

### VOLUME 2

# The proof that endures.

Formats, levels, timestamps, validation, trusted lists and preservation: everything needed so that a signature made today can still be proven thirty years from now.

Signed agreement.pdf

VALIDATION REPORT

Valid · LTA

|                     |                     |
|---------------------|---------------------|
| ✓ Content integrity | unchanged           |
| ✓ Certificate chain | trusted list        |
| ✓ Timestamp         | 04/03/2026 14:07    |
| ✓ Revocation status | embedded in the PDF |
| ✓ Archive timestamp | current             |

ARCHIVE TIMESTAMPS

2026

initial

2034

renewed

2042

renewed

2050

planned

## **Rúbrica. Notebooks on signature, identity and digital trust**

Volume 2. The Proof That Endures

First edition, October 2026.

Montevideo, Uruguay.

This book is a work of general information. It describes technical and legal principles of general scope and does not constitute legal advice. The validity of a signature in any specific case depends on the applicable law, the type of act and the available evidence; before making a decision, consult a qualified professional in the relevant jurisdiction.

References to laws, institutions and companies are made for historical and educational purposes. The trademarks mentioned belong to their respective owners.

Set in Figtree and JetBrains Mono, both under the SIL Open Font License.

RÚBRICA · VOLUME 2

# The Proof That Endures

What happens to a signature after signing, and how its value is preserved over time.

# Contents

|    |                        |    |
|----|------------------------|----|
|    | Preface                | 5  |
| 01 | How a signature dies   | 7  |
| 02 | The formats            | 11 |
| 03 | Signature levels       | 15 |
| 04 | The signature in PDF   | 19 |
| 05 | Timestamps             | 23 |
| 06 | Validation             | 27 |
| 07 | Trusted lists          | 31 |
| 08 | Preservation           | 35 |
| 09 | The evidence record    | 39 |
| 10 | The signature in court | 43 |
|    | Glossary               | 47 |
|    | Further reading        | 51 |

BEFORE WE BEGIN

# Signing takes an instant. Proof is forever.

The first volume of this series ended at the moment someone signs. This one begins right after, when the signature must start to defend itself on its own.

A handwritten signature ages along with its paper. It fades, gets stained, goes missing in a move, but as long as the paper exists it remains the same signature and an expert can examine it. With the digital signature something paradoxical happens: the document can be kept intact, bit for bit, for decades, and yet the signature can become impossible to verify. Certificates expire, the authorities that issued them shut down, algorithms weaken, and the programs that could read a format disappear.

This volume is about how to avoid that fate. It is the most technical in the series, out of necessity: preserving the proof depends on details that never appear on the signer's screen but that decide, years later, whether a document holds up or not. We have tried to explain those details without losing sight of the practical question that justifies them: what an organization needs in order to demonstrate, ten, twenty or thirty years from now, that a document was signed by whom it claims, when it claims and without changes.

The chapters follow a signature through its life. We begin with the risks that threaten it as time passes. We continue with the formats and levels designed to withstand them, with particular attention to PDF, which carries most of the signed documents in the world. We then devote chapters to timestamps, validation and trusted lists, the three pieces that allow a third party to verify a signature without relying on whoever produced it. We close with long term preservation, with the evidence record of the signing process and with what happens when a signature reaches a court.

IF YOU HAVE NOT READ VOLUME 1

This volume assumes some concepts are already familiar: the **hash** or cryptographic digest, the **key pair**, the **certificate** and the **chain of trust**, **revocation** and the **timestamp**, and the three **levels** of electronic signature. All of them are explained in chapters three and four of the first volume, and the glossary of this book summarizes them.

# How a signature dies.

A digital signature is not erased and does not fade. It becomes unverifiable. This chapter lists the six reasons why that happens and the idea that makes it possible to address them.

Imagine a contract signed digitally in 2026 that must be presented before a court in 2046. The file has been kept without a single change. The judge, or the expert assisting the judge, opens the document in a validator. What can go wrong? Much more than it seems. The signature depends on a set of elements external to the document, and each of them has its own life cycle.

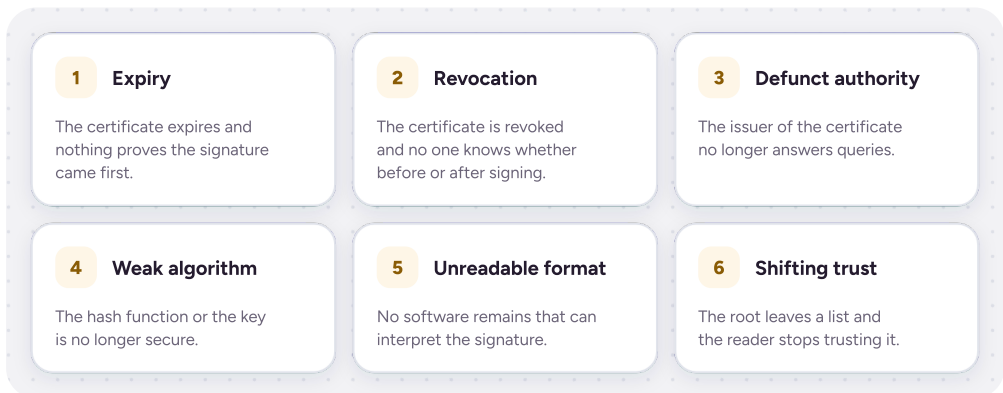


**FIGURE 1** Six ways a digital signature stops being verifiable, even when the file is kept intact.

## **The certificate expires**

Every certificate has an expiration date, usually between one and five years after it is issued. After that date, the certificate can no longer be used to sign. That does not invalidate the signatures made while it was valid, but it makes it necessary to prove that the signature was made before expiration. Without reliable proof of the moment of signing, a validator in 2046 will only see a certificate that expired eighteen years earlier and will be unable to conclude anything.

## **The certificate is revoked**

A certificate can be revoked before it expires, for example because its holder reported the loss of a card. Signatures made before revocation remain valid; those made afterward do not. Once again, everything depends on knowing when the signature was made and what the status of the certificate was at that moment. And there is an aggravating factor: certification authorities are not required to keep information about expired certificates forever. A query made twenty years later may get no answer.

## **The authority disappears**

Certification authorities are organizations, and organizations close, merge or lose their accreditation. When that happens, their revocation status services may stop working and their root certificate may be removed from trusted lists. A signature that depended on querying that authority online is left orphaned.

## **The algorithm weakens**

As we saw in the first volume, cryptographic algorithms have a useful life. A signature made with a hash function that later proves vulnerable to collisions loses evidentiary value: someone could argue that the signed document was replaced by another with the same hash. Standardization bodies periodically publish tables of the algorithms and key lengths they consider secure, and the dates from which they cease to be so.

## The format becomes unreadable

A signed document is useless if there are no programs able to open it and interpret its signature. Proprietary formats, or poorly documented variants of open formats, are a long term risk. The history of computing is full of files that nobody knows how to read.

