

This is an HTML working draft that led to an article publication. A reference to this work should always be done using the following citation:

[Dimitrios Lekkas](#), [Costas Lambrinoudakis](#), "Outsourcing Digital Signatures: A solution to key management burden", *Information Management & Computer Security*, Vol.14, No.5 (2006) pp.435-448 ([doi:10.1108/09685220610707449](https://doi.org/10.1108/09685220610707449))

This material is presented to ensure timely dissemination of research and technical work. Copyright and all rights therein are retained by authors or by other copyright holders. All persons copying this information are expected to adhere to the terms and constraints invoked by all the copyright holders. In most cases, these works may not be reposted or distributed without the explicit permission of the copyright holders.

Outsourcing Digital Signatures: A solution to key management burden

Dimitrios Lekkas

Dept. of Product and Systems Design Engineering, University of the Aegean,
Syros GR-84100, Greece (email: dlek@aegean.gr)

Costas Lambrinoudakis

Dept. of Information and Communication Systems Engineering, University of the Aegean, Samos
GR-83200, Greece (email: clam@aegean.gr)

Abstract

Digital signatures are only enjoying a gradual and reluctant acceptance, despite the long existence of the relevant legal and technical frameworks. One of the major drawbacks of client-generated digital signatures is the requirement for effective and secure management of the signing keys and the complexity of the cryptographic operations that must be performed by the signer. Outsourcing digital signatures to a Trusted Third Party would be an elegant solution to the key management burden. We investigate whether this is legally and technically feasible and we propose a framework for outsourced digital signatures. In our approach a relying party trusts a *Signature Authority* for the tokens it issues, rather than a Certification Authority for the certificates it creates in a traditional PKI scheme. Given that a signing request is strongly authenticated, we argue that passing the control of signature creation to a Signature Authority rather than the signer

herself, is not a stronger concession than the dependence on an identity certificate issued by a Certification Authority.

Keywords

Security, Legal conformance, Trust, Signature Authority, Signature token

1. Introduction

Digital signatures appeared back in 1970s (Rivest, 1978) signifying a very promising evolution: they were better than handwritten signatures, unforgeable, uncopyable and they could not be repudiated. Today, digital signatures are increasingly gaining momentum in our everyday electronic transactions, but their usage is still limited to a minority of users and applications. A digital signature preserves basic security characteristics of a digital document, such as integrity and authenticity of the binary data, while it signifies the intention of the signer at the moment of signing. European law, the electronic signatures directive (European Union, 1999), further increases the value of electronic signatures, by granting them legal validity equivalent to traditional hand-written signatures.

Although the mathematics involved in digital signing are complex, the main concept of the signing procedure is simple. It is based on public key cryptography, where the signer possesses a key pair, one called the private key and the other called the public key. The public key is publicly known to all interested parties through an open directory, while the private key remains a secret for the signer and cannot be mathematically derived from the public key. The signer encrypts (signs) a sequence of data using her private key and the verifier of the signature (called the *Relying Party - RP* hereinafter) ensures the originality of the data by decrypting the signature using the public key of the signer and obtaining the original data (Rivest, 1978).

So why are digital signatures receiving only gradual and reluctant public acceptance, although they are legally acceptable and the technology is present for so many years? A major disadvantage of digital signatures is that people are reluctant to place their trust in a system that requires a high level of mathematical knowledge to understand (Gelbord, 2000). The poor usability of the software and the low comprehensibility of the concept from the average user's perspective is an inhibitory fact (Burmester, 2004). Several questions arise that cannot be answered by a simple user, who is about to perform a legally binding action: *Can the software and hardware used for the signing action be trusted to perform correctly the necessary calculations? Who is using the signature-creation-data? Does the signer perform a willful act? Does the signed binary data represent correctly the semantics that the signer grasps?* (part of the last question is also known as the 'What You See Is What You Sign' problem (Sceibelhofer, 2001). The above questions demonstrate that there is a large semantic gap between a signature

created by a computer and the understanding of the signer (Schneier, 2000). The problems of mistrust and reluctance and the worries of a user are further intensified by the legal consequences that the action of signing may impose.

