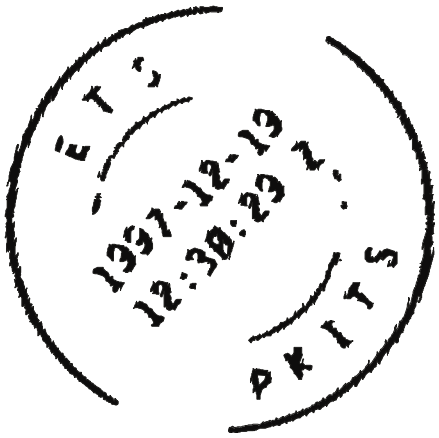


PKITS
Public Key Infrastructure with Time-Stamping Authority

ETS PROJECT: 23.192

Deliverable D3
Architecture of Time-Stamping Service and Scenarios of Use: Services and Features

Produced by: ***FNMT***
Date of issue: 30*th July 1998*
Revision Number: ***30***

TABLE OF CONTENTS

TABLE OF CONTENTS (EXPANDED)

GLOSSARY OF TERMS

Specifications:	
SHALL	Essential requirement. A requirement must be fulfilled or a feature implemented wherever this term occurs. The designer is requested, however, to indicate if one or more “shall requirements” would increase the cost or time unreasonably in relation to the total cost or design cost, in which case the specification may have to be revised.
SHOULD	Important requirement. Shall be implemented without or with minimum extra cost. Valid reasons in particular circumstances may allow ignoring such requirements.
MAY	Optional requirement. From case to case, it should be decided whether implementing it or not, in any case without exceeding the budget planned for the related activity.

ACRONYMS

Technical:	
ASN.1	Abstract Syntax Notation, One
CA	Certification Authority
CPS	Certification Practise Statement
CRL	Certificate Revocation List
DS	Digital Signature
EA	Escrow Authority
EDI	Electronic Data Interchange
ETS	European Trusted Systems
IETF	Internet Engineering Task Force
MTBF	Mean Time Between Fails
MTTF	Mean Time to Fail
NA	Notary Authority
NPS	Notarisation Practise Statement
NTP	Network Time Protocol
PKCS	Public Key Cryptography Standard
PKI	Public Key Infrastructure
RA	Registration Authority
RFC	Request For Comments
STS	Secure Time Source
TDS	Time Data Service
TS	Time-Stamping
TSA	Time-Stamping Authority
TSPS	Time-Stamping Practice Statement
TSS	Time-Stamping Service
TTP	Trusted Third Party

1 EXECUTIVE SUMMARY

This is the first technical deliverable of the PKITS project. Its aim is to establish a working architecture to provide time-stamping services. The elements of service identified in this document shall be used during the rest of the project to analyse the specific requirements of three business situations: unstructured documents, EDI messages, and multimedia information.

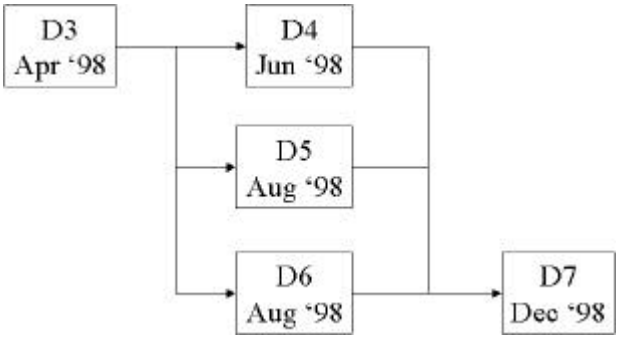
Section 3 defines the time-stamping service, and describes three protocols to provide it: a simple protocol, a linking protocol, and a distributed protocol. Section 4 places the time-stamping service in the context of PKI authorities. Section 5 describes meaningful scenarios of use, dealing with the public and private sectors. Section 6 performs a risk analysis, covering foreseeable threats and their impact, in order to propose preventing measures and curative actions that are checked against the proposed protocols in section 3. Section 7 covers some complementary aspects: levels of service, TSA liability, and service economics.

Appendixes collect material to be used either in this deliverable to specify the service or in the following deliverables to particularise for specific needs that depend on the nature of the data to deal with. Appendix A covers generic elements of cryptography; appendix B, the international standards either approved or in progress; appendix C surveys literature and patents related to time-stamping services, and appendix D presents the available sources of secure time, their availability and quality.

1.1 PROJECT ROADMAP

PKITS structures its technical contributions into a collection of deliverables:

D3. Architecture of Time-Stamping Service and Scenarios of Use: Services and Features
D4. Time-Stamping Service Functional Specification and Protocols for Unstructured Data
D5. TSS Functional Specification and Protocols for EDI Messages and Interchanges
D6. Time-Stamping Service Functional Specification and Protocols for Multimedia Information
D7. Testbed Description and Lessons Learned



D3 studies the definition of the service, identifies applicable protocols, and establishes a working framework. D4 focuses on unstructured documents, that is those documents whose internal structure is unknown to the TSA, and it cannot benefit from any knowledge, nor deal with separate fields in any sensible manner. D5 and D6 consider those cases where there is a knowledge of the syntactic structure of the documents: D5 studies the case of EDI messages, that are strictly formalised, and where there are fields specifically designed for holding time-stamping information; D6 studies multimedia information where there is knowledge of the audio and video stream structure. D5 and D6 propose mechanisms for embedding time-stamping information into already standardised information structures. D4, dealing with unstructured data, proposes an appendix to wrap time-stamping information to raw data. Lastly, a testbed will be used to provide a proof of concept, and start acquiring hands-on experience as a preliminary step towards interaction with time servers, and actual service provision. The lessons derived from this short experience are reported in D7.

2 REFERENCES

[Adams98a] Adams, Cain, Pinkas, Zuccherato, "Internet Public-Key Infrastructure, Part V: Time-Stamp Protocols", PKIX Working Group *Draft*, March 1998. ftp://ftp.ietf.org/internet-drafts/draft-adams-time-stamp-01.txt

[Adams98b] Adams, Zuccherato, "Internet Public-Key Infrastructure: Notary Protocols", PKIX Working Group *Draft*, February 1998. ftp://ftp.ietf.org/internet-drafts/draft-adams-notary-01.txt

[All74] Allan, D.W., J.H. Shoaf and D. Halford, "Statistics of time and frequency data analysis". In: Blair, B.E. (Ed.). Time and Frequency Theory and Fundamentals. National Bureau of Standards Monograph 140, U.S. Department of Commerce, 1974, 151-204.

[BaHaSt92] D. Bayer, S. Haber and W.S. Stornetta, "Improving the Efficiency and Reliability of Digital Time-Stamping", Sequencies'91: Methods in Communication, Security, and Computer Science, Springer-Verlag, 1992, pp. 329-334.

[BeGoGo94] M. Bellare, O. Goldreich, S. Goldwasser, "Incremental Cryptography: The Case of Hashing and Signing", Advances in Cryptology – Crypto 94 Proceedings, LNCS Vol. 839, Springer-Verlag, ed. by Y. Desmedt, 1994.

[BeMa92] J. Benaloh, M. de Mare, "Efficient Broadcast Time-Stamping (Extended Abstract)", Clarkson University Department of Mathematics and Computer Science Technical Report number TR-MCS-92-1. April 1992.

[BeMa93] J. Benaloh, M. de Mare, "One-Way Accumulators: A Decentralized Alternative to Digital Signatures (Extended Abstract)", Proceedings of EuroCrypt `93. Lofthus, Norway. May 1993. ed. by T. Heleseth.

[DAR81] Defense Advanced Research Projects Agency. "Internet Control Message Protocol". DARPA Network Working Group Report RFC-792, USC Information Sciences Institute, September 1981.

[DEC89] Digital Equipment Corporation, "Digital Time Service Functional Specification Version T.1.0.5", 1989.

[EC98] European Commission, DG XIII C/E, Telematics Application Programme, Project RE 1027 GNSS, "Global Navigation Satellite Systems Support", 1998.

[FIPS186] National Institute of Standards and Technology, "Digital Signature Standard", U.S. Department of Commerce, May 1994. http://www.nist.gov:80/itl/div897/pubs/fip186.htm

[G.803] ITU-T Recommendation G.803 (6/97), "Architectures of transport networks based on the synchronous digital hierarchy (SDH)" http://www.itu.int/itudoc/itu-t/rec/g/g800up/g803_23488.html

http://www.itu.int/itudoc/itu-t/rec/g/g800up/g813_36313.html

[G.825] ITU-T Recommendation G.825 (3/93), "The control of jitter and wander within digital networks which are based on the synchronous digital hierarchy (SDH)"
http://www.itu.int/itudoc/itu-t/rec/g/g800up/g825_23272.html

[HaKaSt95] S.Haber, B. Kaliski, W. S. Stornetta, "How do Digital Time-Stamps Support Digital Signatures?", RSA Laboratories' Cryptobytes Newsletter, Volume 1, Number 3, Autumn 1995.
http://www.rsa.com/rsalabs/pubs/cryptobytes/crypto1n3.pdf

[HaSt91a] S. Haber and W.S. Stornetta, "How to Time-Stamp a Digital Document", Advances in Cryptology - Crypto'90 Proceedings, Springer-Verlag, 1991, pp. 437-455.

[HaSt91b] S. Haber and W.S. Stornetta, "How to Time-Stamp a Digital Document", Journal of Cryptology, v.3 n.2, 1991, pp. 99-112.

[HaSt92a] S. Haber and W.S. Stornetta, "Digital Document Time-Stamping with Catenate Certificate", U.S. Patent #5,136,646, 4 Aug. 1992.

[HaSt92b] S. Haber and W.S. Stornetta, "Method for Secure Time-Stamping of Digital Documents", U.S. Patent #5,136,647, 4 Aug. 1992.

[HaSt94] S. Haber and W.S. Stornetta, "Method of Extending the Validity of a Cryptographic Certificate", U.S. Patent #5,373,561, 13 Dec. 1994.

[HaSt95] S. Haber and W.S. Stornetta, "Method for Secure Time-Stamping of Digital Documents", U.S. Patent #Re 34,954, 30 May 1995.

[ISO 10181] ISO/IEC 10181-4:1997, Information technology -- Open Systems Interconnection – "Security frameworks for open systems"
http://www.iso.ch/cate/d23615.html

[ISO 13335] ISO/IEC TR 13335-1:1996, Information technology – Security techniques – "Guidelines for the management of IT security (GMITS)"
http://www.iso.ch/cate/d21733.html

[ISO 14516] ISO/IEC WD 14516, Information technology – Security techniques – "Guidelines on the use and managament of trusted third party services" (N1855).

[ISO 15945] ISO/IEC WD 15945, Information technology – Security techniques – "Specification of trusted third party services to support the application of

[ITU93] ITU-T Recommendation G.825 (3/93), "The control of jitter and wander within digital networks which are based on the synchronous digital hierarchy (SDH)"

[ITU97] ITU-T Recommendation G.803 (6/97), “Architectures of transport networks based on the synchronous digital hierarchy (SDH).”

[KAP96] Kaplan, Elliot D. (ed). (1996), “Understanding GPS: Principles and Applications”, Boston, Artech House Publishers.

[MIL85] Mills, D.L. “Network Time Protocol (NTP)”. DARPA Network Working Group Report RFC-958, M/A-COM Linkabit, September 1985.

[MIL89] Mills, D.L. (09/89), “Network Time Protocol (version 2) - specification and implementation”. DARPA Network Working Group Report RFC-1119, University of Delaware.

[MIL92] Mills, D.L. (03/92), “Network Time Protocol (version 3) - Specification, Implementation and Analysis”. Network Working Group Report RFC-1119, University of Delaware.

[magerit] Spanish Ministry for Public Administrations (MAP), 'MAGERIT Versión 1.0 – Metodología de Análisis y Gestión de Riesgos de los Sistemas de

[PKCS #1] RSA Laboratories, “PKCS #1 - RSA Encryption Standard”, version 1.5, Nov. 1993.
http://www.rsa.com/rsalabs/pubs/PKCS

[PKCS #10] RSA Laboratories, “PKCS #10 – Certification Request Syntax Standard”, version 1.0, Nov. 1993.
http://www.rsa.com/rsalabs/pubs/PKCS

[PKCS #12] RSA Laboratories, “PKCS #12 – Public Key User Information Syntax Standard”, version 1.0, Apr. 1995.
http://www.rsa.com/rsalabs/pubs/PKCS

[PKCS #6] RSA Laboratories, “PKCS #6 – Extended-Certificate Syntax Standard”, version 1.5, Nov. 1993.
http://www.rsa.com/rsalabs/pubs/PKCS

[POS83a] Postel, J., DARPA Network Working Group Report RFC-868, “Time protocol”,USC Information Sciences Institute, May 1983.

[POS83b] Postel, J., DARPA Network Working Group Report RFC-867, “Daytime protocol”, USC Information Sciences Institute, May 1983.

[SU81] Su, Z., DARPA Network Working Group Report RFC-781, “A specification of the Internet protocol (IP) timestamp option”, SRI International, May 1981.

[PKCS #7] RSA Laboratories, “PKCS #7 – Cryptographic Message Syntax Standard”, version 1.5, Nov. 1993. http://www.rsa.com/rsalabs/pubs/PKCS

[PKIX] PKIX Working Group, “Internet Public Key Infrastructure - Part IX.509

http://www.ietf.org/html.charters/pkix-charter.html

[Ray98] Ray, J.R., “The IGP/BIPM time transfer project”, in 1998 IGS Analysis Center Workshop Proceedings, European Space Operations Centre, Darmstadt, Germany, in press 1998.

[RFC 1305] D.L. Mills, “1305 Network Time Protocol (Version 3) Specification, Implementation”. March 1992. ftp://ds.internic.net/rfc/rfc1305.txt

[RFC 1319] B. Kaliski, "The MD2 Message-Digest Algorithm", Apr. 1992. ftp://ds.internic.net/rfc/rfc1319.txt

[RFC 1320] R. Rivest, "The MD4 Message-Digest Algorithm", Apr. 1992. ftp://ds.internic.net/rfc/rfc1320.txt

[RFC 1321] R. Rivest, "The MD5 Message-Digest Algorithm", Apr. 1992. ftp://ds.internic.net/rfc/rfc1321.txt

[RFC 1422] S. Kent, “Privacy Enhancement for Internet Electronic Mail: Part II: Certificate-Based Key Management”. February 1993. ftp://ds.internic.net/rfc/rfc1422.txt

[RFC 781] Su, Z. A specification of the Internet protocol (IP) timestamp option. DARPA Network Working Group Report RFC-781. SRI International, May 1981. ftp://ds.internic.net/rfc/rfc781.txt

[RFC 792] Defense Advanced Research Projects Agency. Internet Control Message Protocol. DARPA Network Working Group Report RFC-792, USC Information Sciences Institute, September 1981. ftp://ds.internic.net/rfc/rfc792.txt

[RFC 867] Postel, J. Daytime protocol. DARPA Network Working Group Report RFC-867, USC Information Sciences Institute, May 1983. ftp://ds.internic.net/rfc/rfc867.txt

[RFC 868] Postel, J. “Time protocol. DARPA Network Working Group Report, USC Information Sciences Institute, May 1983. ftp://ds.internic.net/rfc/rfc868.txt

[RIPE 95] RIPE Project. Ripe Integrity Primitives: Final Report on Race Integrity Primitives Evaluation; LNCS 1007, Springer-Verlag, 1995.

[Schneier96] B. Schneier, “Applied Cryptography”, 2nd ed. John Wiley & Sons, 1996.

[SPKI] Internet Engineering Task Force SPKI Working Group. SPKI Specifications. http://www.ietf.org/html.charters/spki-charter.html

[X.509 v3] ITU Recommendation X.509, “The Directory – Authentication Framework”, version 3, Geneve 1996. http://www.itu.ch/itudoc/itu-t/rec/x/x500up.html

3 TIME-STAMPING SERVICE

Generally speaking, non-repudiation services provide a PKI user with protection against another user later denying that some exchange took place. While these services do not prevent a user from repudiating another user's claim that something occurred or existed, they ensure the availability of either thoroughly irrefutable evidence, or an overwhelming amount of evidence to support the speedy resolution of any such disagreement.

Non-repudiation services are defined in [ISO 10181-4], and involve up to five phases:

1. service request
2. evidence generation
3. evidence transfer/storage
4. evidence verification
5. dispute resolution

An essential part of non-repudiation services is to provide evidence that a given data is in the given hands in a moment in time. The moment in time may be implicit by the sequence of exchanges in a complex protocol, or may be required to be explicit, that is referenced to absolute time, when the protocol is open, and the proof-of-existence may be needed much later, in a different protocol situation.

Non-repudiation always requires a trusted third party to which both parties agree, because no arbitrator could decide on the basis of evidences presented by the disputing parties themselves.

The need to establish relationships between a document and its time of generation, exchange, signature, etc, arises in many situations. The following list enumerates some examples:

- In intellectual property matters the verification of the date on which a person first described the substance of the invention is of paramount importance.
- In public-key infrastructures, it sometimes happens that an identity certificate (the connection between a user's identity and his/her public key) must be revoked (v. g. if the private key has been compromised, or cryptographic advances render the key length too short). It is necessary to be able to determine, at a later time, if a particular document was signed within the period of validity of the certificate.
- In legal matters, the determination of the applicable law is always performed after close examination of the dates of related documents and facts.

The increasingly widespread use of electronic documents (text, multimedia information, electronic commerce transactions, etc.) represents a serious threat to the viability of the document time-stamping process: the easy revision of digital documents, and the absence of marks proving that an electronic document has been revised, arise justified doubts on any statement regarding the document creation or modification date.

Some schemes have been proposed to perform the time-stamping of digital documents; these schemes transfer control of the time-stamping process from the author to an independent, uninterested trusted third party, thus removing from the author the ability to influence the time-stamping agent in the production of other than a truthful time stamp.

PKI authorities issue certificates, revocations, storage, registry, and large amounts of information that have a limited time frame of application. For these services, authorities do require a trustful time-stamping service. This subject is further analysed in the following sections.

A time-stamping service (TSS) may be provided in isolation of other services by a specialised time-stamping authority (TSA), or packed together with services provided by other authorities. Section 4 analyses how to merge authorities into meaningful clusters.

3.1 GENERAL SERVICE ARCHITECTURE

A TSA is not expected to work in isolation. It may rather be expected to find in actual practice that many TSA are deployed by public and private institutions to create a net of trust.

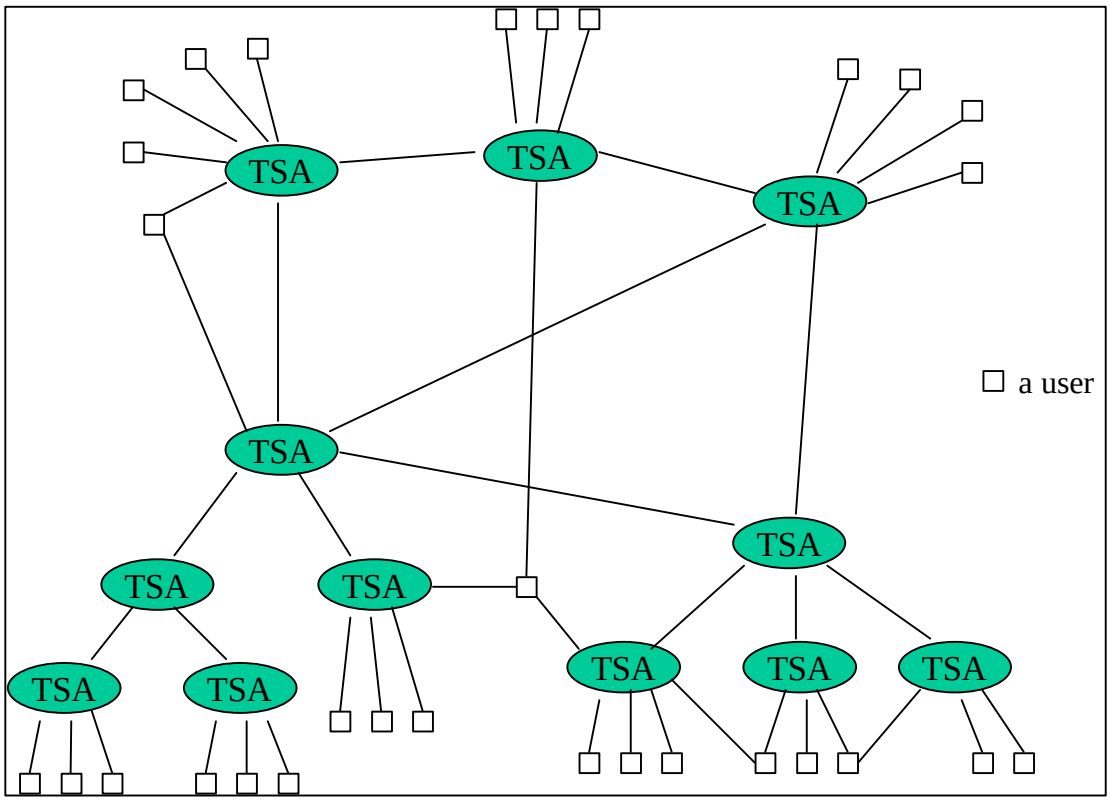
User groups are expected to use one or more TSA intensively for their internal businesses. Using more than one TSA for important data may improve service reliability. Users may expect an acceptable degree of synchronisation between time-stamps issued in parallel by two different TSA.

When documents from different user communities have to be compared, either for exchanging data or for resolving disputes, the corresponding TSA will be required to be synchronised within a given tolerance. Users may not be requested to foresee every possible scenario of use of their time-stamped data, and therefore have to rely on their usual TSA (or small collection of TSA) for a universally applicable service.

User communities may appear for several reasons, the boundaries between them being due to:

- geography
- network topology
- private networks (e.g. intranets)
- closed user groups (e.g. extranets)
- non-permanently connected nodes (e.g. autonomous systems)

For any of these reasons, the topology of a world-wide TSA provision service is likely to cluster service requesters around near TSA, and impose a requirement on each TSA to synchronise with others. The topology of TSA synchronisation is out of the scope of this document, and might be a hierarchical topology, a mess, or a mixture. If TSA proliferate, as may be expected when the private sector participates in the provision of these services, direct connection between all TSA in the world seems highly unlikely, and indirect (i.e. transitive) synchronisation is to be expected.



Synchronisation of stamps has to be addressed from two complementary points of view. The best effort is expected from TSA to use reliable clocks that are kept coherent within certain limits to be defined by the quality of the provided service. However, synchronisation will be also verified *a posteriori* to detect deviations, and allow for rapid remedy. Clock synchronisation is a common requisite for the protocols described in this section, and related protocols will be covered in Appendix D, Secure Time. Service synchronisation is protocol-specific and will be covered within the descriptions below.

Following this expected scenario, three protocols are described below:

Basic Protocol: a simple centralised service

Where the TSA commits itself to a trustful clock, and signs the association of time to data.

Linking Protocol: a centralised service enhanced by means of linking

Where the previous protocol is extended to provide a mechanism that links the documents time-stamped by the TSA, creating an unforgeable link between all the users of the service

Distributed Protocol: several time-stamping entities

Where several signers perform the service, making collusion difficult since all the parties would have to be corrupt.

The basic service may apply to communities where trust is achieved by other means. The linking protocol may apply to large user communities using one TSA, and also to a collection of TSA that use a common ancestor in a hierarchical synchronisation architecture. The distributed protocol may apply to user groups that share no authority, and to collections of TSA without a clear hierarchical topology.

3.2 DEFINITION OF SERVICE

In [HaSt92a, HaSt92b] Haber and Stornetta stated the following very general properties for a digital time-stamping protocol:

- The data itself must be time-stamped, without any regard to the physical medium on which it resides.
- It must be impossible to change a single bit of the document without the change being apparent.
- It must be impossible to time-stamp a document with a date and time different from the present one.

The first requirement is fundamental for dealing with electronic data.

The second requirement is highly restrictive; it protects against the smallest modification, but it may also detect differences that are not semantically relevant. In some scenarios, this requirement could be relaxed.

The third requirement affects to TSA operation procedures, depends on good clock synchronisation, but fails to foresee the open opportunity of any document to be time-stamped at any moment after its creation.

Time-stamping service proves the existence of an electronic document representation at a certain point in time. The time-stamping provider notarises that fact, and guarantees the correctness of the time label.

.

Time-stamping would be a trivial task if systems were trusted and reliable, meaning that:

- There is a unique universally accepted clock.
- Documents may not be back-dated.
- Time-stamps may not be lost.

When the actors may benefit from modifying facts covered by a time-stamp, trust may not be just supposed, but demonstrated either by construction or by verification. These considerations drive us to consider the following models:

- Basic Centralised: a trusted third party commits itself; trust is transitively derived from the authority.
- Linking Centralised: a trusted third party commits itself, and furthermore the protocol prevents malicious forgery of time-stamps; trust is derived transitively from the authority and enforced by construction.
- Distributed: an unforgeable random subset of certifying partners; trust is statistically derived from the accumulation of facts.

Time is an eluding concept that is implemented differently by different people and systems. Very generally speaking, the time is a monotonically increasing value agreed by the parties, such that when data are labelled with a time value, it is always possible to check, given two pieces of time-stamped data, which of them was stamped first, or if the stampings were simultaneous. There is a world-wide use of atomic clocks and other devices that satisfy the above mentioned property as a neutral label that may be trusted by whichever two parties, even without any previous bilateral agreement. Universal tokens may be replaced by private tokens that are trusted in closed environments.

The association between the data to be time-stamped and the time token may provide different services:

- The document existed before-than:

  - Proves that the time-stamped data were not created after the indicated time.
- The document exists after-than:
  - Proves that the time-stamped data were not destroyed before the indicated time.
- The document existed within a given time period.

Labelling data with a time token is the traditional way to get a *before* service. Incorporating the time token within the data is the traditional way to provide an *after* service. Merging both methods provides the *within* service.