## Trust changes hands

Finally, the decision about which authorities deserve trust is not fixed. Official lists are updated, PDF readers modify their own lists, operating systems remove compromised roots. A signature that appears trustworthy in a program today may appear as “not verified” tomorrow, without anything having changed in the document.

## The idea that solves it: freezing the proof

The six threats have something in common: they depend on information that lives outside the document and changes over time. The answer the industry built over two decades is equally simple in principle: **gather that information while it is still available, store it alongside the signature and protect it in turn with proof of time.**

Concretely, that means adding three things to the document shortly after signing: a timestamp that fixes when it was signed, the complete certificates of the chain of trust, and the revocation responses obtained at that moment. With that, a future validator does not need to query anyone: the document itself holds everything required to reconstruct the state of the world on the day of signing. And so that this information cannot be altered or lose value as algorithms age, it is protected with successive timestamps, renewed before the previous ones weaken.

That idea, applied systematically, gave rise to the signature levels we will study in chapter three. First, it is worth getting to know the formats that make it possible.

**KEY TAKEAWAY**

A digital signature is not preserved just by preserving the file. It is preserved by preserving the **proof of when it was signed**, the **certificate chain** and the **revocation status** at that moment, all protected in turn against the passage of time.

## Four formats and a container.

A digital signature has to live somewhere: inside the document, around it or beside it. The answers to that question gave rise to a family of formats standardized in Europe and used throughout the world.

Technically, a signature is a block of data: the cryptographic signature value, the signer's certificate and a set of attributes, such as the declared time of signing or the reason for signing. That block has to be associated with the signed document in some way and with a structure that any program can interpret. During the 1990s each vendor solved this in its own way. From 2000 onward, the European Telecommunications Standards Institute, known as ETSI, began publishing common specifications that today form the basis of the advanced signature in Europe and in much of the world.

### Three ways of relating signature and document

Before discussing formats, it helps to distinguish three basic configurations that appear in all of them.

- ✓ **Enveloped signature.** The signature lives inside the document. This is the case of the signed PDF: the file contains both the content and the signature, and an ordinary reader can display both.
- ✓ **Enveloping signature.** The signature contains the document. The result is a new file, with a different extension, that must be opened with a program able to extract the content.
- **Detached signature.** The signature lives in a separate file, which refers to the document by its hash. The original document remains untouched, but the two files

- ✓ must be kept together.

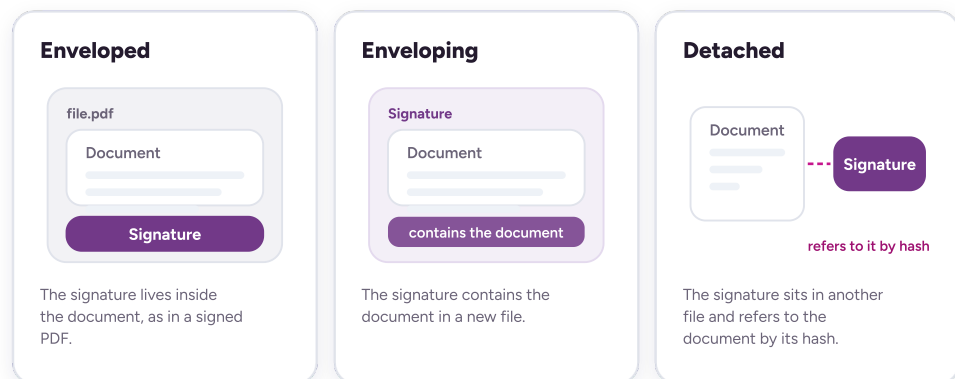


**FIGURE 2** Three ways of relating signature and document. Every format in the European family supports at least one of them.

## CAdES: signing anything

CAdES derives from an older and widely adopted standard, the Cryptographic Message Syntax known as CMS, itself a descendant of specifications published by a security company in the 1990s. It is a compact binary format that is agnostic about content: it can sign a PDF, an image, a compressed archive or any sequence of bytes. It is widely used for detached signatures and in systems that sign large volumes of data, such as electronic invoicing in some countries. Its current standard is EN 319 122.

## XAdES: signing structured documents

XAdES builds on the XML signature defined by the W3C consortium and is designed for documents structured in that language. It can sign an entire document or only some of its parts, which made it very popular for data exchange between systems, in electronic government and in electronic invoicing in many Latin American and European countries. Its current standard is EN 319 132.

## **PAdES: the signature you can see**

PAdES defines how to embed advanced signatures within a PDF. Its great advantage is the user experience: the signed document opens in any reader, the signature can have a visible representation on the page and the reader can show its status. That is why it is the dominant format in processes involving people. It was first published in 2009 and its current standard is EN 319 142. Chapter four is devoted to it.

## **JAdES: signing for interfaces**

JAdES is the youngest member of the family, published in 2021. It is based on signatures for the JSON format, the language in which most web applications and programming interfaces communicate today. It is designed for signing messages between systems and digital credentials, and it is likely to gain importance with identity wallets, which handle credentials in formats of that family.

## **ASiC: the container**

ASiC is not a signature format but a container: a compressed file with a defined structure that brings together one or more documents with their signatures, whether CAdES or XAdES. It solves the problem of detached signatures, which require several files to be kept together, and it allows a set of documents of any type to be signed in a single act. Some countries have adopted it as the standard format for signed documents that are not PDF.

| FORMAT       | TECHNICAL BASIS        | TYPICAL USE                                                          | STANDARD   |
|--------------|------------------------|----------------------------------------------------------------------|------------|
| <b>CAdES</b> | CMS, binary            | Any file, detached signatures, large volumes                         | EN 319 122 |
| <b>XAdES</b> | XML signature          | Structured documents, exchange between systems, electronic invoicing | EN 319 132 |
| <b>PAdES</b> | PDF with CMS signature | Documents read and signed by people                                  | EN 319 142 |
| <b>JAdES</b> | JSON signature         | Messages between systems, credentials                                | TS 119 182 |
| <b>ASiC</b>  | Compressed container   | Sets of documents with their signatures                              | EN 319 162 |

## Why standards matter

These formats share a vocabulary of attributes and a logic of levels, so that a signature produced by one program can be validated by another from a different vendor. That interoperability is the most important safeguard against the threat of the unreadable format we saw in the previous chapter. A document signed in a standardized, open format can be read as long as someone implements the standard, and the European standards have open source reference implementations, maintained with public support, that anyone can use.

The practical recommendation is simple: for any signature solution, ask what format it produces and which standard it complies with, and be wary of formats that can only be validated with the vendor's own tools.

## Four steps toward permanence.

The European formats define successive levels that add, one on top of another, the proof a signature needs to survive over time.

Understanding them means understanding how long a signature can last.

These levels should not be confused with the legal levels of the first volume. Those, simple, advanced and qualified, describe how much the law trusts a signature. These describe how much technical proof a signature carries with it so that it can be verified in the future. They are two different axes: a qualified signature can have a basic technical level and be left defenseless within a few years, while an advanced signature can be perfectly prepared to last for decades.