Another practical problem preventing the widespread employment of digital signatures is the requirement for the existence of a trusted directory of public keys. This is an enormous logistical task, open to both accidental and malicious failures. Today's digital signature technology is useless without the Public Key Infrastructure (PKI) i.e. a trust model that involves authorities trusted by the users (called Trusted Third Parties - TTPs) which certify that a key belongs to a particular physical entity. A TTP acting as a Certification Authority issues digital certificates that bind a public key to a physical entity and publishes this information. Since there are no strong proposals today that eliminate the need for TTPs, the whole procedure of verifying a digital signature heavily depends on the testimony of a TTP. In other words, without the PKI, we can prove that a particular key created a signature but there is no way to prove to whom this key belongs.

Our objective is to investigate whether it is legally and technically feasible to outsource the process of generating digital signatures, on behalf of a client. Outsourcing digital signatures to an external authority would eliminate several problems on the client side, mostly related to the key management procedures. Additionally, the paradigm of *security services outsourcing* is in line with the typical Application Service Provision that is the fastest growing business model today. However outsourcing of signing applications brings new security and privacy threats that have to be identified and mitigated. In section 2 we will exhibit the serious weaknesses and risks that the client-side key management implies. In section 3 we investigate the technical and legal feasibility of our approach. In section 4 we compare various characteristics of client-generated signatures versus the outsourced digital signatures. In section 5 we propose a specific technical approach for the implementation of a signature outsourcing scheme and finally in section 6 we conclude the paper.

2. Key management: A weakness on the client side

A fundamental intrinsic problem of digital signatures (Mauer, 2003) is that the usage of the private key (i.e. the signature-creation-data) for the creation of a digital signature is not directly controlled by a physical entity, but only indirectly through a machine and an application. The risk lies in the fact that the usage of keys for the calculation of a digital signature is performed transparently by hardware and software (the signature-creation-device) that is mostly unknown and non-trusted for the end-user and that may also be malicious or at least unreliable. As a result, one may be held liable for a signature created by his private key, without his awareness or consent.

The management of the keys used to create and verify digital signatures, is an extremely complicated process. Several considerations arise throughout the lifespan of the signing keys,

which cannot be answered in a straightforward manner by a user without deep knowledge on cryptography. As a result, fear and reluctance against this technology is augmented, especially when the action of signing has legal consequences. It is true (Oppliger, 2000) that there is a plethora of cryptographic algorithms embedded in several widely used applications. These algorithms remain strong for years despite their exposure to the public. The vast majority of attacks against cryptographic applications do not exploit a weakness of the algorithm itself, but a vulnerability of the application that handles the cryptographic keys (Ellison 2000). Consequently the worries of the users on the key management aspects are not unjustified. A list of questions that will probably emerge follows:

- *How to create* my key pair? It has been proved that it is possible to have weak keys, despite the strength of the cryptographic algorithm. This kind of problem usually originates from the implementation of random number generators.

- *How to protect* my private key? Cryptographic keys may be stored in various storage means in an encrypted form. However this approach leads back to the same questions on how to handle the key that decrypts the protected key. Tamper-proof devices such as smart cards for creating and storing keys may reduce the risk of key compromise, but there is still the risk of smart card mishandling, theft and PIN tapping.

- *Who is using it?* In spite of the existence of a secure binding between a key and a physical entity, assured by a Trusted Third Party, by means of a digital certificate, it is inherently impossible for a relying party to determine when, how and by whom a digital signature was generated, unless it is time-stamped and notarized by a properly authorized notary within a trusted environment. In other words there is no way to prove that the creator of a digital signature guarantees his awareness and that he performs a conscious and willful act and that the key used is in fact under her absolute control. This fact is the basic weakness of digital signatures comparing to the hand-written signatures – which although are easy to forge, sometimes not-recognizable and not securely bind to one person – their creation is under the direct control of the signer and directly bind to material (a piece of paper).

- *How are the applications handling it?* Most of the key mishandling cases happen on the application level. Several worrying questions are identified here: Is the key used in a protected manner with no leakages to other applications or interfaces? Are there any traces of the key left in the memory after the execution of the signing process? Is the correct data and only this data presented for signature?

