

# Cumulative Notarization for Long-term Preservation of Digital Signatures

Dimitrios LEKKAS<sup>1</sup> and Dimitris GRITZALIS

Dept. of Informatics, Athens University of Economics and Business (AUEB)  
76 Patission Ave., Athens GR-10434, Greece.  
*e-mail:* [dlek@aegean.gr](mailto:dlek@aegean.gr), [dgrit@aueb.gr](mailto:dgrit@aueb.gr)

## Abstract

The long-term preservation of digitally signed documents may be approached and analyzed from various perspectives, i.e. future data readability, signature validity, storage media longevity, etc. The paper focuses on technology and trust issues related to the long-term validation of a digital signature. We exploit the notarization paradigm and propose a mechanism for cumulative data notarization that results in a successive trust transition towards new entities, modern technologies, and refreshed data. A future relying party will have to trust only the information provided by the last notary, in order to verify the validity of the initial signature, thus eliminating any dependency on ceased entities, obsolete data, and weak old technologies. The proposed framework uses recursive XML elements so that a notarization token structure encapsulates an identical data structure containing a previous notarization token.

## Keywords

Security, trust transitivity, metadata, time-stamping, data encapsulation, certification, Extensible Markup Language (XML).

## 1. Introduction

Documents in digital format are increasingly becoming a common means for a wide range of information types, such as transaction records, books, scientific work, contracts, and even governmental decrees. In several cases these documents must be preserved for long periods of time, for future control and accounting purposes, for evidential reasons or for the protection of an entity's interests. The value of the archived documents depends on the existence of a digital signature, which is the principal expression of an author's intent, while it ensures the integrity of the document. The preservation of the readability, the verifiability and the validity of the digital signature are, thus, crucial for the future value of the documents.

In the current situation there is a considerable gap between the potential longevity of a digital document and the longevity of its digital signature [1]. While the longevity of the document itself depends only on the preservation of its readability, the longevity of digital signature depends on multiple factors, which have shorter or short lifespan. In detail:

---

<sup>1</sup> The paper was compiled while the first author was with the Dept. of Informatics of the Athens University of Economics and Business, in a visiting Lecturer's capacity.

- The keys used for signature creation and signature verification must have limited lifespan in order to avoid long exposure to cryptanalysts and other threats. A common practice of Certification Authorities is to impose a limit of one or two years in the lifespan of issued certificates that are based on a 1024 bit RSA key pair. Although the factorization problem is currently not solvable in a two year period, the evolution of technology may be unpredictable and in addition, long-lived keys are more probable to be lost or stolen.
- Signing keys may be compromised before the completion of their lifespan or the algorithms used for signature creation may be broken, rendering the signature of a document vulnerable to modification attacks.
- The information needed for the verification of a digital signature, such as digital certificate chains and certificate revocation status, may be not available in a future time.
- The Trusted Third Party, which binds the signature-verification-data to a specific identity, may be not trusted in a future time, either because it ceased operation or because it does not fulfill the necessary requirements any more.

As of today, several various electronic signature schemes have been proposed. The main procedure is common and it is based on the public key cryptography, where the signer encrypts (signs) a sequence of data using her private key and the verifier of the signature ensures the originality of the data by decrypting the signature using the public key of the signer and obtaining the original data [2]. From the very first steps of public key cryptography till nowadays, several new (since the 1970s) methods have contributed new features to the basic signature capability. The hash algorithms gave a solution to the computational efficiency of the signatures, the digital certificates [3] and the self-certified keys [4] provided the means for effective identification of the signer, the Public Key Infrastructure (PKI) architectures build the necessary trust relationships and, finally, the time-stamping [5] and notarization [6] schemes made a digital signature even stronger.

The paradigms of time-stamping and notarization have been used to extend the lifespan of a digital signature, either by indicating that a signature was created at a time before a subsequent compromise, or by transferring the trust against the signed data to a new entity, the Notary. Yet, timestamps and notarizations consist of digital signatures and therefore will become invalidated in some short period of time.

Technical frameworks for long-term electronic signatures are presented in [7] [8]. These frameworks are mainly based on document time-stamping that proves the existence of a sequence of data before a specific moment in time. An electronic signature is time-stamped, indicating that the signature was created at a moment before a subsequent key or algorithm compromise. This timestamp is refreshed before the used algorithms or keys become (or are likely to be) compromised or rendered vulnerable, ensuring its validity through the years. However, the validation process in such schemes still requires the successful validation of the initial signature and timestamp and therefore is based on information, such as digital certificates and Certificate Status Information (CSI) [9]. Although such authenticating information may be included in the timestamp data, the original signing entities may be unknown or not trusted in the future. In other words, the frameworks give a solution to the problem of algorithm decay, but not to the preservation of trust. Another drawback of the schemes is that the long-term validity of the digital signature depends entirely on a timestamp. However, even if this timestamp is absolutely trusted, a common user may still be unable to identify whether the time indicated in the timestamp was before or after an eventual algorithm compromise.

The objective of this work is to present a digital signature scheme where the signature verification process is based on trust relationships, data and technologies that are available at the moment of verification. The basic idea towards this objective is the elimination of any dependency on obsolete trust relationships, data, and technologies that may have existed in the past, but are subsequently invalidated. The idea focuses on the preservation of trust in the informa-

tion needed to verify the identity of the signer of a document in a ceaseless way. This is achieved by a continuous *successive trust transition* to new entities, data, and technologies.

