Assessing Trust in the Long-Term Protection of Documents

Martín Vigil, Daniel Cabarcas
and Johannes Buchmann
Technische Universität Darmstadt
Hochschulstraße 10, 64289 Darmstadt, Germany
{lastname}@cdc.informatik.tu-darmstadt.de

Jingwei Huang
University of Illinois at Urbana-Champaign
1308 West Main Street, Urbana 61801, USA
jingwei@iti.illinois.edu

***Abstract*—Digital archives rely on trusted parties, such as certification authorities, to ensure authenticity, integrity and proof of existence protection for documents. In this paper, we analyse the trust assumptions that a verifier has to make in order to trust in the protection of a document. We show that trust fades out in the long term due to the ever-growing number of trusted parties. Despite such a dire prospect, current technologies such as X.509 PKI do not assess trust, thereby leaving verifiers in the dark. We present a certification scheme for documents that provides verifiers with a better assessment of trust than in X.509 PKI. In the proposed scheme, trusted parties are rated based on the correctness of their performance. From the ratings, verifiers can assess quantitatively the trust in the trusted parties for the short term, and in the protection of documents for the long term. The proposed scheme encourages trusted parties to work properly.**

***Index Terms*—Authenticity; Digital signatures; Integrity; Long term; Notary; Proof of existence; Timestamping; Trust assessment; Trusted party; X.509.**

I. Introduction

Digital archives are increasingly playing a critical role in preserving data for long periods, such as in the health care sector [1], land registers [2], and patent offices [3]. Digital archives have to protect archived data to ensure authenticity ("the origin of data can be correctly identified") and integrity ("data is uncorrupted"). Additionally, digital archives may have to guarantee a date and time when data existed. This protection is called *proof of existence* and is necessary for patents, for example. We refer to *authenticity*, *integrity* and *proof of existence* collectively as *protection*.

Digital signatures with public key infrastructure (PKI) have been the prominent cryptography-based solution to provide protection for documents. This solution cannot provide long-term security, i.e. for an unbounded period, because digital certificates expire and the security of cryptographic algorithms weakens as computer power and cryptanalytic techniques improve. The most common solution for this problem consists of a trusted party named timestamp authority (TSA) which signs timestamps to extend the lifetime of digital signatures. A timestamp cannot provide long-term security either, because its authenticity fades out. Therefore, the long-term protection of documents needs further timestamps, which accumulate over time, causing a huge storage and processing overhead to digital archives. An alternative consists of notaries which extend the lifetime of a document's signature by verifying and attesting the signature's existence and validity on a regular basis. Because the notary verifies the document's protection, old attestations of existence and validity can be discarded. This approach has the advantage of a minimal overhead, but more responsibility is placed in notaries than in TSAs.

A fundamental question in the long-term protection of documents is how much we trust in such protection. The verifier of a protected document can verify the evidence provided by either approach, but there is a number of assumptions that the verifier can only trust. In particular he has to trust in certification authorities (CAs), TSAs or notaries. As the number of trusted parties increases, the probability that all of them act *properly* decreases exponentially, assuming these are independent events. Consequently, the verifier's trust in the document's protection decreases. However, many of the current schemes such as the X.509 PKI [4] overlook this degradation of trust.

Our contribution is twofold. First, we analyse trust within the context of protecting documents in the long term. We identify and analyse the trust assumptions that a verifier has to make to accept the protection of documents. Second, we propose a reputation scheme for notaries, which provides verifiers with quantitative data for assessing the trust in the protection of documents.

The rest of this paper is organised as follows. Section II presents the related work. Section III reviews the necessary background in regard to the protection of documents and assessment of trust. Section IV analyses the security assumptions and the trust in the protection of documents. Section V provides an extension to an existing notarial scheme, allowing for the trust assessment of a document's protection. Section VI presents our conclusions and future work.

II. Related Work

Vigil et al. [5] provide an overview of trust assumptions for long-term protection schemes. The overview is intended to analyse the assumptions individually, without comparisons.

Zimmermann [6] proposes the so-called PGP Web of Trust. It is a decentralised PKI that allows to assess trust in the

Martín Vigil is supported by CAPES (Brazil) under grant BEX4334/10-8.

authenticity of a participant's public key. The assessment is restricted to only a few qualitative trust levels (viz. ultimate, completely, marginal, none and unknown). Jøsang [7] addresses this restriction. He quantitatively expresses trust as a triple comprising belief, disbelief and uncertainty, and proposes an algebra to operate over trust. Alternatively, Reiter and Stubblebine [8] propose the use of a monetary metric to assess the trust in the authenticity of a public key. These proposals are well suited for assessing trust in authenticity within a short-term context. However, they are not suited for a long-term context, where authenticity relies also on TSAs. In addition, trust in public keys evolves in the long term as key holders interact, which is a fact that PGP Web of Trust is not intended to deal with.

