5 *Signature-creation Function:* The cryptographic method used to create electronic signatures must comply with approved quality standards.

A.6 *Electronic Signature:* Created electronic signatures must be tamper-proof.

A.7 *OTP:* The confidentiality and integrity of one-time passwords (OTP) that are used by the Signatory together with her VAD must be preserved.

A.8 *User Data:* The integrity and confidentiality of Signatory-related data stored in the central database must be preserved.

A.9 *User ID:* The user ID unambiguously identifies the Signatory. This is necessary to select the correct SCD for each signature-creation process and to send the OTP to the correct user. The confidentiality and integrity of the user ID must be preserved.

A.10 *Database:* The database stores all required user-related data. The integrity of the database must be preserved and access to the database must be protected.

A.11 *Signature server:* The signature server provides functionality for accessing internal server components such as the database or the HSM. Physical and electronic access to the signature server must be protected and its integrity must be preserved.

A.12 *HSM:* The HSM (hardware security module) provides cryptographic functionality on a secure and reliable basis. Physical and electronic access to the HSM must be protected and its integrity must be preserved.

A.13 *HSM Master Key:* This is a secret key that is unique for each HSM, and which is used to protect the HSM's functionality and data. This key must always be kept confidential.

A.14 *Session ID:* The session ID is used to uniquely map a signature-creation operation to a specific Signatory and facilitates the parallel processing of requests from multiple Signatories. The integrity of the session ID must be preserved.

5.2 Assumptions

The following assumptions define properties and capabilities of the environment, in which the SuE is operated. The made assumptions delimit the set of potential threats for identified assets. All assumptions must be covered by appropriate security objectives.

AS.1 *CGA_Cert:* The Signature Server and the CA (certificate authority) are able to mutually authenticate each other.

AS.2 *CGA_Init:* The CA implements an appropriate registration and identification procedure and transmits relevant information to the Signature Server.

AS.3 *CGA_Secure:* The Signature Server has a secure channel to the CA, which is able to issue appropriate signing certificates according to given legal requirements and to verify that the user has access to her private keys.

AS.4 *OTP_Trusted:* The OTP Gateway is trusted and can be authenticated by the Signature Server.

AS.5 *OTP_Secure:* OTPs can be transmitted from the Signature Server to the OTP Gateway through a secure channel.

AS.6 *User_Trusted:* The Signatory uses a trustworthy end-user system to access signature-creation functionality provided by the Signature Server and to receive OTPs.

AS.7 *User_Secure:* The user communicates with the Signature Server through a secure channel.

AS.8 *Avail:* All external components are available during the registration and/or the signature-creation process.

5.3 Policies

Similar to assumptions on the environment, the definition of policies helps to delimit the set of potential threats. Again, all defined policies must be covered by appropriate security objectives. For the proposed evaluation model, the following policies have been defined:

P.1 *CA_QCert:* The CA is able to issue certificates for the Signatory's public key (SVD) according to the requirements defined by the relevant legal framework.

P.2 *QSign:* The Signatory uses an electronic-signature scheme that complies with given legal frameworks and requirements.

P.3 *Sig_System:* The Signatory's private key (SCD) is virtually unique.

P.4 *HSM_Secure:* The HSM is able to securely import and export data. The HSM does never export its secret master key. The HSM features secure cryptographic key generation and signing functions.

P.5 *HSM_Sig:* Electronic signatures are created inside the HSM only if all required data are available in the HSM and if all components of the SuE are in a defined state.

P.6 *System_Secure:* The SuE is deployed and operated in an appropriately secured environment and its components are protected from unauthorized access.

5.4 Threats

Based on the architectural model defined in Section 3, several threats can be derived that threaten to compromise the security of identified assets. Considering the above-defined assumptions and policies, the following threats can be derived:

T.1 *Hack_Phys:* An attacker accesses physical interfaces of the SuE to mount physical attacks such as side-channel analysis or fault attacks.

T.2 *SCD_Divulg:* An attacker gains access to confidential SCD stored by the SuE.

T.3 *SCD_Derive:* An attacker derives confidential SCD from public data such as SVD or electronic signatures created with the SCD.

Table 1: Assets targeted by threats.