As example situations, when a conventional notary signs a paper document it provides a *before* service; when the front page of a widely known newspaper is incorporated into a photograph, it provides an *after* service; when the previously mentioned photograph is registered by a notary, it provides the *within* service.

This document focuses on the first service: proving data existed before a given instant in time. The other services may be built out of the basic one, or require an electronic notary to support proof-of-possession (after service). Proving a document is available after a given time may be achieved by (1) getting a time-stamp on any data that shall be used as a reference, and (2) merging, i.e. hashing together, that token with the document to be time-stamped. If this merge is submitted to time-stamping, the within service is achieved.

Cryptographic algorithms may be used for the aforementioned associations in such a way that

- They are easy to verify.
- They are difficult to forge.

There is an issue with respect to which data are subject to time stamping. The data themselves need not to be known to the time-stamping agent, and any evidence of existence may be used to indirectly associate a time-stamp to the actual data. This characteristic will allow for:

- Using a hash of the data to preserve confidentiality of the original data.
- Using a hash of the data to reduce the volume of the messages.
- Using a hash of the data to reduce storage requirements.
- Time-stamp an already time-stamped document to provide further evidence (e.g. renewal).

It is important to understand that the use of a hash of the data instead of the data itself is sufficient for the purposes of the time-stamping service. Since the ability to calculate the hash of certain data implies the availability of that data (see Section 8.3 on cryptographic hash functions), time-stamping the hash of a digital document will prove that the document was available to the user before time-stamping it. Hashes of digital documents are frequently used in cryptography (notably in digital signatures) instead of the original documents.

Access to document contents for inspection is regarded as a notary service, out of the scope of a time-stamping authority.

3.3 ELEMENTS OF SERVICE

In the following sections we will study the elements of service for three time-stamping certificate generation schemes: a basic protocol, that time-stamps documents independently; a linking protocol, that links every time-stamped document in an unforgeable chain; and a distributed protocol, that does not require a central time-stamping authority.

All of these protocols relay on the use of hash functions, as explained above, and the hash value is time-stamped rather than the document itself. There is a wide collection of available hash functions, and more are expected to be discovered and standardised in the near future.

Therefore, the hash function used must be explicitly identified. Furthermore, users may submit more than one hash value in order to:

- to prevent weaknesses in known hash functions,
- to permit different TSA to prefer different hash functions to fulfil their policy requirements,
- and to introduce enough flexibility in the system to smoothly adapt to new hash functions.

For this purpose, let us introduce a bit of notation:

let P(x) be the non-empty list of pairs: $\{<H_i, H_i(x)>, \ldots\}$

where H_i are hash function identifiers, H_i (x) is the hash value of x.

P (M) will be used below to refer to the list of hash values that characterise the message M to be time-stamped.

The election of H_i is subject to service provision policies.

3.3.1 Basic Protocol

A Time-Stamping Authority (TSA) is introduced in the Public-Key Infrastructure (PKI) to centralise the issue of time-stamp certificates upon users' request. The role of this authority will be to produce, store, verify and renew time-stamping certificates. The TSA will be a trusted third party, with no particular interest in the time-stamped documents; therefore, the signature of the TSA in the time-stamps will prove their validity.

This basic protocol is roughly based on [Adams98a].

This protocol allows the time-stamping of any kind of digital information, and protects the confidentiality of the time-stamped data. Despite this, it has a serious drawback: there is no mechanism (except the TSA being a trusted third party) to minimise the possibility of collusion between the TSA and some user to produce a false time-stamping certificate or to falsely attribute a time-stamp to an interested party.

3.3.1.1 Sub-Systems

There are several possible embodiments compatible with this centralised time-stamping model; we will attempt to consider the present sub-systems in a general case, and consider their interrelationships:

- ***Certificate Authority (CA)***

This authority is an essential party in some public-key infrastructures; its role is to certificate the identities of the rest of the parties, and link their identities to their public keys. In non-hierarchical public-key infrastructures, there might be no CAs.

The following requirements shall be imposed on the CA regarding the time-stamping service:

BP1	The CA **shall** link the identity of the users and TSA to their public keys via appropriate certificates.

- ***Time-Stamping Authority (TSA)***

  The TSA will be responsible for the time-stamping service as such; it will stamp the time provided by the STS. Different levels of service may require the use of different STSs.

  The following requirements shall be imposed on the TSA:

BP2	The TSA **shall** select a set of "accepted" hashing algorithms that the users will

	use to produce the hashes of their documents when following the time-stamping protocol.
BP3	The TSA **shall** identify itself to its users via appropriate certificates.
BP4	The TSA **shall** use a secure digital signature scheme (and key size) to produce the time-stamping certificates.
BP5	The TSA **shall** select the Secure Time Source (STS) that will be used in the time-stamping protocol.
BP6	The TSA **should** reply to all time-stamping requests by the users, except when clearly motivated (v. g., when there are evidences of malicious attempts to attack the system).
BP7	The TSA **shall** assist in the verification of any time-stamping certificate provided by it.
BP8	The TSA **shall** publicise the quality parameters of its time-stamping service; in particular, it **shall** periodically publicise information regarding the accuracy of its time source and the distribution of its response times.
BP9	The TSA **shall** not disclose to unauthorised third parties the users of its time-stamping service.
BP10	The TSA **should** publicise among its users information on cryptographic advances that could affect the validity of their time-stamping certificates.
BP11	The TSA **shall** assist its users in the periodic renewal of their time-stamping certificates.
BP12	On cessation of its activities, the TSA **shall** transfer to another TSA all the information that could be needed to verify or renew the time-stamping certificates issued during its activity period. When selecting the TSA to which the certificates will be transferred, attention **shall** be given to the compatibility of the time-stamping policies and quality parameters.

- ***Secure-Time Source (STS)***
  The STS will provide a monotonically increasing value of time that will be synchronised with the rest of the STSs in the public-key infrastructure and with the other time provision services in common use (GPS clock, UTC time, etc.). Different levels of service may require the use of different STSs by a TSA.

  The following requirements shall be imposed on the STS:

BP13	The STS **shall** provide a monotonically increasing value of time that will be synchronised with the rest of standard time sources accepted internationally.

- ***Universe of users (UoU)***

The UoU is the set of people or electronic systems that will use the time-stamping service in the course of their activities. These users may be in possession of an identity certificate issued by the CA, and may use this certificate to prove identity to the TSA.

The following requirements shall be imposed on the UoU:

BP14	Each member of the UoU **should** be in possession of an identity certificate issued by the CA, and **should** use it to prove identity to the TSA.
BP15	Each member of the UoU **shall** use an accepted hashing algorithm when following the time-stamping protocol.
BP16	Each member of the UoU **shall** store the time-stamped documents in a safe place, since they will be needed for verification and certificate renewal.

3.3.1.2 Provision

Service provision will be performed at the user's request. When a user Alice wishes to obtain a time-stamp for a digital document M, she calculates one or more hash values of M, and sends the list P (M) to the TSA. This will produce a small amount of information (with size depending on the hash algorithms used, but usually a few bytes long). The TSA will append the date & time to the received hash list. Finally the whole receipt will be signed with the TSA secret key so that everybody will be able to validate the resulting time-stamping certificate.

Centralised Generation of TS Certificates

1. Alice sends her identity, A, a non-empty list of pairs of hash function identifiers, and the corresponding hash value of the message M to be time-stamped
   let P(x) be the list of pairs: {<H_i, $H_i(x)$>, …}
   A sends: <A, P (M)>

2. The TSA returns to Alice:
   $C = S_K$ (n, t, A, P (M))
   where:
   S_K denotes the digital signature by TSA.
   n is the time-stamp certificate serial number.
   t is the date/time provided by the STS.

3. Alice receives the certificate C, and checks that:
   - It is signed by the TSA.
   - It is the message she asked a time-stamp for.
   - The time is within reasonable limits of precision.

C is the time-stamp certificate.

The identity of the requester is required by the TSA, at least for recording, and optionally for further identification of the client (e.g. for billing).

At least one hash function H must be used. For higher security, more than one hash value may be submitted. The used hash function must be always explicit, and part of the signed response. Provision for a list permits the system to evolve smoothly as old hash functions become obsolete, and new ones are put in place.

3.3.1.3 Verification

Verification consists of checking that the TS certificate has been produced by a valid TSA.

Verification of time-stamping certificates

$_K$ (n, t, A, P (M)), a user Bob would have to:

1. Check that the hash value(s), P (M) on the certificate C corresponds to Alice's document, M.

2. Check that the digital signature on C is valid and corresponds to a TSA. This proves that the time-stamp was produced by the TSA.

If these two conditions were satisfied, Bob would regard Alice's time-stamp valid. Otherwise, he should reject Alice's time-stamp certificate.

Depending on the quality of the service provided, the TSA may store the issued certificates as evidence for later verification, and assist users in the verification process. Activity records further add to this quality of service in providing evidence.

A low quality service would simply delegate evidence storage to service users. This situation is extremely dangerous if the TSA signing key were compromised.

3.3.1.4 Synchronisation

The central TSA issuing time-stamp certificates **shall** synchronise itself with other TSA in the PKI universe in order to permit document data comparisons, and foresee the case of robbery of the private signing key.

Periodically, the TSA shall submit for time-stamping its own records of activity. If R is the record of activity of period Ti of the TSA, <TSA, P (R)> is submitted to another TSA to be time-stamped. This prevents the TSA to back- or forward-date documents. This protection is relevant for users to compare their time-stamp certificates with those issued by other authorities, and to protect certificates from TSA's key compromise.

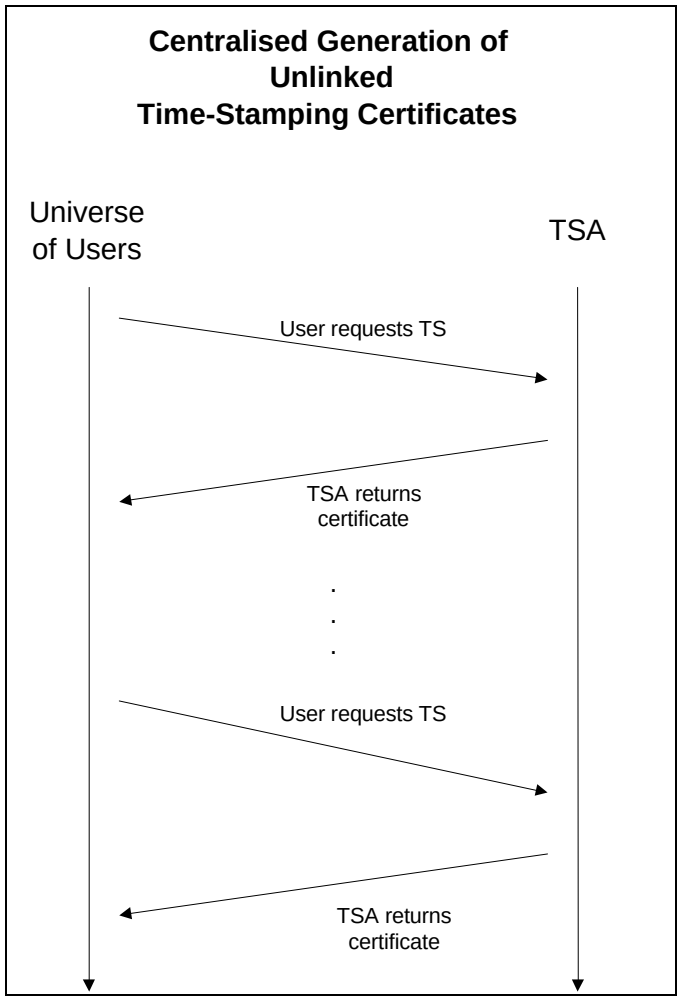
In order to extend trust between many authorities, and permit wide usability of time-stamp certificates, either a hierarchical or a mess of TSA may be foreseen. If the PKI is hierarchically oriented, an alternative is that there are TSA at each level of the hierarchy, and that at each level, the corresponding TSA time-stamps periodically the records of activity of subsidiary TSA.

If there is no such notion of hierarchy, TSA which users wish to interoperate shall time-stamp each other's records of activity. This cross certification may extend trust transitively between two not-directly connected TSA.

Granularity of synchronisation depends heavily on the length of the recording period between synchronisation points, and on the overlapping of periods. Quality of service may degrade very quickly when there are several synchronisation steps between two users using different TSA.

3.3.1.5 Activity Flow

The following diagram shows the roles played by each party, the information exchanges, and the order in which they are performed, in provision of service according to this protocol.



3.3.1.6 Variants

To reinforce the security of the time-stamping process, the following variants of the basic protocols can be considered.

3.3.1.6.1 Prevention of "Man-In-The-Middle" Attacks

There is a risk that the request is intercepted by another user, E, that could impersonate A and request a valid time stamp on the hash of a document not owned by E. There are at least two modifications to the protocol that prevent this impersonation:

1. The user provides enough identification within the document as to guarantee it is his, even if Eve succeeds to time-stamp A's document for his benefit. If the document contents is not enough, it is always possible to incorporate the identity of the user when producing the hash to be signed: P (A+M).

2. The TSA may provide secure channels for requesting TS. Such a secure channel shall authenticate both the user and the service provider, and guarantee integrity and confidentiality of contents.

These modifications to the protocol also prevent malicious (or erroneous) assignments by the TSA of the time-stamping certificate to a party different to the requester. The first variant provides this assurance by itself, while the second variant depends on the authentication characteristics of the secure channel.

3.3.2 Linking Protocol

This protocol was originally proposed by Haber & Stornetta in [HaSt91a, HaSt91b, HaSt92a, HaSt92b]. It adds some security guarantees to the simple centralised protocol.

In the centralised time-stamping scheme a time-stamping authority (TSA) is included within the public-key infrastructure; the role of the TSA will be to produce/verify/renew/etc. time-stamp certificates. This TSA will be a trusted third party, and the protocol will attempt to strengthen the confidence on this authority by minimising the possibility of colluding with the TSA to produce a false time-stamp certificate or a false attribution of a time-stamp to an interested party.

This protocol allows the time-stamping of any kind of digital information, and includes means to protect the confidentiality of the time-stamped data.

This protocol introduces more information in a certificate to involve more parties and disallow collusion between the user requester and the TSA.

3.3.2.1 Sub-Systems

There are several possible embodiments compatible with the centralised time-stamping model; we will attempt to consider the sub-systems present in a general case. We will enumerate the necessary subsystems and their interrelationships:

- ***Certificate Authority (CA)***

This authority is an essential party in the public-key infrastructure; its role is to certificate the identities of the rest of the parties, and link their identities to their public keys.
The following requirements shall be imposed on the CA regarding the time-stamping service:

LP1	The CA **shall** link the identity of the users and TSA to their public keys via appropriate certificates.

- ***Time-Stamping Authority (TSA)***
  The TSA will be responsible for the time-stamping service as such; it will stamp the time provided by the STA. Different levels of service may require the use of different STAs.
  The following requirements shall be imposed on the TSA:

LP2	The TSA **shall** select a set of "accepted" hashing algorithms that the users will use to produce the hashes of their documents when following the time-stamping protocol.
LP3	The TSA **shall** use a secure digital signature scheme (and key size) to produce the time-stamping certificates.
LP4	The TSA **shall** select the Secure Time Source (STS) that will be used in the time-stamping protocol.
LP5	The TSA **shall** reply to all time-stamping requests by the users, except when clearly motivated (v. g., when there are evidences of malicious attempts to attack the system).
LP6	The TSA **shall** assist in the verification of any time-stamping certificate provided by it.
LP7	The TSA **shall** establish a means to publicise the linking information.
LP8	The TSA **shall** publicise the quality parameters of its time-stamping service; in particular, it **shall** periodically publicise information regarding the accuracy of its time source and the distribution of its response times.
LP9	The TSA **shall** not disclose to unauthorised third parties the users of its time-stamping service.

LP10	The TSA **shall** publicise among its users information on cryptographic advances that could affect the validity of their time-stamping certificates.
LP11	The TSA **shall** assist its users in the periodic renewal of their time-stamping certificates.
LP12	On cessation of its activities, the TSA **shall** transfer to another TSA all the information that could be needed to verify or renew the time-stamping certificates issued during its activity period.

- ***Secure-Time Source (STS)***
  The STS will provide a monotonically increasing value of time that will be synchronised with the rest of the STSs in the public-key infrastructure and with the other time provision services in common use (GPS clock, UTC time, etc.).

  The following requirements shall be imposed on the TSA:

LP13	The STS **shall** provide a monotonically increasing value of time that will be synchronised with the rest of standard time sources accepted internationally.

- ***Universe of users (UoU)***

The UoU is the set of people or electronic systems that will use the time-stamping service in the course of their activities. These users will be in possession of an identity certificate issued by the CA, and will use this certificate to prove identity to the TSA.

The following requirements shall be imposed on the UoU:

LP14	Each member of the UoU **shall** be in possession of an identity certificate issued by the CA, and **shall** use it to prove identity to the TSA.
LP15	Each member of the UoU **shall** use an accepted hashing algorithm when following the time-stamping protocol.
LP16	Each member of the UoU **should** store the time-stamped documents in a safe place, since they will be needed for verification and certificate renewal.

3.3.2.2 Provision

Service provision will be performed at the user's request. When Alice wishes to obtain a time-stamp for a digital document, she calculates a hash value or a collection of them. This will produce a small, fixed amount of information that will be sent to the TSA for time-stamping. The TSA will decide if the hash algorithm(s) used is secure enough according to its service policies, and if so it will append the date & time to the received hash. Then the whole data will be signed with the TSA secret key so that everybody will be able to validate the time-stamping certificate.

To prevent the TSA from forging an invalid, backdated certificate, each time-stamped document is linked to the last and next time-stamped documents, so that an unmodifiable chronological chain is obtained.

Centralised Generation of Linked TS Certificates

In this description we assume that user Alice wishes to time-stamp data M, and that this request is the n^{th} request made to the TSA.

1. Alice sends her identity, A, a non-empty list of pairs of hash function identifiers, and the corresponding hash value of the message M to be time-stamped
   let P(x) be the list of pairs: {<H_i, $H_i(x)$>, …}
   A sends: <A, P (M)>

2. The TSA returns to Alice:
   $s = S_K(n, t_n, A, P(M), L_n)$
   where:
   S_K denotes the digital signature by TSA.
   n is the time-stamp certificate serial number.
   t_n is the date/time provided by the STS.

$L_n = (n, t_{n-1}, U_{n-1}, P(M_{n-1}), P(L_{n-1}))$ is the link to the previous certificate that was requested by U_{n-1} for M_{n-1}.

3. When the next request is issued, the TSA will send U_{n+1} to Alice, so that there is also a link in the future direction of time.

4. Alice receives the receipt s, and checks that:
   - It is digitally signed by the authority
   - It is the message she asked a time-stamp for.
   - The time is within reasonable limits of precision.

The pair $C = <s, U_{n+1}>$ is the time-stamp certificate.

The identity of the requester is required by the TSA, at least for recording, and optionally for further identification of the client (e.g. for billing).

At least one hash function H must be used. For higher security, more than one hash value may be submitted. The used hash function must be always explicit, and part of the signed response. Provision for a list permits the system to evolve smoothly as old hash functions become obsolete, and new ones are put in place.

The link to the previous certificate establishes a verifiable sequence of acts. Each link proves that the current certificate is issued after the previous one. Two links (backwards and forwards) proof that the certificate was issued after the previous one, and before the next one. Even if the time source were not reliable, the documents are strictly ordered.

Granularity of time depends on the time source used by the TSA. Granularity of verifiable trust depends on the frequency of requests arriving at the TSA to create the chain of trust.

3.3.2.3 Verification

When someone wishes to verify a TS certificate, there are several possible "security levels" depending on the importance of the TS in the specific situation. The simplest verification would be to check that the TS certificate has been produced by a valid TSA. A higher security level would be obtained by checking the k certificates around: the k previous certificates, and the k posterior ones. The higher the value of k, the greater the confidence in the system.

Verification of linked time-stamping certificates

To verify Alice's certificate $C = (s, U_{n+1})$, a user Bob would have to:

1. Check that the hash(es) contained in the certificate corresponds to Alice's document.

2. Check that the digital signature on the certificate is correct [$s = S_K(n, t_n, A, P(M), L_n)$]. This proves that the time-stamp was produced by the TSA.

3. Select $k \geq 1$, and ask users $\{U_{n-i}, ..., U_{n+i}$, for $i= 1..k\}$ for their certificates, and then check their correctness and the correctness of the links between them.

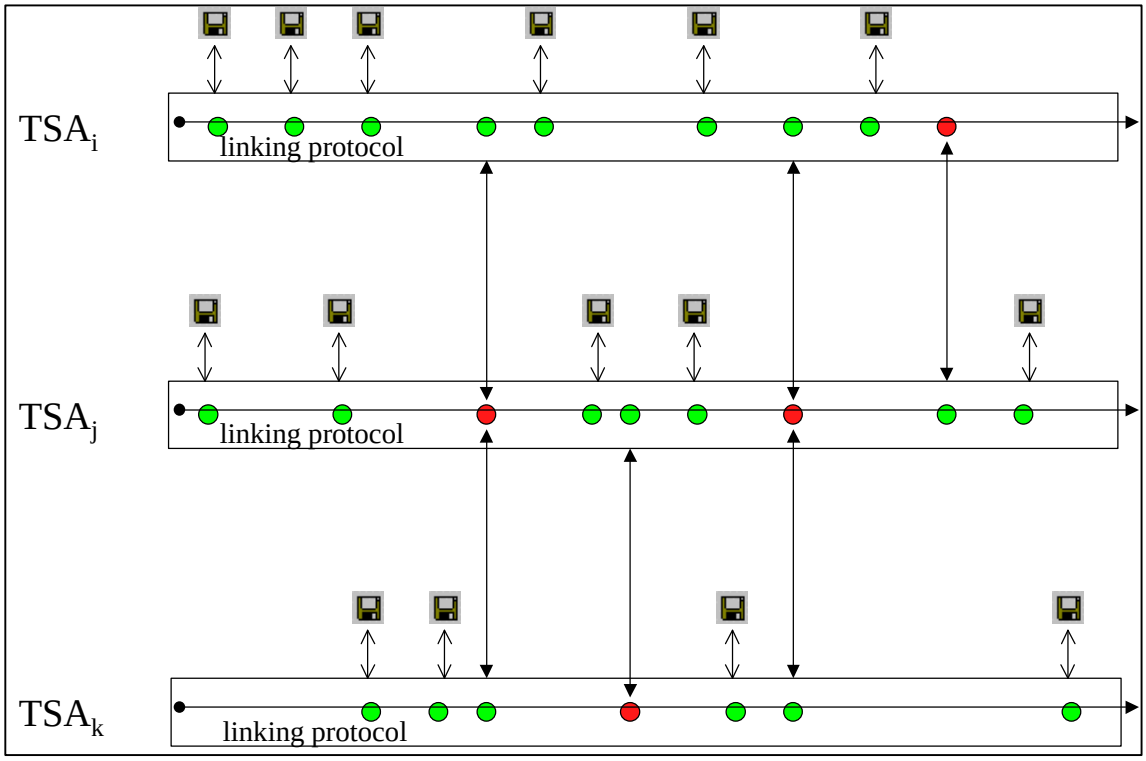
If these conditions meet, Bob would regard Alice's certificate ad valid; otherwise he would reject it.

Depending on the quality of the service provided, the TSA may store the issued certificates for later verification, and assist users in the verification process.

3.3.2.4 Synchronisation

The TSA may synchronise its accumulated hash to extend trust beyond its own universe of users.

To strengthen the system security, and for auditing purposes, the different TSA will periodically time-stamp each other's linking values. This administrative procedure permits the comparison of time-stamps issued by different TSA limiting the risk of clock deviations to the agreed interval between cross time-stamping.



As already described above, TSA may be organised hierarchically or rather live in a mess. Cross-linking may be performed in both cases: either the father TSA creates its own super-hash link recording daughters' link values, or TSA cross time-stamp each other.

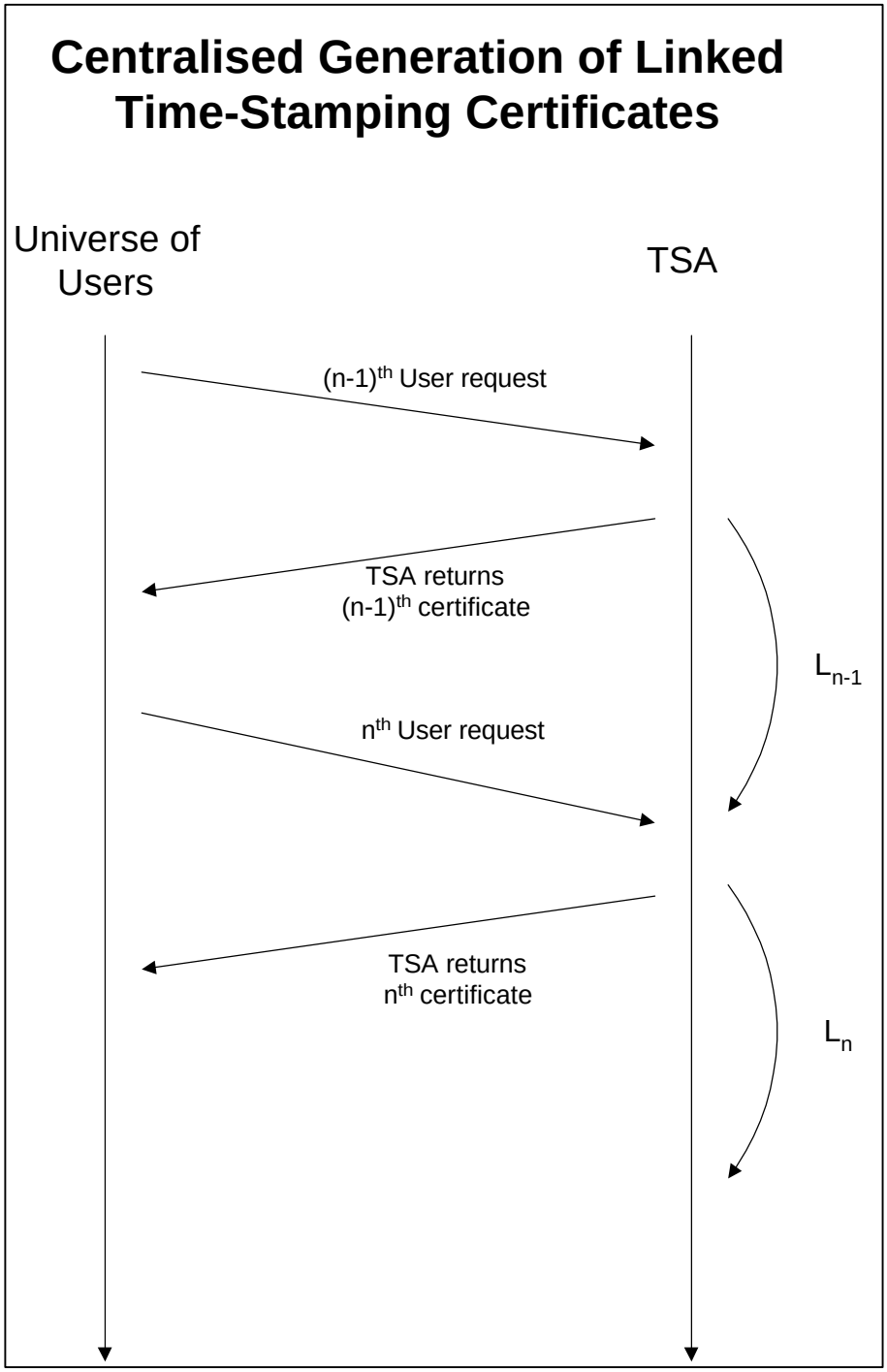
Granularity of synchronisation depends heavily on the length of the recording period between synchronisation points, and on the overlapping of periods. Quality of service may degrade very quickly when there are several synchronisation steps between two users using different TSA.