The solution to this problem is based on a *cumulative notarization scheme*. An overview of the proposed framework is presented in section 2 including the basic terms used, the assumptions made and the requirements identified. The technical part of the framework is presented in section 3. The trust and security considerations raised, are discussed in sections 4 and 5 respectively. In section 6 the framework is compared to other signature schemes and the conclusions are drawn in section 7.

## 2. Framework overview and background

### 2.1 Basic actors and relevant terms

The actors involved in the proposed model of *cumulative notarization* are the following:

- The *initial signer* who creates the *initial digital signature* on the *initial document*.
- The *Certification Services Provider* (CSP) that provides the necessary trust mechanisms (e.g. digital certificates) to authorize the signer to act as such and to bind her keys to her identity.
- The *Notaries* (or *Confirmers*) acting as Trusted Third Parties, which make specific attestation on the content and the characteristics of a data collection (e.g. on the validity of a digital signature at a specific moment) and then digitally sign the data and the attestation.
- A *Relying Party* (RP) (or *Verifier*), who relies on the information provided by a CSP or a Notary, that enables her to verify the validity, the origin and other characteristics of a digital signature. A short-term RP and a long-term RP may be distinguished.

The basic terms used for the description of the data collections used in the proposed framework are the following:

- *Signed Data*: The initial document signed by the initial signer
- *Initial signature*: The value and metadata of the signature of initial signer on the initial document.
- *Signature-creation-data*: Unique data that is under the absolute control of a signer or a notary and enable her to create a digital signature on a digital document.
- *Signature-verification-data*: Data that is used to verify a digital signature and that is public or at least known to relying parties.
- *Digital certificate*: A testimony, issued by a CSP, which binds the signature-verification-data to a specific identity.
- *Notarized data*: A collection of signed data together with their signature, which are subsequently attested and signed by a Notary.
- *Notary attestation*: A collection of declarations made by a Notary on the characteristics of a data collection.
- *Encapsulated Data*: Data that are subsequently enclosed within another data structure.
- *Notarization Token*: A complete collection of Notarized data, Notary attestation and Notary digital signature, together with any required metadata.

### 2.2 Assumptions

The basic assumption made is that at the moments of the creation and of the verification of a notarization, the specific notary must be a trusted entity within the socio-economic system,

either on national level or within a closed group. The notary may be a stand-alone Trusted Third Party (TTP), or an entity properly authorized to provide notary services by another TTP by means of digital certificates with the proper ‘usage extension’ [10]. Legally, the notary should abide by law and regulations concerning a regular notary service and be staffed by authorized personnel.

At the moment of a notarization the notary attests that:

- The signature of the document is algorithmically valid for a given public key.
- The signer possesses a valid digital certificate that binds the signer’s keys to her identification details or at least she possesses a self-certified key pair [4].
- The issuer of the signer’s certificate is a trusted, officially qualified entity, acting legally within a specific scope, adhering to published policies and abiding by the current national or international law and regulations. It is therefore assumed that the notary trusts the CSP of the signer at the moment of the notarization.
- All the relevant information, such as the status of certificates, the validity of trust chains and the policies and regulations in force are properly checked.
- The validity of the optionally included timestamp is properly verified.

### 2.3 Requirements:

The successful verification of the last notarization must indicate that the initial signature is valid. This verification must not have any dependency on past technologies and on data that are not necessarily archived or attached to the initial document, such as keys, certificates and CSI. Therefore, it is required that the verification of the *Cumulative Notarization Token* (CNT) as described in the next sections, must not have any dependency on:

- The existence of the CSP, which issued the certificate for the initial signature.
- The existence and the operational status of intermediate Notaries.
- The algorithm and key characteristics of the initial signature or the intermediate notarizations
- The existence of keys, certificate chain and CSI for the validation of initial signature and intermediate notarizations.
- The subsequent expiration or compromise of the keys of the initial signature or the intermediate notarizations.

The future verification of a CNT must be based on technologies considered as valid and strong at that time and on the minimum possible data that are easily archived and recognized. The data that is required for the CNT verification process are at least:

- The initial document or its hash value
- Identity data of the initial signer
- Metadata that describe the initial document and the signature process of the last notary
- The CNT, with the information necessary for the validation of the last notarization such as certificate trust chains, policies and CSI.

The framework does not require:

- A secure archiving system. Notarization tokens are tamperproof and may be stored in personal storage media or in publicly accessible directories.
- The inclusion of the initial document in the CNT. Such inclusion is optional, but at least a hash value of the initial document must be included. In this case the relying party must hold the complete initial document in order to complete a verification process.

## 2.4 Long-term preservation of data

Besides the core subject of this paper to preserve the verifiability and the validity of digital signatures for long periods, the readability of both the document and the signature data must also be addressed [11]. Technology obsolescence, different data representation formats, limited storage media longevity, special software and hardware required, are some factors that may reduce the future readability of digital information. Various solutions have been proposed to address this issue; they are based on the existence of metadata [12][13]. Metadata is a collection of key attributes and information that describe an object, enabling future access and processing on this object. The underlying principle for metadata is to link and integrate heterogeneous, multi-platform digital information collections that are contributed by different sources, into a unified form, so that information is accessible by anyone, independently of location, time and technology.