III. Background

In this section, we review the background necessary for the rest of the paper. In Section III-A, we present two approaches to protect documents. We then provide in Section III-B a trust model to assess the trust placed in the protection of documents over time.

A. Creating and Verifying Cryptographic Evidence

There are four main players in the lifetime of a document's signature: a signer, who signs digital documents; an archivist, who is responsible for the protection of documents; a verifier, who verifies the protection of documents; and TSAs or notaries, which provide the archivist with up to date cryptographic evidence. In the following, we detail the role of each player.

The signer selects and signs a document using his private key. For the corresponding public key, there is an X.509 certificate issued by a CA. We assume that there is only one certificate chain for the public key. By certificate chain we mean a sequence of certificates from the signer to a trusted CA. The signer submits the selected document, the signature, his certificate chain and corresponding revocation data to the archivist.

The archivist receives the document and corresponding cryptographic evidence from the signer. Figure 1 depicts the received data at time t_0. The archivist updates cryptographic evidence before cryptographic algorithms fade out or the evidence creator's certificate expires or is revoked. Figure 1 illustrates evidence updated at time $t > t_0$. The update procedure relies on either timestamping or notarisation, which both *sequentially* attest the document's protection by creating new evidence on request. We detail each approach in the following.

In the timestamping approach [9], trusted TSAs sign timestamps on the archivist's request. The archivist collects the cryptographic evidence that a signature is valid. He requests a timestamp to date and bind the document and the existing evidence to the collected evidence. The top half of Fig. 1 depicts this process. It shows evidence collected for the document's signature Sig_s, the first timestamp Ts_1's signature, and the second timestamp Ts_2's signature at times t_1, t_2, and t_3, respectively. Notice that evidence accumulates over time.

Lekkas and Gritzalis [10] propose notarisation as an alternative to timestamping as follows. The archivist requests a notary to attest the validity and proof of existence of a document signer's certificate. The notary evaluates the certificate validity. If valid, the notary dates and signs an *assertion* which attests that the certificate existed and was valid to verify the document's signature on the assertion's date. The archivist replaces the document's signature, certificate chain, and corresponding revocation data by the assertion and the notary's certificate chain. On the archivist's request, one notary reattests the other notary's assertion. This allows the archivist to replace old assertions. The bottom half of Fig. 1 depicts this process. It shows at time t_1 the replacement of the document's signature Sig_s, signer's certificate chain CC_s and revocation data RR_s by an assertion A_1 and corresponding notary's certificate chain CC_1. At times t_2 and t_3, it shows the replacement of assertions A_1 and A_2 and corresponding notaries' certificate chains CC_1 and CC_2. Notice that evidence is discarded over time.

The verifier obtains a document and corresponding cryptographic evidence. He then verifies the signatures on the document and on either timestamps or notarial assertion. Signatures are verified with the corresponding public keys. The verifier checks the certificate chain(s) and revocation data using the so-called *certification validation path algorithm* [4]. The failure of any verification calls into question the document's protection.

The difference between timestamping and notarisation is that the former accumulates evidence while the latter discards it. The advantage of timestamping is that a verifier can evaluate the document's protection at any time reference between the first timestamp and the current time. The disadvantage is that the overhead to verify and store evidence grows linearly as new timestamps, certificate chains and revocation data are necessary. The advantage of notarisation is that the overhead depends on the cryptographic algorithms rather than the number of evidence updates. The disadvantage is that a verifier lacks evidence to evaluate the protection of a document at past time references.

B. Trust Model

Among many definitions of trust, Huang and Nicol [12] define it as a mental state comprising: (a) *expectancy*: the trustor expects a specific behavior of the trustee (such as providing valid information or effectively performing cooperative actions); (b) *belief*: the trustor believes that the expected behavior occurs, based on the evidence of the trustee's competence and goodwill; (c) *willingness to take risk*: the trustor is willing to take the risk for that belief.

Trust and belief are tightly related, but different. Trust is placed in an autonomous entity; belief is placed in information. It is common to see from literature the expression of *trust in information*. Such an expression should be understood as *firm belief in that information*. Trustworthiness of a system is about "assurance that a system will perform as expected" [13].