The current ETSI standards define four levels, common to CAdES, XAdES, PAdES and JAdES. They are designated by the letters B, T, LT and LTA, often preceded by the name of the format.

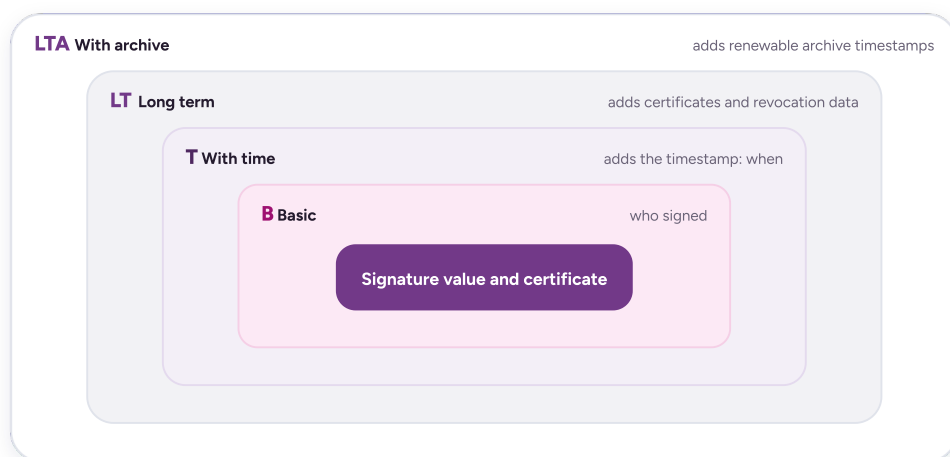


**FIGURE 3** The four technical levels of an advanced signature. Each layer wraps the previous one and adds the proof it needs to endure.

## Level B: the basic signature

This is the signature as it comes out of the act of signing: the cryptographic value, the signer's certificate and some mandatory attributes, in particular a protected reference to that certificate, which prevents it from being replaced by another with the same key. A level B signature proves that someone holding the private key signed that content. It does not prove when, and its verification depends on whether, at the time of validation, the certificate chain and its revocation status can still be checked.

A level B signature is sufficient for immediate validation, but it begins to lose strength as soon as the certificate expires.

## Level T: the signature with time

To level B is added a timestamp issued by a time stamping authority, applied over the signature value. From that moment on there is independent proof that the signature existed at a specific date and time. This solves the problem of a certificate that expires or is revoked after signing: it is enough to check that the timestamp predates the expiration or the revocation.

Level T is the reasonable minimum for any document that must retain its value beyond a few months.

## Level LT: the self sufficient signature

To level T is added the validation material: all the certificates in the chain, up to the root, and the revocation responses, whether lists or OCSP responses, obtained after the timestamp. The letters stand for *long term*.

With level LT, the signature no longer depends on external services. A validator twenty years from now will not need to query the certification authority, which may have ceased to exist: it will find in the document the evidence that the certificate was valid when the signature was made. This is the answer to the threats of the revoked certificate and the disappearing authority.

## Level LTA: the archivable signature

One threat remains unresolved: that of the aging algorithm. Everything level LT gathered is protected by signatures and timestamps made with today's algorithms. If those algorithms weaken fifteen years from now, someone could argue that the validation material was fabricated afterward.

Level LTA, for *long term with archive*, adds an archive timestamp covering everything that came before: the signature, its timestamp, the certificates and the revocation responses. That archive timestamp can be renewed periodically, with stronger algorithms, before the previous one weakens. Each renewal covers everything that existed before, including the previous timestamp, forming a chain of timestamps that extends the life of the signature indefinitely.

### A USEFUL IMAGE

Think of a letter kept in a safe. Level B is the signed letter. Level T adds the postmark. Level LT adds, inside the envelope, certified copies of the documents proving that the signer was who they claimed to be on the day of the postmark. Level LTA keeps all of that in a box closed with a dated band, and every few years a notary checks the band and places a new, stronger one on top.

## Who adds each level

The levels need not all be added at the moment of signing. Level B is produced by the signer. The level T timestamp can be requested by the signer or by the platform immediately afterward. The level LT validation material should be added as soon as possible, but ideally after a short grace period, to allow time for any revocation made just before the signature to be published. The level LTA archive timestamp can be added by the platform when the process closes or by a preservation service later on.

This distribution has design consequences. A platform that coordinates signing processes is in a privileged position to raise the level of the signatures it collects: it knows the moment of each signature, it can request timestamps and revocation responses from third party services, and it can apply the archive timestamp when the last signer finishes.

## When an external service does not respond

Raising the level requires contacting external services: time stamping authorities and revocation services. Those services can fail. A good practice, and one worth requiring, is for the platform to have alternatives configured for each service and, if it still cannot reach the intended level, to leave the document at the level actually achieved, report it clearly and record it, instead of failing silently or appearing to have a level it does not have. A level T document declared as such is far more useful than one that claims to be LTA and is not.

### KEY TAKEAWAY

**B** proves who signed. **T** adds when. **LT** keeps within the document the proof that the certificate was valid at that moment. **LTA** protects all of the above with renewable timestamps against the aging of algorithms.

# What happens inside a PDF.

PDF carries most of the signed documents in the world. The way it holds signatures has particular features that explain many of the messages, sometimes puzzling, that readers display.

## A file that grows in layers

The most important feature of PDF for understanding the signature is the incremental update. When a PDF is modified, the program does not need to rewrite the file: it can append at the end a new section describing the changes, leaving everything before it untouched. The file grows like an onion, in layers, and each layer is a *revision* of the document.

The signature takes advantage of this mechanism. When someone signs a PDF, a revision containing the signature is added, and the signature covers all the bytes of the file up to that point, except the space reserved for the signature value itself. If another person signs later, a new revision is added that covers everything before it, including the first signature. Each signature protects the version of the document that existed at the moment it was made.

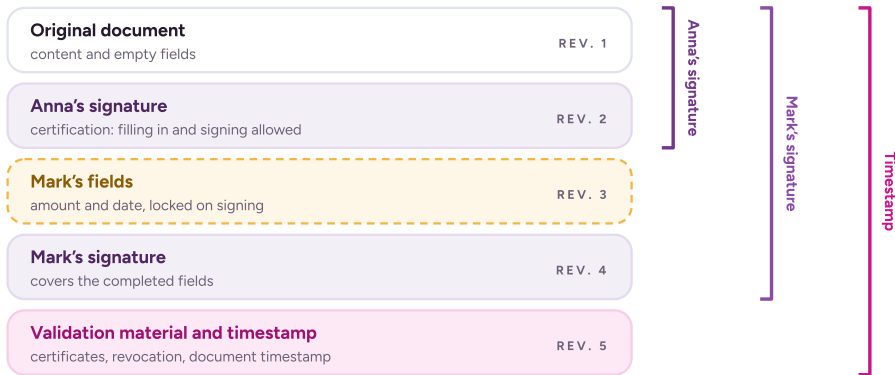


**FIGURE 4** A signed PDF grows by revisions. Each signature covers everything up to it; the document timestamp covers the whole file, including the validation material.