Towards this objective, the Extensible Markup Language (XML), which has an open text-based format and contains metadata by design, is selected for structuring the data used in the presented framework. Additionally, a metadata field for the description of the initial document is always included in the data structures, as well as the metadata needed for the recognition and the process of signatures and notarizations (e.g. algorithms used and digital certificates).

## 3. Technical framework

### 3.1 Data Structures

The power of public key cryptography [2], combined with the wide acceptance of the Extensible Markup Language (XML) [14], provide the means for the construction of an open cumulative notarization scheme that ensures effectiveness, readability, and applicability for a long period of time. An indicative XML schema is presented in the sequel and the usage of the data structures are explained in the next section. The schema uses a simplified form for the digital signatures, which can be further developed to integrate the XML-signature standard. It is worth noticing that the realization of a cumulative data structure is achieved by means of *recursive XML elements*. Specifically, the element ‘cumulativeNT’ contains the element ‘previousCNT’ which in turn is a choice of either another ‘cumulativeNT’ or a ‘firstNT’ element.

```
<?xml version="1.0" encoding="UTF-8"?>
<schema targetNamespace="http://mynotary.org"
xmlns="http://www.w3.org/2001/XMLSchema" xmlns:n="http://mynotary.org">

  <!-- signed initial document -->
  <complexType name="signedData">
    <sequence>
      <element name="initialDocument" type="base64Binary"/>
      <element ref="n:documentMeta-data"/>
      <element name="signatureValue" type="base64Binary"/>
      <element ref="n:signatureMeta-data"/>
      <element name="signerDN" type="string"/>
      <element name="timeStamp" type="base64Binary" minOccurs="0"/>
      <element name="notaryAttestation" type="string" minOccurs="0"
maxOccurs="unbounded"/>
    </sequence>
  </complexType>

  <!-- metadata for the recognition of the signed document -->
  <complexType name="documentMeta-data">
    <sequence>
      <element name="mimeType" type="string"/>
      <element name="version" type="string"/>
      <element name="otherAttributes" type="string"/>
    </sequence>
  </complexType>
</schema>
```

```

</complexType>

<!-- metadata for the recognition of a digital signature -->
<complexType name="signatureMetaData">
  <sequence>
    <element name="digestAlgorithm" type="string"/>
    <element name="signatureAlgorithm" type="string"/>
    <element name="certificateChain" type="base64Binary"/>
    <element name="CRLs" type="base64Binary"/>
  </sequence>
</complexType>

<!-- the first notarization on the initial signed document -->
<complexType name="firstNT">
  <sequence>
    <element ref="n:signedData"/>
    <element name="notarySignatureValue" type="base64Binary"/>
    <element ref="n:signatureMetaData"/>
    <element name="notaryDN" type="string"/>
  </sequence>
</complexType>

<!-- previously notarized data that will be notarized again -->
<complexType name="previousCNT">
  <sequence>
    <choice>
      <element ref="n:cumulativeNT"/>
      <element ref="n:firstNT"/>
    </choice>
    <element name="timeStamp" type="base64Binary" minOccurs="0"/>
    <element name="notaryAttestation" type="string" minOccurs="0"
      maxOccurs="unbounded"/>
  </sequence>
</complexType>

<!-- the resulting cumulative notarization token -->
<complexType name="cumulativeNT">
  <sequence>
    <element ref="n:previousCNT"/>
    <element name="notarySignatureValue" type="base64Binary"/>
    <element ref="n:signatureMetaData"/>
    <element name="notaryDN" type="string"/>
  </sequence>
</complexType>

<!-- data element definitions -->
<element name="cumulativeNT" type="n:cumulativeNT"/>
<element name="previousCNT" type="n:previousCNT"/>
<element name="signatureMetaData" type="n:signatureMetaData"/>
<element name="documentMeta-data" type="n:documentMeta-data"/>
<element name="signedData" type="n:signedData"/>
<element name="firstNT" type="n:firstNT"/>

</schema>

```

### 3.2 Generic Mechanism

The mechanism of cumulative notarization is summarized in the following steps:

1. The *initial document* is signed by the *initial signer*. The document, the metadata describing its content and format, the signature, the signature metadata, and the distinguished name of the signer form the ‘signedData’ element.
2. The metadata included in ‘signedData’ consists of two elements that refer to the document and the signature, respectively. Since the objective is the long-term preservation of data, the accurate description of the subject data is crucial, as already described in section

2.4. The signature metadata may optionally include a complete certificate chain up to a trusted authority and the relevant Certificate Revocation Lists (or alternatively the OCSP responses) at the moment of signing.