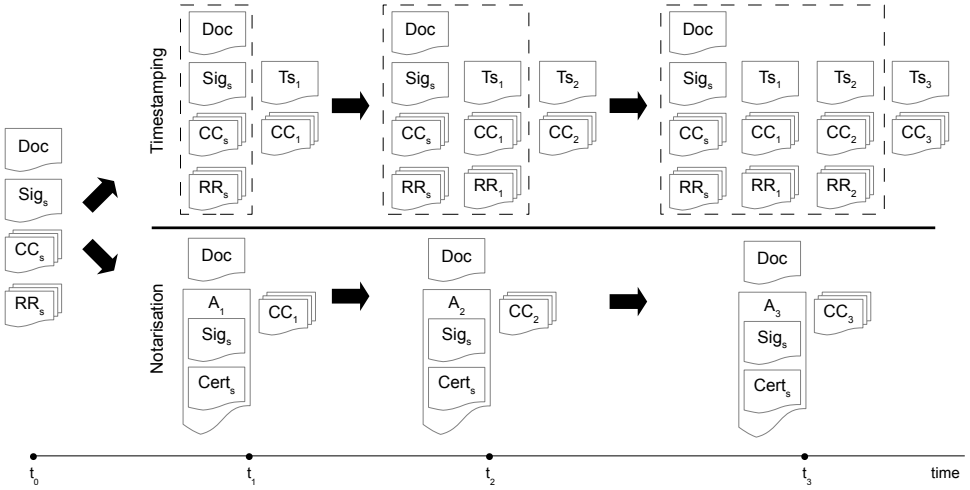

Fig. 1. Long-term protection of a document Doc by timestamping (on top) and notarisation (at the bottom), where Sig_s is a signature on Doc, $Cert$ is a certificate, CC is a certificate chain, RR is revocation data, *Ts* is a timestamp, and A is an assertion. These data are identified by index s for the signer and numbers for timestamp authorities or notaries. The dashed rectangle depicts the data that is timestamped.

In general, we consider the trustworthiness of a system or a thing as firm belief in that the system or thing has expected attributes. Trust is an adequate approximator of trustworthiness [14].

Trust and reputation are also related. Roughly, trust is between two entities; but reputation is the aggregated opinion of a community on an entity; an entity that has high reputation is frequently trusted by many entities in that community; an entity outside that community may also use the reputation or data to evaluate the trust in a trustee.

Because there is frequently insufficient data available to evaluate trust, we leverage the reputation data on some "public roles" in a community, such as a CA, a TSA, or a notary, to measure our trust placed in them. Assume that the reputation data of a trustee $\mathcal{T}$ is represented by a triple $R(\mathcal{T}) = \langle p, n, e \rangle$, where $p, n, e \in \mathbb{N}$ are the totals of positive, negative and undecidable samples of direct interaction experience with $\mathcal{T}$ by a community respectively. In Section V, we refer to p, n and e as positive, negative and neutral opinions.

Following the *Opinion Model* [15], trust is measured by a triple $TM(\mathcal{T}) = \langle t, d, u \rangle$, where $t, d, u \in [0, 1]$ represent the degree of trust, distrust, and uncertainty, respectively. Their values can be calculated from $R(\mathcal{T})$ by using the following equations:

$$t = \frac{p}{p+n+e}, d = \frac{n}{p+n+e}, u = \frac{e}{p+n+e}. \quad (1)$$

Note that $t + d + u = 1$.

The *Subjective Logic* [15] defines the *conjunction* operation on two trust measures $TM(\mathcal{T}_1)$ and $TM(\mathcal{T}_2)$. The operation reflects the conjunctive trust placed in $\mathcal{T}_1$ and $\mathcal{T}_2$ by

$$\begin{aligned} TM(\mathcal{T}_1) \odot TM(\mathcal{T}_2) =& \langle t_1, d_1, u_1 \rangle \odot \langle t_2, d_2, u_2 \rangle \\ =& \langle t_1 t_2, d_1 + d_2 - d_1 d_2, \\ & 1 - t_1 t_2 - d_1 - d_2 + d_1 d_2 \rangle. \end{aligned} \quad (2)$$

Note that the conjunction operation is commutative and associative, and it corresponds to the logical binary "AND". We claim that (2) is adequate to approximate the trust placed in the protection of a document. As seen in Section III-A, the protection relies on the sequential creation of cryptographic evidence. Therefore, a trustor *believes* in the protection if he *believes* in each trustee (viz. CA, TSA or notary) in the process. The intersection of each trustee's trust degree represents such *belief*. The trustor *disbelieves* in the protection if he *disbelieves* in any trustee. The disjunction of each trustee's trust degree represents such *disbelief*. The complement of the sum of trust and distrust degrees represents the trustor's *ignorance* of the protection.