If several TSA share a common father TSA to synchronise their link chains, there may be a clustering effect if many of them do their requests simultaneously: the first and last ones get loose time frames. To benefit from the smaller granularity of time in those TSA that serve to

many users, linking value time-stamping should be mutual between fathers and daughters ina hierarchical architecture.

3.3.2.5 Activity Flow

The following diagram shows the roles played by each party, the information exchanges, and the order in which they are performed.



3.3.2.6 Variants

Several variations of the centralised linking protocol can be suggested, in order to further reinforce the security of the time-stamping process.

3.3.2.6.1 Use of a tree structure to compute the linking information

This improvement was introduced by Bayer, Haber and Stornetta in [BaHaSt92], and is used in the implementation of Surety's time-stamping service (described in Appendix C).
The idea is that, if the TSA receives n time-stamping requests within the same time unit, the TSA computes the hashes of the documents using a tree structure. The hash at the root of the tree is made public so that everybody can get it (and nobody can change it). The time-stamping certificate is formed by the information needed to rebuild the branch to which the document belongs, and the hash of the opposite branch. Validity checking is performed rebuilding the tree of hashes and checking that the root value corresponds to the publicly available one.

3.3.2.6.2 Prevention of "Man-In-The-Middle" Attacks

There is a risk that the request is intercepted by another user, E, that could impersonate A and request a valid time stamp on the hash of a document not owned by E. There are at least two modifications to the protocol that prevent this impersonation:

1. The user provides enough identification within the document as to guarantee it is his, even if Eve succeeds to time-stamp A's document for his benefit. If the document contents is not enough, it is always possible to incorporate the identity of the user when producing the hash to be signed: P (A+M).

2. The TSA may provide secure channels for requesting TS. Such a secure channel shall authenticate both the user and the service provider, and guarantee integrity and confidentiality of contents.

These modifications to the protocol also prevent malicious (or erroneous) assignments by the TSA of the time-stamping certificate to a party different to the requester.

3.3.3 Distributed Protocol

Distributed time stamping makes sense when there is no clear central authority to provide the service. Either because the clock or the security previsions are below needed quality, or because there is not enough trust in the relations between the involved partners. This distributed protocol is proposed in [HaSt91a, HaSt92b, HaSt95].

3.3.3.1 Sub-Systems

- ***Certification Authority (CA)***

This entity will only be present if she surrounding public-key infrastructure requires it.
The requirements on the CA regarding the time-stamping service shall be the following:

DP1	The CA **shall** link the identity of the users and TSA to their public keys via appropriate certificates.

- ***Secure Time Sources (STS)***
  In the distributed generation of time-stamping certificates, time provision can be either decentralised (each certifier chooses its time source) or centralised (there is a common time source that all the certifiers use). In both cases the time sources shall satisfy the following requirements:

DP2	Upon request of the members of the UoC, a STS **shall** provide a monotonically

	increasing value of time that will be synchronised with the rest of standard time sources accepted internationally.

- ***Universe of Certifiers (UoC)***
  The UoC can be the same as the universe of users, in the case that the users are performing the certification activities.
  The requirements on the UoC shall be:

DP3	The UoC **shall** select a set of "accepted" pseudo-random number generation algorithms that the users will use to select the time-stamping authorities.
DP4	The UoC **shall** select a set of "accepted" hashing algorithms that the users will use to produce the hashes of their documents when following the time-stamping protocol, and to obtain the seed to the certifiers-selection random process.
DP5	The UoC **shall** agree on the secure digital signature algorithm (as well as on a recommended key size) that will be used in the time-stamping protocol.
DP6	The members of the UoC **shall** cooperate with the user in the processes of verification and renewal.

- ***Universe of Users (UoU)***

The UoU is the set of people or electronic systems that will use the time-stamping service in the course of their activities. These users will be in possession of an identity certificate issued by the CA, and will use this certificate to prove identity to the TSA.

The requirements on the UoU shall be:

DP7	Each member of the UoU **should** be in possession of an identity certificate issued by the CA.
DP8	Each member of the UoU **shall** have a means to obtain the list of the UoC, so that a random selection of certifiers can be performed.
DP9	Each member of the UoU **shall** keep a copy of each of its time-stamped documents so that the verification and renewal can take place.
DP10	Each member of the UoU **shall** keep in a safe place a copy of the time-stamping certificates received from the randomly selected certifiers.

3.3.3.2 Provision

The provision of the time-stamping certificate is performed at user's request. When user Alice wishes to time-stamp a piece of information, she begins by hashing the information with one of the allowed hashes, thus obtaining a fixed length representation of it. This hash is input as seed in a pseudo-random number generator that will produce a sequence of numbers that will be used to select the members of the universe of certifiers that will time-stamp the document.
In this case the trust is strengthened by the fact that the user cannot predict or choose the certification agents, thus reducing the probability of collusion; the fact that there are several certifiers (as many as the particular implementation of the service considers appropriate) also prevents collusion.

Distributed Time-Stamping Protocol

In this description we assume that user Alice wishes to time-stamp data M.

1. Alice inputs the hash value H (M) as seed to a pseudo-random number generator (R) (that has been accepted as "standard PRNG" by the universe of certifiers).

2. Using the R, Alice obtains k numbers V_1, V_2, …, V_k, that she interprets as identities of k

certifiers.

3. Using the directory of certifiers, Alice sends the list of hash values, P (M) to the certifiers with identities V_1, V_2, …, V_k
   <A, R, P (M)>

4. Each certifier V_j time-stamps the received data independently, issuing a time certificate s_j, that is sent back to Alice.

5. Alice checks the receipts s_1, …, s_k and builds the certificate $C=(s_1, …, s_k)$.

$C= (s_1, …, s_k)$ is the time-stamping certificate.

3.3.3.3 Verification

To verify Alice's certificate, Bob would check that the certification parties were obtained with the random selection algorithm, and that their digital signatures were valid.

Verification of Distributed Time-Stamping Certificates

To verify Alice's certificate, Bob would:

1. Compute the hashes (h_1,…,h_n) with same hashes that Alice did. If these do not agree with Alice's values, Bob would reject the certificate suspecting that Alice changed the document.

2. Use the hash h as input to the pseudo-random number generator to obtain k numbers V'_1, V'_2, …, V'_k.

3. If these $\{V'_j\}$ do not agree with Alice's $\{V_j\}$, he would reject Alice's certificate, suspecting that Alice chose favourable certifiers.

4. If the $\{V'_j\}$ agree with Alice's $\{V_j\}$, Bob will check the independent time certificates from as many Vj as required for him to get enough confidence.

If the digital signatures are OK, he would accept Alice's claim; otherwise he would consider it invalid.

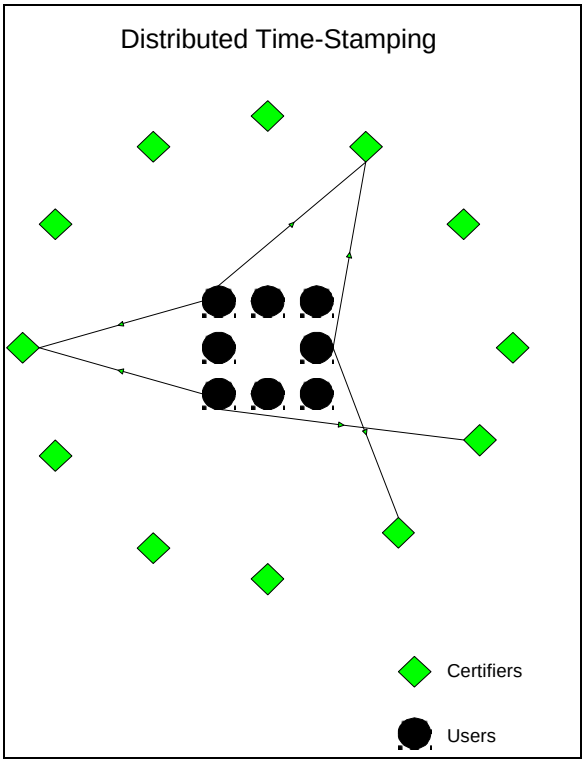
3.3.3.4 Synchronisation

The distributed protocol bases its trust on the addition of the amount of trust provided by an unpredictable collection of certifiers. If one of the certifiers further synchronises itself with other time-stamping providers, this certifier becomes a centralised authority, and the other certificates are not needed any more. Furthermore, the use of the random function to select certifiers becomes void. The distributed protocol is reduced to the basic protocol.

Nevertheless, users of a distributed protocol may need further assurance and achieve this through the combination of more than one time-stamping service. See the paragraph on combining several protocols below.

3.3.3.5 Activity Flow

The following diagram shows the roles played by each party, the information exchanges, and the order in which they are performed.



3.3.3.6 Variants

Several variations of the centralised linking protocol can be suggested, in order to further reinforce the security of the time-stamping process.

3.3.3.6.1 Threshold scheme

Since the number of certifiers can be significant, there is a non-negligible probability that some of them will be unable to respond to the time-stamping request, making it impossible for the user to obtain the time-stamping certificate. To prevent this, a “k out of n” threshold scheme could be applied that requires that at least k of the n time-stamping requests to be valid. Use of this approach must be carefully instrumented in order to avoid abuse. If users are free to select the k certifiers out of the n ones pointed to by the random function, there is an opportunity to corrupt the protocol.

In the case that it is the verifier that uses the “k out of n” rule, the election of k must also be random.

In the case that it is the user that just presents “k out of n” receipts, this user must present as well an audit trail that covers why the other ones are not available. Verifiers may request further insight into users records to assure that there is no systematic abuse.

3.3.3.6.2 Prevention of “Man-In-The-Middle” Attacks

The distributed protocol makes very difficult that someone could intercept the messages sent to the certifying parties. However, a small increase in complexity may provide further assurance.

There is a risk that the request is intercepted by another user, E, that could impersonate A and request a valid time stamp on the hash of a document not owned by E. There are at least two modifications to the protocol that prevent this impersonation:

1. The user provides enough identification within the document as to guarantee his ownership, even if Eve succeeds to time-stamp A's document for his benefit. If the document contents is not enough, it is always possible to incorporate the identity of the user when producing the hash to be signed: P (A+M).

2. The TSA may provide secure channels for requesting TS. Such a secure channel shall authenticate both the user and the service provider, and guarantee integrity and confidentiality of contents.

These modifications to the protocol also prevent malicious (or erroneous) assignments by the certifying parties of the time-stamping certificate to a party different to the requester.

3.3.3.6.3 Combination of Protocols

Users of a distributed protocol may need further assurance and achieve this through the combination of more than one time-stamping service. Two cases are briefly described.

Case 1. Use the distributed protocol to accumulate evidence. The certifiers are sound TSA, and the exchanges of data between them and Alice follow either the basic or the linking protocol. This approach prevents service interruptions or malfunctions when connecting to one of the TSA.

Case 2. The distributed protocol may be used routinely and eventually rely on a central authority to time-stamp a file of certificates. This scenario may apply when connectivity to a central authority is not permanent, or is too dear, or too slow to be used for each time-stamp. The distributed protocol is used locally for fine grain time-stamping, and one of the other protocols is used for coarse grain time-stamping.

3.3.4 Protocol Comparison

The following table summarises the pros and cons of each of the proposed time-stamping protocols.

Protocol	Pros	Cons	Appropriate scenario
Simple Centralised	• Simplicity • Small storage requirements • Fast operation • Simple verification	• Undetectable misbehaviour by TSA • Strong requirements on clock synchronisation	• Simple scenarios • High trust on TSA
Linking Centralised	• Unforgeable time-stamping certificates • May work without real-time	• Heavy storage of evidence • May involve several users to verify	• Many users • Low trust on the TSA
Distributed	• No central authority • Distributed trust • Synchronises partners • May complement the other protocols	• Involves several signers • Heavy requirements on evidence storage • Slow stamping	• No central authority • A threshold scheme, permits to tune reliability to needs.

3.4 ADDITIONAL PROTOCOLS

Previous protocols deal with the issue of time-stamp evidences. Two related activities are considered below, the renewal of time-stamp certificates, and how to demonstrate that a document is available after a given instant in time.

3.4.1 Certificate Renewal

Time stamps may be valid for a limited period of time, that may be explicit in the policy statement, or in the certificates that support the signing key of the TSA, or implicit if cryptography advances introduce reasonable doubts about the soundness of the elements (either a hash function is subject to successful attacks, or the signing protocol, or the keys, or …). For one or other reason, a time-stamp certificate may become void, and need to go under a renewal process to extend the non-repudiation period. It must be clear that the aim of renewal is not to get a new time-stamp for another period (no new protocol is needed for that), but to effectively extend the validity period of an already time-stamped document.

Haber and Stornetta [HaSt94] proposed an interesting way to renew certificates so that their validity can be extended indefinitely.

Time-stamp certificate renewal shall be performed by rehashing the original document together with the existing certificate, and digitally signing the resulting receipt; that is, the user will submit to the TSA

<U, previous_certificate, P (M + previous_certificate)>

Since the time-stamping certificates were renewed at a time where the hashing and digital signature primitives were safe, the renewal process will extend the validity of the original time-stamping certificate. The justification of this is that the renewed certificate incorporates the original one at a time when that certificate could only be obtained by legitimate means.

The previous certificate is explicit in the renewal process to permit later verification along a chain of renewal operations.

3.4.2 After Service

The protocols describe above demonstrate the existence of the hash value(s) of the document at a given instant in time. Since the hash value is derived from the document, it may be concluded that the document was not created after the time-stamping token.

But those protocols cannot prove that the user has the document after a given instant in time.

Achieving this purpose is easy if a time token T is merged with the document:

P (M + T)

this token T may be any unpredictable easily verifiable fact (any news), or it may be a time-stamp issued by the TSA on any void message, or an unpredictable fact time-stamped by the TSA.

When the merged information, <U, T, P (M+T)>, is further submitted for time-stamping, the service called “within” is provided: the owner has the document after t1 and before t2, being t1 the time-stamp merged with the document (T above), and t2 the last time-stamp.

This kind of services is likely to be used to provide non-repudiation services in scenarios where there are unique tokens, such as hardware tokens, that cannot be replicated.

4 AUTHORITIES INVOLVED

This chapter reviews the different authorities that have been proposed in the context of public-key infrastructures, and describes their roles and interrelationships, with special attention to the TSA.

4.1 DEFINITIONS

A PKI involves several authorities and service providers in order that are expected to operate in good harmony to provide quality services to end-users. This section described the different roles, their mutual relationships, and the impact of the architecture of public-key authorities on the design and operation of a time-stamp authority (TSA).

The following roles are considered:

1- CA (Certification Authority)
2- NA (Notary Authority)
3- RA (Registry Authority)
4- EA (Escrow Authority)
5- TSA (Time-Stamp Authority)
6- STS (Secure Time Service)
7- TDS (Time Data Service)

4.1.1 CA (Certification Authority)

An application based on public key technology requires that users of a public key to be confident that the public key really belongs to the subject with which they want to use this kind of technology. Basically, there are two models of confidence: Direct Trust and Third-Trusted Party.

The Direct Trust model bases confidence between users in direct contact. On public key environment this means that users interchange their public keys face to face or users use an additional physical transmission channel to ensure that the public key received really belongs to the correct user. This model is appropriate if it is used in a few users environment and has the advantage of simplicity. However, if the number of users increases, it becomes very difficult to manage this kind of confidence.

In a Third-Trusted Party model, users trust on an entity (Certification Authority) that assumes the responsibility to relate public keys with subjects. In this model, each user only has to communicate with this entity, so the management is easier than a Direct Trust model when a big number of users are involved. Moreover, in case a pair of users disagrees or has any problem in security, the Third-Trusted Party can solve these situations because his management role.

Certification Authorities achieve this binding, relating subjects to public keys and then signing each of these structures.

There are some requirements that a Certification Authority must satisfy:

A11	Confidence: a CA **shall** work properly to his role and must respect his own security policies.
A12	User identification and authentication: a CA **shall** identify users in an infrastructure accord to his security policies.
A13	Certificate generation and distribution and local certificate management.
A14	Certificate revocation notification: a CA **shall** support secure revocation notifications.
A15	CRL (Certificate Revocation List) generation and distribution and local CRL

	management: a CA **shall** generate, refresh and distribute the list of certificates that have been revoked.
A16	Management auditing: a CA **shall** store all the important events that can help to resolve future security problems.
A17	Certificate and CRL storing a CA **shall** store all the certificates and CRL it has issued.

4.1.2 NA (Notary Authority)

The Notary Authority is a Trusted Third Party (TTP) that can be used as one component in building reliable non-repudiable services. Useful Notary Authority responsibilities in a PKI are to validate signatures and to provide up-to-date information regarding the status of certificates.

A Notary Authority verifies the correctness of specific data submitted to it. The Notary Authority provides the notary service in order to non-repudiation evidence may be constructed relating to four types of service:

NPD (Notarise Possession of Data): the validity and correctness of an entity's claim to possess data. The NA **shall** verify the correctness of the enclosed digital signature using all appropriate status information and public key certificates and produce a signed token attesting to the validity of the signature. This means that NA authority **shall** authenticate the requesters of the service.

ND (Notarise Data): the validity and correctness of various types of data at particular instant in time. The NA **shall** verify the correctness of the enclosed data through the stated policies and produce a signed notary token attesting to the validity of the data

NB (Notarise Both): both the signature and the data **shall** be verified.

NC (Notarise Certificate): the validity and revocation status of an entity's public key certificate the NA **shall** verify according to [PKIX] of the enclosed certificate and its revocation status at the specified time using all appropriate status information and public key certificates and produce a signed notary token attesting to the validity and the revocation status of the certificate

In all cases, the trust that PKI entities have in the Notary Authority is transferred to the contents of the notary token that the Notary Authority returns, which always includes a trusted time. This trusted time **may** be obtained from a Time Stamp Service.

NA **shall** sign each notary token using a key generated exclusively for this purpose and have this property of the key indicated on the corresponding certificate

Adams and R.Zuccherato from Entrust Technologies define notary protocols in IETF Drafts [Adams98b].

4.1.3 RA (Registry Authority).

In addition to end-entities and CAs, many environments call for the existence of a Registration Authority (RA) separate from the Certification Authority. The functions which the registration authority may carry out will vary from case to case but may include personal authentication, token distribution, revocation reporting, name assignment, key generation, archival of key pairs… but does not sign or issue certificates.

The term Local Registration Authority is used elsewhere for the same concept

There are some technical and organisational reasons which justify the existence of these authorities.

Technical reasons include the following:

- If hardware tokens are in use, not all end entities will probably have the required equipment to initialise these, due to economical reasons. The RA equipment may include the necessary functionality and supply it to a group of end entities.

- In case end entities do not have the capability to publish certificates or the capability of issue revocation requests (if the key pair is completely lost), RA may be placed for this.

Organisational reasons:

- RA entities can reduce the number of CAs required in an organisation, and a CA can directly interact with less end-entities so, CA management can be easier.

- For many applications there will already be in place some administrative structure so that candidates for the role of RA are easier to find than the CA role.

- RAs may be better placed to identify people, specially if the CA is physically remote from the end entity.

RA acts as a gateway for certification requests and receipts, certification revocations, CRL request … , because the RA will redirect all the requests it receives to the CA it is working with, only if RA is able to authenticate the requester users under the conditions imposed by the related CA. Moreover, a RA has extra-functionalities to simplify to CA some of these tasks. The following cases exposes some examples of how a RA can interact with a CA to simplify the management CAs have to do:

Proof of possession: in some cases and to prevent some kind of attacks to a CA, a proof of private key possession **may** be requested by the CA policy to the requester end-entity. If a RA exists between them, it **may** check this requirement and attests to the CA that proof of possession has been received and validated. All cryptographic check has important time cost, and if RAs do some of them, CAs overhead is drastically reduced. Of course, a very restrict policy can disable this feature.

Location of key generation: if an end entity does not have the resources to generate its own cryptographic key pair, the CA or the RA **may** do it. Again, if the RA generates the keys, the CA overhead is reduced

Confirmation of successful reception: in some cases it would be interesting that an end entity can issue a receipt to indicate that it has received the certificate the CA issued. This receipt **shall** be signed by the end-entity, so the RA instead of the CA can do its verification.

4.1.4 EA (Escrow Authority)

An Escrow Authority is third party that participates in cryptography infrastructures where some agencies want to have the possibility of accessing to end users communications. For example, the law-enforcement agencies wish to have access to the communications of suspected criminals by wire-tapping.

The idea is that communications are encrypted with a secure cryptographic algorithm, but the keys would be kept by one or more escrow authorities. Each of these keys (unit key U) would be escrowed and each escrow authority would safely store a different fragment of the key.

End users use a not escrowed key K (session key) to encrypt their messages, which **may** be generated by a public method such as RSA. Then other information is appended to the message before it is transmitted. Basically, this information includes the session key K encrypted with the unit key U. The receiver check the added information and decrypts the message with the key K.

If a law-enforcement agency wishes to tap the line, it presents an authorisation warrant to the escrow authorities, and these give the parts of the unit key U to the agency. Then the agency decrypts the session key K transmitted with the message. Now the agency can use K to decrypt the actual message.

4.1.5 STS (Secure Time Service)

A Secure Time Service is an entity that supplies trusted time to other entities in a public-key infrastructure. A Secure Time Authority is required to synchronise its clock to the others STS in the community to reduce dispersion between clocks which can have a negative influence to Time Stamp Service requirements or other services. It is recommended that Secure Time Authorities also synchronised its clock with other standard time source organisations (in UTC Time standard for example) through a high precision communication media (GPS time for instance).

Nowadays, Network Time Protocol (NTP) v3 is one of the most used protocol to synchronise systems over distributed control networks and supports both standards mentioned above.

4.1.6 TDS (Time Data Service)

A Time Data Service is a Trusted Third Party that acts as a source of data strongly related to external unpredictable events and issues temporal data tokens. Temporal data tokens from one or more TDS **may** be included in time stamp tokens providing added evidence for the times stamped.

The following requirements shall be imposed on the TDS:

A61	The temporal data which a time stamp tokens is associated to **shall** be unpredictable in order to prevent forward dating of tokens.
A62	A TDS **shall** verify only temporal data.
A63	A TDS **shall** include the current data associated with a specific and unpredictable event in each temporal data token
A64	A TDS **shall** interact with CA to obtain a key pair certificate that will be used to sign each time data token.
A65	A TDS **shall** sign each temporal data using a key generated exclusively for this purpose and **shall** indicate the policy used in this signature.
A66	Possible types of temporal data: stock market information, sports and lottery results, headlines in specific newspapers, official weather in a specific location…
A67	The TSA involved in the transaction **shall** verify the correctness of the signature to include the time data token into the time stamp token.

Adams and Zuccherato define TDS and TSA in IETF drafts [Adams98a]

4.2 ARCHITECTURAL ORGANISATION OF AUTHORITIES

Public key infrastructures which use a Third-Trusted Party model can be constructed basically on two possible architectures: a hierarchical infrastructure or a cross-certified infrastructure.

4.2.1 Hierarchical Architecture

All the authorities involved in the provision of security services have to be more or less integrated into a hierarchical architecture. This hierarchy establishes who guarantees the authenticity of the authority and the security policies applied during its operation.

This architecture is based in a tree of trust. A user of a security service requiring knowledge of a public key often needs to validate the certificate containing the key, and if he does not have the public key of the CA which issued that certificate, the user might need an additional certificate. In these situations, chains of multiple certificates are needed by end-users, and in a simple case, each certificate can be related to a different level in a hierarchical architecture. These chains of certificates are called Certification Paths

ITU-T X.509 [X.509 v3] proposes a trusted environment to provide user identification service based in the X500 Directory. X.509 also defines the certificate, CRL formats and all the terminology related and, in the proposed scheme, each Directory user has a certificate issued by a Certification Authority that is stored in the X500 Directory. X509 also recommends a hierarchical CAs architecture based X500 Directory distinct names.