	A.1	A.2	A.3	A.4	A.5	A.6	A.7	A.8	A.9	A.10	A.11	A.12	A.13	A.14
T.1	X		X			X						X	X	
T.2	X				X	X								
T.3	X				X	X								
T.4			X			X								
T.5	X					X								
T.6		X												
T.7			X			X								
T.8					X									
T.9					X		X				X			
T.10							X				X			
T.11					X		X	X			X			
T.12									X					
T.13								X						
T.14								X	X	X			X	
T.15			X	X	X	X	X	X	X	X	X	X		X
T.16					X						X	X	X	
T.17	X				X						X	X	X	
T.18				X										
T.19			X		X	X		X						X
T.20			X		X	X		X						X
T.21			X		X	X	X			X	X	X		X

T.4 *Sig_Forgery:* An attacker forges signed data or the corresponding electronic signature.

T.5 *Sig_Repud:* An attacker compromises the non-repudiation of an electronic signature by modifying signature-relevant data.

T.6 *SVD_Forgery:* An attacker forges the Signatory's public key (SVD) during transmission to the CA in order to compromise the SVD's integrity.

T.7 *DTBS_Forgery:* An attacker modifies the DTBS that are displayed to the user. This way, the Signatory can be tricked into signing unintended data.

T.8 *SigF_Misuse:* An attacker misuses functionality provided by the system to sign arbitrary data on behalf of the legitimate Signatory without her knowledge.

T.9 *OTP_Forgery:* An attacker modifies or blocks delivered OTPs. This way, successful user authentication and hence signature-creation processes can be prevented by the attacker.

T.10 *OTP_Replay:* An attacker uses an intercepted OTP to authorize subsequent signature-creation processes.

T.11*OTP_Derive:* An attacker uses an intercepted OTP to derive useful information such as subsequent OTPs.

T.12 *UserID_Forgery:* An attacker modifies or deletes the unique ID of a Signatory to redirect the delivery of authentication data or to prevent successful signature-creation processes.

T.13 *Userdata_Forgery:* An attacker copies, modifies, deletes, or publishes user-related data to compromise their confidentiality.

T.14 *DB_Forgery:* An attacker steals or destroys the internal database of the SuE to compromise the system's integrity.

T.15 *Server_Forgery:* An attacker steals or destroys the Signature Server to compromise the system's integrity.

T.16 *HSM_Forgery:* An attacker steals or destroys the HSM of the SuE to compromise the system's integrity and to reveal confidential data.

T.17 *HSM_Compr:* An attacker compromises the master key of the HSM to gain access to assets stored inside the HSM.

T.18 *PIN_Compr:* An attacker gains access to the Signatory's VAD. This partly compromises the authentication scheme, which protects the Signatory's centrally stored private keys.

T.19 *Sess_Forgery:* An attacker guesses, modifies, or creates the ID of a session between the Signatory and the SuE. This enables an attacker to terminate an established session or to impersonate the Signatory.

T.20 *Sess_Hijack:* An attacker hijacks an established session between the Signatory and the SuE to gain the same privileges as the legitimate Signatory.

T.21 *System_Malfunc:* A malfunction of one or more components of the system leads to errors during the signature-creation process.

Each identified threat potentially applies to one or multiple assets. The concrete mapping between assets defined in Section 5.1 and threats defined in this section is provided in Table 1.

5.5 Security Objectives

A secure and reliable server-based signature solution must be able to counter all potential threats and meet all made assumptions and defined policies. From the identified threats, made assumptions, and defined policies listed above, we can therefore derive the following set of security objectives:

O.1 *Lifecycle_Security:* The SuE must be able to detect errors during initialization, personalization, and operation.

O.2 *SCD_Secrecy:* The confidentiality of the Signatory's private key (SCD) must be guaranteed.

O.3 *SCD_SVD_Corresp:* The SuE assures the correct relation between the Signatory's private and public key. Upon request, the SuE must be able to verify the relation between private and public key.

O.4 *SVD_Auth:* The SuE provides means that enable the CA to verify the authenticity of the Signatory's public key (SVD).

O.5 *Tamper_ID:* The SuE is able to detect physical manipulations of its components.

O.6 *Tamper_Resistance:* The SuE is resistant against manipulations of its components.

O.7 *Tamper_Response:* The SuE performs appropriate actions to protect components or data when tampering of the SuE has been detected.

O.8 *Init:* The SuE provides appropriate means to assure that private and public keys can be created by authorized persons only.

O.9 *SCD_Unique:* The SuE guarantees an appropriate level of quality for cryptographic keys. Private keys (SCD) must be virtually unique and must not be derivable from corresponding public keys (SVD).

Table 2: Threats, assumptions, and policies covered by security objectives.