IV. Trust in Timestamping and Notarisation

In this section we analyse the trust in the protection provided by timestamping and notarisation. The analysis is based on the assumptions a verifier has to make to trust in a protection approach. We identify and associate the trust assumptions to the parties involved in both approaches. For the long term, we show that the number of involved parties is the key parameter to assess trust and that the trustworthiness of both approaches tends to be equal.

The trust a verifier places in the protection of documents is derived from his assumptions that the involved parties and the security components work as he expects. Table I presents a list of assumptions that we recognize as necessary for the protection approaches. The list is not intended to be exhaustive. We will next explain the listed assumptions.

Cryptographic algorithms must remain secure to ensure integrity and authenticity. The verifier can check the security status of these algorithms at any arbitrary time reference if the history of cryptographic evidence is available and timestamped. This is not the case for notarisation, where archivists discard old evidence. Therefore, notaries must properly verify

TABLE I
ASSUMPTIONS FOR TIMESTAMPING AND NOTARISATION.

Assumptions	Timestamping		Notarisation
	CA	TSA	Notary
The security of cryptographic algorithms is properly verified			✓
The time source is trustworthy		✓	✓
Keys are not compromised	✓	✓	✓
Credentials are properly verified	✓		
Certificate chains are properly verified			✓

the security of algorithms before creating and evaluating cryptographic evidence.

A verifier cannot verify whether the time value used in timestamps and notarial assertions is accurate. Therefore, TSAs and notaries must use trustworthy time sources.

Signature keys must be trustworthy to ensure authenticity. That is, a key must belong to and be used only by the entity that claims to be the key's subject. In this sense, there are the following assumptions. A CA must ensure that a signature key belongs to a subject by properly verifying the subject's credentials. A notary must properly verify the certificate's validity to guarantee that the signature key belongs to the intended subject. Signature keys must not be compromised. In contrast to the other assumptions, a signature key compromise may be reported in the corresponding certificate's revocation status. However, there is no guarantee that the certificate revocation is timely.

In order for a verifier to establish how much he trusts in a protected document, it would be desirable to compare these assumptions with respect to the probability that they hold. Unfortunately, we lack meaningful data from existing PKIs. For example, revocation data published by CAs conveys reasons for the revocation of signature keys like "CA Compromise", "Key Compromise", "Privilege Withdrawn" or "Unspecified". However, the existing reasons are insufficient to precisely identify the violated assumptions. Moreover, most organizations provide no revocation reasons as seen in [16]. From 1.96 million revocations collected in the Web, 53,55% of the revocation reasons were "Unspecified" or not provided, in contrast to only 3% for "Key Compromise" and less than 1% for "CA Compromise".

Therefore, a verifier can at best assess the trust in each involved party as a black box. Consecutively, the verifier assesses the trust in the protection of documents as a function of the number of involved parties and the *conjunction* of the probabilities that the parties act properly. Since the number of parties that need to be trusted in both approaches increases linearly over time, trust decreases *exponentially* in the long term.

The older the document's signature is, the more timestamps or notarial assertions are applied. Each timestamp and notarial assertion require a TSA and a notary respectively. Additionally, at least a superordinate CA is necessary because of the TSAs' and notaries' certificate chains. Thus, the number of involved parties increases linearly and the trust in both approaches tends to zero in the long term.

V. PROVIDING NOTARISATION WITH TRUST ASSESSMENT

In this section we describe our scheme for assessing trust in the long-term protection of documents. We extend the notarisation scheme proposed by Vigil et al. [11] by adding reputation for notaries. It thus provides verifiers with quantitative data for assessing trust in the protection of documents. We model trust as described in Section III-B. In Section V-A we give a general overview of the scheme and then provide details in sections V-C to V-F.

We chose to construct our scheme over notarisation rather than over timestamping because it is quite natural and necessary. A notary verifies only cryptographic evidence, a process that can be routinely audited, while CAs verify individuals' credentials in an ad hoc process. Moreover, the notarisation approach discards cryptographic evidence, hence being less accountable than timestamping. Our scheme aims at balancing the lack of accountability with trust assessment while keeping the notarisation's low overhead.

A. Overview

A master authority of a community of signers, verifiers and archivists defines a set of notaries. The role of the master authority is limited to this initial setup. The community does not necessarily trust the notaries, but rather relies on the notaries' reputation to decide whether to believe in them. Notaries are interested in having a high reputation, for instance, to sell their notarial services. The service a notary offers is to issue an assertion certifying that on a specific date: a) a signed document and corresponding signer's X.509 certificate existed; and b) this certificate was valid. Other notaries rate the issuing notary on how hell he provides his service.