4.2.1.1 PEM Infrastructure

Nowadays, Internet community has adopted a completely hierarchical architecture that is known as PEM that do not require the X500 Directory. For PEM (Internet Privacy Enhanced Mail), [RFC 1422] defined a completely rigid hierarchical structure of CAs, and defines three types of these authorities:

a) Internet Policy Registration Authority (IPRA): This authority acts as the root of the hierarchy, at level 1. All certifications path originally have to start with the IPRA. The main purpose of IPRA is to registry and to certify PCA, and to check that PCAs respect their own and IPRA security policies. Other services supported by IPRA are the following:

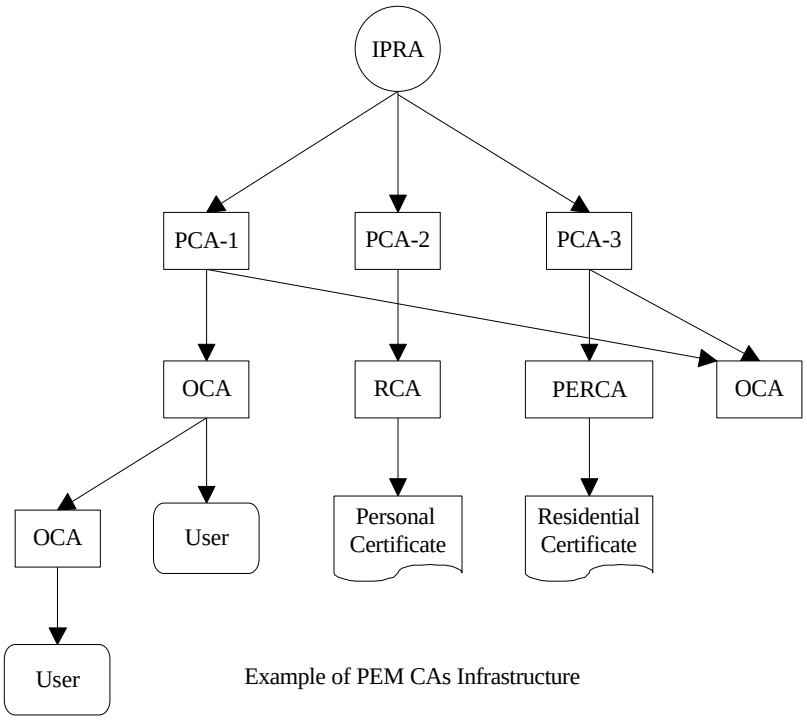
   - Defines which processes a new PCA have to use to registry to IPRA
   - Defines a X500 name based scheme
   - Requires PCAs to provide and manage a CRL database that must be accessible by e-mail for all users in case the X500 Directory is not used.

b) Policy Certification Authorities (PCA): are at level 2 of the hierarchy, and each is certified by the IPRA. Distinct PCA aim to satisfy different user needs, and because of this each of them must define and publish a statement of its policy. For example, a PCA might support the e-mail needs of an organisation, while another PCA with a more restrict policy might be used in economic transactions.
   Each PCA defines his own pair keys generation processes, information privacy and security policies in the CA authenticity process, and also imposes security requirements to their related CA.

c) Certification Authorities: are at level 3 of the hierarchy and can also be at lower levels. A CA can be certified by two or more PCA, which means that CA will have two or more key pairs and his users can registry to the CA with different security policies.
   There are three kinds of Certification Authorities at this level:

   - Organisational Certification Authority (OCA): represents a particular geographical area, a particular organisation or particular organisational units (e.g. departments, groups, and sections)
   - Residential Certification Authority (RCA): provides certification services to the users which are not in any organisation.

- Personal Certification Authority (PERCA): it is used if a user wants not to deliver his real name when using a certification service.



Example of PEM CAs Infrastructure

However, with X.509 V3, some of the restrictions imposed by this hierarchy may be omitted. For example, PCA are not necessary with the certificate extensions relating to certificates policies and policy mappings. The application can determine if the certification path is acceptable based on the concepts of the certificates instead of priori knowledge of PCAs.

Moreover, certifications path may start with a public key of CA in user's own domain, or with the public key of the top of the hierarchy.

There are also other studies in certification infrastructures that are based or have different common points with PEM architecture. For example, PASSWORD infrastructure, NIST infrastructure or ICE (Infrastructure of Certification for Europe) are very similar to PEM but differs in some aspects related to the services that offers PCA, names conventions or the possibility of having more than an unique hierarchy for all users.

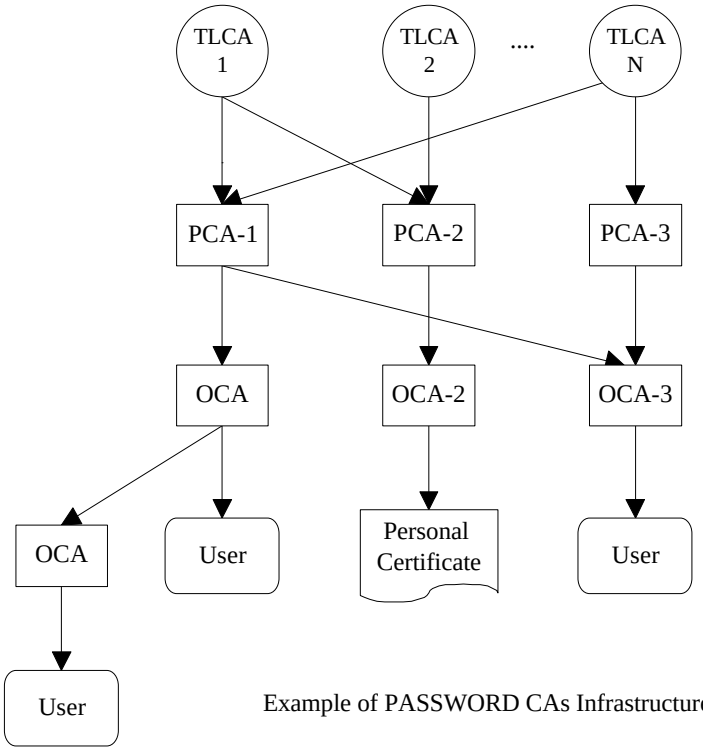
4.2.1.2 PASSWORD Infrastructure

PASSWORD Infrastructure is a multi-hierarchical PEM based infrastructure that proposes a different hierarchy for every European country and uses X500 Directory. Each European country can have a TLCA (Top Level Certification Authority) that is a root Certification Authority, and there is no limitation on the number of TLCA per country.

A TLCA is an authority that certifies PCAs with similar security policies. A TLCA in a country can certify PCAS from other countries with an equivalent policy, and because of this, certification paths between users from different countries can exist under similar security policies. A PCA can certificate OCA (Organisational Certification Authorities) from the same country, or foreign OCA, but it can not certificate others PCA.

Certification paths between countries could also exist with cross-certification. This point will be commented later, but basically two TLCA certify each other which simplifies the number of certificates that a TLCA must issue.

As PEM, PASSWORD infrastructure imposes restrictions on names in authority identifications. Authority names in X500 Directory must also have a hierarchical order.



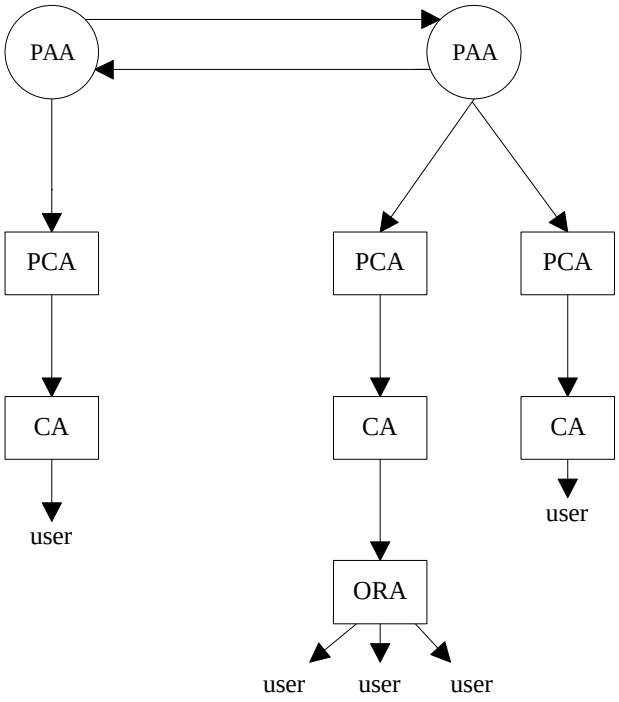
Example of PASSWORD CAs Infrastructure

4.2.1.3 NIST Infrastructure

NIST Infrastructure is a multi-hierarchical infrastructure similar to PASSWORD. Every federal state or country have a root authority called PAA (Policy Approving Authority) which has a similar role than IPRA in PEM. Each PAA can relate to other PAA (from different countries) by cross-certification.

On the second level of the hierarchy there are PCA, which certifies level three authorities CA. In this scheme, a CA can not certify others CAs and there is a single kind of CA, but also a new authority at level fourth of the hierarchy called ORA (Organisational Registration Authority). An ORA is a RA (Registration Authority) that can appear if a large number of users depend on an unique CA.

NIST propose a directory service to certification delivery, not necessary X500 Directory, without any limitations on authorities directory names.



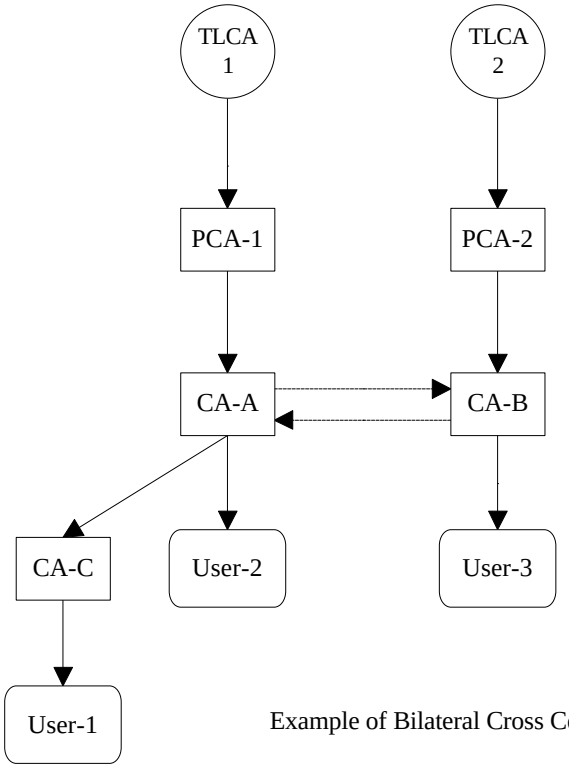
Example of NIST CAs Infrastructure

4.2.2 Cross-Certified Authorities

Due to the lack of roots for hierarchical architectures trees, and also in order to simplify the relationships between authorities using the same security policy, it has been allowed to establish "horizontal" links of trust. These links can relate authorities with the same level of trust with respect to its users and/or parent authorities, but they can also link authorities of different level of trust.

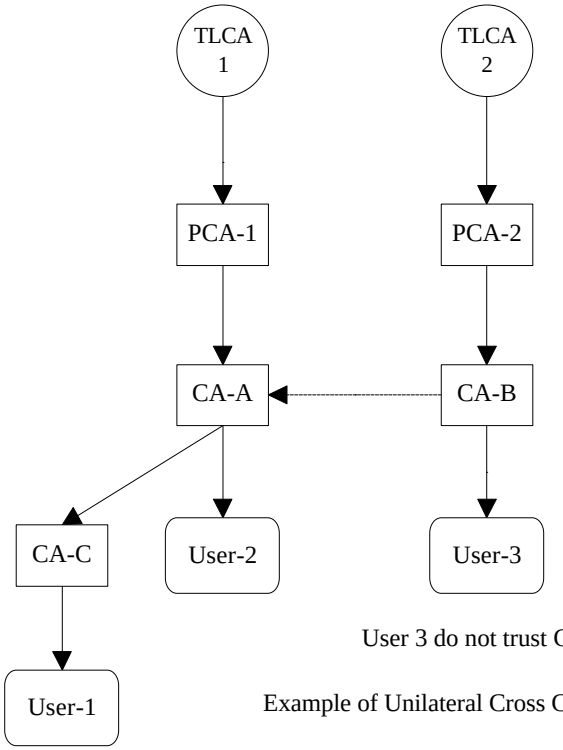
These horizontal links of trust which connect different hierarchies are build trough cross-certification between the two involved entities. We can consider two different types of cross-certification: bilateral cross-certification and uni-lateral cross-certification

In a bilateral (or two-way) cross-certification scenario two CA (CA-A and CA-B) wants to interchange their trust. This means that the two CA trusts the other one, which implicitly implies that their security policies are equivalent. Because of this, all entities which are in lower levels in the hierarchy under CA-A will trust this CA-A, CA-B and also CA-B parent entities, and also the adverse case, entities under CA-B will trust CA-B, CA-A and also CA-A parent entities.



Example of Bilateral Cross Certification

In an uni-lateral (or one way) cross-certification scenario one CA-A trusts the other CA-B, but this do not trust the first (CA-A), so in this scheme the security policies of CAs are not equivalent. For example, CA-B security policy may be more severe than CA-A security policy. In this scheme, all entities under CA-A will trust CA-A, CA-B and all entities under CA-B, but not the adverse case.



User 3 do not trust CA-C

Example of Unilateral Cross Certification

Both “two-way” and “one-way” cross-certification implies to issuer certificates between CAs involved. In “two-way” cross-certification, each Ca must issue a certificate to the other CAs, while in “one-way” only one certificate is needed and it must be issued by the CA that is trusted by the other. In the example, CA-B must issue the certificate.

[PKIX #3] describes processes involved in obtaining cross-certificates, data formats and restrictions of a cross-certification request compared to a simple certification request.

4.2.3 Hierarchical vs. Cross Certified Architectures

Hierarchical architectures are probably the best infrastructure to represent pyramidal organisations, and one of the simplest infrastructures to manage. For example, PEM infrastructure has a single root authority, which manages all the other authorities, which means that every certification path must start on the root entity, and is quite easy to verify this property and that the certification path is well constructed.

This scheme is also technically simple to implement and easy to administer in a small community of users. However, this scheme presents the following negative points:

It is very difficult to find a general entity that can support all the different interests or needs for a large community. For example, PASSWORD or NIST infrastructure considers that international and national differences between users are important enough to consider different hierarchy roots to simplify certificate management. Moreover, it would be interesting that similar interest organisations could define their own policies and hierarchies, and it would be very difficult to centralise these different scheme in a single authority

In a huge single hierarchical infrastructure the certification path verification process can have a high cost in performance. If an end-user or authority is trying to verify a certificate which comes from other part of hierarchy it will need more additional certificates than the same process in a multi-hierarchical scheme where certification authorities interchange confidence.

On the other hand, cross-certified architectures solve these problems because horizontal links of trust allow exist a large number of small hierarchies which can be easily administered. However, in this scheme is very important to manage horizontal links correctly, and it can be complicated because the large number of relations that can have a root authority and because it may be difficult to match completely security policies from different hierarchies. Moreover, the management involved in cross certification expiration is also more complicate because different certification authorities have to interchange their CRL without compromising security at any moment.

To sum up, a hierarchical architecture is easy to implement and represent very well real world pyramidal organisations but it is progressively impossible to administer as the number of users increase and is not flexible enough to their needs. A cross-certified architecture is more complicated to implement and to manage at all, but much more flexible and adaptable to different organisations and user needs.

4.3 TSA CO-ORDINATION WITH OTHER PKI AUTHORITIES

After revising the different authorities that may play a role in a PKI scenario to provide trust services, it makes sense to ask whether these authorities make sense by themselves or have to be clustered together to provide useful services.

4.3.1 CA alone

A CA has to mention time in its certificates and revocation lists.

A certificate includes a number of dates that specify intervals of validity. These apply to public keys, private keys, and other attributes that users might need. While public CA are expected to be rather restrictive in the use of extensions, private CA may make a wide use of attribute certificates for their particular use. However, there is no obvious or universally applicable reason for the TSA to time-stamp these certificates to be able to demonstrate that the certificate was ready at a given instant in time.

However, a CRL includes a date of issue, and a date of certificate termination. Deviations in these dates do make a difference:

- if the CRL is actually issued with a [fake] older date, verification of trust paths performed in the time between becomes void; that is, the CA cannot say today that yesterday it published that and that
- if the issued CRL declares an older time for a certificate to be terminated, the date of issue takes precedence to determine the instant of termination; that is, the CA cannot say today that this certificate is not valid since two days ago

The date of issue of the CRL becomes of paramount relevance to determine the validity of a certificate verification. Therefore, issuing CRL without the assurances provided by a time-stamping service makes the CRL subject to fraud.

4.3.2 CA + TSA

The TSA shall be used to back CA assertions related to time. This is an optional requirement for certificates, and is a must requirement for revocation lists.

The TSA role may be embedded within CA activity; that is there is a single signing act, and a single signature, that demonstrates CA commitment to mentioned times.

If the simple protocol is used, there is only a requirement on the extendedKeyUsage attribute of the signing key to state the role(s) of the CA, that shall be explicit as well in a section of the CPS.

If the linking protocol is used, the linking elements shall be part of the certificate, as further extensions. The CA shall store evidence data (the hashes, and the identities of the requesters, as needed).

If the distributed protocol is used, the chosen witnesses shall be recorded in a certificate extension, and the CA shall store evidence signatures, and made them available on request.

If the CA uses a segregate TSA to time-stamp certificates and CRL separately (another signature), data in the certificates are subject to interpretation: the proof of existence of the certificate or CRL is the time-stamping time, rather than the time of issuance. This approach may introduce a noticeable amount of noise in the system, since the very basic process of certificate validation is subject to two signed data: the CRL itself, and the time-stamped one. This situation may prove unacceptable for normal practice.

4.3.3 NA alone

The exact role of NA is subject to wide debate, and further experience is needed; but it may be foreseen that NA statements regarding some information shall have time limits. Possession of data, and data notarisation must be reliable as to say that the notarisation is valid at a certain moment in time, and for a finite time hereafter. Certificate notarisation is valid at a precise instant in time, when the NA has checked the path of trust.

Therefore, a notary service without a trustful time may be meaningless for most foreseen services.

Nevertheless, there might be other notary services that do not require a trusted time, such as all those protocols that use a notary as a TTP in the middle, basically as an evidence storing agent, within an end-to-end protocol that is self-timed.

4.3.4 NA + TSA

Time Stamp Service is basically a service that only matches a representation of a content (hash of a document) with a secure time. However, a Time Stamp Service is not required to know the content of the document nor the requester of the service (if linking protocol is not supported), which can limit potential uses of this kind of service.

A Time Stamp Service has much more sense if we consider an authority that can use this service but is also able to authenticate requester users and to optionally analyse the contents of the message to time stamp. The Notary Authority plays this role in the scheme we are considering, so NA will probably be the most common TSA client.

For most of its services, a notary signing a statement should state as well the time of signing, and this time must be supported by a secure time server. A TSA embedded within the notary may play a role, thus providing a single digital signature for the whole service pack.

Same considerations apply as to CA for binding a TSA within a NA.

4.3.5 RA alone

A RA may play a large number of functions that may require a secure time, or not. Time stamps are required when the RA plays a non-repudiation role, but time stamps are not required when the RA acts as a transmission agent for some remote CA or NA. In other words, reliable time stamps are not needed when the RA plays an intermediate role between users and other authorities, and the protocol is self-timed.

There is no extensive experience in designing, nor using RA in actual PKI scenarios.

4.3.6 RA + TSA

An RA with a linked TSA may provide non-repudiation services, and may usefully record all its services for auditing, and as evidences in court: the logs of activity themselves become trustful.

Same considerations apply as to CA for binding a TSA within a RA.

4.3.7 EA alone

The role of EA has nothing to do with respect to time. It is a purely storage service. Nevertheless, activity recording may greatly enhance EA auditing.

4.3.8 TSA alone

A TSA alone may time-stamp documents without taking into consideration their contents. No notary service may be provided by these means.

A time-stamp on a document that is no other way known is mostly useless. As far as a collection of [possibly contradictory] documents may be time-stamped under the unique control of the requester, the value of these stamps is void: any document might be disclosed as convenient.

Time-stamping services require some storage facilities to make sense. As soon as there is a verifiable storage system, linking documents, identities and time-stamps, these components may

be used as evidence with trustful time. Notice that the contents of the documents themselves may remain encrypted; the key issue is that the collection of documents subject to consideration is known, and non-repudiable.

4.3.9 TSA + TSA

Time-stamps provided by a TSA must be trustful, either by themselves, stemming trust from conventional auditing procedures, or by establishing synchronisation links. Documents time-stamped by different TSA may need to be compared, therefore synchronisation is a must.

TSA **shall** synchronise each other in a secure context to prevent security attacks, and because of this is, an additional (and different) key pair **should** be used and a different certificate **should** be obtained from a CA in these communications.

4.3.10 TDS alone

A TDS is an entity that is related to TSA in a public key infrastructure. It is difficult to find an authority or end entity that requires TDS services if it does not require a TSA or NA services. In case it needs to obtain some kind of untrusted time data information it can always get it from other sources. Otherwise, this authority or end entity requires a TSA or NA service.

4.3.11 TDS +TSA

In order to add supplementary evidence for the time included in time stamp tokens a TSA **may** interact with one or more TDSs. Time Stamp Tokens then include some unpredictable data time that prevents forward dating of tokens.

Because the time stamp service requests a TSA may get in a period of time can be considered unpredictable information, a TSA **may** also acts as a TDS for others TSA in the hierarchy. Because of this, a TSA may include accumulated hashes from others TSA as a time data tokens if the requirements to a TDS and it behaviour are satisfied by the TSA that supplies these data time tokens.

4.3.12 STS alone

Nowadays there are a large number of time sources that we can consider as STS in a public key infrastructure, for example, GPS, UTC time sources or all time servers which are synchronised through NTP (Network Time Protocol v3)

Different Secure Time Services in a public infrastructure **shall** interact each other to avoid (micro) time differences which can cause time related problems and to prevent security attacks

In security terms, STS synchronisation is a warranty that the time a STS supplies has not been manipulated from an external agent. In the STSs synchronisation process, if any STS detects that one of the others supplies a time value that differs too much from the rest it may indicate that there is a security problem in terms that the time the STS has issued can not be trusted. It is expected that it must be difficult to trick all the STS in a public infrastructure.

4.3.13 STS + TSA

TSA will interact with STS in order to obtain time data that can be considered the real time at a moment in the related public infrastructure.

TSA **shall** interact with at least one STS but it is strongly recommended that TSA select one from a set of STS supported by the TSA on time stamp transactions. There are basically two reasons related to security and service availability.

If a TSA interacts with different STS on time stamp transactions, the TSA may easily detect security attacks. The TSA can detect them earlier through the time stamped token chain TSA check during the transaction. If only a STS is used for all the requests the TSA receives and the STS has been manipulated, it will have to wait to TSA synchronisation to detect a possible attack.

Also, if a TSA interacts with two or more STS and one STS is unavailable, the time stamp service may use the others in order to maintain the service running.

However, because communications between STS and TSA are not trusted another implementation scheme is possible. A TSA **may** not interact with external STS if a TSA is a STS itself. In this case, TSA is directly synchronised with other STS in the infrastructure and will probably get a more exactly real time than if it has to communicate to STS through networks.

4.3.14 Any authority alone

Any authority alone may provide services that either unaffected by time, or services which timing is neatly defined by other activities or exchanges carried out by the parties that may require the service. For those activities, no TSA is needed.

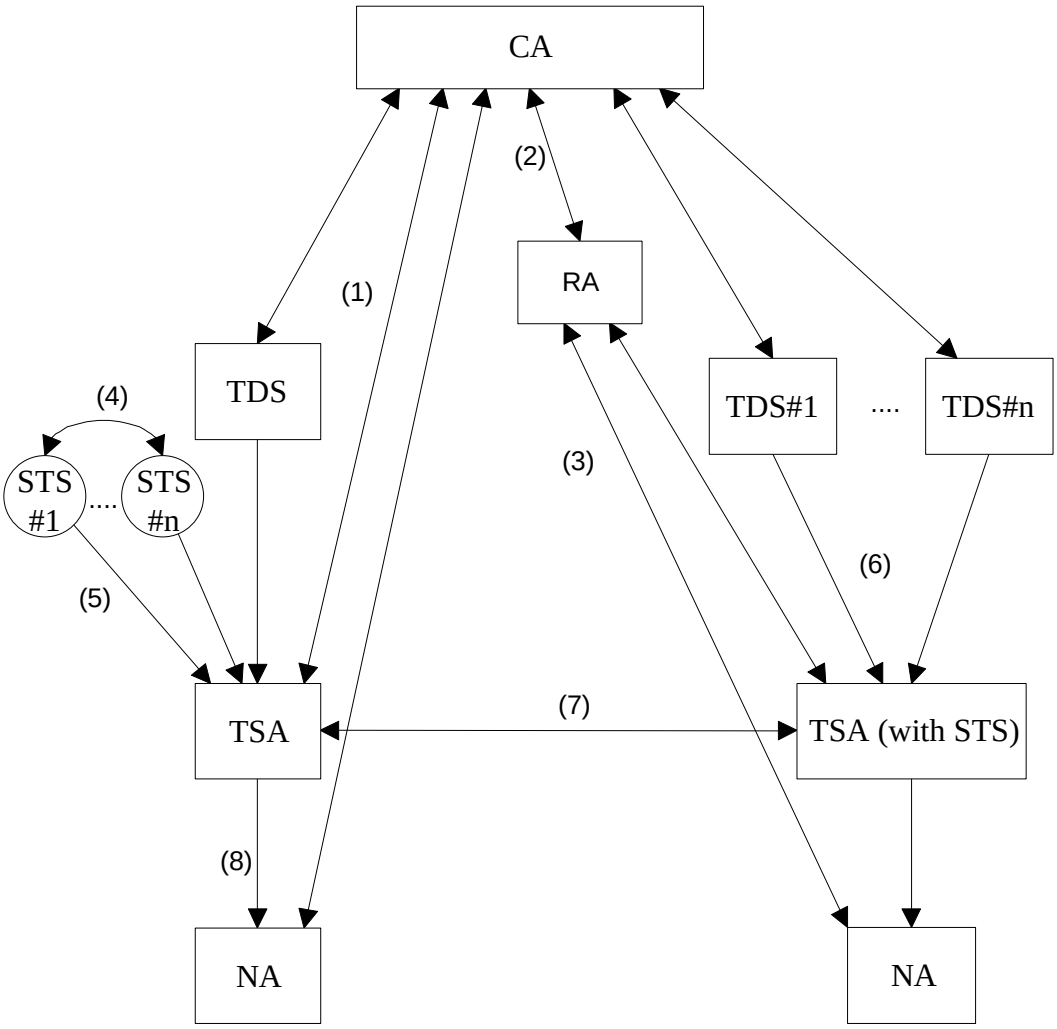
4.3.15 Any authority + TSA

Any authority with a TSA adds value to its statements, whichever they are. A trustful time embedded in authority statements establish a proof of existence at the instant of signing. This fact may be used as additional evidence in protocols that are otherwise self-checking, and in situations where the protocol is abruptly terminated, either by a misbehaving party or by accident.