Threats/Assumptions/Policies vs. Objectives	O.1 Lifecycle_Security	O.2 SCD_Secrecy	O.3 SCD_SVD_Corresp	O.4 SVD_Auth	O.5 Tamper_ID	O.6 Tamper_Resistance	O.7 Tamper_Response	O.8 Init	O.9 SCD_Unique	O.10 DTBS_Integrity	O.11 Sig_SigF	O.12 Sig_Secure	O.13 HSM_Secure	O.14 HSM_Trusted	O.15 HSM_Feature	O.16 HSM_Sig	O.17 Server_Trusted	O.18 System_Secure	O.19 VAD_Secure	O.20 DB_Access	O.21 DB_Secure	O.22 DB_Encrypt	O.23 DB_Bound	O.24 System_Avail	O.25 CA_QCert	O.26 SVD_Auth_CA	O.27 CA_Auth	O.28 VAD_Auth	O.29 VAD_Secure	O.30 User_Trusted	O.31 User_Secure	O.32 Avail
T.1 Hack_Phys		X			X	X	X																									
T.2 SCD_Divulg		X												X		X				X	X	X	X									
T.3 SCD_Derive									X			X		X																		
T.4 Sig_Forgery	X	X	X	X	X	X	X					X		X		X									X	X						
T.5 Sig_Repud	X	X	X	X	X	X	X		X	X	X	X		X		X				X	X	X	X		X	X						
T.6 SVD_Forgery				X																						X						
T.7 DTBS_Forgery										X			X	X			X													X	X	
T.8 SigF_Misuse										X	X		X	X		X	X			X	X	X										
T.9 OTP_Forgery																			X									X	X	X	X	
T.10 OTP_Replay																	X		X									X	X	X	X	
T.11 OTP_Derive																	X		X													
T.12 UserID_Forgery																	X			X	X	X										
T.13 Userdata_Forgery																	X			X	X	X										
T.14 DB_Forgery																		X					X	X								
T.15 Server_Forgery																	X	X						X								
T.16 HSM_Forgery																		X					X	X								
T.17 HSM_Compr														X																		
T.18 PIN_Compr													X								X									X	X	
T.19 Sess_Forgery																	X							X						X	X	
T.20 Sess_Hijack																														X	X	
T.21 System_Malfunc																	X							X								
AS.1 CGA_Cert																											X					
AS.2 CGA_Init																									X							
AS.3 CGA_Secure																											X					
AS.4 OTP_Trusted																												X				
AS.5 OTP_Secure																													X			
AS.6 User_Trusted																														X		
AS.7 User_Secure																															X	
AS.8 Avail																																X
P.1 CA_QCert			X																						X							
P.2 QSign											X	X													X							
P.3 Sig_System								X	X		X																					
P.4 HSM_Secure													X	X	X						X	X										
P.5 HSM_Sig														X		X																
P.6 System_Secure																		X														

O.10 *DTBS_Integrity:* The SuE must assure that DTBS are not modified while transmitted to the HSM. Furthermore, DTBS must not be modified when being displayed to the Signatory. The SuE must assure that exactly the same DTBS are displayed and signed by the HSM.

O.11 *Sig_SigF:* The SuE provides signature-creation functionality to legitimate users only and protects a Signatory's private key from access and use by others.

O.12 *Sig_Secure:* The SuE creates electronic signatures, which cannot be forged without knowledge of the private key (SCD). The private key cannot be derived from created signatures.

O.13 *HSM_Secure:* The HSM provides appropriate means to securely transmit security-critical data such as SCD or DTBS.

O.14 *HSM_Trusted:* The trustworthiness of the HSM and its cryptographic operations must be guaranteed.

O.15 *HSM_Feature:* The HSM must be designed and implemented such that an error-free and correct operation is assured.

O.16 *HSM_Sig:* The HSM is able to carry out all required cryptographic operations in a protected environment.

O.17 *Server_Trusted:* The signature-server component must be implemented according to its specification and must process signature-creation requests as expected.

O.18 *System_Secure:* It must be assured that the SuE and all of its components are protected against physical and electronic intrusion and that signature-creation processes are carried out correctly.

O.19 *VAD_Secure:* Authentication data (VAD) that are used to authenticate Signatories prior to signature-creation processes must be protected against misuse by unauthorized persons.

O.20 *DB_Access:* Access to the SuE's database must be restricted to authorized persons and components.

O.21 *DB_Secure:* Data transmissions to the database from other components must rely on secure communication channels to prevent eavesdropping and modification of transmitted data.

O.22 *DB_Encrypt:* The SuE must assure that user-related data stored in the database can be accessed by the correct user and can be used for signature-creation processes of this user only.