3. The inclusion of a 'timeStamp' element in 'signedData' though optional is desirable, since it may strengthen further the reliability of the tokens by indicating that the referred data existed before an event that could reduce its validity. A timestamp should normally refer to the initial document together with the initial signature. Once a timestamp is included, the notary will be obliged to include in its attestations the verification of the validity of the timestamp, as well as a declaration of trust against the *TimeStamp Authority* (TSA) [5], [15].
4. A Notary attestation is also included in the 'signedData' element. This field contains all the details of the verification process that the notary performs before the signing (notarization) of the data structure. The 'notaryAttestation' field may appear multiple times and its indicative values are:
  - a. Signer's CSP trusted
  - b. Algorithm valid and strong
  - c. Key length acceptable
  - d. CSI checked
  - e. Certification chain valid
  - f. Signature algorithmically valid
  - g. CSP policy is compatible and/or acceptable
  - h. TSA trusted and timestamp validity checked
5. The first Notary signs the 'signedData' forming the first Notarization Token ('firstNT' element) structure, which includes the notarized data ('signedData') the notary signature, its metadata and the distinguished name of the first Notary.
6. The second Notary forms the 'previousCNT' element, which contains the 'firstNT' and its attestation on the verification process it performed on the previous notarization token. It may also optionally include a new timestamp.
7. The second Notary signs then the 'previousCNT' element, forming the Cumulative Notarization Token (CNT - 'cumulativeNT' element) structure, which includes the notarized data ('previousCNT') the notary signature, its metadata, and the distinguished name of the second Notary.
8. Consecutive Notaries similarly encapsulate the last CNT into the 'previousCNT' element (instead of the 'firstNT' element) and perform the steps 6 and 7, producing each time a new CNT.

*Notice:* The entities referred to as 'second Notary' or 'consecutive Notaries' are not necessarily distinct and different from the previous ones. They are named as such in order to emphasize the fact that each and every notarization process is a stand-alone and independent procedure that may be performed by repeated or distinct notaries.

The described mechanism of cumulative notarization and the relevant transition of trust are illustrated in Figure 1.

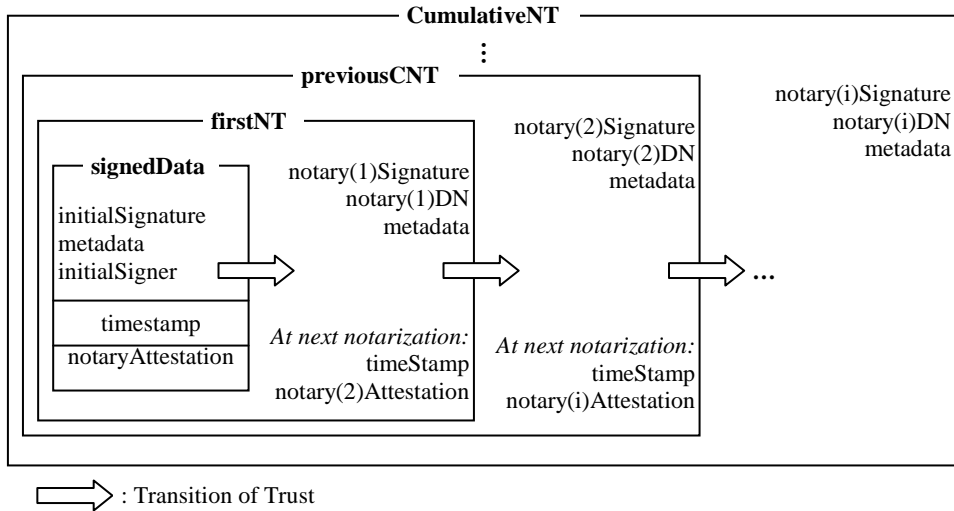


Figure 1: Successive data encapsulation in Cumulative Notarization

### 3.3 Sequence of actions

The proper sequence of actions during the creation and the lifespan of a CNT is important, particularly the moment of notarization of a previously signed data object. A problem that must be addressed is the revocation interim, being the time elapsed between actual key compromise, key revocation and next CRL publication. Even in online CSI [9] systems such as the Online Certification Status Protocol (OCSP), the time interval between actual compromise and system update is enough for the creation of a forged digital signature that will be considered as valid (see Figure 2).

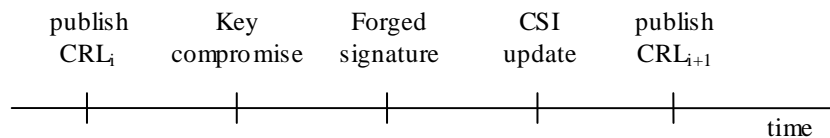


Figure 2. The problem of revocation interim

To solve this problem, latency between signature and notarization is required, if the notary is based either on online (e.g. OCSP) or periodically updated (e.g. CRL) CSI systems. The Notary receives signed document, but verifies its validity only after next CRL is published by the signer's CSP. The time of signature will be after the publication of CRL<sub>i</sub> and the time of notarization will be after the publication of CRL<sub>i+1</sub> as illustrated in Figure 3. The same sequence applies also to subsequent notarizations, whereas the signer is replaced by the Notary<sub>i</sub> and the first Notary is replaced by Notary<sub>i+1</sub>.

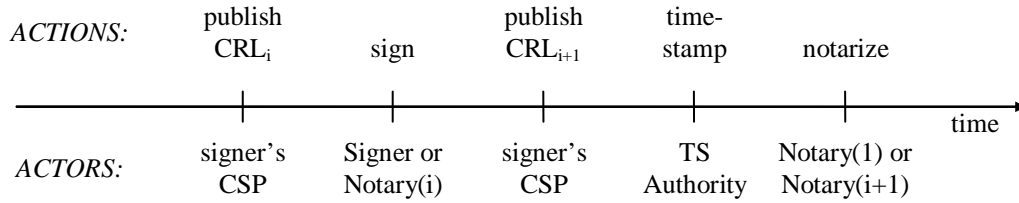


Figure 3. Latency between signature and notarization

Time-stamping precedes the notarization in order for the Notary to subsequently verify and attest the validity of the timestamp before the notarization. This sequence will eliminate the need, on the relying party's side, to trust an additional entity, namely the timestamp authority.