It is clear that key management on the client side is a serious drawback that severely increases the mistrust of the users against the signing procedures. In the next sections we will examine whether it is possible to avoid the existence of client keys at all and have the signatures created in an external trusted and secure environment.

2.1 *Secure Signature-creation-devices*

Although regular computing systems may be considered rather untrustworthy for the creation of digital signatures, it is generally accepted that PKI-enabled smart cards provide a high level of security in the cryptographic operations needed for digitally signing data. They conform to the European directive on digital signatures that requires secure signature-creation-devices that remain under the control of the signer (European Union, 1999). Smart cards are generally considered as tamper-proof devices, while their usage is authorized by means of a personal identification number (PIN). The private key is generated and securely stored within the smart card, declining any kind of export or usage of the private key outside the chip of the smart card.

However, a smart card has no direct I/O interface to the user and its operation depends on a computer with the necessary APIs. Due to this fact, we may identify two major problems that lead us again to the problem of untrustworthy systems:

- When a signer creates a digital signature she must trust the operating system, the user interface and the card API, in terms that they present to the smart card the correct data for signing.
- Although the private key is used within the trusted environment of the smart card, the PIN is given to a dialog presented by the terminal. Thus, malicious software may be able to reuse the PIN to sign another portion of data within the smart card.

As a result, the problem of untrustworthy key management (even with the use of smart cards) may lead the signer to the repudiation of having created a signature. There are several reasons why one may *deny the creation of a signature*, such as:

- The signature is created without the consent of the owner of the private key, due to a virus, a malicious application or an unauthorized usage of her personal computer.
- The private key is compromised and a signature is created by an unauthorized entity, before the key compromise is detected, reported to the authority and published by means of key revocation mechanisms.
- A digital certificate proved to be invalid, due to CA fault on identity verification or due to algorithm compromise, thus binding a digital signature (precisely, the signature-verification-data) to the wrong physical person.

3. Feasibility of digital signature outsourcing

The feasibility of outsourcing digital signatures to an external entity mainly depends on two factors: Whether such a scenario is technically feasible and whether it is legally acceptable. We will demonstrate how both considerations can be waived.

3.1 Technical approach

The objective is to eliminate the requirement of managing key pairs and performing complex signature creation calculations on the client side. The mechanics of our proposed solution are simple: The client wishing to produce a digital signature on a specific document, prepares a relevant request containing the document to be signed (or equivalently its hash value) together with her identity details. The request is sent to a trusted entity --called the *Signature Authority (SA)* hereinafter, which will generate the signature on behalf of the signer.. A requirement is that the signer has been previously registered with the Signature Authority and strongly authenticated each time she sends a signing request (issue addressed later in section 5.3).

The SA in turn will produce, store and distribute a digital signature on behalf of the signer who requested it. A digital signature in this case will not be the traditional signature where a document's hash value is encrypted with the signer's private key, since there is no such key. A general, legally acceptable definition (European Union, 1999) states that a digital signature is data attached to a document, connected to a unique signer and to the specific document. Accordingly, when a data structure that contains the identity of the signer and the identity of the document (i.e. its hash value) is signed by the SA, it binds the two entities (i.e. the signer and the document) in an unambiguous and non-repudable way. We will call this data structure as *Signature Token (ST)* hereinafter, to differentiate it from the traditional notion of digital signature.

The entity wishing to verify the validity of the signature token - the Relying Party - will have to trust the Signature Authority, in exactly the same way a RP trusts the issuer of digital certificates within the Public Key Infrastructure. The verifier checks the signature of the ST taking into consideration the trust relationship with the SA and the algorithmic validity of the signature value. She finally checks the signer's identity details and whether the hash value contained in the ST matches the one of the signed document.

Signatures are usually bind to the signed content either in the form of an integrated file containing both content and signatures (attached signature) or in two separate files distributed simultaneously (detached signature) (Bartel, 2002). The basic advantage of attached signatures is that the signature accompanies the document and thus no effort is needed for retrieving or distributing its signature from an external source. The benefit from the detached scheme is that the signed document may be kept in its initial format and its signature can be archived and retrieved when needed. The same two alternatives (attached or detached) apply to the proposed signature tokens. Since the communication of a signed document is not performed only between a signer and a RP, but a third party is also involved , the detached form of signature tokens is strongly preferred. The reason is that the document may be communicated between the interested parties without disclosing its contents to the Signature Authority, while its signature may be independently created, stored and distributed.