A notary and a client (e.g. an archivist) are both interested in that the assertion is rated by other notaries. The notary wants to build a good reputation. The client wants a further guarantee that the assertion is correct. Therefore, after issuing the assertion, the notary requests the set of notaries to give opinions on the assertion. The set randomly selects $m > 1$ rating notaries, who examine and rate the assertion's correctness. A positive opinion is given if correctness is verified, otherwise a negative opinion is given. An opinion is neutral if the rating notary cannot examine the assertion. The notary's reputation is based on the opinions given on his assertions. The set is interested in storing the given opinions to prevent any notary from tampering with them.

Before believing in a protected document, a verifier verifies the cryptographic evidence and assesses the trust in the

document's protection. The verification consists of verifying signatures and certificates. The verifier assess trust from the reputation of the notaries using the trust model presented in Section III-B. The trust placed in a document's protection is the conjunction of each involved notary's trust. The verifier believes in the document if: a) cryptographic evidence is correct; and b) he accepts the measure of trust in the protection.

B. The Infrastructure

A notary signs his own certificates and applies to the master authority for joining the set of notaries. The master authority signs and distributes to the community a list [17] comprising the authorised notaries' certificates. The list also identifies where a notary's services and repository are available. There is a public peer-to-peer network that the notaries join for rating and sharing reputation data. The notaries use a distributed random number generator [18] to randomly select m authorised notaries to rate a given assertion. For each opinion that a selected notary gives, the set confers him the chance to have one of his assertions rated. This exchange of opinions motivates notaries to contribute to the reputation scheme. The set of notaries replicates reputation data among themselves to prevent tampering. Each notary publishes replicated reputation data in his public repository.

C. The Notarial Assertion & Reputation

A notary creates an assertion that a document's signature and corresponding signer's certificate existed and that this certificate was valid. One notary can renew (i.e. reissue) another notary's assertion. An assertion is defined as follows.

Definition 1. An assertion in regard to a signature σ on document d is a tuple $A_{\mathcal{N}_v}(\sigma) = \langle H, \sigma, c, \tau, v, W, pk, \delta \rangle$. Here H is a set of tuples of the form $\langle h, y \rangle$, with h being a hash function, $y = h(d||\sigma)$, and $||$ is the binary concatenation. Also, c is the document signer's certificate, τ is the date and time when the validity of c and the existence of d and σ were attested, $v \geq 1$ is the assertion's version, W is the conjunction of the trust placed in the notaries $\mathcal{N}_1, \mathcal{N}_2, \ldots, \mathcal{N}_{v-1}$ that issued the previous versions of the assertion. The public key of notary $\mathcal{N}_v$ is pk and δ is $\mathcal{N}_v$'s signature on $A_{\mathcal{N}_v}(\sigma)$. For $v = 1$, we define $W = \langle 1, 0, 0 \rangle$.

An assertion $A_{\mathcal{N}}(\sigma)$ is evaluated by $m > 1$ notaries other than notary $\mathcal{N}$. Each rating notary $\mathcal{N}_j$ gives an opinion $R_j(A_{\mathcal{N}}(\sigma)) = \langle p_j, n_j, e_j \rangle$ such that $p_j + n_j + e_j = 1$. The rating notaries combine the m given opinions in the assertion's reputation

$$R(A_{\mathcal{N}}(\sigma)) = \left\langle \sum_{j=1}^{m} p_j, \sum_{j=1}^{m} n_j, \sum_{j=1}^{m} e_j \right\rangle. \quad (3)$$

Given $k \geq 1$ assertions A_1 to A_k issued by a notary $\mathcal{N}$ and corresponding reputations $R(A_i) = \langle p_i, n_i, e_i \rangle$ for $1 \leq i \leq k$, the reputation of $\mathcal{N}$ is

$$R(\mathcal{N}) = \left\langle \sum_{i=1}^{k} p_i, \sum_{i=1}^{k} n_i, \sum_{i=1}^{k} e_i \right\rangle. \quad (4)$$

D. Assertion Creation

In this section, we describe how a notary creates an assertion and reputations are updated.

A notary $\mathcal{N}$ receives assertion requests that consist of:

- a signature σ on a document d;
- a tuple $\langle h, y \rangle$, where h is a hash function and $y = h(d||\sigma)$;
- the signer's certificate chain CC, which includes the signer's certificate c; and
- the up to date revocation status RR for each certificate in CC.