The duration of the intervals between each action depend mainly on the policy for the period of CRL publication. In any case it must be reasonable and that means it must not exceed a single-digit number of days.

### 3.4 Verification of a CNT

Provided that the assumptions described in section 2.2 are in effect, and the requirements described in section 2.3 are satisfied, the verification of the initial signature contained in a CNT mainly depends on the verification of the last notarization and will be restricted to the following steps:

1. Retrieval of a CNT relevant to a subject document, based either on the document hash value or on its contents.
2. If the CNT contains the complete document, then this step is omitted. If the CNT contains only the hash value of the document, then the relying party must possess the initial document and has, also, to both a) regenerate its hash, and b) confirm that it is equal with the one stored in the CNT.
3. The relying party checks the validity of the last notarization. The verification of the notarization consists of the following basic checks:
  - a. The signature of the last notary is algorithmically correct.
  - b. The notary is trusted to perform the operation, as described in section 2.2 'Assumptions' and it is using current valid technology.
  - c. The certificate of the last notary is validated, by checking its chaining up to a Certification Authority and the CSI for this certificate.
  - d. An additional forgery protection described in section 5.1 exists and is valid.
  - e. The attestation declaration and the optional timestamp included in the token are satisfactory for the relying party.

Performing successfully the above steps will lead the relying party to conclude that the initial signature is valid and the initial signer is the one indicated in the 'signerDN' field. A further analysis of the intermediate notarization tokens up to the initial signature can be made for historical reasons only rather than for verification purposes, since it may rely on technology or data that is not trusted any more.

## 4. Trust Considerations

### 4.1 Transition of Trust

The presented solution, apart from its technical aspects, is mainly based on the notion of trust within the Public Key Infrastructure. Trusting an entity such as a Certification Services Provider or a Notary is essential for the development of the proposed cumulative notarization. It is therefore important to present some basic considerations on how trust is established and managed and under what circumstances it may be transferred towards new entities.

Trust in the Information Society may be built on various different grounds. In order to define the notion of trust against a notary and analyse the mechanism of trust transition, a summary of the possible trust types follows:

*Calculus-based Trust:* The parties involved assess the degree of their dependency on the other entity, the expected profit and the possible risks in order to decide whether they will finally develop a trust relation with each other. This kind of trust is a result of careful calculations usually based on financial criteria [16].

*Information-based Trust:* The trusting party collects as much information as possible on the other party, in order to enhance the ability to predict the behaviour of the trustee. The sense of uncertainty decreases, the possible risks are minimized and therefore, a trust relation can be developed.

*Transitivity-based trust:* In this category we classify various trust architectures developed in the PKI. The consolidation of a strong trust relation with a third entity enhances the transitivity property of that relation. Therefore, a relying party who trusts a third entity is also inclined to trust the entities pointed out or suggested by that third entity. However the transitivity of trust depends on additional conditions and restrictions that a relying party has to take into account. [17]

*Trust within the social system:* Trust derives from the participation of a human being in a socio-economic system. In this case, the meeting, the familiarisation or the knowledge of details between the parties involved in a transaction, is not required. A typical example of this case is the monetary system and the bank transactions.

A notary in the proposed framework acts a TTP for the services it offers. The trust to a notary service is grounded on a combination of the abovementioned trust types. Specifically, the notion of trust against a third party, assumes that the TTP is part of the social system, expresses the faith of the relying party in specific operational, ethical and quality characteristics, while it also includes the acknowledgement of a minimum risk factor [18].

Trust has the properties of *transitivity* and *selectivity*. A trust relationship is transitive [19] when the fact that an entity A trusts an entity B and the entity B trusts an entity C, leads us to the conclusion that entity A is bound to trust entity C. The property of trust transitivity is essential for the proper functionality of the proposed framework; however there is a need of establishing specific restrictions and requirements in trust transition (see section 4.2). In fact, it is hereby assumed that when a relying party trusts a Notary, it is bound to trust any other party that the Notary declares as a TTP and consequently it trusts the information that this TTP publishes. However, the acceptance of this trust transitivity remains to the discretion of the relying party and it may also be based on other factors such as calculus or information, as described previously.

Selectivity applies when trust in one entity is different and independent for every property that entity possesses and for every activity it executes. It is desirable that trust also maintains its selectivity property. Trust of a Notary should be restricted to specific autonomous activities, such as identity attestation, digital signing and time-stamping and it must not extend to other irrelevant activities.

The trust model resulting as a consequence of trust transitivity is illustrated in Figure 4. A short-term relying party trusts directly the CSP that certifies and authorizes the initial signer. As a result, it trusts the signature-verification-data and consequently the created signature data. The first Notary validates and notarizes the signature data, since it trusts the signer's CSP. Subsequent Notaries are continuously refreshing the trust against the signature data by applying the cumulative notarization. A long-term relying party trusts directly only the last notary and as a result of trust transitivity, a trust relationship against the signature data is derived.