This has an important consequence: a signed PDF can keep changing without breaking the earlier signatures, because each one continues to protect its own revision. A validator can show, for each signature, exactly which version of the document that person signed and what changed afterward. The question then becomes whether those later changes were permitted.

## What can be changed after signing

The format provides two mechanisms for declaring which changes are allowed after a signature.

### Document certification

The first signature on a document can be a *certification signature*, which in addition to signing declares which subsequent changes are accepted. There are three options: no changes; only filling in forms and adding signatures; or, in addition, adding annotations and comments. A reader that follows the standard will flag as altered a document that received changes beyond what was permitted, even if each individual signature remains mathematically valid. In the specification this mechanism is called DocMDP.

## Field locking

The second mechanism, called FieldMDP, allows each signature to lock certain form fields. Thus, when a signer fills in the amount and signs, they can prevent the next signer from changing that amount, even though the document still allows others to fill in their own fields. This is the tool that makes possible the processes in which several people enter data and sign on the same document.

### WHY A READER SAYS “MODIFIED”

Many of the warnings that alarm users are explained by these rules. A document that someone opened and saved again with a program that rewrites the file, or to which a comment was added when the certification did not allow it, can appear as modified even if the visible content is identical. The signature is not lying: the file changed.

## The appearance is not the signature

A signature in a PDF can have a visible representation: a box with the signer’s name, the date, a logo or the image of a *rúbrica*, the handwritten flourish. That representation is just that, an image. The real signature is the invisible cryptographic block associated with the field. Copying the image to another document does not copy the signature, and a document can be signed without any visible representation.

This distinction, obvious to a technician, is not obvious to many users, who trust what they see. That is why good practice is for the visible representation to point to something verifiable, such as a code that leads to a validator, and never to present an image as if it were the signature.

## Where the long term proof is stored

To reach levels LT and LTA from the previous chapter, the PDF needs a place to store the certificates and revocation responses. For this purpose the standards define a Document Security Store, a structure added in a revision after the signature that

gathers all the validation material. The archive timestamp, in turn, takes the form of a *document timestamp*: a special signature, issued by a time stamping authority, that covers the entire file up to that point, including the store.

The most widely used readers identify documents that contain this material as “LTV enabled”.

## **PDF/A: PDF for archiving**

The PDF format allows elements that hinder preservation: fonts that are not embedded, content that depends on external resources, executable code. The ISO 19005 family of standards, known as PDF/A, defines subsets of PDF designed for long term archiving, which require, among other things, that all fonts be embedded and prohibit dynamic content.

For documents that must be preserved for decades, the document should be PDF/A before the first signature. Converting it afterward is not possible without breaking the signatures, because the conversion rewrites the file. This detail, which seems minor, is one of the most frequent reasons why signed documents fail to meet an organization's archiving requirements.

## **Signing is not flattening**

One last practical warning. Some tools, at the end of a signing process, generate a new PDF that reproduces the content and the visible signatures, but without the cryptographic signatures, or with a single platform signature covering everything. That document may be useful for reading, but it does not preserve the individual proof of each signer. For any solution, it is worth asking whether the final document contains the signatures of each participant, each one over its own revision, or a reconstruction.

# The clock that everyone trusts.

The question “when?” runs through this entire volume. Timestamps are the technical answer to that question, and their history begins with an intellectual property problem.

## A laboratory problem

In 1991, two researchers at a US telecommunications research laboratory, Stuart Haber and W. Scott Stornetta, published a paper whose title posed the question plainly: how to timestamp a digital document. The problem that interested them was that of laboratory notebooks: how to prove that an idea or a result existed on a certain date, when digital files can be modified without leaving a trace and a computer clock can be set at will.

Their proposal had two parts. The first was to rely on a third party that receives the fingerprint of a document, adds the time to it and signs the whole. The second, more original, was to chain the timestamps together, so that each one includes information from the previous ones. That way, not even the third party itself could fabricate a backdated timestamp without altering the entire chain that follows. Years later, a company founded on those ideas began periodically publishing a summary fingerprint of all its timestamps in the classified ads of a widely circulated newspaper, an anchor that no one could alter once printed. This idea of anchoring a chain in a public, immutable medium reappears today in many evidence systems.

## The protocol that prevailed

In 2001, RFC 3161 was published, defining the time stamping protocol that almost all signature systems use today. The way it works is simple.

- 1 The party that needs the timestamp computes the fingerprint of the data to be stamped, which may be a document or, more often, the value of a signature. Only the fingerprint is sent: the authority never sees the content.
- 2 The time stamping authority receives the fingerprint, takes the time from a reliable clock, builds a structure containing the fingerprint, the time, a serial number and the declared accuracy, and signs it with its own key.
- 3 The timestamp is returned to the requester, who incorporates it into the signature or stores it alongside the document.
- 4 Anyone can later verify that the timestamp corresponds to that fingerprint and that it was signed by the authority, whose certificate has a specific usage for time stamping.



**FIGURE 5** How a timestamp is obtained. The authority receives only the hash, never the document, and returns a signed structure with the time from its clock.

## Where the time comes from

A timestamp is only as good as the authority's clock. Serious authorities synchronize their clocks with sources traceable to Coordinated Universal Time, such as national metrology laboratories or satellite signals, and continuously monitor drift. The European standards for time stamping authorities require an accuracy of at least one second with respect to Coordinated Universal Time, and require the authority to stop issuing timestamps if it loses synchronization.

## The qualified electronic timestamp

The eIDAS Regulation recognizes the qualified electronic timestamp, issued by a supervised trust service provider that meets those requirements. Its legal effect is a presumption: the accuracy of the date and time it indicates, and the integrity of the

data to which it refers, are presumed. As with the qualified signature, this presumption is recognized in all Member States. Outside Europe, several countries have their own accreditation schemes for time stamping authorities.

## What it proves and what it does not

A timestamp proves that a piece of data existed **before** a certain moment. It does not prove that the data did not exist earlier, nor who created it, nor that the time declared in the signature is the true one. If a signature claims to have been made at ten o'clock and the timestamp says five past ten, the only thing we know for certain is that it existed at five past ten. For most purposes that is exactly what matters: that the signature existed while the certificate was valid.

## Redundancy and degradation

Relying on a single time stamping authority is an operational risk: if it does not respond, there is no timestamp, and the signature remains at the basic level. That is why mature platforms configure several authorities in an order of preference, and document what they do if none responds. Some retry later and raise the level as soon as they can; others halt the process. What should never happen is for the system to carry on as if it had obtained the timestamp.

There is also a question of cost. Qualified timestamps are usually charged per unit, and a process with many signatures may require many timestamps. That is why it is common to distinguish between the timestamp on each signature, which yields level T, and the archive timestamp on the complete document, which is applied only once when the process is closed and yields level LTA.