O.23 *DB_Bound:* Private keys (SCD) must be bound to the HSM. Furthermore, the use of private keys must be limited to the signing of DTBS.

O.24 *System_Avail:* Technical and organizational means must be in place to assure the availability of the SuE and to guarantee that no data are lost in case of system errors or crashes.

O.25 *CA_QCert:* The CA creates certificates according to the given legal requirements.

O.26 *SVD_Auth_CA:* The CA verifies the integrity of the obtained public key (SVD), checks the origin of the public key, and verifies the relation between public key and private key.

O.27 *CA_Auth:* The CA must authenticate at the SuE. Furthermore, the SuE and the CA must communicate over an authenticated and secure channel.

O.28 *VAD_Auth:* The OTP Gateway must be able to prove its trustworthiness.

O.29 *VAD_Secure:* The SuE and the OTP Gateway must communicate over an authenticated and secure channel.

O.30 *User_Trusted:* The Signatory must assure that the used end-user system is trustworthy and free from malware and other sources of interference.

O.31 *User_Secure:* The Signatory must assure that her communication with the SuE relies on a secure and trustworthy channel.

O.32 *Avail:* Technical and organizational means must be in place to assure the availability of external components.

By meeting the derived security objectives, which finally represents the developed evaluation model, server-based signature solutions are able to counter all identified threats and meet all made assumptions and defined policies. Each security objective covers one or multiple threats, assumptions, or policies. The concrete mapping between security objectives, threats, assumptions, and policies is provided in Table 2.

6 CONCLUSIONS

In this paper we have proposed an evaluation model for the systematic assessment of arbitrary server-based signature solutions. The proposed model basically defines a set of implementation-independent security objectives. Security objectives have been derived following an elaborate methodology aligned upon the approved concept of Common Criteria. The security of a concrete server-based signature solution can be assessed with the help of the proposed evaluation model by determining its capability to meet the set of defined security objectives.

Application of the proposed evaluation model to existing server-based signature solutions is regarded as future work and will kill two birds with one stone. First, the soundness of the proposed evaluation model will be evaluated. Second, existing server-based signature solutions will be systematically assessed in order to identify potential security vulnerabilities. By providing a universal evaluation model for arbitrary implementations, the proposed evaluation model helps to assess and assure the security of future server-based signature solutions. This facilitates a future adoption of server-based signature solutions also in security-critical fields of application and paves the way for secure *and* usable online services based on electronic signatures.

REFERENCES

A-Trust (2010). Activate mobile phone signature. http://www.buergerkarte.at/en/activate-mobile.html.

Bicakci, K. and Baykal, N. (2003). Saots: A new efficient server assisted signature scheme for pervasive computing. In Hutter, D., Mller, G., Stephan, W., and Ullmann, M., editors, *SPC*, volume 2802 of *Lecture Notes in Computer Science*, pages 187–200. Springer.

Bicakci, K. and Baykal, N. (2005). Improved server assisted signatures. *Computer Networks*, 47(3):351–366.

CEN/ISSS (2001). Protection profile - secure signature creation device type 3. http://wwww.commoncriteriaportal.org/files/pp files/pp0006b.pdf.

Common Criteria (2013). Common criteria. http://www.commoncriteriaportal.org/.

Ding, X., Mazzocchi, D., and Tsudik, G. (2002). Experimenting with server-aided signatures. In *NDSS*. The Internet Society.

European Committee for Standardization (2013). Security requirements for trustworthy systems supporting server signing. https://shop.austrian-standards.at/Preview.action?preview=&dokkey=478405.

European Council (2004). Multidisciplinary ad hoc group of specialists on legal, operational and technical standards for e-enabled voting (ip1-s-ee) b. explanat.

European Union (1999). Directive 1999/93/ec of the european parliament and of the council of 13 december 1999 on a community framework for electronic signatures.

European Union (2012). Proposal for a regulation of the european parliament and of the council on electronic identification and trust services for electronic transactions in the internal market.

Forum of European Supervisory Authorities for Electronic Signatures (2005). Public statement on server based signature services. http://www.fesa.eu/public-documents/PublicStatement-ServerBasedSignatureServices-20051027.pdf.

Orthacker, C., Centner, M., and Kittl, C. (2010). Qualified mobile server signature. In *Proceedings of the 25th TC 11 International Information Security Conference*.

Zefferer, T. and Krnjic, V. (2012). Usability evaluation of electronic signature based e-government solutions. In *Proceedings of the IADIS International Conference WWW/INTERNET 2012*, pages 227 – 234.