3.2 *Trusting the Signature Authority*

In both the traditional client-generated signature and the signature outsourcing schemes, a

Relying Party trusts directly a Third Party (known also as the ‘trust anchor’) for the information it provides, as shown in Figure 1. This direct relationship may be the result of following a trust path that includes several TTPs, where the trust is transited from one entity to another (e.g. a root TTP and a sub-TTP in hierarchical trust architecture) (Atreya, 2002). For simplicity reasons we assume that a Relying Party trusts directly the TTP that provides the final services.

The trust to the signature is a derived trust relationship. For the case of signature outsourcing the RP trusts the TTP acting as a Signature Authority and consequently she trusts the signature token that is signed by the SA. For the case of client-generated signatures the RP trusts the TTP acting as a Certification Authority and thus she trusts the digital certificate issued to the signer. As a result, she finally trusts the digital signature produced by the keys related to the certificate. We may notice that in the case of client-generated signatures one additional data structure (the certificate) and one additional entity (the signer) are involved in the procedure, compared to the outsourced signature. This fact increases the complexity of client-generated signatures, requires the definition of additional policies and procedures and it may raise additional threats.

One may argue that we give too much authority to one entity, by allowing a third party to produce a signature on behalf of a signer. This is true, but this is not different from the client-generated signatures case. The dependence on a Certification Authority, in the latter case, is as strong as the dependence on a Signature Authority, since a CA certifies who is the owner of a key and therefore who is the creator of a signature. Additionally, on the negative side of client-generated signatures we must notice that we place too much trust to the capabilities of a client to manage cryptographic keys and produce robust cryptographic results. In fact, the only significant difference between the two schemes is that we transfer the liability of managing keys and performing cryptographic actions at the authority’s side, leaving to the client only the concern to communicate securely with the authority (an issue addressed later in the paper).

Figure 1: Trust models in the two alternative schemes

3.3 *Legal conformance*

The delegation of the signature capacity by a legitimate signer to another entity (the *proxy signer* or the *trustee*) is common in our society and acceptable in most national laws. The same concept (signing on behalf of...) can be applied and accepted in digital signing. The action of digital signing is a personal action protected by National and European laws. The basic consideration of the European directive 99/93/EC is the protection of the signatures against forgery and their binding to a unique person and a unique document. Although the content of the digital signatures directive is strongly related to the classic approach, where a digital signature is generated by a client who possesses a digital identity certificate, a detailed examination will show that our proposal conforms to the law as well.

According to the directive, the basic requirements for qualified electronic signatures are:

1. The signature data is uniquely linked to the signer
2. It is possible, through the signature, to determine the identity of the signer
3. It is created by means that the signer preserves under her absolute control
4. It is connected to the signed data so that any subsequent modification on the signed data can be traced

The conformance of our approach to the requirements 1,2 and 4 is straightforward. The Signature Token is a signed data structure and thus the integrity of its contents is preserved. It contains the distinguished name of the signer, so it identifies the signer and it is unambiguously connected to her. The same signed data structure contains the hash value of the signed data, so it is unambiguously connected to the original signed data as well.

The conformance to the 3rd requirement is under question. The signer preserves under her control the preparation of a signing request, but the final creation of the signature is executed by another party. Consequently the TTP acting as a Signature Authority is subject to fraudulent action and specifically to the creation of a binding signature without the consent and the awareness of the signer. We may argue however, that the non-conformance to this requirement is exactly the same as in the traditional digital signing process with personal keys and digital certificates. It is true that a TTP acting as a Certification Authority in this case is also subject to fraudulent action. Specifically, a CA is capable to create a key pair, issue a relevant digital certificate in the name of any person and then generate a valid digital signature on a document, without the slightest awareness of the subject signer. Additionally, the possession of keys on the client side is subject to misuse, as explained in section 2 and further worsens the conformance to the 3rd requirement.