In general, time-stamping records of activity of a TSA enhances the value of logs both for auditing and for providing extra evidence in disputes.

4.4 A POSSIBLE PKI AUTHORITY SCENARIO

The following chart represents a possible scenario involving the entities discussed.



Example of interactions between involved entities

(1) Interactions between a CA and involved entities. CA acts as a certificate issuer (root of the hierarchy)
(2) Interactions between a CA and a RA related to it
(3) Interaction between end entities and a RA to get CA services
(4) STS interacts to syncrhonise time information
(5) TSA get time information from STS
(6) TSA optionally get time data information from TDS
(7) TSA interactions to introduce syncronisation points in time stamp token chains
(8) NAs optionally use Time Stamp services

5 SCENARIOS OF USE

Time-stamping technology is quite new and therefore there is little experience in real service provision. However, it is of vital importance to identify potential service users, their needs and requirements. This section deals with the analysis of these scenarios of use, applying a methodology to find requirements through the study of provision of Time Stamping services in different situations.

The process of analysis on the scenarios of use leads to a better knowledge on the potential market and helps us to define better architecture and operative design to meet requirements demanded in each given scenario.

The use of time stamps, as they are considered in this document, is an innovative use of technology. The generic entities involved in a time stamping transaction are a TSA (Time-Stamping Authority) and the users of the service. Each entity plays the same role in all imaginable scenarios of use but in every scenario each entity is of different nature (public, private, national, trans-national, etc.) and has different characteristics (type of event/document, structure, confidentiality, time accuracy, etc.). A simple analysis of entities will give us a way to structure the various types of scenario and will help us in exploring them.

The basic questions to answer in each scenario are

1. who are the actors?
2. what is the activity carried out by each actor to get a time-stamp?
3. what is the activity carried out by each actor to verify a time-stamp?
4. who stores what evidences?

In those scenarios where more than one TSA is involved, there is one further question to answer:

5. how are services synchronised?

5.1 GLOBAL OVERVIEW

The launch of structures making possible SGI, opening doors for a world-wide commerce through open networks, is regarded as a unstoppable process.

In Europe, many efforts have been done last years in order to achieve economic integration of countries conforming the EU. This will allow free exchange of goods, services, capital, and people among the member countries that will use the same legal tender: the EURO. It is clear that this process means a massive interchange of digital information in its different forms (text, graphics, video, audio...). This interchange involves different actors from different scopes: public administration, private sector and individuals.

Digital information introduces complex problems related to its security and protection, among them raises that one of guaranteeing the integrity of a digital document in a given moment in time. This section will cover scenarios where this problem needs to be solved, and the particular characteristics that are demanded to the service provision, paying special attention to the European scope and relationships with other certification services such as PKI´s.

5.2 PUBLIC ADMINISTRATION

In first step, we must define “Public administration” in order to be able to place and classify scenarios where entities regarded as public administration take part. When we name “Public

Administration" we are referring to government entities in the most general sense, this means any kind of local, regional, national or international administrations.

Yet another definition is required; TSS applications in public sector must be defined as relationships among citizens and public organisms, industry and public organisms, and different public administration organisms.

The use of TSS will take place in all traditional procedures where the time variable is required as a physical imprint. This is in documents when applicable laws demand it or when administrative units demand it with the final objective of providing proof of existence, integrity and/or delivery making reference to expiration dates, terms, or any other effects.

The Public Administration may play several roles related to time-stamping:

1. providing the service to individuals
2. providing the service to private companies
3. providing the service for other sectors of the administration
4. using the service when serving individuals
5. using the service when trading with private companies
6. using the service when interacting with other administration units
7. using the service for its internal operations

5.3 PRIVATE SECTOR

Applications of TSS in private sector cover activities where the use of the time variable is required but takes place off-side official organisms participation whether acting as witness or taking active part. Therefore we can define "Private Scenarios of Use" as those where TSS is provided to support relationships among individuals, among commercial entities, and among individuals and commercial entities.

This sector will use extensively TSS for it will decrease cost in commercial operations, but some strict minimum requirements will be demanded:

- Legal validity (Universal if possible) for Time-stamping certificates.
- Minimum service quality to meet each specific requirement.

The private sector may play several roles in a PKI with respect to time-stamping services:

1. provision of service for itself
2. provision of service under contract for other parties
3. using the service for external operations
4. using the service for internal operations

5.4 THE BORDER BETWEEN PUBLIC AND PRIVATE SECTORS

It is expected that TS services provision will be deployed in a complex multi-domain context.

Services will be available either from public authorities or from private companies. Furthermore, proved that in most cases these services depend on CA's since digital signature scheme is required in certain services, we encounter another source of heterogeneity. We will have private and public CA operating together and providing service to public and/or private TSA.

Depending on applicable laws, time-stamping services provided by private entities may not have legal value in court, or its value may need further assessment by the public administration.

This complexity relates to TSA operating within a national domain providing services to processes or applications in the real world such as examples depicted in this section, namely: electronic commerce, teleworking, SAP, etc. The situation becomes even more complex when considering supra-national scopes such as the European one. The need for synchronisation among Authorities operating in different countries and/or domains leads to layer structure with different topologies, architectures and protocols...

Within this multi-domain context, applications using TSS will require a network of co-operating management processes to support the management and provision of the interchanges of secure date and time information, conforming an European network of TS services trusted within European scope

5.5 SOME EXAMPLES

There is a wide spectrum of situations that lead to information transactions over open networks. Most of these may need the services provided by a TSA. It is important to note that these services can be requested by completely different users with completely different motivations.

In order to follow a systematic approach to analyse possible scenarios of use, we must group examples of use so that we can identify usage patterns representing homogeneous user groups:

- TSS users from Public Sector.
- TSS users from Private Sector.

By searching in both user groups we can find many (almost infinite) possible applications of TSS. A selection has been made among them in order to point out significant examples that conform the launching platform for identification and analysis of scenarios of use.

5.5.1 Examples of Application in the Public Sector

5.5.1.1 Administrative registry

Every official organism, regardless its domain, provides a registry service whose aim is to admit and register documents that citizens submit to public administration on “motu propio” or under administration demand. It is also in charge of registering the outcome of official documents.

Modernisation of public service provision is forcing governments to adapt their bureaucratic structures to the new growing technologies. This process will introduce open networks as an effective tool in communication among Public administration bodies and citizens. This fact will lead to creation of electronic registers that would use security services provided by PKI and specially would require Time stamping services to operate in a secure way.

In this direction, we must name some initiatives aimed to deliver SAP (Single Access Point) at different domains; In national domain we can highlight Spanish Single Access Point Project [http://www.igsap.map.es/sgpro/conven/conven1.htm] which objective is to provide a system for interconnection among different public organisms easing the way the citizens “talk” to Public Administration. In a European scope, we must name SPACE project (Single Access Point for Citizens in Europe) which “aims to create one point of access to administrative services, allowing a civil servant to retrieve the necessary information from different sectors in another member state. To release this single point of access, the project will develop the necessary application to make the electronic transfer of the required information between members states possible. It will involve a range of different Ministries in members states: Interior, Health, Justice”.

5.5.1.2 Notary & property registry

So far, application of new technologies to Notary has been kept to automation of administrative processes and aid in preparation of documents elaborated to corroborate acts where physical presence of the notary and the parties is required.

The use of digital signature schemes in notary tasks will allow remote signing introducing a new electronic notary scheme where the notary digitally signs the documents once their contents have been verified. This new scheme needs the existence of a TSA to time stamp documents setting their creation, modification and/or signing date and time.

5.5.1.3 Copyright & intellectual property

Copyrights and Intellectual Property Rights are commonly related to the moment the author presents his/her work to the corresponding register. This allows discriminating among disputing parties under the basis of always awarding the first requester if equality is settled regarding other matters. In this sense, the TSA becomes of mandatory importance since it will provide trustful evidence on the date of protected work delivery.

On the other hand, TSS offers to librarians, bookkeepers, editors, publishers and authors of digitally supported works an effective (and cheap) tool to protect their work against unauthorised alteration.

5.5.1.4 Postal services

Postal services around the world provide to their users a certificate which enables citizens to make use of wide territorial networks to send over packets, letters, documents, etc. with the guarantee of a certificate that assures the date, time and location of sending. Telex and telegram services also provide time stamping of messages. It is therefore expected that such networks will be progressively substituted by the use of open networks as transport medium. This fact will make postal offices important users of TSS: a natural evolution in order to provide their traditional certificate services.

Regarding these matters, it is convenient to mention initiatives from USPS (United States Postal Service) that has recently announced its intention to provide an electronic postmarking service within The States, for e-mail and more generic electronic documents.
Yet another application within Post Service field comes along with digital postmarking techniques which are being developed in cryptographic domain and that are expected to be complemented with time stamping features (services).

5.5.1.5 Stock exchange

This is another potential user of TSS. Buy/sell orders supplied in electronic format must be time stamped independently from parties involved in transactions.

5.5.1.6 Justice administration/procurement

TS applications in this domain are countless. Since communication among courts and judicial agents (prosecutors, lawyers, juries, witnesses, judicial police, ...) are subject to strict terms that must be accomplished in its execution, electronic documents containing this information demand time stamps that assure their validity and integrity within the given terms. Furthermore, electronic documents regarded as evidences in civil/criminal trials are usually subject of doubt since authenticity and integrity is easily forgeable. TSS will eliminate this doubt making time stamped evidences stronger.

5.5.1.7 Public health care

The expected increase on the number of medical interventions made through telematics such as teleexploration, telediagnosis or teleanalysis will force public and private health systems to be

main users of TSS. In this sector, time stamping will provide security and coherence in diagnosis and images that would be inserted in historic records for each patient.

5.5.1.8 Police and internal security forces

TSS will strengthen evidence obtained by authorised recording used in crime prosecution.

5.5.2 Examples of Application in the Private Sector

Some examples follow covering foreseeable use by private companies and individuals.

5.5.2.1 Electronic commerce

There are a large number of scenarios of use of electronic commerce, all of them will require time-stamping services in order to reach an adequate level of security, and evidence to face potential disputes. These services will be usually bound with digital signatures for further guarantee in those cases where acceptance and non-repudiation of documents, when the secret key was revoked, either because of compromise or expiration, and still the documents are needed.

This is an added value to certificates issued by CAs, and applies as well in the public sector.

5.5.2.2 Teleworking

Distance working, without physical presence, by its own nature, requires the regular exchange of documents and messages via electronic mail or other electronic means. Working in groups usually requires confidence with respect to the identity, integrity, and the date and time of its creation. A time-stamping service becomes a must. Time-stamping may also be used to back proof of reception of documents developed remotely, and delivered on e-mail.

5.5.2.3 Teleconference and video-conference

These technologies aim to replace face-to-face meetings. In order to secure the conclusions reached along these sessions, and to avoid afterwards manipulation, it is necessary to time-stamp both the documents, and the audio and video exchanged between the participants. Time-stamped documents become the classical meeting minutes.

5.5.2.4 Transport

This is a very important sector in current economy. It is rather usual that contractual terms are referenced to delivery times, and to the satisfaction of time constraints along intermediate stages along the logistic chain, specially when different means of transport are involved. Time-stamping may be used to certify the pass across intermediate and extreme points.

5.5.2.5 Finances (banking and assurance)

The very own nature of the financing sector presents a large amount of opportunities for time-stamping services. Among others, let us mention its application in new payment protocols such as SET (Secure Electronic Transfers), and electronic money such as Millicent and Digicash.

In the assurance sector, there is another large amount of opportunities for TSA activities, as referred to on-line contracts, and assurance in general.

5.5.2.6 Broadcasting

There are many applications of time-stamping services in radio, TV, cable, satellite, and so on. Let’s mention those related to emission control, that has an impact on publicity costs and revenues, since the instant of emission makes a deep difference in its impact.

5.5.2.7 Other examples

Secure time-stamping is very often needed in industrial processes, and in scientific experiments. Also in high performance sport events, to register with precision the arrival time, or the exact occurrence of certain events (swimming, athletism, horse races, car races, etc.)

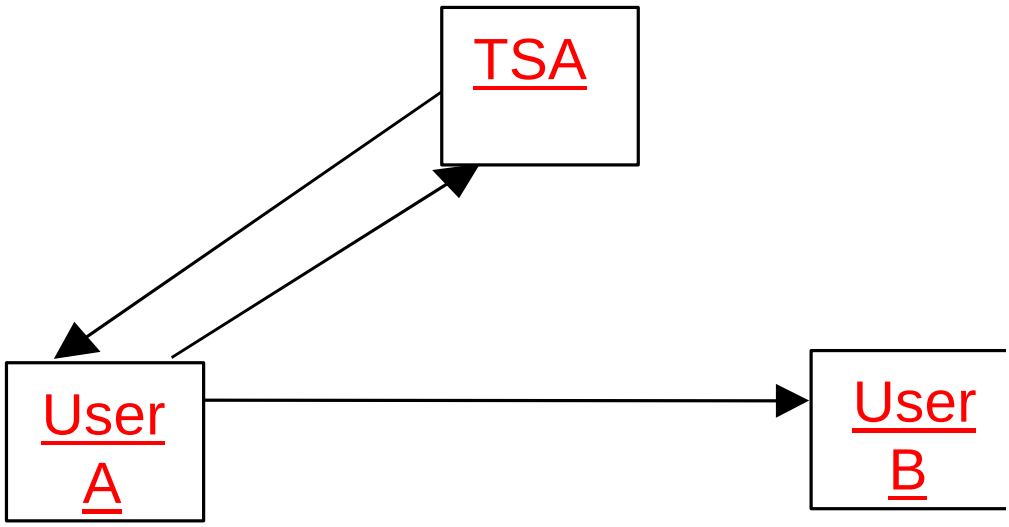
5.6 BASIC SCENARIOS

This section analyses scenarios of use from the functional point of view.

5.6.1 Non repudiation of origin

Actors: 2 users, 1 TSA

Description: User A sends a document to user B. Non-repudiation of origin is requested by user B. When a user A wants to be able to demonstrate the ownership, authorship or just the knowledge of some information at a given time, then its s/he who must contact the TSA, prior to the publishing of that information or document. This service may also be required by the recipient of the document or message, to use the proof of origin as a tool for non-repudiation of origin.



Procedure:

1. User A sends a signed hash of the document to TSA
2. TSA answers with the TS token
3. User A may distribute the document with the attached TS token to demonstrate its origin/ownership/authorship or whatever is required.

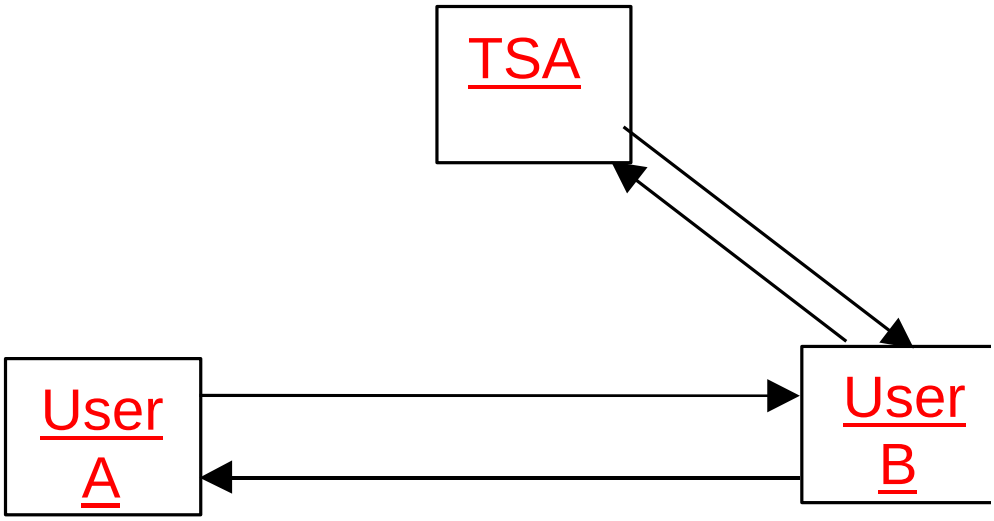
Applications:

- Summons.
- Legal ruling.
- Official communications.

5.6.2 Non repudiation of reception

Actors: 2 users, 1 TSA

Description: User A sends a document to user B. Non-repudiation of receipt is requested by user A.



Procedure:
This scenario corresponds to the situations where it is requested the non-repudiation of receipt of a document at a given time.

1. User A sends a Document to user B.
2. User B sends a hash of the document to TSA
3. TSA answers with a Token, including the hash and the time-stamp, as described in section 3.
4. User B sends an acknowledgement of receipt to user A, with the Token of its document, so that User A will have a proof of reception of Document by User B, who will not be able to repudiate the reception.

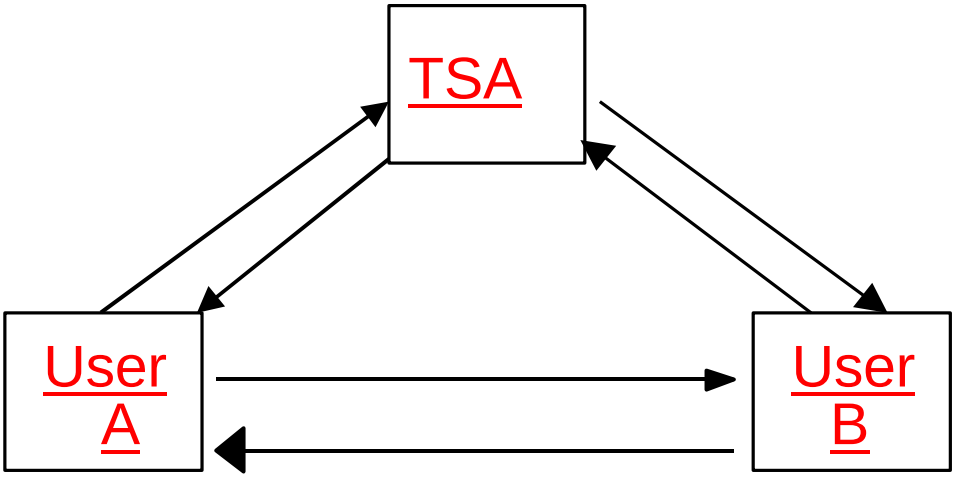
Applications:

- Grant request.
- Income tax return.
- Official certificates request.

5.6.3 Mutual non-repudiation

Actors: 2 users, 1 TSA

Description: When both users require non-repudiation of the other's action, a combination of the two previously described scenarios may be necessary.



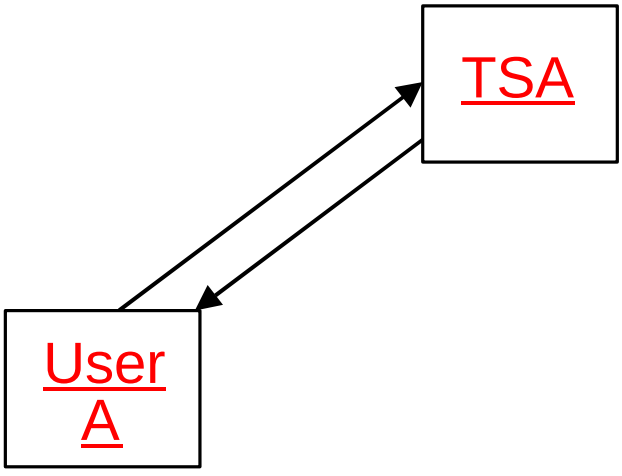
Procedure:

1. User A sends a (signed) hash of the document to TSA
2. TSA answers with the TS token
3. User A sends the document with the attached TS token to B, to demonstrate its origin.
4. User B sends a (signed) hash of the document to TSA
5. TSA answers with a Token, including the hash and the time-stamp, as described in section 3.
6. User B sends an acknowledgement of receipt to user A, with the Token of its document, so that User A will have a proof of reception of Document by User B, who will not be able to repudiate the reception.

In this case, the signature of the hash may not be needed, if the required proof makes reference only to the generation and reception of the message, with the attached document, in which case the whole message itself, may be signed.

5.6.4 Time-stamping in isolation

Actors: 1 user, 1 TSA
Description: A user wants to time-stamp a document for future, yet unknown use.

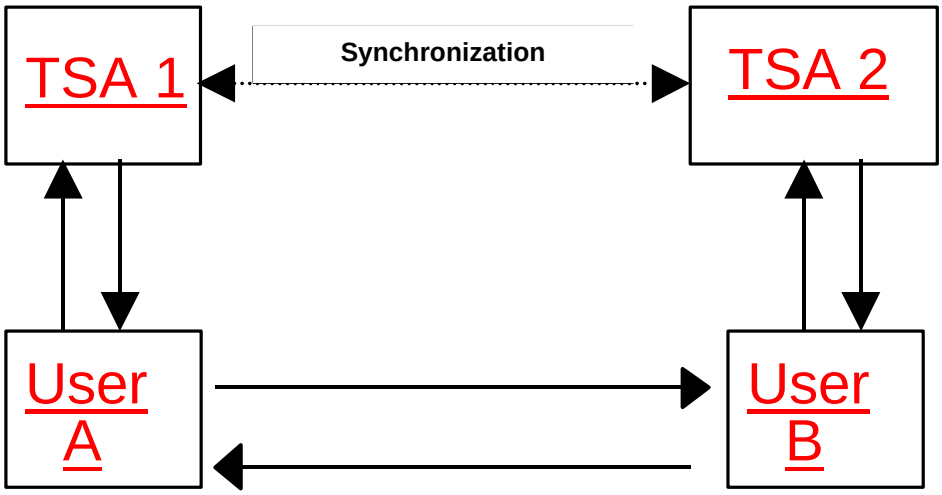


Procedure:

1. User A sends a (signed) hash of the document to TSA
2. TSA answers with the TS token
3. User A stores the document with the attached TS token

5.6.5 Use of two time-stamping authorities

Actors: 2 users, 2 TSA
Description: 2 users being clients of two different TSA need non-repudiation services.



Procedure:

1. User A sends a (signed) hash of the document to TSA1.
2. TSA1 answers with the TS token.
3. User A sends the document with the attached TS token to B, to demonstrate its origin.
4. User B checks TS token validity by sending it to TSA2
5. TSA2 checks it by using cross-certification with TSA1.
6. TSA2 answers with a Token, including the hash and the time-stamp, as described in section 3.
7. User B sends an acknowledgement of receipt to user A, with the Token of its document, so that User A will have a proof of reception of Document by User B, who will not be able to repudiate the reception.

5.7 REAL SCENARIOS

This section analyses situations from the point of view of service users.

5.7.1 When citizens submit documents to the Public Administration

This is a situation of mutual recognition where both parties store evidence.

There are a number of variants of this service:

- the citizen uses a notary authority or a registry authority
- the end recipient office has its own authority or not
- the citizen and the recipient office use the same TSA or not
- the authority closed to the citizen incorporates its own TS service or not
- the authority closed to the administration uses its own TS service or not

The different scenarios of authorities involved imply

- mutual recognition of separate authorities
- synchronisation of separate time-stamping authorities

Examples:

- tax payment
- submission for grants
- document deposit
- request of public documents (e.g. driver's license)
- use of public notaries

5.7.2 When private companies submit documents to the Public Administration

This is a similar scenario of mutual non-repudiation, where both parties store evidence.

But the variants of the service differ from the case of individuals when the private company runs its own authorities that must be recognised (asymmetrically) by the public authorities involved. this recognition may be direct (with respect to the local authority) or indirect (if the remote authority receives an authentity certificate issued by the local authority).

Examples:

- tax payment
- bidding for public procurement

5.7.3 When the Administration issues public documents

This is a case of time-stamping in isolation, where the public administration uses its authorities to notarise and time-stamp a document that becomes public.

Examples:

- official journal
- publication of laws
- publication of justice verdicts

5.7.4 Relations between Public Administrations

This is a case of mutual non-repudiation with mutual store of evidence.

The variants of the service are:

- both offices share a common authority or not
- if there is more than one authority, do they share a common TS service?

When there are different authorities, mutual recognition is a must.

When different TSA are used, mutual synchronisation is a must.

5.7.5 Trading between the public administration and private companies

This is an scenario that combines the submission of documents to the public administration, and the issue of public and private documents from the administration to the company. Public issue is typical of open procurement situations, while private issue is typical or negotiations, ordering, warranty claims, after-sales services, etc.

The private company mey run its own authority and time stamping service for internal recording of activity. The private TS service must be verifiably synchronised with the public service for its records to have legal value in dispute resolution.

5.7.6 Private trading

Private trading may be run without involving the public administration, or requiring a public notary authority. The election depends on the value of the trade, on applicable law, and on the desire of the parties of adding additional evidence to the trade.

There may be involved one (agreed), two (private) or three authorities involved, that must establish a relation of trust. This relation may be a tree, a connected graph, or a complete set of mutual cross-certifications.

If more than one TSA is involved, verifiable synchronisation becomes a must.

Examples:

- electronic commerce in general

5.7.7 Private contracts

Same as private trading, internal agreements may require external support.