### KEY POINTS

A timestamp is a signature by a third party over a fingerprint and a time. It proves that something existed before that moment, without revealing its content. Its value depends on trust in the authority and in its clock, which is why it should be qualified or accredited, and why more than one should be available.



## Valid, invalid or indeterminate.

Validating a signature looks like a yes or no question. In reality it is a procedure with several stages, a policy that decides what is required, and three possible answers, one of which is “it cannot be known yet.”

When a PDF reader shows a green check mark next to a signature, it is summarizing a fairly complex process in a single icon. In Europe that process is standardized by EN 319 102, which describes step by step how an advanced signature must be validated and what results it can produce. Understanding it helps in interpreting what validators say and, above all, what they do not say.

### The questions a validator asks

- 1 Is the document intact?** The fingerprint of the signed content is recomputed, and the signature is checked against that fingerprint using the public key in the certificate.
- 2 Is the certificate the one it claims to be?** The protected reference to the certificate inside the signature is checked against the certificate included.
- 3 Does the chain reach a trusted root?** The certificate chain is built up to a root, and that root is checked against the trust anchors the validator accepts.
- 4 What is the reference time?** If there is a valid timestamp, the signature is evaluated at the time of the timestamp. If not, it can only be evaluated at the present time.
- 5 Was the certificate valid at that time?** It is checked that the certificate was neither expired nor revoked, using the revocation information included in the

document or retrieved online.

- 6 **Were the algorithms acceptable, and are they still?** It is checked that the algorithms and key lengths were permitted at the reference time, and that the protection of the signature does not depend on algorithms now considered broken.
- 7 **Are the requirements of the policy met?** For example, if the policy requires a qualified certificate, the certificate is checked to confirm that it is one.

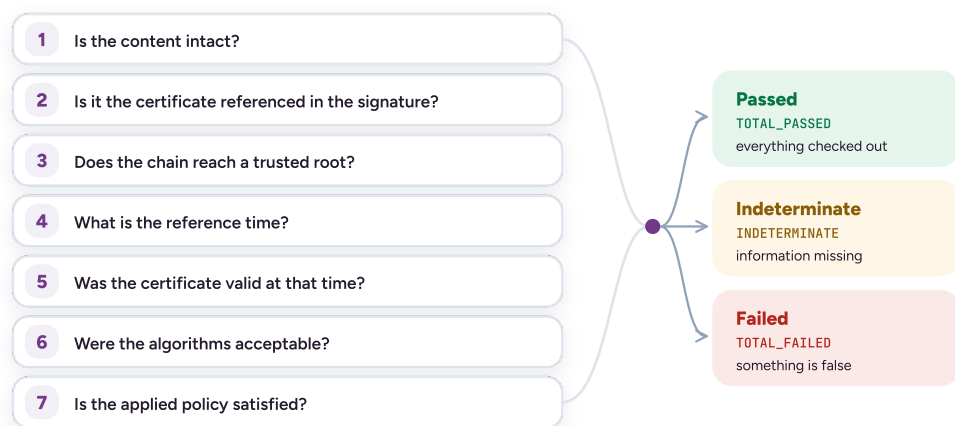


**FIGURE 6** The path of a validation and its three possible outcomes. Indeterminate does not mean false: it means information is missing to decide.

## Three results, not two

The standard provides for three possible conclusions, whose technical names have become common in validation reports.

- ✓ **Passed** (TOTAL\_PASSED). All checks were positive under the policy applied.
- ✓ **Failed** (TOTAL\_FAILED). Some check was definitively negative: the document was altered, the signature does not match, the certificate was revoked before signing.
- ✓ **Indeterminate** (INDETERMINATE). With the information available, no conclusion can be reached. Revocation information may be missing, the root may not be among the trusted ones, or there may be no proof that the signature predates a revocation.

The indeterminate result is the most frequent in practice and the most misunderstood. It does not mean the signature is false. It means something is missing in order to decide. Often that something can be obtained: by adding a trusted list, retrieving the revocation information or finding a timestamp. A good validation report states exactly what was missing.

## The validation policy

Validating always means validating *against something*. The validation policy defines which trust anchors are accepted, which algorithms are allowed and until when, what type of certificate is required, whether a timestamp is needed, and what tolerance applies to revocation information. Two validators with different policies may give different results for the same document, and both may be right.

This explains many discrepancies between programs. A general purpose PDF reader applies a policy designed for most of its users, with its own list of roots. An official European validator applies a policy based on the trusted lists of the Member States. A national validator may accept only the trust service providers accredited in its own country. When a signature appears valid in one and indeterminate in another, the first step is to ask which policy each one applied.

### A TYPICAL CASE

A qualified signature issued by a trust service provider in a Latin American country is opened in a European reader and appears as “not verified.” The signature is perfectly valid in its own country. The reader simply does not have that root among its anchors. The solution is not to distrust the signature but to validate it against the appropriate trusted list.

## Validators

There are validators of different kinds. PDF readers include one, convenient but limited to its own policy. Public bodies in several countries offer online validators based on their official lists. The European Commission maintains an open source library for

creating and validating signatures according to European standards, which serves as a reference for many products and which anyone can use. And signature platforms usually offer validators of their own.

The eIDAS Regulation also provides for a qualified validation service, provided by a supervised third party, whose result carries particular evidentiary weight. For high value documents, validation should not depend solely on the tools of whoever produced the signature.

## What a good validator should show

- The overall result and, for each signature, the individual result.
- The revision of the document covered by each signature and any later changes, indicating whether they were permitted.
- The complete certificate chain, with the root and the trusted list that vouches for it.
- The reference time and the evidence on which it is based, such as the timestamp.
- The revocation information used, with its source and date.
- The technical level reached, from B to LTA, and the legal level, if it can be determined.
- What could not be checked, and why.

The last point is what most distinguishes an honest validator. In a field where trust is everything, stating clearly what was not checked is worth as much as stating what was.

# Who decides whom to trust.

Every chain of trust ends in a human decision. Trusted lists turn that decision into a public document, signed and verifiable by machines.

In the first volume we saw that the certificate chain ends in a root that is trusted by explicit decision. The question is who makes that decision. For a long time it was, in practice, made by software vendors: every operating system, every browser and every PDF reader includes a list of roots it considers trustworthy, compiled according to its own criteria. For web browsing that model works reasonably well. For signatures with legal effect it has an obvious problem: the decision about which signatures are valid before the law should not depend on a private foreign company.

## The European answer

The European Union solved the problem with a system of trusted lists, whose format is standardized in the ETSI specification TS 119 612. Each Member State publishes a signed list of the trust service providers it supervises, the services each one provides, their current status and the history of that status. The European Commission in turn publishes a List of Trusted Lists that indicates where each country's list is located and with which certificate it is signed.

A validator that wants to know whether a signature is qualified starts from the List of Trusted Lists, downloads the list of the relevant country, verifies its signature, looks up the trust service provider, checks that the service that issued the certificate had qualified status at the time of signing and, only then, reaches a conclusion. The entire process is automatic and does not depend on any private list.