Regarding the legal requirements for a qualified certification provider, the same requirements must apply to the signature provider. These requirements relate to the liability of the provider, its accreditation, the protection of the data used for the creation of the signature tokens, the properness of the personnel and other procedural details. The above requirements still apply and must be met in the case of a signature authority, as proposed in our approach.

Thus, we may conclude that the act of outsourcing digital signatures conforms to the legal requirements as much as the client-generated digital signatures do, since in both cases the conformance depends to a large extent on the fairness and the trustworthiness of a TTP.

4. Outsourced vs. Client-generated digital signatures

In this section we examine several characteristics of the outsourced digital signatures and we compare them to the traditional client-generated signatures. As it will be shown, the outsourcing of digital signatures can be a secure, efficient and cost-effective solution, legally equivalent to the

client-generated signatures and in several cases even superior in terms of administration difficulty and complexity for the end user.

Outsourced signature tokens and client-generated signatures both satisfy the principal requirement for binding the identity of a person to a portion of data in an unambiguous and non-repudable way. Both alternatives are based on PKI trust models (e.g. hierarchical, web-of-trust, bridge) and the verification of signatures depends on the trust relationship against a Trusted Third Party. For an outsourced signature the RP trusts the information contained in a signature token issued by the TTP, while for a client-generated signature the RP trusts the information contained in a digital certificate issued by the TTP, as described in section 3.2. Finally, in both cases the signer must be previously reliably registered with the TTP (acting as Signature Authority or Certification Authority respectively) and the verification of both digital signatures and signature tokens can be performed offline.

One of the major advantages of outsourced digital signatures is the elimination of the need for possessing and managing cryptographic keys on the client-side, avoiding all related problems of client-generated signatures as described in section 2. As a result, there is no need for special signature-creation-devices (e.g. smart cards, card readers and trusted systems) on the client side and therefore the cost for implementing a signature infrastructure is vastly reduced. Digital certificate issuance and management is avoided as well. Another positive consequence for the signer is that the procedure has lower complexity and as such it is more comprehensible. Since signing constitutes a legally binding action, the signer feels more confident and willing to execute a simple comprehensible action, rather than to perform complex cryptographic functions without having the relevant knowledge.

A positive characteristic of outsourced DS is that since they are centrally produced, it is easy to implement a signature repository that could also be globally accessed as a directory. The involvement of an online TTP in the signature creation procedure makes the simultaneous provision of additional services possible, such as timestamping and notary (Adams, 2001). At the pros of outsourced signatures we may also add the possibility of long-term preservation of the signatures since the signing infrastructure at the authority's side is better, the keys can be longer and trust refreshing methods (Lekkas, 2004) may be used. Finally, in order to face possible fraudulent actions the SA may easily implement signature revocation services (Lekkas, 2005) and provide Signature Status Information as part of the centrally generated and archived signatures.

On the negative side for the outsourcing of digital signatures we may notice that this scheme requires the existence of an online, highly available Signature Authority, while client-generated signatures are created offline. Additionally, a secure channel between the signer and the SA is needed for protecting signer's requests and also the signer must be strongly authenticated (see section 5.3). As in every centralized system, system and network performance considerations may arise. However, given that digitally signing is not a real-time intensive application and that the full-text of signed documents is not transmitted, performance is not a real problem for modern systems. An argument against the outsourcing of signatures may be that the Signature Authority

consists a single-point-of-failure, since a compromise of the SA keys would result in the cancellation of every ST it has produced. This problem may be mitigated by using different keys for groups of subscribers or even for every individual subscriber.

The pros and cons of signature outsourcing against the client-side signing are summarized in Table 1. As it is concluded, there is no strong argument against the signature outsourcing, but in the contrary we identified several advantages of our proposed signature outsourcing scheme.