The notary $\mathcal{N}$ performs verifications `2` and `3` (Table II). If `2` or `3` fails, then notary $\mathcal{N}$ aborts. Otherwise, the notary $\mathcal{N}$ creates an assertion $A_{\mathcal{N}_v}(\sigma) = \langle H, \sigma, c, \tau, v, W, pk, \delta \rangle$, with $H = \{\langle h, y \rangle\}$, τ the current date and time, $v = 1$, $W = \langle 1, 0, 0 \rangle$. The notary $\mathcal{N}$ returns $A_{\mathcal{N}_v}$.

TABLE II
VERIFICATIONS FOR CREATING AN ASSERTION.

Id.	Description
1	is δ valid under pk?
2	is the cryptographic algorithm of σ still secure?
3	is c currently valid given CC and RR?
4	is τ close to the current date and time?
5	is c the same in CC and $A_{\mathcal{N}_v}(\sigma)$?
6	are the cryptographic algorithms of $A_{\mathcal{N}_v}(\sigma)$ still secure?

Notary $\mathcal{N}$ sends a message comprising CC, RR, and $A_{\mathcal{N}_v}(\sigma)$ in the peer-to-peer network. Each rating notary performs verifications `1` to `6` (Table II) to rate $A_{\mathcal{N}_v}(\sigma)$ as follows. If `1` fails, a rating notary gives no opinion and aborts. If `1` to `6` succeed, he gives a positive opinion. If any of `2` to `6` fails, he gives a negative opinion. He gives a neutral opinion if any of `2` to `6` cannot be performed rather than failing. For instance, he has problems with his internal clock and cannot compare the current time with τ in `4`. The rating notaries build and broadcast $R(A_{\mathcal{N}_v}(\sigma))$ (3) in the peer-to-peer network.

Notice that the rating notaries cannot verify the submitted tuple $\langle h, y \rangle$. This is because they do not have access to document d to check whether $y = h(d||\sigma)$. Verifiers are responsible for such verification (Section V-F).

E. Assertion Renewal

In this section, we describe how a notary renews an assertion and reputations are updated. The renewal is necessary to ensure the security of the cryptographic algorithms used in the assertion.

A notary $\mathcal{N}_{v+1}$ receives a renewal request that consists of

- an assertion $A_{\mathcal{N}_v}(\sigma) = \langle H, \sigma, c, \tau, v, W_v, pk_v, \delta_v \rangle$; and
- a new tuple $\langle h', y' \rangle$, where h' is a secure hash function, $y' = h'(d||\sigma)$ and σ is a signature on document d.

The notary $\mathcal{N}_{v+1}$ performs verifications `1` and `2` for $A_{\mathcal{N}_v}(\sigma)$, and `2` for $\langle h', y' \rangle$ (Table III). If `1` or `2` fails, then he aborts the renewal. Otherwise, he creates and returns $A_{\mathcal{N}_{v+1}}(\sigma) = \langle H', \sigma, c, \tau, v+1, W_{v+1}, pk_{v+1}, \delta_{v+1} \rangle$, where $H' = H \cup \{\langle h', y' \rangle\}$ and $W_{v+1} = TM(\mathcal{N}_v) \odot W_v$ (2). Notice that $TM(\mathcal{N}_v)$ (1) is derived from $R(\mathcal{N}_v)$ (4).

TABLE III
VERIFICATIONS FOR RENEWING AN ASSERTION.

Id.	Description
1	is δ valid under pk?
2	are the cryptographic algorithms still secure?
3	is $H' \cap H = H$?
4	are τ, c and σ the same in $A_{\mathcal{N}_v}(\sigma)$ and $A_{\mathcal{N}_{v+1}}(\sigma)$?
5	is W_{v+1} correct considering W_v and $R(\mathcal{N}_v)$?

The notary $\mathcal{N}_{v+1}$ sends a message comprising $A_{\mathcal{N}_v}(\sigma)$ and $A_{\mathcal{N}_{v+1}}(\sigma)$ in the peer-to-peer network. Each rating notary performs verifications 1 to 5 (Table III) to rate $A_{\mathcal{N}_{v+1}}(\sigma)$ as follows. A rating notary gives no opinion and aborts if 1 fails for $A_{\mathcal{N}_v}(\sigma)$ or $A_{\mathcal{N}_{v+1}}(\sigma)$. He gives a positive opinion if 2 for $A_{\mathcal{N}_v}(\sigma)$ and $A_{\mathcal{N}_{v+1}}(\sigma)$ succeeds, and 3 to 5 succeed. Otherwise, he gives a negative opinion. He gives a neutral opinion if he cannot perform 2 to 5. For instance, he has insufficient information about the security of a particular cryptographic algorithm in 2. The rating notaries build and broadcast $R(A_{\mathcal{N}_{v+1}}(\sigma))$ in the peer-to-peer network.