Examples:

- private service contracts
- registry of deliverable reception
- employee activity records

5.7.8 Internal use

Any entity, either public or private, may use time-stamping services to add value to activity records and logs. The value of these records when disclosed to help in dispute resolution, depends heavily of the amount of synchronisation between the internal service and external time references.

Use of unpredictable events in a linking protocol may add sound evidence, since it makes hard or very hard to forge the normal system operation. The trustability of the service depends on the grain of trust, and the amount of service requests (a system with a low request rate is easy to falsify; while a high request rate in a linked system is hard to forge).

Synchronisation with an external TSA adds trust to the private service. Same considerations apply with respect to the density of service requests.

6 RISK ANALYSIS

Risk analysis introduces a rigorous and consistent approach to the analysis of the potential threats to and vulnerabilities of assets which comprise the architecture of the time stamping provision service and the scenarios of use, and to the determination of risks of non-availability or interruption, destruction, unauthorised disclosure or modification. It also assists in the identification of appropriate and justified specific security countermeasures.

This is carried out through a formal process of risk analysis using MAGERIT methodology [magerit]. Also guidelines [ISO 13335] have been followed.

By using a risk analysis methodology the detailed considerations of the relationship between assets identified, their importance to the system, the threats to which they are subject, vulnerabilities and adverse impacts leads to security practices and to the selection of effective safeguards commensurate with the assessed risks.

6.1 RISK MANAGEMENT PROCEDURE

Briefly, the main steps considered in the risk analysis procedure are the following:

1. Identification and valuation of assets.
2. Threat assessment.
3. Vulnerability assessment.
4. Adverse impact assessment.
5. Risks assessment.
6. Safeguards. Selection of security IT services.
7. Safeguards. Selection of security IT mechanisms.

6.2 OBJECTIVES

The objectives to be reached by this analysis are described below:

6.2.1 Main objectives

1. Introduce rational knowledge over security aspects related to the provision of the TSS and introduction of security measures in order to reach a given security level.
2. Reach a total coverage. Dealing with all TSS elements and studying each one with homogeneous in-depth review.
3. Provide embedded security mechanisms within TSS. Integration of security features into the TSS from the beginning.

6.2.2 Particular objectives

1. To ensure time stamped document integrity (If needed).
2. To provide evidence and take account of each event occurred in the system (document creation, message delivery, ...) on a given point in time (accountability).
3. To ensure indefinite validity for every time stamp issued.
4. To ensure integrity and availability of audit trails for possible use in court.
5. To ensure that the specified level of service is fulfilled.
6. To ensure that software used conforms to its specifications.

6.3 ASSETS

An asset is a component or part of the total system to which the organisation directly assigns value (to represent the level of importance to the operations of the organisation), and hence for which the organisation requires protection. Thus, assets are those resources which either compose the time stamping service or conform its environment.

Assets can be classified following different approaches. In this analysis the assets of the time stamping service have been grouped into five groups, namely:

1. The information system environment.
2. The information system itself.
3. Data handled by time stamping service.
4. System functionality. It requires subsystems to be in working order to be accomplished.
5. Other assets (trust, image, liability)

The classification of assets following this approach allows easily the identification of hierarchies and the definition of fault-trees. After the identification of assets risk analysis concentrates on assets which are perceived as being exposed to high risks or which are considered critical for the TSS.

Assets inventory:

1. The information system environment. It includes those assets required to ensure proper operation of assets included in other groups.
   - Personnel working in the provision of time stamping service.
   - Business premises and workshops.
   - Equipment and supplies.

2. Information system itself.
   - Hardware: At this level, hardware assets have not been identified yet.
   - Software: At this level, software assets have not been identified yet.
   - Communications. Own resources.
   - Technological environment of communication networks.

3. Data handled by time stamping service.
   - Documents to be time stamped: Data sent by system user.
   - Time stamped documents: Data sent to system user.
   - Stored information: Data contained within TSA databases, regarding either internal information or user documents.
   - Audit trails data.
   - The set of keys used by time stamping service (e.g. key pairs used by the TSA).
   - Other information: data structures, format, codes.

4. System functionality.
   - Time stamping service provision.
   - Clock provision mechanism (Time data provision)
   - Levels of service.
   - Response to all time stamping requests.
   - Linking of time stamped documents.
   - Verification of time stamping certificates produced by the TSA.
   - Renewal of time stamping certificates.
   - Digital signature scheme of the TSA to produce the time stamping certificates.
   - Hashing scheme of the time stamping service provision.
   - Secure time source used in the time stamping protocol (to know time source status by continuous coherence checks).
   - Quality of the time stamping service (accuracy of the source, response times).
   - Clock synchronisation.
   - Audit trail service.
   - Storage procedure.

- Publication of linking information.
- Documents reception.
- Universe of users. The set of people or electronic systems that will use the time stamping service as part of its activities:
  - Secure Time Authority.
  - Certification Authority.
  - Notary Authority.
  - Registry Authority.
  - Escrow Authority.
  - Other providers of time stamping services.
  - Citizens, businesses, administrations, other public or private organisations.

5. Other assets (we should be able to evaluate any risks identified in the system regardless its kind).
   - Liability.
   - Public Trust.
   - Image.

The valuation of assets is related to the loss of authentication, confidentiality, integrity and availability.

ASSETS	Authentication	Confidentiality	Integrity	Availability
Hashing scheme	critical	free	medium	less than an hour
Clock provision mechanism	high	protected	medium	less than an hour
Secure Time Source	high	protected	medium	less than an hour
The set of keys used by TSS	high	confidential	low	less than an hour
Clock synchronisation	critical	protected	medium	less than an hour
Universe of users	high	protected	high	less than a day
Time stamping service provision	critical	confidential	high	less than an hour
Time stamped documents	critical	confidential	medium	less than an hour
Documents to be time stamped	high	confidential	medium	less than an hour
Document reception	high	protected	medium	less than an hour
Credit	high	restricted	medium	less than a day
Public trust	high	restricted	medium	less than a day
Image	high	restricted	medium	less than a day

Authentication: critical, high, medium
Confidentiality: confidential, protected, free
Integrity: high, medium, low
Availability: less than an hour, less than a day, less than a week.

6.4 THREATS

Threats are possible sources of danger for the time stamping service. The realisation of a threat may result in loss of confidentiality, integrity, authentication or availability: disclosure, corruption, modification, temporary loss of service or permanent destruction.
There is a list of threats identified. Then there is a table that shows the relationships between specific threats to the time stamping service and the assets they would affect (see end of this chapter).

Threats have been classified in several categories:

1. Accidents:
   - Physical accident of industrial origin: fire, explosion, flood by pipe breaking, contamination, radio-electric emissions.
   - Breakdown: of logical or physical origin, caused by a built-in defect or happening while system operation.
   - Physical accident of natural origin: natural flood, earthquake, volcano eruption, electric storms, avalanche, landslide, collapse, ...
   - Essential services supply failure: electric supply, water supply, telecommunication breakdown, ...
   - Mechanical or electromagnetic accident: Impact, drop, strange body, electrostatic radiation, ...
2. Errors:
   - Usage mistakes while collecting, transmitting or processing data.
   - Design mistakes in software development phases.
   - Miss-routing sequencing and/or mistaken delivery of data streams.
   - Inadequate monitoring and/or registry of information traffic.
   - Loss of synchronism.
     TSA provides a kind of service that lies strongly on a trusted clock provision. This means that any deviation in clock used in time-stamping service may lead to a miss-function of the system in the sense that documents time stamped are not “so well” time-stamped. Synchronism is of vital importance in critical time-stamping services such as a surveying video time-stamping.
   - Time provision service malfunction.
     In the case of malfunction of the source of time, Time Stamping service can no longer be provided. This is an essential asset to be protected. The system is completely inoperative in the absence of it.
   - Key Loss.
     Whenever time stamping service relies on a PKI, an important risk arises since private key can be lost or stolen.
   - Databases Corruption or deterioration.
3. Local deliberate threats or Local malicious attacks:
   - Unauthorised physical access causing system malfunction by destruction or theft.
   - Unauthorised logical access leading to passive eavesdropping.
   - Unauthorised logical access leading to modifying/theft of either transit data or set up information, generation of a false time-stamp, manipulation of data to be time-stamped before and during time-stamping.
   - Unauthorised logical access leading to corruption/destruction of either transit data or set up information.
   - Lack of technical/human resources availability (strike, ...) that prevents the delivering of a time-stamp, the recording of time-stamping events.
4. Remote deliberate threats or remote malicious attacks:
   - Unauthorised logical access leading to passive eavesdropping.
   - Unauthorised logical access leading to corruption/destruction of either transit data or set up information.
   - Unauthorised logical access leading to modifying transit data, generation of a false time-stamp, manipulation of data to be time-stamped before and during time-stamping.
   - Faked sender or identity forgery (source impersonating).
   - Sender repudiation or data reception repudiation.
   - Hash function break.

6.5 RISK SCENARIOS

The risk analysis strategy selected focuses on risk scenarios critical to the time stamping service. For each scenario there is a description, detailed considerations of assets, threats, vulnerability, impact and risk, and a number of preventive and corrective safeguards.

Preventive safeguards reduce vulnerability whereas corrective safeguards reduce impact.

6.5.1 Desynchronisation

The service provided by a given TSA is allowed to differ an specified "delta" from the "real time" from other TSA. The given delta will be clearly stated in TSPS and/or Level of service TSA statement. When difference between our time source and time delivered by synchronisation procedure [3.3.1.4, 3.3.2.4 and 3.3.3.4 Synchronisation] goes beyond that "delta" we can talk about desynchronisation. Desynchronisation can be a problem itself or point out a more serious problem such as time source malfunction.

- Damaged assets:
  - A1: Documents to be time stamped.
  - A2: Time stamped documents.
  - A3: Clock provision mechanism.
  - A4: Document reception.
  - A5: Credit.
  - A6: Public trust.
  - A7: Image

- Threat: Loss of synchronism

- Vulnerability: Desynchronisation could occasionally happen.

- Impact level: Moderate loss because of:
  - Non-availability.
  - Disturbance or political-administrative embarrassing situation (e.g. prestige, credibility gap, distrust, ...)

- Risk level: Low

- Preventive Measures:
  1. Synchronisation procedure must be launched periodically and its frequency must be high "enough" to detect and correct possible deviations.
  2. Synchronisation edges must be chosen in a pseudo-random way (v.g.: Derived from last_hash_value modulus num_of_TSA). This is so to prevent from collusion among given TSA.
  3. Synchronisation operations must be recorded in audit trail.

- Corrective measures:
  Once desynchronisation has been detected:
  1. Event must be recorded in audit trail.
  2. TSA must not issue any time stamp from the detection moment.
  3. TSA must determine the cause of desynchronisation (temporal or malfunction).
  4. TSA and CA must revoke certificates from the date of the desynchronisation used to time stamp.

6.5.2 Time Provision Service malfunction

Time source must be regarded secure [Appendix D] and its management is a TSA's liability [7.3 Liability]. Eventually, this trusted secure source can fail or go inoperative.

- Damaged assets:
  A1: Documents to be time stamped.
  A2: Time stamped documents.
  A3: Clock provision mechanism.
  A4: Document reception.
  A5: Credit.
  A6: Public trust.
  A7: Image.

- Threat: Time provision service malfunction

- Vulnerability: Time provision service malfunction could exceptionally happen.

- Impact level: Large loss because of:
  - Non-availability.
  - Disturbance or political-administrative embarrassing situation (e.g. prestige, credibility gap, distrust, ...)

- Risk level: Medium.

- Preventive Measures:
  TSA must fulfil certain requirements regarding security on the link between the time source and TSA [7.3 Liability]:
  1. Access to time source receptor must be strictly restricted and placed in a secure location.
  2. Receptor location must provide a stable signal reception and avoid shade or dark reception zones.
  3. TSA must be provided with a mechanism that allows detection of time deviations within small intervals. Such devices could be atomic clocks or any other high precision time measurements.
  4. Duplication of valid time sources (v.g. ntp & gps). This way we assure that if one source fails we can rely temporarily in the other. Note that this requires a mechanism allowing us to detect deviations. This is an additional feature regarding desynchronisation.
  5. Testing and evaluation of components used to detect time deviations and to combine the time with the data to be time stamped.

- Corrective measures:
  In case of time source failure:
  1. TSA must not issue any time stamp from the detection instant.
  2. TSA is not required to respond any request in any way.
  3. Detection time must be recorded in audit trail.
  4. Any request must be recorded in audit trail but not replied in any way.

  On time source recovery:
  1. Event must be recorded in audit trail.
  2. Synchronisation procedure must be launched in order to synchronise with other TSA's (proof of normal operation).
  3. TDS request must be issued.

4. When TDS token arrives, link backwards the last valid time stamp must be established (via hash function) and normal operation can be restarted.

6.5.3 Key Compromised

The TSA private key can be compromised. If this occurs, TSA identity can be forged and so can time certificates.

- Damaged assets:
  A1: Documents to be time stamped.
  A2: Key pairs used by the TSA.
  A3: Digital signature procedure.
  A4: Document reception.
  A5: Credit.
  A6: Public trust.
  A7: Image.

- Threat: Key loss

- Vulnerability: Key compromising could exceptionally happen.

- Impact level: Large loss because of:
  - Non-availability.
  - Disturbance or political-administrative embarrassing situation (e.g. prestige, credibility gap, distrust, ...)
  - Non-physical elements recovery (procedures, documentation, ...)

- Risk level: Medium.

- Preventive Measures:
  1. TSA must be provided with a special security length key for time stamp tokens signing purposes. The length of the key must be secure "enough" regarding the state of art for signature forgery.
  2. TSA signing key lifetime must be restricted to a given "short" period of time. This prevents from forgery since the longer a key is operative the easier it is to forge.
  3. TSA signing key lifetime must be restricted to a maximum number of digital signature operations.
  4. When a key is considered insecure or its lifetime has expired, TSA must record the event in its audit trail.
  5. When a key is considered insecure or its lifetime has expired, TSA must "time stamp" the fact.
  6. When a key is considered insecure or its lifetime has expired, TSA must notify that to the corresponding CA and an ARL must be issued.
  7. Testing and evaluation of components used to generate signatures for time-stamps.

6.5.4 Breaking of the hash function

Hash function can be broken. This means that *hackers* can reach a hash from a document different of the original one.

A collision for the compression function consists of an arbitrary H_{i-1} and two different inputs X_i and X_i' such that:

$$f(X_i, H_{i-1}) = f(X_i', H_{i-1})$$

Some attacks find *pseudo collisions* for the compression function. This is a collision for two different chaining variables H_{i-1} and H'_{i-1} and two (possibly equal) inputs X_i and X_i':

$$f(X_i, H_{i-1}) = f(X_i', H'_{i-1})$$

Although this doesn't bring us closer to finding collisions for the hash function, it is considered as a weakness of the compression function and a failure of one of the design principles.

When the hash values of two messages differ in only a few bits we speak of a *near-collision*.

To invert a one-way hash function the opponent can select random messages and check if they are hashed to the given value. For T trials and a hash code of bit length n the probability of success equals $T/2^n$.

Collision resistant hash functions are susceptible to a **birthday attack.** If the opponent calculates the n-bit hash value for $2^{n/2}$ arguments, the probability of a collision equals 63%.

In practice these attacks are impossible if the bit length of the hash code is sufficiently large. For one-way hash functions a length of 64 to 80 bits is required. If the function has to be collision resistant (e.g. in Time Stamping Service) one needs a hash code of 128 to 160 bits.

- Damaged assets:
  - A1: Documents to be time stamped.
  - A2: Time-stamped documents.
  - A3: Hashing algorithms.
  - A4: Document reception.
  - A5: Credit.
  - A6: Public trust.
  - A7: Image.

- Threat: Hash function break

- Vulnerability: Hash function breaking could exceptionally happen.

- Impact level: Destructive loss because of:
  - Non-availability.
  - Disturbance or political-administrative embarrassing situation (e.g. prestige, credibility gap, distrust, ...)
  - Documents confidentiality and integrity.

- Risk level: High.

- Preventive Measures:
  1. In order to increase hash function security, TSA must renew time stamped documents with an updated hash function when the state of art requires so.
  2. TSA must check if the hash function is stated in its TSPS. If not, TSA should not accept time stamp request.
  3. Users could use simultaneously more than one hash of the same document, and send them as a request to the TSA.
  4. Users could use only one of some hash functions stated in the TSPS of the TSA.

5. When hashing a document we could use a padding procedure that adds the length of the message. This prevents opponents from finding collisions for messages of different length.
6. Testing and evaluation of components used in hash function processing.

- Corrective measures
  1. Event must be recorded in audit trail.

6.5.5 Burst of time-stamp requests

TSA could receive request burst coming from the same user. This means that user could try to:

a) saturate TSA activity.
b) forge time stamp of some documents: If the user send a lot of request (request burst), he can forge the time stamp of a document since he possesses documents that go before and after it.

It's necessary to define the maximum number of documents (N) belonging to the same user that TSA should accept.

- Damaged assets:
  A1: Documents to be time stamped.
  A2: Time stamped documents.
  A3: Document reception.
  A4: Credit.
  A5: Public trust.
  A6: Image.

- Threat: Burst attack

- Vulnerability: Request burst could occasionally happen.

- Impact level: Moderate loss because of:
  - Non-availability.
  - Disturbance or political-administrative embarrassing situation (e.g. prestige, credibility gap, distrust,...)
  - System malfunction.

- Risk level: Low.

- Preventive Measures:
  1. TSA should insert the hash of an unpredictable event every N documents.
  2. TSA should insert the hash of an unpredictable event periodically.

- Corrective measures
  1. Event must be recorded in audit trail.

6.5.6 Data Corruption

Data stored in databases may be corrupted. There are two possibilities:

1. Corruption of present linking information. If this occurs no more documents can be added to linking chain until the value is recovered.

   - Damaged assets:

A1: Documents to be time stamped.
A2: Document reception.
A3: Credit.
A4: Public trust.
A5: Image.

- Threat: Data corruption.

- Vulnerability: Data corruption of present linking information could occasionally happen.

- Impact degree: Limited loss because of:
  - Non-availability.
  - Disturbance or political-administrative embarrassing situation (e.g. prestige, credibility gap, distrust,...)
  - System malfunction.

- Low risk degree.

- Preventive Measures:
  1. No preventive measures accounted.

- Corrective measures
  1. Reconstructing linking information starting from last hash value and last request.
  2. TSA must not issue any time stamp from the detection instant.
  3. Events must be recorded in audit trail.

2. Corruption of historic linking information. This is a fatal failure.

- Damaged assets:
  A1: Documents to be time stamped.
  A2: Time stamped documents.
  A3: Document reception.
  A4: Credit.
  A5: Public trust.
  A6: Image.

- Threat: Data corruption.

- Vulnerability: Data corruption of historic linking information could exceptionally happen.

- Impact degree: Large loss because of:
  - Non-availability.
  - Disturbance or political-administrative embarrassing situation (e.g. prestige, credibility gap, distrust,...)
  - System malfunction.
  - System corruption.

- Large risk degree.

- Preventive Measures:
  1. Making backups of historic information every time.

- Corrective measures
  1. Event must be recorded in audit trail.

6.6 SAFEGUARDS. TECHNICAL AND NON-TECHNICAL SAFEGUARDS

First of all, there must be a security policy of the time-stamping provision service that should include at least the following elements: security objectives and strategy, directives and procedures, roles and responsibilities, risk management, trustworthiness and assurance, safeguards implementation, awareness training, security compliance checking, incident handling, accountability, legal obligations, ethics, standards, auditing and proportionality.

Specific preventive and corrective measures have been selected for critical risk scenarios. Now there is a list of main technical and non-technical security services and mechanisms.

6.6.1 Preventive Measures

- Uninterrupted power supply (UPS) systems preventing shut-downs and other natural or industrial accidents. By means of this we assure continuous system operation against electric services supply failure.
- Physical countermeasures. Subsystems providing service must be protected from theft, fire or other physical damage.
- Preventive safeguards such as personnel education & formation or guide-lines issuing.
- Liabilities. Events granting rights to attend service request will be recorded in audit trail.
- Fault tolerant systems.
- Use of a secure operating system.
- Object re-usability. Replace or remove operation of unnecessary subsystems must be carried out in a secure module based procedure, decreasing risks to sensible data.
- Access control. Every information system must have an access control mechanism to permit only access to trusted personnel.
- Redundancy of access channels.
- Operation control logs: Once the user has been authorised to enter the system, it should be advisable to record his/her actions within the system and revert this info to a file. This file can be later examined by security services.
- Audits. Failed access trials either to the application or databases files will be recorded.
- Recording of time-stamping events.
- Authentication & identification
  - Internal:
  - Identification of operation personnel
  - Identification of maintenance personnel.
  - Identification of security personnel.
  - External:
  - Authentication of entity/user requesting service.
- Data exchange. Communication protocols used should be provide confidentiality features, error detection & correction measures in order to verify message integrity.
- Key data expiration control. No key should have an undefined lifetime. Keys should expire automatically because:

- ✓ If the key is frequently used, then there is a notorious possibility to compromise key.
- ✓ It's "easier" to obtain a secret key when there are a lot of documents encrypted with the same key.

- Testing and evaluation of components employed in the provision of the time-stamping service.

6.6.2 Corrective Measures

- Periodical synchronisation according to any standard clock. Time Stamping mechanism is based in a date stamp, that must come from a trusted clock, and that could be in the proper TSA. For this reason, TSA must periodically proceed to synchronise its clock so that the existing deviations between the different TSA will be minimal and invaluable.
- Correction, restoration and curative detection.

Assets / threats:

	Desynchronisation	Time provision service malfunction	Key compromised	Hash breaking	Request burst	Data corruption	
						(1)	(2)
Clock provision mechanism	✓	✓					
Time Stamped documents	✓	✓		✓	✓		✓
Documents to be time stamped	✓	✓	✓	✓	✓	✓	✓
Document reception	✓	✓	✓	✓	✓	✓	✓
Hashing algorithms				✓			
Key pairs used by the TSA			✓				
Digital signature procedure			✓				
Credit	✓	✓	✓	✓	✓	✓	✓
Public trust	✓	✓	✓	✓	✓	✓	✓
Image	✓	✓	✓	✓	✓	✓	✓

Countermeasures / threats:

	Desynchronisation	Time provision service malfunction	Key compromised	Hash breaking	Request burst	Data corruption	
						(1)	(2)
Synchronisation	✓	✓					
Event recorded in audit trail	✓	✓	✓	✓	✓	✓	✓
Stop issuing time stamps.	✓	✓				✓	
Duplication of valid time sources	✓	✓					
Issue TDS request		✓			✓		
Secure length of the keys			✓				
TSA signing key lifetime restricted			✓				
Renew time stamped documents	✓	✓		✓			
Use more than one hash				✓			
Add message length to the hash				✓			
Reconstructing linking information						✓	
Making backups of historic							✓
Testing and evaluation of components		✓	✓	✓			

7 MISCELLANEA

7.1 LEVELS OF SERVICE

The quality of the time-stamping service provided to the user depends on multiple architectural, functional and non-functional factors. This section will introduce those aspects significant to TS Service quality and show the way of building different levels of service by constructing layers with added value and functionality to the service.

Quality of service will be analysed first from the point of view of the user, then from the point of view of the means needed to provide the service. Lastly, three levels of quality are drafted, just to exemplify possible implementations of the service.

7.1.1 User's Requirements

Usual parameter define the perceptible quality of service:

1.Reliability.

That should be high. It is measured in terms of service availability, and low rate of service denial (excluded incorrect requests)

2.Price.

That should be balanced to the other quality of service parameters, and TSA assumed liability.

3.Speed.

That should be high, measured in terms of response time for time-stamp provision, and for verification.

4.Accuracy.

It is measured in terms of time grain. The smaller the grain of time (guaranteed precision of time tokens), the best.

5.Availability.

That should be high. It is measured as the rate of service denials for reasons that are different from errors in the request.

6.Lasting certificates.

The validity period for time stamps should be as long as possible. Renewal process is regarded as uncomfortable to the user.

7.Ease of use.

No special abilities should be required to the user and software (if needed) should be provided.

8.Minimum equipment requirements (storage...).

Users should have no requirement on extra equipment, and software.

7.1.1.1 Interactivity

The interactivity degree of stamps procurement/verification is important to the level of service and is greatly related to transport mechanism used to communicate with TSA.

Interactive service could be achieved using HTTP protocol (e.g. WWW forms with on line response). MIME objects would be transported using common HTTP processing engines over WWW links.

On-line service provided by socket based protocols (e.g. sockets interface) are also fast and offer a good response time.

Batch services could be based in file based protocols (e.g. ftp) or mail protocols (SMTP) and allow for off-line signing.

7.1.1.2 Time-stamps renewal advise

It is a matter of level of service to define the policy for stamps renewal. Basic services will yield responsibility of renewal to the user, this means that an already time-stamped document may be subject of renewal upon user's request. A higher level of service could be provided by informing clients about expiration of time stamps, and offering renewal with/without charge.

7.1.1.3 TSA Knowledge

TSA knowledge degree [7.2 Knowledge] is an important element in the construction of the service to be provided. The control (knowledge) over the elements of the service is always of benefit to the user. That information can be made available to the user upon request increasing greatly the level of service. This is mandatory in some situations such as verification procedure used in the linking protocol [Appendix C], when the user must have access to the K previous/posterior hash values to prove his receipt validity. On other hand, providing means to monitor TSA activity entails a best service and trust promotion among users.

7.1.1.4 TSA operating protocol

The operating protocol chosen will affect in a very significant way the level of service that a given TSA is able to provide. Implementing the Surety® scheme [Appendix C] (hash trees linked by the root), in other words, using some variation of the protocol "by rounds" allows the TSA to cope with a great number of requests at an instant in time. However, a problem arises for there is no order-relationship among time-stamps within a single round. All documents within a single round have the same time assigned. This should not be a problem if the round is "short enough"(e.g. half a second). Therefore we are forced to implement a very fast round service to achieve a certain satisfactory service level. Even if the round is too short, critical situations arising (e.g. two documents regarding a copyright) could result in a serious damage for TSA [7.4 Liability].