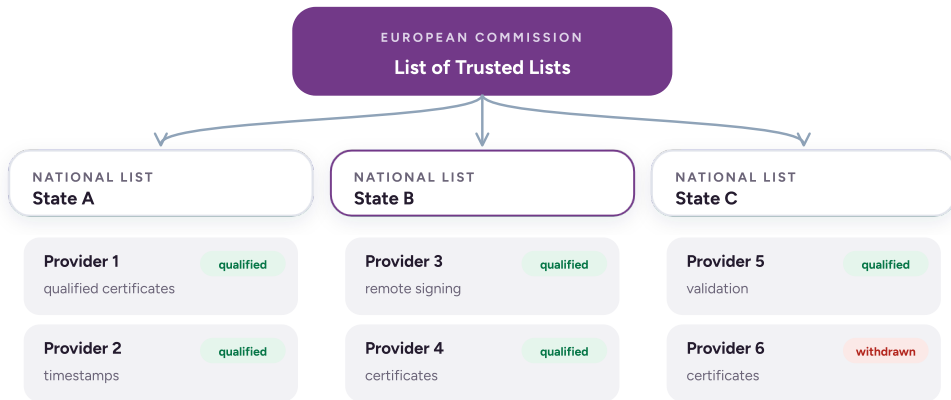


**FIGURE 7** The European trusted list system. A list of lists points to the national lists, which describe each provider, its services and their status over time.

## More than a list of roots

Trusted lists contain more information than a simple list of root certificates. They describe each service by its type, whether certificate issuance, time stamping, validation or preservation, and its status over time. This makes it possible to answer questions such as “was this trust service provider accredited to issue qualified certificates in March 2024?”, which is exactly what a validator needs when it evaluates a signature at the time it was made.

## Vendor lists

Software vendors did not stay on the sidelines. The most widely used PDF reader has for years maintained its own program of approved roots for document signing, and over time it also incorporated the European list. This greatly improves the experience of European users, but it leaves an asymmetry: a perfectly valid signature issued by a trust service provider accredited in a country that is not on any of those lists may appear as not verified on millions of computers.

## Latin America and regional recognition

Several Latin American countries publish official lists of their accredited trust service providers, and some already do so in the standardized European format, which allows validators to process them automatically. The Latin American and Caribbean Network for Electronic Government (Red GEALC) has promoted work to harmonize those lists and facilitate mutual recognition, and agreements with the same goal have been signed within Mercosur. The path toward automatic cross border recognition, like the European one, is under way but not complete.

For a platform operating in several countries, this means that trusting the operating system's roots is not enough. It needs to read and verify the official lists of each jurisdiction in which it operates, keep them up to date and apply, for each signature, the list that corresponds to the country of the act.

## A list is also signed, and also ages

A trusted list is a document signed by the authority that publishes it, and as such it must be validated. Its signature uses certificates that in turn expire and are renewed; the European List of Trusted Lists indicates which certificates are valid for signing each national list. In addition, the lists are updated frequently and carry a scheduled date for the next publication. A rigorous validator verifies the signature of each list, rejects those that have expired and keeps the version used for each validation, so that the result can be reproduced years later.

### KEY POINTS

Trusted lists shift the decision about whom to trust from software vendors to public authorities, in a signed, machine readable format. They are the foundation of cross border recognition of qualified signatures and an indispensable part of any rigorous validation.



# Thirty years in a box.

Preserving a signed document is not the same as storing it. This chapter describes what it takes for a signature to remain provable when no one remembers any longer how it was made.

Many organizations must keep documents for very long periods: mortgage loan contracts, for the entire life of the loan and a few years beyond; medical records, for decades; certain administrative acts, permanently. Throughout that time, the signature must remain provable. The previous chapters showed the pieces that make this possible. This one arranges them into a strategy.

## Three layers of preservation

### Preserving the bits

The first requirement is that the file exists and does not change. This calls for copies in different locations, periodic integrity checks using fingerprints, and migration to new media as the old ones age. It is the classic discipline of digital preservation, and signatures make it more demanding: a single altered bit invalidates the signature, even if the document still looks the same. Storage that prevents an object from being modified or deleted for a defined period, known as write once (WORM) storage, is a common tool in this layer.

### Preserving readability

The second requirement is that the file can be read. That is why open, standardized formats are preferred, such as PDF/A for documents and the European formats for signatures, which have multiple independent implementations.

## Preserving the evidence

The third requirement, and the one specific to signatures, is that the evidence remains provable. This requires each signature to reach at least level LT, with its validation material embedded, and to be protected by archive timestamps that are renewed before the algorithms supporting them weaken.

## Renewing archive timestamps

Renewal is the heart of long term preservation. It works as follows: periodically, or when the authorities announce that an algorithm or key length will no longer be considered secure, a new archive timestamp is applied that covers everything already present, including the previous timestamps, using stronger algorithms and keys. Before it is applied, the validation material for the previous timestamp is embedded, so that this timestamp can also be verified in the future.

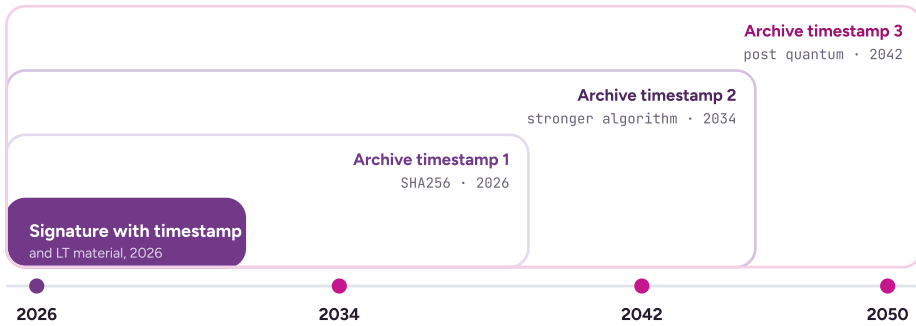


**FIGURE 8** Renewing archive timestamps. Each new timestamp, with stronger algorithms, covers the signature and every earlier timestamp.

The result is a chain of timestamps in which each link protects the previous ones. Even if the algorithm of the first timestamp is broken, the second timestamp proves that everything it covered existed before that happened, and therefore could not have been fabricated using the attack. The chain can be extended indefinitely, as long as it is renewed in time.

## Evidence records

Renewing timestamps document by document is costly when there are millions of documents. To solve this, the Internet standards body published in 2007 the specification for Evidence Record Syntax records, known as RFC 4998. The idea is to group the fingerprints of many documents into a tree structure, so that a single timestamp on the root of the tree protects them all. Each document keeps only the path that connects it to the root, a small proof that makes it possible to show that it was part of the stamped set. Renewal is performed on the tree, not on each document.

This technique, based on the trees that Ralph Merkle proposed in the late 1970s, is the same one used today by public certificate logs and many evidence systems. We will return to it in the next chapter.