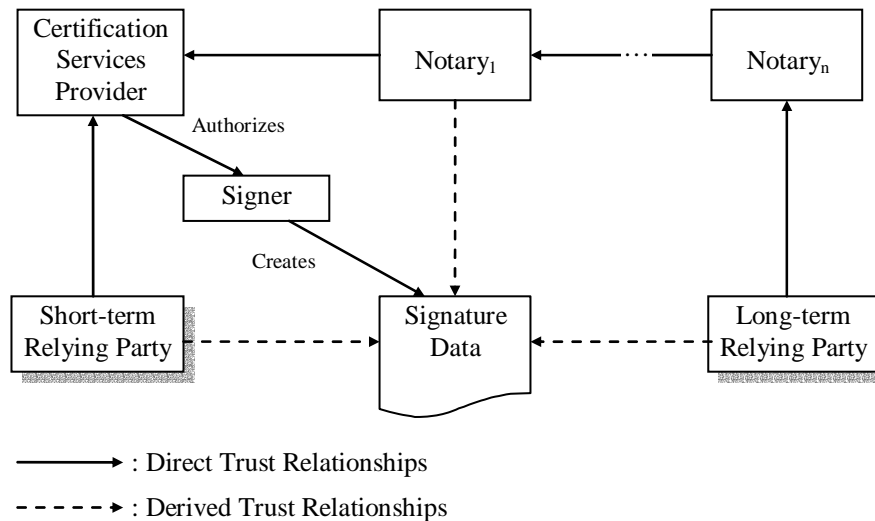


Figure 4: Trust Model

#### 4.2 Limiting Trust transitivity

After a long-term cumulative notarization on a digital signature, it is likely that the resulting trust chain between the initial signer and the relying party will become unexpectedly long. This long 'trust distance' may introduce new risks and weaken the trust relationships, since the relying party has no control on this successive transitivity and in most cases cannot follow up the policies and practices of all the intermediate Notaries and CSPs. The restriction of the length of trust chain is a non-solution, since a limit on the steps of trust transitions would reduce the longevity of the initial signature. The rules that may be applied in order to limit trust chains should have two targets:

- *Restriction of security domains:* Each Notary may identify a set of other TTPs, which are considered as trusted and thus restricting the trust relationships of its users within this set of security domains. It requires that all the TTPs involved in a trust chain belong in this set, in order for the chain to be valid.
- *Restriction based on policy compatibility:* A notarisation is taking place according to specific Notary policies and practices, which describe the rules and the conditions of this procedure. Typical example of these conditions is the procedure of signature validation before the notarisation. Stating these policy terms as critical, the trust transitivity will not extend to other TTPs that have policies with incompatible or conflicting procedures [20].

### 5. Security considerations

#### 5.1 Forgery protection

Although the entity performing the notarization is assumed to be a widely trusted and incorruptible authority (see the assumptions at section 2.2) additional mechanisms are required for

the protection of the integrity of the already issued notarization tokens. An eventual compromise of a notary or an internal corruption could lead to the deletion, modification or re-issuance of previously issued notarization tokens. The insertion of out-of-date or out-of-sequence tokens, namely the breach of '*sequential consistency*' is also considered as a forgery.

Another possibility is the forgery by cryptanalysis, such as by guessing or inferring a cryptographic key of a notary or by finding a flaw in the cryptographic algorithm used. The prevention of issuing new forged tokens may be faced by various key revocation and CSI dissemination mechanisms [9]. An interesting approach would also be the development of a '*notarization token revocation*' mechanism, similar to the certificate revocation mechanisms. A mechanism that assures the integrity and the sequential consistency of the previously issued tokens is also limiting the possibility of forgery by cryptanalysis.

A solution to the above problems and especially to the '*sequential consistency*' requirement may be based on two proposed mechanisms:

According to the first mechanism, all the tokens issued by a specific notary are forming a *one-way linked list* [21]. This is achieved by including in each NT a unique reference to the previous NT. The unique reference is the hash value of the previous NT and therefore any deletion, modification or insertion in the list will be immediately detectable. In the proposed schema the solution is implemented by adding the following element at any structure it is notarized, being the 'signedData' and the 'previousCNT' structures:

```
<element name="lastNTHash" type="base64Binary"/>
```

The value of 'lastNTHash' refers to the last notarization token issued by the same notary for any kind of data and must not be confused with the 'previousCNT' element that includes the last issued cumulative NT for a specific document, probably issued by a different notary.

The second mechanism uses the *binary hash trees* or other similar variations [22], [15] where the leafs of the tree contain the hash values of all the issued NTs of a notary. The parents of leafs are periodically calculated and contain the hash of the concatenated leaf values (or alternatively the result of a XOR operation) issued for a specific period of time (e.g. daily). The root node is also periodically calculated as the hash of all the second level nodes. Every new root value is published by a high-integrity means such as the daily press, making the existence of all the notarization tokens a historical fact. As a result, any modification in the sequence of the issued tokens or their content will - in a very high probability - render the root value of the tree different from the one already published and therefore the modification will be detectable. Since a hash is a many-to-one function there is always the possibility of collisions, despite the fact that the calculation of two different values that return the same hash value is considered as computationally infeasible. In order to further reduce the risk of a hash collision, in the highly secure environment of a notary, the binary hash trees may be combined with other mechanisms that are not based on hash functions, such as the one-way linked lists described previously.