If we decide to implement some variation of the linking protocol, the problem of relationships between any two given documents time stamped by the same TSA is out of the question. In exchange, the protocol is slower since we cannot take advantage from parallel processing. Further discussion is required to determine temporal relationship between two documents time-stamped by different TSA [3.2.2.4 Linking Protocol Synchronisation]. Another drawback to this protocol is the fact that collaboration among TSA users is required in verification procedure. This forces TSA to authenticate users and to store some information about them decreasing response time.

The basic protocol offers a quick response time. The request is not linked to the previous one and therefore there is no requirement on the storage of hashes or requests. It is also cheaper since less assets are to be protected. In exchange, its dependency on Key pairs is extreme; If key is somehow compromised, even if its usage period has expired, Time stamps can be forged easily and great damage can be done to TSA and its users. This solution accepts a great risk rate and does not offer the minimum security level for long lifetime time-stamps, but could be suitable for cheap short life time-stamps.

Distributed protocols enforces a uniform level of service among UoC since random election of the certifier could impose a “not desired” level of service to the user. This protocol will lead to definition of *Groups of Certifiers* providing an equal level of service.

7.1.1.5 TSA operating environment

On one hand, it is clear that TSA services are not “stand alone” services and conform important service clusters in Integration with other PKI functions such as time stamping revocation dates. In this direction, TSA can be embedded within other authorities and adapt the level of service provided to them.

On the other hand, different TSA could operate in different scenarios of use. The requirements identified in those scenarios determine the level of service to be provided:

A TSA operating within a public organism registry is not required to provide high precision timing (e.g.: a minute round will be enough) nor to establish temporal relationship among documents within a round for they regard the same subject (e.g: tax payment) and will not be dispute.

When operating within a NA, when human participation is required, level of service regarding response time is not so important, if the document is valid will be delivered after a period of time specified in NA PS (practice statement).

7.1.1.6 TSA security features

Security practices mean in most cases an increase of level of service. The higher security level the TSA offers to its clients the stronger the evidence provided by it will be in case of dispute.

- The number of events recorded in audit trail and links with other Authorities make evidences stronger.

- The use of more than one hash in the request (e.g. MD-5 & SHA-1) is a security mean that has double-side effect: on one hand it decreases user satisfaction since he/she must run two hash algorithms (time lost) but on the other hand, time stamps issued this way obtain a much longer lifetime before renewal. This additional security feature also provides the system with legal recognition from court regarding the system as securer than others.

- Off-line signing is a common security practice to protect private key, this increases TSA security but forces a longer response time since the key is not available On-line.

7.1.1.7 TSA storage liability

It is a TSA requirement to deliver means to the user so that he/she can prove time stamps validity to the rest of the universe for ever. However, it is a matter of the level of service to state the storage liability for the TSA. TSA should store information a given period of time as stated in CPS. When that period expires, the TSA may keep information, inform users or just throw it away.

Another matter related to the service is whether the TSA is to store the receipt once it has been signed and sent to the requester. In this case TSA liability [7.3 Liability] is affected. If not, then a tool for discharging TSA responsibility is required. I mean that the TSA must receive from requester a receipt acknowledgement. A non-repudiation service is therefore required.

7.1.2 Provision of service

The Time stamping service can be provided in a number of different ways. Aspects involved in the provision of the service are listed below:

7.1.2.1 Secure Time source

The level of service strongly depends on clock precision. Clock regards service accuracy and availability. Several time sources are available [Appendix D. Secure Time], the choice of one of them determines a quality/costs relationship:

- Satellite
- Radio
- Net
- Atomic clocks

Satellite and Radio systems offer high accuracy and allow synchronising. Its use leads to strategic positioning of the TSA for a satisfactory signal reception. Nevertheless benefits, occasionally systems can go out of service (e.g. military conflicts, critical atmospheric conditions, ...). Another option can consist of a high precision clock (e.g. atomic clocks) and synchronise it every "long" interval (stated in TSA CPS [7.3 Liability] and depending on clock quality). In order to increase availability, two secure time sources can be used so that if one fails, the service will be provided in a normal way.

If time is distributed to the TSA through the net via some secure protocol (e.g. NTP), quality of service is decreased since time accuracy is lower. But it is also cheaper.

Service accuracy results affected by the clock precision. A "delta" must be defined as the maximum deviation from "real time". "Delta" size determines quality of service regarding accuracy.

7.1.2.2 TSA processing power

Process capacity of TSA equipment is essential in TSS activity. Cryptographic operations such as Digital signature algorithms or hash algorithms need a given process power in order to provide a fast service and to prevent from queuing situations.

7.1.2.3 Operational Security

TSA must provide a given security level in its operation:

- Physical.
- Documental (storage of evidence).
- Personnel.
- Back-up policy.
- Provision for disaster recovery...

Security practices get the service dearer and therefore modify the cost/quality relationship that is an important parameter in level of service estimation.

7.1.3 Service Profiles

7.1.3.1 Low-end service

A low-end service could be defined as follows:

- Using naïve protocol (digitally signed non-linked time stamps).
- Off-line signing for security means.

7.1.3.1.1 Benefits

- Fast
- Cheap

7.1.3.1.2 Limitations

- There is no authentication on user's identity and therefore it is subject to attacks (spoofing).
- TSA receipt has limited legal validity in time. After a short key verification period [9.1 X.509], time certificates signed with the given key are no longer valid.
- Renewal is not available.
- Colluding between client and TSA is feasible and therefore there are not trust basis.
- Weakness in case of attack, no countermeasures can be accounted if key becomes compromised.

7.1.3.2 Medium quality service

Medium quality could be achieved by providing some variation of the linking protocol or "by-rounds" protocol and some kind of secure notice board. TSA publishes link information in a secure publishing board. The way the notice board is changed and the period that "news" remain in the board must be defined within the level of service statement. Notice that this also affects TSA liability for it is responsible for maintaining the board as stated in specifications.

7.1.3.2.1 Benefits

- High Trust Level. The linking between documents provide a temporal chain increasing security against stamp forgery and turning operation auditable.
- No key is used and therefore there is no risk of loss/theft.
- Renewal is available upon user's request.
- Quite fast.

7.1.3.2.2 Limitations

- User and TSA are not authenticated, therefore ownership and receipt validity are subject to attacks.
- User receives TSA receipt which has no legal validity as long as it is not signed. User may want to consult the notice board and see if his request has effectively arrived.

7.1.3.3 Superior quality service

Superior service could be regarded as an implementation of a linking or "by-rounds" protocol along with added value and functionality. In this case On-line signing of receipts is provided to support a fast response time and comfort to the client (no need to consult on notice boards). The liability for stamps renewal lies on the TSA; it would notify its clients for renewal purposes.

7.1.3.3.1 Benefits

- User is authenticated and therefore ownership of time stamp is clearly stated.
- TSA is authenticated and user cannot be fooled by a "man-in-the-middle".
- High Trust Level. The linking between documents provide a temporal chain increasing security against stamp forgery and turning operation auditable.
- Fast response time.
- Renewal available. TSA notification.

7.1.3.3.2 Limitations

- Authentication of users slows down TSA operation.

7.2 KNOWLEDGE

Knowledge is of two kinds.
We know a subject ourselves,
or we know where we can
find information on it.
Samuel Johnson

This section refers to the amount of information that TSA should deal with. TSA knowledge about elements involved directly in its operation is essential to protect itself against threats [6.1 Threats] and provide and appropriate level of service to its users [7.1 Levels of service]. It is clear that not regarding information speeds up TSA operation but introduces a higher risk rate over TSA operation.

We identify the elements subject of knowledge present in every TSA architecture:

7.2.1 Secure time provision

It is one of TSA liabilities to **know** time source status by continuous coherence checks. The quality of time provision service **could** be increased by a "sufficiently" secure protocol between TSA and time source.

K1	The TSPS **shall** specify the origin and nature of its time source.
K2	The TSPS **shall** specify date/time format within its receipts.
K3	The TSA **should** provide means in order to check status of time source and detect deviations.

7.2.2 Time data provision

TSA **must** know where to find time data whenever necessary. Unpredictable information **must** be available and a secure protocol allowing its recovery in every moment.

K4	The TSPS **shall** specify the origin and nature of unpredictable events within the link chain, as well as the number of unrelated bits used to encode these events.

7.2.3 Security primitives

TSA **knowledge** on the state of art on cryptology is mandatory in order to launch preventive measures in case that progress on that field threatens TSA assets and credibility.

Hash functions used for calculating message digests should be **known** by TSA so that they can be regarded as valid or not.

K5	The TSPS **shall** specify hash algorithms that shall be accepted in the corresponding request field, as well as the formal identifiers of them.

7.2.4 Data submitted to the TSA

Data submitted to the TSA is out of the question. Standardisation initiatives are being carried out in this direction. It is clear that some unique representation of the original document is required. An identification of the requester is also required. So far, we have defined "blind" time stamping, TSA does not know anything about the original document, that job is for the NA.

Therefore a usual TSA's client would be a NA. At this stage, it is important to point out some dangerous situations derived from this operation scheme:

e.g.: It is feasible that a scientist comes to a point where his/her work follows two different paths depending on different premises which validity he/she is not able to demonstrate. In this case he/she can describe his conclusions in two completely different documents. It is clear that at least on of them is completely erroneous. However this does not worry him/her since he/she will send the digest of both of them to a TSA and wait till the correct answer comes out to light in order to claim for his/her rights.

It is important to notice that cases like the one depicted here constitute an important argument to put in against operation of "a bare TSA". TSA operation is best understood embedded in meaningful authority clusters as defined in Chapter 4 [4.4 Impact...].

7.2.5 Requesting entity

Requesting entity **should** be authenticated and recorded in audit trail. This is required (**shall**) in the linking protocol in which the assurance of identities is essential. On the other hand, payment to the TSA must be addressed to someone, and this cannot be achieved if there is no signed request.

K6	The TSA **shall** authenticate requester making use of CA services or any other authentication techniques in those protocols where identity of the requester is essential (v.g., centralised linking protocol).

7.2.6 Linking information

The TSA is responsible for maintaining the linking information, If this value is lost [6 Risk Analysis] serious liability is derived form this fact.

K7	The TSA **shall** provide adequate countermeasures to preserve linking information from corruption, loss or theft.

7.2.7 Random/Deterministic function

When TSA operates as part of a distributed time-stamping infrastructure, it is to **know** the random/deterministic function used by TSS users to determine which TS agents among UoC [3.2.2 Distributed Time Stamping] shall attend their request. TSPS must specify that function and receipts shall make reference to it.

K8	The TSPS **shall** specify random mechanism that **shall** be used by users to choose among TS agents within a distributed TSS environment.

7.2.8 Evidence

The TSA must maintain **knowledge** on evidences generated for verification purposes. Here knowledge directly translates to storage, TSA shall **know** how to prove validity for time stamps and provide that knowledge to the user in order for him/her to claim for related rights. This kind of knowledge must be kept for a given period of time specified in the TS Level of Service Statement [7.1 Levels of service].

K9	The TSPS **shall** specify term limiting mandatory evidence storage.
K10	The TSA **shall** kept information (evidences) for verification purposes as stated in TSPS.

7.3 LIABILITY

The operation of a TSA is critical and sensible to complaints due to system malfunctions, technical flaws and operation defects, even if these situations **should** occur very seldom. Transactions involving time stamping services are usually related with activities that, in those cases, are likely to end in a situation where the TSA must deal with liability issues.

To define in what cases the TSA **shall** or **shall** not be liable for any problem and must answer to a demand is rather difficult and out of the scope of this document. However, we will define what are the practices that **should** be followed by any entity delivering a Time-Stamping Service in order to ensure that its liability will be limited.

7.3.1 Time Stamping Practice Statement

Any entity delivering a Time Stamping Service **shall** have a Time Stamping Practice Statement (TSPS). In case the TSA is part of another type of Authority (e.g., a CA or a NA) this TSPS **shall** be a subset of the Practice Statement corresponding to that Authority, i.e. the Certification Practice Statement (CPS) or Notarisation Practice Statement (NPS).

The TSPS **shall** define all the points related to:

- Service Provision and Level of Service.
- Security Practice.

The Time Stamping Practice Statement **shall** consider (but need not be limited to) the following Practice Statements (PS*i*)

Service Provision and Level of Service.

PS1	*Cryptographic primitives.* The TSPS **should** enumerate the cryptographic primitives used by the TSA (hash functions, or public key signature schemes, etc. used).
PS2	*Duration of the pair of keys used to time-stamp*. (6.3 Countermeasures). When a key is used to sign a large number of messages the risk of having it compromised grows. The key used to time-stamp messages must be changed every specific period or every specific number of messages time-stamped. This period of time or number of messages **shall** be stated in the TSPS.
PS3	*Publicity of evidences.* TSA **shall** specify the means to made evidence material available to service users. This includes access procedures, response time and timeliness of information.
PS4	*Period of storage of evidences.* The period of time during which the time-stamped evidences will be maintained in the TSA shall be stated in the TSPS.
PS5	*International standards applicable to the Time Stamping Service Provision followed by the TSA.* This item is important for the interoperability and synchronisation with other TSA. Actually there is no standard applicable to this subject.
PS6	*Communication channel with users.* TSA **shall** state the communications channels available for the communications between the TSA and its users.
PS7	*Identification and Authentication Procedures.* TSA **shall** state the methods to identify and authenticate the intervening parties in the time-stamping protocols.
PS8	*Secure Time Source.* TSA **shall** state the source from which it obtains the time reference and the procedures to avoid deviations from this time base.
PS9	*Accuracy of the time-stamping service.* TSA **shall** state the precision (or precisions in case there are more than one alternative) that can be expected in the time stamping service according to the level of services agreed with the TSA.

7.3.1.1.1 Security Practice

Most of these practice statements are shared with those of a CA or a NA and are related to the high security nature of these type of services.

The Security Practice Statements shall consider:

PS10	*Identification and authentication practices.* TSA **shall** state how all the intervening parties (users or other authorities) shall be identified and authenticated.
PS11	*Non Technical Security Practices.* TSA **shall** use non technical security controls to deliver its services in a secure way. Physical, personnel and procedural controls are included.
PS12	*Technical Security Practices.* TSA **shall** use technical security controls to deliver its services in a secure way. Security control of the equipment, of the life cycle of the software, of the networks and of the cryptographic modules are included.
PS13	*Security Audit Procedures.* TSA **shall** state how and when audits are carried out and who must do it (external or internal).
PS14	*Log Procedure.* TSA **shall** state what events shall be recorded, the type of log files, the procedure to do it, who is the responsible, and how long the log files shall be maintained.
PS15	*Communication of abnormal operation.* TSA **shall** state how all deviation from the normal operation shall be communicated to the customers and published to make them known to any interested party. Deviations includes loss of synchronisation, deviations from the time base, compromised keys, etc.
PS16	*Disaster recovery.* TSA **shall** state its internal and external procedures to recover after a disaster of any type has occurred.
PS17	*Termination of the TSA activity.* TSA **shall** state how it shall proceed in case of going out of the Time Stamping Service Provision activity.
PS18	The TSPS **shall** also include any other statement delimiting the relationship between the TSA and its users that can be source of disputes.

7.3.2 Liability limitation.

The TSPS **shall** define the operation framework for the Time Stamping Service Provision for a TSA. User demanding Time Stamping Services from a specific TSA **shall** accept the items defined within the TSPS. If the TSA operates out the conditions defined in the TSPS **shall** be liable for any complaint.

Consequences of a TSA malfunction can be of a very wide range and, in many cases, can cause serious damages to TSA's customers. The damage can be of a very wide range of types (economic, image...) and very difficult to quantify. Two ways can be used to delimit TSA's liability against complaints:

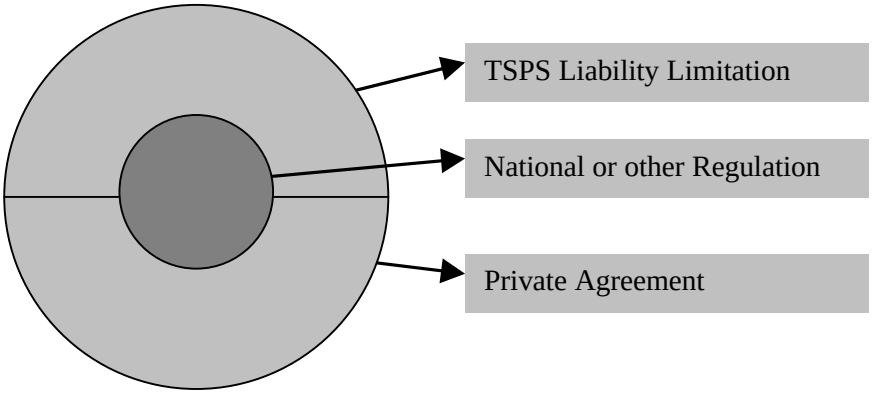
- Legal
- Contractual

In the first case, TSA **shall** behave according to that stated in the national regulations of each country or community. In the second case, and even as a complement of the first, the TSA **may** agree a contract with each customer defining the limits of its liability. This agreement **should** be previous to any operational relationship between the TSA and the customer.

As a general situation, TSA can include the liability limitation in its TSPS so any user **may** know the position of the TSA in case of complaints due to fails during the service provision. In this case, users **shall** be able to choose the TSA that offers better satisfaction in case of trouble, among other decision elements.

In some cases, when a special customer demands special liability limits, out and greater of those contained in the TSPS, the TSA **may** negotiate this situation with the customer.

Basically, the definition of the Liability Limitation **should** be according to the following figure:



Both the TSPS Liability Limitation and the Private Agreements **shall** comply with all the regulation applicable to the TSA.

7.4 ECONOMICS

The objective of this section is to analyse the economic factors that would influence the provision of a Time Stamping service. Special emphasis will be placed on fixed expenses, variable expenses and possible sources of income.

7.4.1 Investment Considerations

7.4.1.1 General Considerations

This section will set out the factors considered and the hypotheses assumed in calculating the investment required for the implementation of a Time Stamping service. Obviously, all TS processes should be effected under the strictest integrated security conditions. These security requirements may be classified as:

- Physical, accessing and operating security conditions
- Logical and cryptographic operating security conditions.

The basic supporting elements for these security conditions are:

- Physical infrastructure
- Document dating infrastructure
- Storage infrastructure for meticulous archiving of all the transactions and events recorded.

In addition consideration must be given to other technological infrastructures that will be required as auxiliary components. These will include:

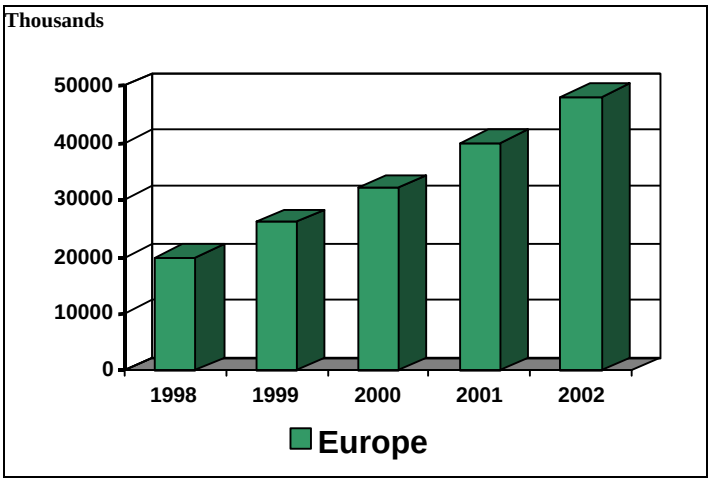
- User support and external relationships infrastructure
- Office automation and internal management infrastructure.

It is anticipated that the initial investment will be followed by others for equipment upgrading or replacement (due to obsolescence of the technological infrastructure).

7.4.1.2 Assumptions

7.4.1.2.1 Number of users

A pessimistic assumption regarding the progression of the number of Internet users over the next few years may serve as the starting point for estimating a feasible number of users of this infrastructure.



7.4.1.2.2 Assumptions Regarding Use Of Services.

The following general assumptions have been made regarding the application of the Time Stamping infrastructure:

- Working days per year: 240
- Average number of transactions per user/year: 5

From the above assumptions we get the following estimated count of operations to be performed:

Year	Potential Users	Time Stamping Certificates
1998	20 million	100 million
1999	26 million	130 million
2000	32 million	160 million
2001	40 million	200 million
2002	48 million	240 million

7.4.1.3 Investment Components

7.4.1.3.1 Physical infrastructure

- Building
- Security conditioning
- Physical security equipment
- Basic furnishing (furniture and fixtures)

7.4.1.3.2 Processing equipment (hardware)

- Processing units (CPUs)
- Basic peripherals
- Cryptographic peripherals

7.4.1.3.3 Storage equipment

- Storage units
- Peripherals

7.4.1.3.4 Communications equipment

- External lines
- Internal data networks
- Transmission systems
- Communications management equipment
- Linking equipment
- Security equipment
- Equipment installation and design

7.4.1.3.5 Development software

- Testing and Validating
- Productivity
- Security
- Development
- Process management
- Collaboration among working teams
- Management of development environment

7.4.1.3.6 Operating software

- User interface
- Generic data access
- Communications
- Task flow support
- Generic cryptographic methods
- Office automation
- Client

7.4.1.3.7 Resource planning software

- Production control
- Monitoring
- Failure recovery
- Environment managing
- Security
- Configuration control

7.4.1.3.8 Projects

- Research and Development

7.4.1.4 Starting Hypotheses

7.4.1.4.1 General hypotheses

The technology needed to develop a Time Stamping system is an immature, changeable technology, which means that hypotheses have been assumed with a certain degree of uncertainty.

7.4.1.4.2 Physical infrastructure

This section should evaluate the situation of the Time Stamping service in a single location. This will require an initial investment in making the space suitable for the requirements of the project as regards both security of the infrastructures and personnel, as well as the normal facilities for conducting the daily work. In the event of the TSA sharing a location with another authority, the relevant proportional part will have to be taken into account.

It should be considered that the **dual-operating or redundancy equipment** that has been anticipated to resolve any crash of the mainframe will be located in the same facility since there will be no alternative backup centre. If this is not the case, the relevant investment should be considered.

As from the first year it is anticipated to reinvest 10% of the original amount.

7.4.1.4.3 Processing equipment

Equipment should be available which will provide the service, in addition to the relevant redundant equipment. Assuming a CPU usage of 2 Kb per operation conducted, if the requests received per second number 1000, a processing capacity of 2Mb per second will be required.

7.4.1.4.4 Storage equipment

Assuming that each Time Stamping operation uses 2 Kb, the storage capacity needed will be dependent on the time period they are kept stored. If we assume a storage time of 1 year and a daily TSA real operating time of 1 hour, the storage capacity needed would be approximately 60 Gb.

It will be necessary to have redundant storage units located in different places, in such a way that if any of them crash, another can make up for it.

7.4.1.4.5 Reinvestments in technological infrastructures

Due to the features of the hardware and software utilised, which have a high obsolescence rate, it is anticipated that as from the third year reinvestment in this infrastructure will be necessary. The percentage recommended is 33%, which is what the computer sector usually observes.

7.4.1.4.6 General investment expenses

An expense figure of 5% of total investments should be estimated for transport, assembly, installation, etc.

7.4.1.4.7 Projects. Personnel

Expenses must be considered for project personnel required for implementation of the service.

7.4.2 Considerations Regarding Operating Expenses

7.4.2.1 General considerations

In order to calculate the costs in this section, the normal functioning and operating expenses of the Time Stamping activity need to be considered.

Advertising expenses have not been taken into account in this section.

7.4.2.2 Cost elements

7.4.2.2.1 Operations

7.4.2.2.1.1 External services

- Specific developments
- Consulting and specialised companies
- Agreements with Universities and other institutions.

7.4.2.2.1.2 Services and licences

- External licences
- Patents

7.4.2.2.1.3 Maintenance and repairs

- Physical infrastructure
- Physical security conditions
- Internal networks
- Processing equipment (hardware)
- Storage equipment (hardware)
- Communications equipment (hardware)
- Development software
- Operating software
- Process software

7.4.2.2.1.4 Personnel

- Cryptographic specifications and control
- User equipment specifications and control
- External technical representation
- Equipment operating
- Equipment maintenance
- Security
- Quality control
- Sales
- Auxiliary services
- Management developments
- Corporate relations
- General administration

7.4.2.2.1.5 General Expenses

7.4.2.3 Starting hypotheses

7.4.2.3.1 Operations manpower

This section should anticipate Structure and Operating personnel costs.

7.4.2.3.2 Outside lines

The capacity of the lines to be hired should be a function of the number of operations per second that it is desired to handle at peak work time. If we take 2 Kb as the standard length per request stamped and we need to attend to a maximum of 1000 requests per second, the capacity of the line to be hired should be a minimum of 2 Mbps.

7.4.2.3.3 Maintenance and Repair

As a general rule, this is estimated to be 10% of the investment.

This percentage will be applied to the initial investment, since it is understood that reinvestments compensate for the obsolete infrastructure.

7.4.2.3.4 Services and licences

Operating software licences would have to be acquired, in addition to the software for office automation, depending on the number of users you have.

7.4.2.3.5 General expenses

These should be estimated as 10% of total expenses, not counting contingencies. This hypothesis has taken into account normal expenses in this business sector. This section contemplates the physical security expenses (watchmen), electricity expenses, air conditioning, etc.

7.4.3 Considerations Regarding Income

7.4.3.1 Segmenting the market

The income forecasts are formulated on sales expectations and prices. First, therefore, we are going to develop the starting framework with regard to possible users of the Time Stamping infrastructure.

As the starting point for the Income study a classification was made of the clients with whom the Time Stamping Authority is going to interact.

Public Administrations. These will be divided into Presidency, General State Administration, Autonomous Communities and Local Administrations, to get a more reliable approximation of the number of users.

Corporations. Distinctions should be made between companies whose turnover is such that the activity they generate is similar to Citizens (SME) and companies which have a turnover that is much greater than Citizens.