	Characteristics of alternative schemes	
	Outsourced Signatures	Client-generated Signatures
<i>Advantages for outsourced signatures</i>	No key management at the client's side	Client keys required
	No client certificate issuance and management	Identity certificates for the signer required
	No need for trusted systems and secure signature-creation-devices	Trusted signature infrastructure required (smart cards, readers and trusted system)
	Much more comprehensible procedure and readable results	Complex cryptographic functions on the client side
	Global access to signature info, via a centralized directory	Signatures remain local
	Long duration of signatures	Signature lifespan restricted by client key lifespan
	Signature revocation possible	Additional signature archiving required
	Timestamping and notary services can be embedded in the signing procedure	Extra infrastructure, protocols and authorities required
	Reduced cost and learning cycle	Considerable cost for client infrastructure and training
<i>Advantages for client-generated</i>	Online service – High availability required	Offline signature creation
	Strong client authentication and secure channel required	Local authentication

<i>signatures</i>	Performance of central system considerations	Local processes
<i>Common for both schemes</i>	PKI trust model must be established	PKI trust model must be established
	Client registration required	Client certification required
	Offline signature verification is possible	Offline signature verification is possible
	Signature process not absolutely controlled by client (signature creation by third entity)	Signature process not absolutely controlled by client (certificate creation by third entity)

Table 1: Comparison of the characteristics of outsourced signatures vs. client-generated signatures

5. A solution framework for outsourcing digital signatures

Towards the design of a technical solution framework for outsourcing digital signatures, we have to identify the major user requirements and to describe the important procedures and data structures that have to be implemented.

5.1 *User Requirements*

- **Registration:** The client must be registered with the Signing Authority, following the procedures described in the relevant policies and practice statements made by the SA.
- **Authenticity of the signing request:** One of the critical requirements of the signing procedure is the authenticity of the signing request sent by a signer to the SA.
- **Secure binding:** The signature must be uniquely bound to a physical entity (the signer) and the signed document. Since there are no digital certificates binding keys to physical entities, the signature itself must contain an unambiguous reference to both the signer and the signed document.
- **Signature verification:** The Signature Token must be signed by its creator, i.e. the Signing Authority. A Relying Party must be able to decide on the validity of the ST by verifying the authenticity of the signed token.

5.2 *Data structure of the outsourced Signature Token*

We exploit the Extensible Markup Language (XML) and the XML-signature syntax (Cowan, 2001) to present a generic outsourced signature XML schema that ensures openness, readability, and applicability for a wide range of applications. The following schema includes the necessary data structures to implement a detached signature token. The schema uses some elements from the xml-signature namespace (Bartel, 2002).

Based on the previously recorded user requirements and the described technical framework we propose the basic element of the schema, the *oST* (Outsourced Signature Token) structure, which includes the *oSTSignedPart* structure and the relevant signature of the entity that creates the signature token. It optionally includes the *keyInfo* containing information (keys, certificate chains etc.) that enables the relying party to easily obtain the information needed to validate the signature of the outsourcing authority. The element *ssiDistrPoints* includes one or more URLs where a relying party may retrieve information about the status of the signature, as it is required in case signature revocation is permitted (Lekkas, 2005). It may also optionally include a secure timestamp (*timeStampValue*) obtained by a trusted time-stamping authority, in case that a proof on the existence of the SRT at a specific time is critical.

The data that is encapsulated and signed within a oST (*oSTSignedPart*) includes first of all a serial number uniquely identifying the ST, the date of its issuance and the distinguished name of the issuer (the SA). It includes of course the distinguished name of the signer (*signerDN*) and the digest (hash value) of the document (*documentHash*) that the ST binds together. The structure includes also some supporting information, such as the *signatureMethod* of the oST signature (as defined in XMLdsig) the document hashing algorithm (*docHashMethod*) and a reference to the applicable signing policies (*signingPolicy*). The Distinguished Names (DN) for both the signer and the signature authority must preferably follow the global naming rules described in the X.500 directory services standard.

```
<?xml version="1.0" encoding="UTF-8"?>
<schema targetNamespace="http://signatureoutsourcing.org" xmlns="http://www.w3.org/2001/XMLSchema"
xmlns:ds="http://www.w3.org/2000/09/xmlsig#"
xmlns:sr="http://signatureoutsourcing.org">

  <!-- Outsourced Signature Token -->
  <complexType name="oST">
    <sequence>
      <element ref="sr:oSTSignedPart"/>
      <element name="oSTSignatureValue" type="base64Binary"/>
      <element ref="ds:keyInfo"/>
      <element name="ssiDistrPoints" type="anyURI" maxOccurs="unbounded"/>
      <element name="timeStampValue" type="base64Binary" minOccurs="0"/>
    </sequence>
  </complexType>

  <!-- Signature Token Signed Part -->
  <complexType name="oSTSignedPart">
    <sequence>
      <element name="serialNumber" type="integer"/>
    </sequence>
  </complexType>