## The qualified preservation service

The eIDAS Regulation defines a qualified preservation service for qualified electronic signatures and qualified “electronic seals”, provided by a supervised third party that undertakes to extend the trustworthiness of signatures beyond their technological validity period. The European technical standards describe different models: the service may store the documents, store only the evidence, or augment the signatures that the client continues to store. For an organization, contracting such a service transfers the technical responsibility for renewal to an audited specialist.

## A written policy

None of these techniques is of any use without a policy that states which documents are preserved, for how long, at what level, where, who monitors algorithm obsolescence and who decides when to renew. That policy must be reviewed periodically, because the environment changes: new standards appear, new weaknesses are announced, providers change.

#### CHECKLIST

- ✓ Documents are converted to PDF/A before the first signature.
- ✓ Each signature reaches at least level LT, and the closed document carries an archive timestamp.
- ✓ Copies are kept in at least two locations and their integrity is checked periodically.
- ✓ Someone monitors announcements about algorithms and is responsible for renewing timestamps in time.
- ✓ The trusted lists and other information used to validate the documents are preserved alongside them.
- ✓ The preservation policy is written, approved and reviewed.

# The memory of the process.

The signature proves that someone signed a given content. The evidence record proves everything else: how the document arrived, who saw it, how they identified themselves, what they accepted. Designing it well is a craft with its own rules.

In the first volume we saw that signing workflow platforms changed the unit of analysis: from the signed document to the signed process. If what matters is the process, then proof of the process is as important as the signature itself. That proof is the evidence record: the ordered sequence of events that the platform logs from the moment the submission is prepared until it is closed.

For a simple electronic signature, the record is often the main proof. For a qualified signature, it complements the signature with information the signature does not contain: which version the signer saw before signing, what consents they gave, what data they entered, from which device they acted.

## What to record

A good record answers, for each participant, the questions a judge or an auditor would ask. Common events include:

- the creation and dispatch of the submission, with the list of participants and their roles;
- each notification sent and through which channel;
- each access to the document, with the date, the time and technical data about the access;
- each identification step, with the method used and its result;

- the acceptance of the terms and of any consent, with the exact text that was accepted;
- each field completed and its value;
- each signature, with the method, the provider and the level achieved;
- rejections, delegations and cancellations, with their reasons;
- the closing of the process and the issuance of the certificate of completion.

Each event must identify its author, whether a person or a system, and carry a time mark taken from a synchronized clock. Events produced by external services, such as an identity provider or a signature provider, should include the service's response or a verifiable reference to it.

## Identification and authorization, kept separate

A subtle and very important distinction: identifying oneself and authorizing a signature are different acts. A person may identify themselves when opening the document through an identity provider and, minutes later, authorize their signature with a code sent to their phone or with the PIN of their card. If the record merges both acts into a single event, it loses the ability to show that the person who signed is the same person who identified themselves, and when each thing happened.

## How to protect the record

A record is only worth something if it can be shown that no one altered it. And the most serious threat comes not from outside but from inside: from whoever operates the platform, from an administrator with access to the database, from a software error. That is why protection techniques aim to ensure that not even the operator can modify the record without leaving a trace.

### Chaining

The first technique is to chain the events: each one includes the hash of the previous one, like the links of a chain. Modifying or deleting an event changes its hash and breaks every subsequent link. Verifying the chain is as simple as recalculating the hashes from the beginning.

## Database constraints

The second is to prevent, within the storage itself, events from being modified or deleted once written. Modern databases make it possible to define tables that only accept additions, and isolation rules that prevent one organization from seeing or touching the events of another.

## External anchoring

Chaining has a limit: whoever controls the system could recalculate the entire chain after a modification. The defense against this is to periodically publish the hash of the last link, or the root of a tree that summarizes many events, in a place the operator does not control: a timestamp from a third party, immutable storage belonging to another organization, or even a public medium. From that moment on, rewriting the earlier history would also require forging the anchor, which is no longer within the operator's reach.

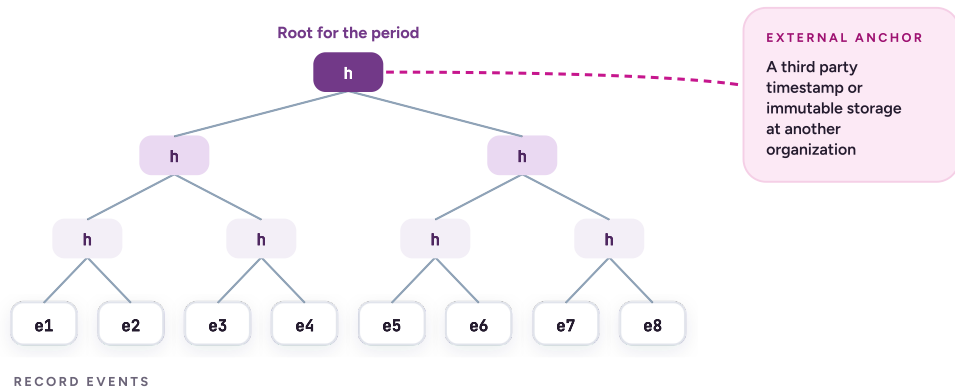


**FIGURE 9** Anchoring an evidence record. Event hashes are combined in pairs up to a root, and the root is published outside the operator's control.

## Privacy and minimization

An evidence record contains personal data: names, email addresses, identity document numbers, network addresses, sometimes biometric data. Data protection laws require that only the necessary data be collected, that they be protected and that they be kept only for as long as justified. There is a real tension between the inclination of evidence to record everything and the principle of minimization. Good practice resolves that

tension by recording what is necessary for proof, storing hashes instead of data when the hash is sufficient, and defining retention periods consistent with those of the document.

## The certificate of completion

The full record is usually long and technical. That is why platforms produce a readable summary, the certificate of completion, which accompanies the document and presents, for each participant, the main facts: when they received the document, how they identified themselves, when they signed. For that summary to have value, it must be signed by the platform or carry its platform seal and be consistent with the full record, which must be exportable in a verifiable format.

### KEY POINT

A sound evidence record logs every relevant fact with its author and its time, separates identification from authorization, is protected through chaining and through external anchors the operator does not control, and respects the privacy of the people who appear in it.

# When the signature reaches a court.

Everything that came before exists for a moment that, ideally, never arrives: someone denies having signed. This chapter describes how a disputed signature is examined and what is worth having ready.

The vast majority of signed documents are never disputed. But the value of a signature is measured in the exceptional case: the debtor who claims never to have signed the loan, the employee who denies having accepted a policy, the company that maintains the contract was altered. In those cases, the electronic signature has to convince someone who is not a technical expert, and do so with evidence that person can assess.

## What is in dispute

Disputes over electronic signatures usually revolve around four questions.

- 1 Was it this person?** This is the question of identity and authorship. It is answered with the evidence of identification and authorization: how it was verified who they were, which factor activated the signature, whether that factor was under their control.
- 2 Is this the document?** This is the question of integrity. It is answered with cryptographic validation, which shows whether the content changed after signing.
- 3 When?** This is the question of timing. It is answered with timestamps and, failing those, with the event log.