## 5.2 Threat scenarios

Some threats, which may arise within the proposed framework and the way they are dealt with, follow:

*Alteration of a cumulative notarization token:* The integrity of the whole structure is protected by the signature of the last notary.

*Forged initial signature:* The sequential consistency described above and the attestations of the first notary ensure that the initial signature was created at a time where the signature algorithms and the signature-creation-data were strong and secure.

*Compromise of intermediate notary keys or algorithms:* Does not affect the validity and the overall security of the CNT, since it depends only on the validity of the last notarization.

*Compromise of last notary keys or algorithms:* One solution is the rollback to a previous notarization in the chain, provided that there exists at least one valid and strong past notarization in the CNT. If this does not apply, the CNT must be notarized as soon as possible by another valid and operational notary service, adding one more step to the trust and technology transition as described in section 3.2.

*Lost public key of a notary:* Such a misfortunate event may happen in case the public keys of notaries are not widely disseminated in distributed public directories. In case of an ‘intermediate’ notary, again the overall validity of the CNT is not affected, since the validation of an intermediate signature is not necessary. In case the public key of the last notary is lost, the validation of a CNT will be not possible, unless the relying party can be based on an earlier strong and valid notarization (rollback) that is included in the cumulative notarization token.

*Faulty attestation by a notary:* Since we have assumed that the notary is a properly trusted and authorized authority, a fault attestation on the characteristics of the notarized data would breach laws and regulations. As a purely legal issue, this case falls out of the scope of the paper.

## 6. Signature schemes comparison

Examining the specific characteristics of various signature schemes, we can identify a comparative advantage of the proposed cumulative notarization scheme in terms of security and usability. The comparison of various schemes presented in Table 1 demonstrates that the proposed scheme keeps the strong security characteristics of digital signatures, while it addresses the issues of trust and technology refreshing as a whole, resulting in a long lifespan.

<b>Scheme</b>	Handwritten signature	Notarized handwritten signature	Plain digital signature	Self-certified digital signature	PKI based digital signature	Notarized digital signature	Timestamp refreshing	Trusted Archive Protocol	Proposed Cumulative Notarization
<b>Property</b>									
Corresponds to unique physical entity	Yes	Yes	Likely	Yes	Yes	Yes	Yes	Yes	Yes
Small size	Yes	Yes	Yes	No	Yes	No	No	No	No
Difficult to forge	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Publicly verifiable	Partial	Yes	No	Yes	Yes	Yes	Yes	Yes	Yes
Signer recognizable by signature data	Partial	Yes	No	Yes	Yes	Yes	Yes	Yes	Yes
Requires trust on signer’s CSP	N/A	N/A	No	Yes	Yes	Yes	Yes	Yes	No
Requires trust on a TSA	N/A	N/A	No	No	No	No	Yes	Yes	No
Requires trust on a Notary	N/A	Yes	No	No	No	Yes	No	No	Yes
Requires public directory	No	No	No	No	Yes	Yes	Yes	No	Yes
Requires secure archiving	No	Yes	No	No	No	No	No	Yes	No
Technology refreshing	N/A	N/A	No	No	No	No	Yes	Yes	Yes
Trust refreshing	No	No	No	No	No	No	No	No	Yes
Lifespan	Med	Long	Short	Short	Short	Med	Med	Med	Long

Table 1. Comparison of signature schemes

The criteria used for the comparison of the characteristics of the signature schemes are explained below, while the notation ‘N/A’ stands for ‘Not Applicable’:

- The correspondence of a digital signature to a unique physical entity is a common characteristic, except of the plain digital signature (without identification linkage mechanism) where the keys used are not necessarily linked to a specific identity.
- The small size of the signature is desirable, however it is not addressed in all schemes, without necessarily affecting the efficiency. The forgery difficulty refers to the algorithmic part of the schemes, assuming that the signature-creation-data is kept secret.
- The property of public verifiability holds when the signature-verification-data (e.g. signature dumps or public keys) are publicly available. The signer is recognizable when the signature-verification-data are unambiguously linked to a physical entity (e.g. digital certificates).
- The relying party is required to trust either the Certification Services Provider (CSP) of the initial signer or a Time-Stamping Authority (TSA) or a Notary.
- The storage and dissemination of signature data and signature-verification-data may require either a publicly accessible directory or a secure archiving system. The first option is obviously more practical and efficient.
- Technology refreshing is met at schemes where a signature is periodically refreshed by new emerging technologies, eliminating the risk of algorithm weakening. Trust refreshing is a unique characteristic of the proposed framework and it is further explained in section 4.1.
- The lifespan of a signature depends on the lifespan of signature-creation-data, of trust relationships and of technologies. A PKI-based signature for example depends on all the three factors, while a cumulative notarization depends only on a refreshed trust relationship.

## 7. Conclusions

The need for a mechanism that ensures long-term preservation of digitally signed documents arises from the fact that digital signatures have a short lifespan compared with the content of the documents which may be required to exist for long periods of time.