```

```

    <element name="dateIssued" type="datetime"/>
    <element name="issuerDN" type="string"/>
    <element name="signerDN" type="string"/>
  <element name="documentHash" type="base64Binary"/>
    <element ref="ds:docHashMethod"/>
    <element ref="ds:signatureMethod"/>
    <element name="signingPolicy" type="anyURI"/>
</sequence>
</complexType>

<!-- XMLdsig data element definitions -->
<element name="signatureMethod" type="ds:signatureMethodType"/>
<element name="docHashMethod" type="ds:hashMethodType"/>
<element name="keyInfo" type="ds:keyInfoType"/>

<!-- New data element definitions -->
<element name="OST" type="sr:OSTType"/>
<element name="OSTSignedPart" type="sr:sRTsignedDataType"/>

</schema>

```

5.3 Security considerations

We may identify several security issues related to the implementation of a Signature Authority. Except of the trust establishment described in section 3.2, we may refer to the protection of the keys of the authority, the trustworthiness of the infrastructure used, the physical protection of its premises, the key revocation procedures and the properness of the personnel. However these issues are already faced in traditional public key infrastructures and we do not need to re-examine them.

We have to focus on an important security issue in the proposed framework, which is the authenticity of a signature request sent by a signer to the SA. Failure to correctly authenticate a signing request would result in fraudulent actions of signing or to the repudiation of the signing action by the signer. It is required that the identity of the signing requester is strongly proved, the integrity of the request is preserved and optionally its confidentiality is assured. However we have to avoid using cryptographic keys on the client-side for performing the required strong authentication, since this would lead back to the key management problems described in section 2 and the feasibility of our approach would be in question.

Client authentication procedures must be strictly designed, but they have to remain simple without using certificate-based authentication. To satisfy this requirement we may exploit the paradigm of web banking applications with some additional features, by implementing the

following steps:

- Secure channel is implemented between the signer and the authority, using for example the SSL protocol with server-side authentication (Ferguson, 2003).
- Client authentication is password based, adhering to strict policies and procedures similar to web banking applications. The S/KEY one-time password system would be a nice and secure alternative solution for client authentication, still avoiding the usage of keys (Haller, 1994).
- Secondary manual confirmation of each signing request would further strengthen the authentication procedure. This is implemented by sending an e-mail confirmation request to the officially registered e-mail address of the client, who, in turn, has to enter to a one-time URL confirming her action.

For a centralized service such as the proposed digital signatures outsourcing the performance of the Signature Authority is a security issue, since it affects the availability of the information. Network, storage and processing capabilities must meet some specific standards for servicing bulk signature requests; however this tends to be a minor problem for modern systems. As a rough estimation, the size of a signature request is no more than 1 Kbyte, a signature token with the signature value produced by a 1024-bits key would be at most 3 Kbytes and the CPU cost in a Pentium4 processor for signing the ST is at the level of 1 second.

5.4 *Verifying an authority-generated signature*

It is generally accepted that since the relying party is the entity who is running the risk of a wrong decision, she is the one to pose the requirements on the necessary evidence that must be supplied in order to decide for the validity of a signature (Rivest, 2001). As a general guide, a relying party wishing to verify the validity of a signature token must perform the following actions:

- She is aware of the policies and conditions applying to the outsourced digital signature creation and she trusts the signature authority.
- She checks the algorithmic correctness of the authority's signature on the ST and she confirms that the keys used match the certificate of the trusted authority and they are not revoked.
- She checks the contents of the signed part of the signature content and confirms that the DN of the signer is the one expected and the hash of the document matches the document under examination.
- She checks some additional information, such as the possibility of signature revocation and that the signature time indication is reasonable.