- 4 **Did they know what they were signing?** This is the question of intent. It is answered with the evidence of the process: what they were shown, what they accepted, how long the document was in view, whether they could download it.

## The role of the levels

As we saw in the first volume, in many jurisdictions the qualified signature or its equivalent enjoys presumptions that shift the burden of proof to whoever challenges it. That does not make it indisputable: whoever denies it may show, for example, that their card was stolen or that they were deceived into activating the signature. But it does change the dynamics of the proceedings. With a simple signature, by contrast, whoever relies on it must prove its authenticity, and in that effort the evidence record becomes decisive.

## The forensic expert

In a complex dispute, a digital forensic expert is usually involved. Their task is to validate the signatures with independent tools, examine the evidence record, verify its integrity and explain to the court, in understandable language, what these show and what they do not. A forensic expert works far better when the platform can provide:

- ✓ the original signed document, with all its revisions and individual signatures;
- ✓ the complete evidence record, exported in a format that can be verified without access to the platform;
- ✓ the technical description of how the record is protected and how its chain is verified;
- ✓ the external anchors that make it possible to confirm the record was not rewritten;
- ✓ the responses from external identity, signature, timestamp and revocation services;
- ✓ documentation of the configuration in effect at the time of the events.

## What experience teaches

Without claiming to summarize the case law of any country, some lessons recur. Courts tend to give more weight to coherent and complete process evidence than to generic claims about the security of a system. A cryptographically flawless signature

can lose strength if the preceding identification process was weak. And a simple signature backed by a sound record, with reasonable identification and explicit acceptance, can be fully effective.

A less technical lesson also recurs: clarity. An understandable certificate of completion, a validator that explains what it checked, a platform that can describe in a few pages how its evidence works: all of this makes the task easier for whoever has to decide.

#### WHAT TO REQUIRE OF A SOLUTION

Signatures in standardized formats that can be validated with independent tools. LT or LTA levels for long term documents. An evidence record that is chained and anchored outside the operator's control, exportable and verifiable. Identification and authorization recorded separately. A validator that states what it checked and what it did not. And the ability to deliver all of this, years later, to a forensic expert who has never seen the platform.

A signature is not proven with a green check mark on a screen. It is proven with a case file that someone outside the system can review and understand.



# Glossary

The main terms in this volume, in the sense in which they are used here. The basic concepts are in the glossary of Volume 1.

## **Archive timestamp renewal**

The periodic application of a new timestamp, with stronger algorithms, over a signature and all its previous proofs.

## **ASiC**

A standardized container that gathers one or more documents with their CAdES or XAdES signatures into a single file.

## **B, T, LT and LTA levels**

The technical levels of European advanced signatures: basic, with time, long term, and long term with archive.

## **CAdES**

An advanced signature format based on the CMS syntax, binary and applicable to any type of file.

## **Certificate of completion**

A readable summary of the evidence record that accompanies the signed document.

## **Document certification (DocMDP)**

A signature on a PDF that declares which subsequent changes are permitted: none, filling in forms and signing, or also adding annotations.

## **Document timestamp**

A special signature by a timestamping authority that covers an entire PDF up to that point.

**Enveloped, enveloping and detached signatures**

The three ways of relating signature and document: the signature inside the document, the document inside the signature, or both in separate files.

**Evidence record (RFC 4998)**

A structure that protects many documents with a single timestamp over the root of a hash tree, and that is renewed periodically.

**External anchor**

The publication of the hash of a record in a place its operator does not control, to prevent the earlier history from being rewritten without detection.

**Field locking (FieldMDP)**

A PDF mechanism that allows a signature to prevent certain form fields from being modified.

**Hash tree (Merkle tree)**

A structure in which the hashes of many pieces of data are combined in pairs up to a single root, so that any piece of data can be proven to belong to the set with a short path.

**Incremental update**

A PDF mechanism by which changes are appended to the end of the file without rewriting what came before, forming successive revisions.

**Indeterminate result**

The outcome of a validation in which the available information does not make it possible to decide whether the signature is valid or not.

**JAdES**

An advanced signature format based on JSON signatures, designed for messages between systems and for credentials.

**List of lists**

A list published by the European Commission that indicates where the trusted list of each Member State is located and with which certificate it is signed.

**PAdES**

An advanced signature format for PDF documents.

**PDF/A**

The ISO 19005 family of standards, which defines subsets of PDF suitable for long term archiving.

**Qualified preservation**

A trust service, provided for in eIDAS, that extends the reliability of qualified signatures and qualified “electronic seals” beyond their technological validity.

**Reference time**

The moment at which the validity of a signature is assessed, normally the one set by its timestamp.

**Validation material**

Chain certificates and revocation responses that make it possible to validate a signature without querying external services.

**Validation policy**

A set of rules that defines which anchors, algorithms, certificates and proofs a validator requires.

**XAdES**

An advanced signature format based on the XML signature, used in structured documents.



## REFERENCE

# Further reading

The laws, papers and standards cited in this volume, grouped by topic.

## History and foundations

Haber, S. and Stornetta, W. S. Paper on the timestamping of digital documents. *Journal of Cryptology*, vol. 3, no. 2, 1991.

Merkle, R. Work on hash trees and signatures, Stanford University, 1979.

## Formats and levels

ETSI. EN 319 122, CAdES. EN 319 132, XAdES. EN 319 142, PAdES. TS 119 182, JAdES. EN 319 162, ASiC.

ISO. ISO 32000, Portable Document Format (PDF). ISO 19005, PDF/A.

## Timestamping

IETF. RFC 3161, Time Stamp Protocol, 2001.

ETSI. EN 319 421 and EN 319 422, requirements and profiles for timestamping authorities.

## Validation and trust

ETSI. EN 319 102, procedures for signature creation and validation. TS 119 612, trusted lists. TS 119 441, validation services.

European Commission. Open source library for digital signature services (DSS) and the list of trusted lists.

European Union. Regulation (EU) 910/2014 (eIDAS), Articles 32, 33, 34, 41 and 42, and Regulation (EU) 2024/1183.

## Preservation and evidence

IETF. RFC 4998, Evidence Record Syntax, 2007. RFC 6283, its XML version, 2011. RFC 6962, public certificate logs, 2013.

ETSI. TS 119 511 and TS 119 512, signature preservation services.

**Rúbrica.**

NOTEBOOKS ON SIGNATURE, IDENTITY AND DIGITAL TRUST

## **Signing takes an instant. Proving that something was signed can take decades.**

This second volume explains what happens to a signature after signing: the formats and levels that prepare it to endure, timestamps, validation, trusted lists, long term preservation, the evidence record of the process, and what happens when a signature is disputed before a court.

### **Volume 1 · From Rúbrica to Digital Signature**

History, foundations, devices, doctrines, uses and platforms.

### **Volume 2 · The Proof That Endures**

Formats, levels, timestamps, validation and preservation.

### **Volume 3 · Orchestrating the Signature**

Processes, identity, integration, operation at scale and the future.