We propose an open, practical and efficient framework, based on the digital notarization paradigm, which ensures that a relying party at the distant future will efficiently verify the validity of the digital signature of a document, without the need to rely on ceased entities, obsolete trust relationships, data and technologies that may have already been invalidated.

The solution to the problem is the continuous successive trust transition to new entities, data and technologies. This is achieved by a mechanism of cumulative notarization that encapsulates and refreshes previously signed or notarized data. As a result, a relying party will have to verify only the newest notarization and thus it will always depend on current trusted entities, on modern technologies and on available data for the verification of a digital signature.

The proposed framework is resistant to forgery and copes efficiently with various possible threats. It is also demonstrated that it has a comparative advantage against other signatures schemes, since it preserves all the security characteristics of a digital signature and additionally it eliminates the requirement to trust the CSP of the signer, it does not require a secure archiving system and it has the ability of technology refreshing.

Future work may focus on the definition of a metadata collection that will efficiently describe a signed document for long periods of time, as well as on a mechanism for the preservation of the history of document revisions and signatures, instead of a single signature. Another topic is the development of a complete protocol for requesting, archiving and retrieving cumulative notarization tokens.

## References

- [1] Maniatis P., Baker M., “Enabling the Archival Storage of Signed Documents”, in *Proc. of the FAST 2002 Conference on File and Storage Technologies*, pp. 31-45, January 2002, USA.
- [2] Rivest R., Shamir A., Adleman L., “A method for obtaining digital signatures and public-key cryptosystems”, *Com. of the ACM*, pp. 120-126, Vol. 21, No. 2, 1978.
- [3] Kohnfelder L., *Towards a practical public-key cryptosystem*, Ph.D. Thesis, MIT, 1978.
- [4] Girault M., “Self-certified public keys”, in *Advances in Cryptology*, pp. 490-497, LNCS 547, Springer-Verlag, 1991.
- [5] Adams C., Cain P., Pinkas D., Zuccherato R., “Internet X.509 Public Key Infrastructure Time-Stamp Protocol”, *IETF Request For Comments* 3161, August 2001, available at <http://www.ietf.org/rfc/rfc3161.txt>
- [6] Ansper A., Buldas A., Roos M., Willemson J., “Efficient long-term validation of digital signatures”, in *Proc. of 4<sup>th</sup> International Workshop on Practice and Theory in Public Key Cryptography (PKC 2001)*, pp. 402-415, Korea, 2001.
- [7] Pinkas D., Ross J., Pope N., “Long term electronic signatures”, *IETF Request For Comments* No.3126, September 2001, available at <http://www.ietf.org/rfc/rfc3126.txt>
- [8] Wallace C., Chokhani S., “Trusted Archive Protocol (TAP)”, *IETF Internet Draft*, August 2003, available at <http://www.ietf.org/internet-drafts/draft-ietf-pkix-tap-00.txt>
- [9] Iliadis J., Gritzalis S., Spinellis D., De Cock D., Preneel B., Gritzalis D., “Towards a framework for evaluating certificate status information mechanisms”, *Computer Communications*, pp. 1839-1850, Vol. 26, No. 16, 2003.
- [10] ITU - International Telecommunication Union, X-509 | ISO/IEC 9594-8, *The directory: Public-key and attribute certificate frameworks*, ITU, X-Series, 2001.
- [11] Wright T., “Secure digital archiving of high-value data”, *BT Technology Journal*, pp. 60-66, Vol. 19, No. 3, 2001.
- [12] Alemneh D., Hastings S., Hartman C., “A metadata approach to preservation of digital resources: The University of North Texas Libraries’ experience”, *First Monday Journal*, Vol. 7, No. 8, August 2002.
- [13] Lupovici C., Masanes J., *Metadata for long-term preservation*, Biblioteque Nationale de France, NEDLIB Consortium, July 2000.
- [14] Cowan J., Tobin R., *XML Information Set*, W3C Recommendation, October 2001, available at <http://www.w3.org/TR/xml-infoset>
- [15] Massias H., Avila X., Quisquater J., “Timestamps: Main Issues on Their Use and Implementation”, in *Proc. of 8th Workshop on Enabling Technologies (WETICE '99)*, pp. 178-183, IEEE Computer Society, USA, June 1999.
- [16] Sulin B., “Establishing online trust through a community responsibility system”, *Decision Support Systems*, pp. 323-336, Vol. 31, 2001.
- [17] Fernandes A., “Risking Trust in a public key infrastructure: Old techniques of managing risk applied to new technology”, *Decision Support Systems*, pp. 303-322, Vol. 31, 2001.
- [18] Lekkas D., “Establishing and managing trust within the Public Key Infrastructure”, *Computer Communications*, pp. 1815-1825, Vol. 26, No. 16, 2003.
- [19] Buskens V., “The social structure of trust”, *Social Networks*, pp. 265-289, Vol. 20, No. 3, 1998.

- [20] Kokolakis S., Kiountouzis E., “Achieving interoperability in a multiple-security policies environment”, *Computers and Security*, pp. 267-281, Vol.19, No. 3, 2000.
- [21] FNMT Group, *PKITS: Public Key Infrastructure with Time-Stamping Authority*, Final Report, EU project no. 23.192, December 1998.
- [22] Merkle R., “A certified digital signature”, in *Advances in Cryptology (CRYPTO’89)*, pp. 218-238, Springer, USA 1989.