6. Conclusions

The creation of digital signatures by a client involves the difficult and mostly incomprehensible tasks of creating, storing, protecting and securely using cryptographic keys. Key management is a burden for a signer and it raises several risks for the digitally signing procedure. Failure to effectively protect a signing key may lead to the unauthorized usage by malicious users or untrustworthy systems, resulting in the creation of legally binding signatures without the awareness and the consent of the signer.

Outsourcing the process of generating digital signatures is technically and legally feasible. A Relying Party can trust a third party, i.e. a Signature Authority, for creating special data structures (the Signature Tokens) that bind securely and unambiguously a signer and a document. This approach conforms to the majority of legal requirements for digital signatures. We argued that passing the control of signature creation to a Signature Authority rather than the signer herself, is not a stronger commitment than the dependence on an identity certificate issued by a Certification Authority.

Except of the elimination of key management on the client side, we identified several additional advantages for our signature outsourcing approach, such as the reduced cost of the infrastructure needed for signature-creation-devices, the reduced complexity for a regular user, the existence of a centralized archive of signatures and the easiness to embed timestamping and notary services. The most challenging security issue that must be faced is the assurance of the authenticity of the signing request sent by a signer to the authority. This can be achieved by combining strong authentication technologies, avoiding the need to use client keys.

References

- Adams C., Cain P., Pinkas D., Zuccherato R., (2001), "Internet X.509 Public Key Infrastructure Time-Stamp Protocol", IETF Request For Comments 3161, available at <http://www.ietf.org/rfc/rfc3161.txt>
- Atreya M., Hammond B., Paine S., Starrett P., Wu S., (2002) "Digital Signatures", RSA Press – McGraw-Hill, Berkeley, ISBN 0072194820
- Bartel M., Boyer J., Fox B., LaMacchia B., Simon E. (2002), "XML-Signature Syntax and Processing", W3C Recommendation, available at <http://www.w3.org/TR/xmlsig-core/>
- Burmester M., Desmedt Y. G. (2004) "Is hierarchical public-key certification the next target for hackers?", Communications of the ACM, Vol.47, No.8, pp.68-74
- Cowan J., Tobin R. (2001), "XML Information Set", W3C Recommendation, available at <http://www.w3.org/TR/xml-infoset>
- Ellison, C., Schneier, B. (2000) "Ten risks of PKI: What you're not being told about", Computer Security

Journal Vol.16, No.1.

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.

Ferguson N., Schneier B., (2003) "Practical cryptography", Wiley, Indianapolis, ISBN 0471223573

Gelbord B. (2000) "Signing your 0111001010", Communications of the ACM Vol.43, No.12, pp.27-28.

Haller N., Morristown N.J., (1994), "The s/key one-time password system", Symposium on Network and Distributed System Security, 3-4 February 1994, San Diego, California.

Lekkas D., Gritzalis D. (2004), "Cumulative notarization for long-term preservation of digital signatures", Computers & Security, Vol.23 No.5, pp.413-424.

Lekkas D., Gritzalis S., Mitrou L., (2005) "Withdrawing a declaration of will: Towards a framework for Digital Signature Revocation", Internet Research, Vol.15, No.4, pp.400-420

Maurer U. (2003), "Intrinsic limitations of digital signatures and how to cope with them", in Proceedings of the 6th Information Security Conference (ISC'03), LNCS-2851, pp.180-192

Oppliger R. (2000), "Security Technologies for the World Wide Web", Artech House Publishers, USA.

Rivest R.L., Shamir A., Adleman L. (1978), "A method for obtaining digital signatures and public-key cryptosystems", Communications of the ACM, Vol.21 No.2, pp.120-126.

Rivest R. (2001), "Issues in Cryptography", MIT Laboratory for Computer Science, available at <http://theory.lcs.mit.edu/~rivest/Rivest-IssuesInCryptography.pdf>

Sceibelhofer K., (2001) "What You See is What you Sign – Trustworthy display of XML documents for signing and verification", In Communications and Multimedia Security, CMS'01, pp.3-13, Darmstadt.

Schneier B., (2000) "Why digital signatures are not signatures", HIPAA Advisory, available at <http://www.hipaadvisory.com/tech/whynot.htm>.