F. Protected Document Verification

We now detail how a verifier checks the protection of a document. The verification comprises two phases: (a) evaluating the protection using the given cryptographic evidence; and (b) assessing the trust in the protection.

An archivist provides a verifier with:

- a document d;
- a signature σ on d;
- an assertion $A_{\mathcal{N}_v}(\sigma) = \langle H, \sigma, c, \tau, v, W, pk, \delta \rangle$.

In phase (a), the verifier performs verifications 1 to 5 (Table IV). Verifications 1 and 2 ensure authenticity for $A_{\mathcal{N}_v}(\sigma)$. Verifications 3 to 4 ensure proof of existence for d and σ. Note that 4 is necessary because notaries cannot verify whether each submitted hash y equals $h(d||\sigma)$. Verifications 3 to 5 ensure authenticity for σ. Note that only 5 is not sufficient to ensure authenticity, because the cryptographic algorithm of σ becomes insecure in the long term.

TABLE IV
VERIFICATIONS OF PROTECTION EVIDENCE.

Id.	Description
1	are the cryptographic algorithms of δ still secure?
2	is δ valid under pk?
3	is at least one hash algorithm h in H still secure?
4	does each hash value y in H match $h(d||\sigma)$?
5	is σ valid under the signer's public key in c?

In phase (b), the verifier evaluates the conjunction of the trust placed in notaries $\mathcal{N}_1$ to $\mathcal{N}_v$. Recall that W is the conjunction of trust in $\mathcal{N}_1$ to $\mathcal{N}_{v-1}$. Therefore, the verifier evaluates $W \odot TM(\mathcal{N}_v)$, where $TM(\mathcal{N}_v)$ is derived from $R(\mathcal{N}_v)$ available in the notaries' repositories. Based on the conjunction's value and a trust threshold, the verifier decides whether to believe in the document's protection. For example, the verifier believes in the protection if the trust degree t (1) of the conjunction's value is greater than 0.5.

Optionally, the verifier decides whether to believe in $A_{\mathcal{N}_v}(\sigma)$ based on the balance of opinions in $R(A_{\mathcal{N}_v}(\sigma))$. For instance, if there is an overwhelming number of negative opinions, the verifier considers $A_{\mathcal{N}_v}(\sigma)$ untrustworthy and the corresponding notary malicious or compromised. Correspondingly, the verifier does not believe in the protection of d.

G. Discussion

Expecting that notaries behave properly and their signature keys are not compromised is a strong trust assumption in notarial schemes. These assumptions are softened in our proposal thanks to its design and right incentives. Reputation is an incentive for notaries to behave as a community expects. The reputation of an assertion allows to detect bad behavior or even the compromise of a notary. Interestingly, the compromise of the notary does not affect the trust in a document's protection as long as the assertion has been positively rated. Moreover, because rating notaries are chosen at random, collusion among notaries against signers or verifiers is unlikely. Collusion to manipulate reputation data is also unlikely, because such data is replicated among notaries and any tampering can be easily detected. A notary is likely to contribute with opinions on other notaries' assertions in order for the notary to ask for opinions on his assertions.

The master authority signs the list comprising the authorised notaries' certificates. The list prevents bogus notaries from offering services and giving opinions on assertions, thereby avoiding sybil attacks [19].

All in all, our proposal only requires the following weak trust assumptions:

- most notaries give fair opinions on other notaries' assertions; and
- a rating value is not tampered with after it has been published.

VI. CONCLUSIONS

We analysed trust in the context of protecting documents in the long term. We identified the trust assumptions that a verifier has to make in order to trust in authenticity, integrity and proof of existence. We showed that trust decreases and tends to zero as the number of trusted parties increases over time. This leads us to the conclusion that cryptographic evidence becomes useless as the trust in the accumulated evidence fades out in the long term. Therefore, reducing the number of trusted parties over time is as important as renewing of cryptographic evidence.

Although the degradation of trust is still an issue in the long term, the proposed scheme assesses trust better than timestamping and previous notarial schemes. Instead of treating trust as a boolean variable or qualitative levels, the proposed scheme renders trust as a function of the trusted parties' reputation. This leads verifiers to conscious trust decisions instead of simply accepting CAs, TSAs or notaries as fully trustworthy because software or operational system vendors trust in them.

ACKNOWLEDGMENTS

The authors thank Ciaran Mullan and the anonymous reviewers for the comments which helped to improve the final version of this manuscript.

REFERENCES

[1] Department of Health, "Annex D1: Health Records Retention Schedule." [Online]. Available: http://www.dh.gov.uk/prod_consum_dh/groups/dh_digitalassets/documents/digitalasset/dh_093027.pdf

[2] Centre of Registers and Information Systems, "e-Land Register." [Online]. Available: http://www.egov-estonia.eu/e-land-register

[3] Intellectual Property Office, "Retention and Disposal Policy for Patent Related Records," jun 2012. [Online]. Available: http://www.ipo.gov.uk/p-retentiondisposal.pdf

[4] D. Cooper, S. Santesson, S. Farrell, S. Boeyen, R. Housley, and W. Polk, "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile," RFC 5280 (Proposed Standard), Internet Engineering Task Force, May 2008. [Online]. Available: http://www.ietf.org/rfc/rfc5280.txt

[5] M. A. G. Vigil, D. Cabarcas, A. Wiesmaier, and J. Buchmann, "Authenticity, integrity and proof of existence for long-term archiving: a survey," Cryptology ePrint Archive, Report 2012/499, 2012, http://eprint.iacr.org/.

[6] P. Zimmermann, *The official PGP user's guide*, 1995.

[7] A. Jøsang, "An Algebra for Assessing Trust in Certification Chains," in *NDSS*. The Internet Society, 1999.

[8] M. Reiter and S. Stubblebine, "Authentication metric analysis and design," *ACM Transactions on Information and System Security*, vol. 2, no. 2, pp. 138–158, 1999.

[9] D. Bayer, W. S. Stornetta, and S. Haber, "Improving the efficiency and reliability of digital time-stamping," in *Sequences II: Methods in Communication, Security and Computer Science*. Springer-Verlag, 1993, pp. 329–334.

[10] D. Lekkas and D. Gritzalis, "Cumulative notarization for long-term preservation of digital signatures," *Computers & Security*, vol. 23, no. 5, pp. 413 – 424, 2004. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167404804001014

[11] M. A. G. Vigil, C. T. Moecke, R. F. Custódio, and M. Volkamer, "The Notary Based PKI – A Lightweight PKI for Long-term Signatures on Documents," in *Public Key Infrastructure, 9th European PKI Workshop: Theory and Practice, EuroPKI 2012, Pisa, Italy, Septemper 13-14, 2008, Proceedings*, ser. EuroPKI '12. Berlin, Heidelberg: Springer-Verlag, 2012, p. to appear.

[12] J. Huang and D. Nicol, "A formal-semantics-based calculus of trust," *Internet Computing, IEEE*, vol. 14, no. 5, pp. 38 –46, sept.-oct. 2010.

[13] A. Avizienis, J.-C. Laprie, B. Randell, and C. Landwehr, "Basic concepts and taxonomy of dependable and secure computing," *Dependable and Secure Computing, IEEE Transactions on*, vol. 1, no. 1, pp. 11 – 33, jan.-march 2004.

[14] S. Hauke, F. Volk, S. Habib, and M. Mühlhäuser, "Integrating indicators of trustworthiness into reputation-based trust models," in *Trust Management VI*, ser. IFIP Advances in Information and Communication Technology, T. Dimitrakos, R. Moona, D. Patel, and D. McKnight, Eds. Springer Berlin Heidelberg, 2012, vol. 374, pp. 158–173.

[15] A. Josang, R. Hayward, and S. Pope, "Trust network analysis with subjective logic," in *Proceedings of the 29th Australasian Computer Science Conference - Volume 48*, ser. ACSC '06. Darlinghurst, Australia, Australia: Australian Computer Society, Inc., 2006, pp. 85–94.

[16] P. Eckersley and J. Burns, "The (decentralized) ssl observatory," in *USENIX Security Symposium (Invited Talk)*, 2011.

[17] ETSI, "Electronic Signatures and Infrastructures (ESI); Provision of harmonized Trust-service status information," ETSI, Tech. Rep., 2006.

[18] B. Awerbuch and C. Scheideler, "Robust random number generation for peer-to-peer systems," *Theor. Comput. Sci.*, vol. 410, no. 6-7, pp. 453–466, 2009.

[19] J. Douceur, "The sybil attack," in *Peer-to-Peer Systems*, ser. Lecture Notes in Computer Science, P. Druschel, F. Kaashoek, and A. Rowstron, Eds. Springer Berlin Heidelberg, 2002, vol. 2429, pp. 251–260. [Online]. Available: http://dx.doi.org/10.1007/3-540-45748-8_24

[20] J. Benaloh and M. de Mare, "Efficient broadcast time-stamping," *Clarkson University Department of Mathematics and Computer Science TR*, pp. 91–1, 1991.