Citizens. Since differentiation by segments has been included, the number of citizens likely to utilise Time Stamping will be calculated by taking the difference between the initial dimensioning and the number of users from Administrations and Corporations.

7.4.3.2 Price strategies

7.4.3.2.1 Pricing methods

There are three possible strategies for obtaining income to finance the certification services provided:

- Flat rate
- Rate per transaction
- Combined model, which applies a flat rate and a per transaction rate (for companies and administrations)

Within these possible options we will indicate those which will subsequently be analysed in the scenarios selected.

	flat rate	variable rate	combined rate
citizens	X		
SME	X		
large companies	X		X
Administration	X		X

a) The Flat Rate is based on an annual fee which will be billed to each of the users. This option facilitates billing and prevents the need for an individualised invoice-generating system.

b) The Variable Rate is determined as a function of the degree of utilisation that is made of the service by each person.

c) The Combined rate brings together a fixed part and a variable part which will depend on the utilisation made of the service. This income-generating strategy requires a technical architecture capable of recording each transaction, calculating the charge involved, and issuing the relevant invoice. However, it enables each user to be charged a more realistic fee in keeping with the costs that his activity generates.

In scenario 2, which proposes a Combined fee for users from the Administrations and Corporations, citizens would always be charged the flat rate described in a). A flat annual fee will be set which will be lower than the annual rate set out in the previous scenario.

The variable part of the Combined rate will be calculated - for Corporations and Public Administrations - as a function of the utilisation of the service.

7.4.4 Profitability Analysis

7.4.4.1 Scenarios

In order to have two different views, we have analysed two scenarios which give different results, since different hypotheses were established for the variables in question.

7.4.4.1.1 Scenario 1

Scenario 1 proposes a flat annual rate for each user of the infrastructure, as follows:

citizens	flat rate
SME	flat rate
large companies	flat rate per user
Administration	flat rate per user

The service rate is set lower for citizens and PYMEs (over 50% are companies with no salary earners), since the activity they will generate will be lower than that produced by a user from the Public Administration and large companies.

7.4.4.1.2 Scenario 2

In this scenario a combined pricing structure is established for Public Administrations and Corporations. In other words, billing will include a flat annual rate plus variable fees depending on the use made of certain services.

Citizens will continue to be billed the annual flat rate, since this hypothesis significantly simplifies the relationship with them.

For Scenario 2, then, we have assumed the following price-structure:

citizens	flat rate
SME	flat rate
large companies	Combined rate (flat rate per user plus use of certain services)
Administration	Combined rate (flat rate per user plus use of certain services)

A price structure following the considerations of this scenario would enable a more balanced sharing out of the infrastructure costs, greater costs affecting users who generate more activity.

8 APPENDIX A: CRYPTOGRAPHIC BACKGROUND

For this appendix the general reference is [Schneier96].

8.1 SYMMETRIC KEY CRYPTOGRAPHY

Symmetric key cryptography (also known as secret key cryptography) uses essentially the same encryption and decryption key (i.e., either the encryption and decryption keys are the same, or they are different but are easily computed from each other). The encryption and decryption operations are usually expressed with reference the common key K (M and C denote the plaintext message and the encrypted message, respectively):

Encryption: $E_K(M) = C$	["the message M is enciphered using key K, and the obtained cyphertext is C"]
Decryption: $D_K(C) = M$	["the cyphertext C is deciphered using key K, and the obtained message is M."]

Important examples of symmetric key cryptography algorithms are the IDEA and DES encryption algorithms, and the message authentication codes (MACs).

8.1.1 IDEA

The IDEA (International Data Encryption Algorithm) algorithm was developed by European researchers between 1990 and 1992. It is a block algorithm (it processes the input message in blocks of 64 bits) and its key is 128 bits long. In IDEA the decryption keys are either the additive or multiplicative inverses of the encryption keys.
Some experts consider it the best and most secure block algorithm available to the public [Schneier96]. The IDEA algorithm is part of the widely used PGP encryption software. For non-commercial use, no license fee is required to use IDEA, but commercial users should contact Ascom-Tech AG, the firm holding its patent (the algorithm is patented in Europe and in the US).

8.1.2 DES

The DES encryption algorithm was developed around 1975 by IBM (International Business Machines Corporation) and the US NSA (National Security Agency), and has been the most widely used symmetric encryption algorithm, especially in financial institutions.
DES processes the message in 64 bit blocks, and its keys are 56 bits long. It uses the same encryption and decryption keys.

8.1.3 Message Authentication Codes (MACs)

A message authentication code (MAC) is a hash function (see section 8.3 of this document for an overview of hash functions) that is performed on a message and depends on a key. For example, the hash could be computed on the concatenation of the document to be hashed and the key used. These codes are computed to verify that the information exchanged by two parties has not been modified, and indirectly authenticates the source.

The parties share a common key K. When a message M is sent from A to B through an insecure communications channel, the sender appends a hash H (K, M) to the message, called the MAC of the message. The receiver computes the MAC H (K, M') with the received M', and checks if the obtained value equals the received MAC; if these values agree, he regards the received M' as valid. Should these two values differ, either the message M or the MAC (or both) have been modified, and the message should be sent again.

8.2 PUBLIC KEY CRYPTOGRAPHY

Public key cryptography was invented by Whitfield Diffie and Martin Hellman in 1976. Their important discovery is that encryption and decryption keys could be different, and could be obtained in a way that prevented to obtain one from the other.
In public-key cryptography, each user has two different keys, one made available to the public (the "public key") and the other kept secret (the "private key" or "secret key"). One of the keys is used to encrypt a message, and the other is used to decrypt it.

The enciphering operation is denoted by the E_K operation, with K being the secret K_S or the public K_P key:

$E_{KP}(M) = C$ ["the message M is enciphered with the public key, producing a cyphertext C."]

$E_{KS}(M) = C$ ["the message M is enciphered with the secret key, producing a

Similarly, the deciphering operation is denoted by the D_K operation, with K being the secret key K_S or the public key K_P:

$D_{KP}(C) = M$ ["the cyphertext C is deciphered with the public key K_P, producing the message M."]

$D_{KS}(C) = M$ ["the cyphertext C is deciphered with the secret key K_S, producing the message M."]

Important public key cryptography algorithms are public key encryption and digital signatures.

8.2.1 Public-Key Encryption

If every user in a system generates a private-key/public-key pair, and distributes his public key while keeping the private key secret, everybody could use the public key to encrypt a messages that only this user is able to decrypt, even if the encrypted message is intercepted. Even the sender of the message is unable to decrypt the encrypted message.

An important public key encryption algorithm is RSA, named after its inventors Rivest, Shamir and Adleman. This encryption scheme gets its security from the difficulty of factoring large integer numbers: the private and public keys are generated from two large random prime numbers, and it is conjectured that breaking RSA is equivalent to factoring the product of the two primes.

Due to its relatively slow performance, RSA is often used in conjunction with symmetric encryption algorithms. The message is encrypted with a symmetric encryption algorithm, and the secret key used in the symmetric algorithm is encoded with the public key of the receiver. The receiver of the message can use his private key to obtain the symmetric key, and decrypt the message with the symmetric algorithm.

The RSA algorithm is patented in the US (and only in the US).

8.2.2 Public-Key Digital Signatures

Every public-key encryption system can be used to perform digital signatures. Section 8.7 of this document is dedicated to the study of digital signature schemes.

8.3 HASH FUNCTIONS

Hashes, or message digests, are functions that take an arbitrary size message as input and produce a fixed-length (typically 16 or 20 bytes) output value. It is evident that many different input messages will generate the same hash value; one of the properties of successful hash functions is that they generate *statistically unique* output values, so that collisions (two or more messages with a common hash) are computationally hard to find. Other important requirement is that they must be one-way: it must be computationally unfeasible to find a message whose hash is a given value.

Hashing algorithms are used in cryptographic systems for a number of purposes, including the creation of message digests for digital signatures, authentication protocols, and integrity checking.

The defining properties of a one-way hash function are:

1. It is computationally easy to compute the hash H (M) of a given message M.

2. Given a certain hash value h, it is computationally infeasible to find a message M with that hash value (i.e., with H (M) = h).

3. Given a message M_1, it is computationally infeasible to find another message M_2 with the same hash value (i.e., such that H (M_1) = H (M_2)).

4. It is computationally infeasible to find two messages M1 and M2 with the same hash value.

The first three properties warrant that the hash function is easily computed and one-way, so that nobody will be able to obtain a message with a certain hash value. The fourth property ensures that nobody will be able to randomly generate a pair of messages with the same hash value. This is an important property in the context of contract signing: generally, signatures are performed not on the contract text but on a hash of it. …

In practical implementation, public-key algorithms are too slow to sign long messages. To save time, digital signature protocols are generally implemented with one-way hash functions: instead of signing a document, one signs the hash of the document. Since hash functions are non-invertible, nobody will be able to generate a different document with the same hash value. The collision resistance property of the hash function will also ensure that nobody will generate two different contracts with the same hash value, and later exchange the agreed contract with the modified one. This is of crucial importance to time-stamping.

8.3.1 MD2

MD2 was designed by Ron Rivest for RSA and is described in [RFC 1319]. It produces a 128-bit digest.

Although no weaknesses in MD2 have been found, it is slower than most other suggested hash functions, and seldom used.

8.3.2 MD4

MD4 was designed by R. Rivest for RSA and is described in [RFC 1320]. The algorithm produces a 128-bit hash. It was designed with the following goals in mind:

- **Security**. It is computationally infeasible to find two messages that hash to the same value. No known attack is more efficient than simple brute force.
- **Direct security**. MD4's security is not based on the computational hardness of any mathematical problem, like the difficulty of factoring.
- **Speed**. MD4 is suitable for high speed software implementations. It is based on a simple set of bit manipulations on 32-bit operands.
- **Simplicity and compactness**. MD4 is very simple, without large data structures or complicated subroutines.
- **Favour little-endian architectures**. Optimised for little-endian microprocessors (e.g. Intel 80x86 family).

The algorithm has been subject to intense cryptographic attacks with some success, and it is not recommended any longer. However, it has been the starting point for MD5, SHA and RIPEMD.

8.3.3 MD5

MD5 was designed by R. Rivest for RSA and is described in [RFC 1321]. It is an improvement of MD4. It produces a 128-bit hash.

MD5 processes the input in 512-bit blocks. First, it pads the message so that its length is a multiple of 512 bits, including the length of the message before padding. The main loop has 4 rounds of 16 operations each.

Some cryptographic attacks have succeeded against some parts of MD5, but the algorithm as a whole has not been violated. It is widely used nowadays, although SHA appears as more secure.

8.3.3 RIPEMD

The RIPEMD 128-bit hash function was developed under the European Community's RIPE project (1988-1992) by researchers from the Catholic University of Leuven (Belgium) [RIPE 95]. These hash functions were designed to resist known attempts to attack MD4. In 1995 the authors considered that 128-bits hash functions provided insufficient protection against brute force collision attacks, and that cryptographic attacks to MD4, MD5 and RIPEMD were raising serious doubts on the strength of these hash algorithms. Therefore, they designed a strengthened 160-bit version called RIPEMD-160, that they expected to be strong enough for 10 years.

The RIPEMD hash functions are not patented and, in contrast to SHA-1, their design criteria are public.

8.3.4 SHA-1

U.S. NIST designed the Secure Hash Algorithm (SHA) for use with the Digital Signature Standard (DSS). When a message M of length $< 2^{64}$ bits is input, the SHA produces a 160-bit output called a message digest. The SHA is based on principles similar to those used in MD4 message digest algorithm (see below), and is closely modelled after that algorithm.

SHA starts by appending the message length at the end of the message; a padding is performed on the resulting message (appending between zero and 511 bytes at the end) to make the message length a multiple of 512 bytes. Then the algorithm processes the message 512 bytes at a time. The main loop has 4 rounds of 20 operations each.

The original SHA specification was revised to "correct a technical flaw that made the standard less secure that had been thought," according to NSA sources, but no details were provided on the flaw. The resulting algorithm is called SHA-1.

There are no known cryptographic attacks against SHA.

8.3.5 Attacks on hash functions

Since the time-stamping (and digital signature) solutions proposed up to date make heavy use of hash functions, it is important to analyse the impact of attacks to hash functions on time-stamping protocols, as well as to find ways to minimise this impact.

To begin with, let us restate the defining properties of a hash function:

1. It is computationally easy to compute the hash H (M) of a given message M.

2. Given a certain hash value h, it is computationally infeasible to find a message M with that hash value (i.e., with H (M) = h).

3. Given a message M_1, it is computationally infeasible to find another message M_2 with the same hash value (i.e., such that H (M_1) = H (M_2)).

4. It is computationally infeasible to find two messages M_1 and M_2 with the same hash value.

Once a hash algorithm is devised, property (1) will hold forever: if something is easy to compute today, it will be easy to compute in the future, and there is every reason to assume that it will be even easier to compute in future times. The validity of the properties (2), (3) and (4) for a hash algorithm will depend on the advances in cryptography.

At first sight one could think that (2) and (3) are equivalent, but after careful examination one concludes that a scheme to break (2) is always useful to break (3), but a scheme to break (3) does not necessarily break (2). Let us explicitly state the reasoning behind this:

A scheme to break (2) does always break (3)

Assume we are able, given a hash value h, to use an algorithm S find a message with hash =h. Now let's see that we can use S to break (3). If we are given a message M_1, we obtain its hash h = H(M_1). Then we use S to obtain another message M_2 with hash h, and (with all probability, since there are infinitely many more messages than hashes, and thus many different messages hash to each possible hash value) this M_2 will be different to M_1. Thus, we have found another message with the
same hash as M_1, breaking (3).

A scheme to break (3) does not necessarily break (2)

If a scheme is devised to break (3), it *can* use as inputs *both* a hash value h and a message M with H(M) = h. If the algorithm uses the message M (for anything different than simply computing its hash value h), then this scheme is not useful to break (2), since in (2) one has access to the hash, but not to any message with the given hash.

Analogously, one can see that an attack to (2) or to (3) always results in an attack to (4), but that an attack to (4) does not necessarily constitute an attack to either (2) or (3).

A scheme to break (2) or (3) always breaks (4)

If a scheme S is found that breaks properties (2) or (3) (or both), one can take an arbitrary message M_1, compute its hash $h = H(M_1)$, and then use the scheme S (if the scheme breaks (2) only h will be needed as input, but if the scheme only breaks (3) then both h and M_1 will be necessary) to obtain another message M_2 hashing to the same value. Thus, two messages M_1 and M_2 with a common hash h are found with the aid of S, that therefore breaks (4).

A scheme to break (4) does not necessarily break (2) or (3)
In both (2) or (3), the hash of the message that has to be found is fixed (in (2), this hash is the single information we have, in (3) we have also a message that hashes to this value), but (4) finds two messages with a previously undetermined hash.

After the previous reasoning, we can conclude that the most powerful attacks of all are those that invalidate property (2); the following most important attacks are those to property (3), and finally the least powerful attacks are those to property (4). It is important to keep in mind that all these types of attack would have enormous consequences for time-stamping and for the surrounding public-key infrastructure (digital signatures, etc.).

In the time-stamping protocol, the user knows both the document to be time-stamped and its hash. Therefore, an attack of type (2) or (3) could be attempted. Someone tapping the communications channel between the user and the time-stamping service provider could only gain access to the hash of the documents (unless the document is also transmitted, but this could be done in encrypted form), and thus could only attempt type (2) attacks, the most difficult.

To minimise the probability of strong attacks (to properties (2) and (3)) to the hash functions used in the time-stamping protocols, the following recommendations should be followed:

1. The TSA (Time-Stamping Authorities) select a set of several accepted hash functions, and require the users to choose several algorithms from the set. If one of the accepted hashes is broken (properties (2) or (3), and therefore also (4)), only those documents time-stamped with the broken hash will be affected, and only in the case that only one hash was used. This will reduce the impact of a hash becoming obsolete.

2. If the TSA require their users to compute *several* different hashing algorithms of the documents, no severe problems will appear if one hashing algorithm is broken. Even in this case a simple renewal of the affected certificates will be sufficient to guarantee their validity.

Attacks to property (4) could be prevented by following the previous recommendations, and by choosing long hashes.

8.4 INCREMENTAL CRYPTOGRAPHY

Incremental cryptography [BeGoGo94] attempts to accelerate the computation of all kinds of cryptographic primitives when they are applied to modified documents. The idea is that if we have already computed the primitive (hash function, digital signature, MAC…) on a certain document, and then the document is modified, the time needed to compute the function on the modified document should be proportional to the amount of modification.
The goal of incremental cryptography, then, is to devise cryptographic schemes with the property that, having applied the scheme on a document, it is possible to update the result for a modified document, instead of having to recompute it completely.

The foreseen applications of incremental cryptographic primitives range from virus protection to the time-stamping of continuous streams of information, such as multimedia. In the course of the project we will explore the possibilities of these algorithms in this latter setting.

8.5 ONE-WAY ACCUMULATORS

One-way accumulators are one-way hash functions that can be used to hash several messages whose processing order is irrelevant (a kind of commutativity). The first works on one-way accumulators were produced by J. Benaloh and M. de Mare [BeMa93].

Suppose there is a sequence of messages M_1, M_2…, M_n that are to be hashed together in order to obtain the accumulated hash:

$h_1 = h(0, M_1)$
$h_2 = h(h_1, M_2)$
$h_3 = h(h_2, M_3)$
…
$h_n = h(h_{n-1}, M_n)$

This last hash value can later be recomputed to verify that none of the messages M_1, …, M_n has been modified. Using a conventional hash function, the hash value must be accumulated in the same order; with a one-way accumulator, however, this computation can be performed in any order, and the outcome will be the same.

Example of operations with the quasi-commutativity property are addition, multiplication and exponentiation modulo a number n. Of these, only exponentiation modulo n is believed to be difficult to invert (under some conditions on the modulo n).

One-way accumulators can be used in several contexts. In particular, their use has been proposed in membership checking and decentralised time-stamping [BeMa92].

8.6 ZERO-KNOWLEDGE IDENTIFICATION

Zero-knowledge identification schemes provide proofs of identity that yield nothing beyond the validity of the identity assertion. That is, a verifier obtaining such a proof only gains conviction in the identity of the other party, but no information is gained that would allow him to impersonate the identified party, even after an arbitrarily large number of verifications.
This contrasts with more basic identification protocols, such as those based on constant passwords, in which the act of proving one's identity gives the verifier full potential to impersonate the user.
With zero-knowledge identification schemes, the verifier cannot even prove to a third party that the prover has been successfully identified, since the third party cannot be sure that the prover and the verifier are not cheating (for example, preparing the questions and answers beforehand).

8.6.1 Feige-Fiat-Shamir Identification Scheme

The Feige-Fiat-Shamir protocol is a zero-knowledge proof of knowledge of a modular square root of the user's public key y_U (square roots modulo a large number N are computationally hard to compute if the factorisation of N is unknown). Since the user U is supposedly the only party knowing its private key x_U (the modular square root of the public key y_U), succeeding in the protocol is taken as evidence that the prover is U. The protocol guarantees that the interaction with the user does not leak knowledge which might be used to impersonate U.

Feige-Fiat-Shamir Identification Protocol

Universal Parameter (chosen by a TTP): A uniformly chosen number N, product of two (secret) primes.
Private Key of each user: A uniformly chosen $x \in \{1,\ldots, N\}$.
Public Key of each user: $y = x^2 \bmod N$.

Protocol to verify identity of user Alice to verifier Bob:

1. Alice uniformly selects $r \in \{1,\ldots, N\}$ and sends $s=r^2 \bmod N$ to Bob.
2. Bob uniformly selects a challenge $\sigma \in \{0,1\}$, and sends it to Alice.
3. Alice replies with $z = r.x^{\sigma} \bmod N$.
4. Bob accepts the response if and only if $z^2 = s.y^{\sigma} \bmod N$.

The probability of the user fooling the verifier is ½ in each round of the protocol. Repeating the protocol a large number of times reduces the risk of impersonation to as low a probability as desired.

This identification protocol can be turned into a signature scheme. The resulting Fiat-Shamir signature scheme is very fast, requiring only about a 4% of the modular multiplication needed for a RSA signature. The Fiat-Shamir identification and signature scheme is patented in the US (US Patent #4,748,668).

8.6.2 Guillou-Quisquater Identification Scheme

This identification protocol minimises the number of exchanges between the prover and the verifier, at the expense of greater computation requirements for an equivalent level of security. The Guillou-Quisquater identification protocol is suited for applications where the exchanges between the verifier and the prover must be reduced, such as smart cards.

Guillou-Quisquater Zero-Knowledge Identification Protocol

Universal parameters: A number *v* and a large number *n* which is the product of two (secret) primes.
Public Key of each user: Large number J.
Private Key of each user: A number B, calculated so that $JB^v = 1 \pmod n$.
Protocol for Alice to prove its identity to Bob:

1. Alice uniformly selects $r \in \{1,\ldots, n-1\}$, and sends $T = r^v \bmod n$ to Bob.
2. Bob uniformly chooses $d \in \{0,\ldots, v-1\}$, and sends it to Alice.
3. Alice computes $D = rB^d \pmod n$, and sends it to Bob.

Bob computes $T' = D^v J^d$; if $T = T' \pmod n$, then the identity assertion is accepted.

The mathematical justification of this protocol is quite obvious:

$$T' = D^v J^d = (rB^d)^v J^d = r^v B^{dv} J^d = r^v (JB^v)^d = r^v = T \pmod n$$

(remember that $JB^v = 1 \bmod n$)

This scheme can be turned into a signature algorithm suitable for use in smart cards.

8.7 DIGITAL SIGNATURE ALGORITHMS

These algorithms create a digital signature for arbitrary data using a private key value. In addition, they support the ability to verify a digital signature, given both the signature object and the associated public key.

8.7.1 DSA

The Digital Signature Algorithm (DSA) has been standardised by the US government as part of the Digital Signature Standard (DSS). Compared to RSA, verification using DSA is computationally more expensive. Unlike other dual digital signature schemes, DSA can only be used for generating digital signatures, and no encryption is possible using it.

8.7.2 RSA

The RSA public-key scheme can be used (as every other public-key system) to digitally sign messages. For RSA digital signatures, a minimum modulus key length of 512 bits should be used; a length of 1024 bits is recommended.

RSA Digital Signature Scheme

To generate a digital signature for a message M:

- The message M is digested with a message-digest algorithm, either SHA-1 or MD5 are recommended.
- The digest H (M) is padded to form a value D of the same length as the key modulus; the de facto standard for D is described in PKCS #1.
- The signature is generated by applying the signer's private key to the data D
  $S= D^{Ks} \bmod n$, where Ks is the private (secret) key, and n= pq is the modulus.

To verify a digital signature:

- The signed data D is recovered from the signature S using the signer's public key
  $D= S^{Kp} \bmod n$
- The message digest contained in D is compared with the message digest of the original message M. If they match, the signature is verified. Otherwise it is rejected.

9 APPENDIX B - STANDARDS BACKGROUND

9.1 X.509 AUTHENTICATION FRAMEWORK

ITU (previously called CCITT) has issued the recommendation X.509 [X.509 v3] to provide authentication across networks. No particular algorithms are specified or recommended, but there are provisions for multiple algorithms and hash functions.

The most important part of X.509 is its structure for public certificates. A trusted certification authority (CA) assigns a unique name to each user and issues a signed certificate containing the name and the user's public key. This association of identity to the public part, transitively associates the identity to the knowledge of the private part.

The following fields are part of an X.509 certificate: a serial number that is unique within the CA; the algorithms used to sign the certificate are included within it, as well as the identification of the issuing CA; the period of validity is a pair of dates: not before, and not after.

The subject is identified by its name. Its public key information includes the algorithm, any necessary parameters, and the public key itself.

The last field is the CA's signature.

Certificates may introduce additional information such as intended use of the key, policy or policies that apply or are regarded as similar for trust extension, alternative names, identification of CRL distribution points, etc. etc. There is provision for extended certificates as needed by applications.

Certificates can be stored in databases. Users can access to them. When a certificate expires, it should be removed from public directories. The issuing CA, however, should maintain a copy of expired certificates: should a dispute arise later, it will be required. Premature expiration (e.g. due to key compromise) is handled by means of certificate revocation lists (CRL).

9.2 PKCS - PUBLIC KEY CRYPTOGRAPHY STANDARDS

The Public Key Cryptography Standards [PKCS #1, PKCS#6, PKCS #7, PKCS #10, PKCS #12] are RSA Data Security Inc.'s attempt to provide an industry standard interface for public cryptography. It is a private collection of standards developed working with a variety of companies. They have become a *de facto* standard, wide accepted in industry.

Out of 10 standards, the following ones are related or applicable to time-stamping services.

PKCS #1 describes a method for RSA encryption and decryption, primarily for constructing the digital signatures and digital envelopes described in PKCS #7. PKCS #1 also describes a syntax, identical to the syntax of X.509, for RSA public and private keys and three signature algorithms -MD2 with RSA, MD4 with RSA, and MD5 with RSA- for signing certificates.

PKCS #6 describes a standard syntax for public key certificates. The syntax is a superset of an X.509 certificate, so that X.509 certificates can be extracted if necessary. This is not widely used.

PKCS #7 is a general syntax for data that may be encrypted or signed, such as digital signatures. The syntax also allows other attributes, such as time-stamps, to be authenticated. PKCS #7 is compatible with PEM so that signed messages can be converted to/from PEM without any cryptographic operations.

PKCS #10 describes a syntax for certification requests. These are sent to a certification authority (CA) for signing.

PKCS #12 describes a syntax for storing in software a user's public keys, protected private keys, certificates, and other related cryptographic information.

PKCS are used as a format for transferring data based on public-key cryptography and an infrastructure to support that transfer.

9.3 PKIX - PUBLIC-KEY INFRASTRUCTURE (X.509)

[PKIX] is a working group under IETF umbrella to provide practical profiles for effective use of public keys in the framework of Internet. The charter describes itself as:

> Many Internet protocols and applications which use the Internet employ public-key technology for security purposes and require a public-key infrastructure (PKI) to securely manage public keys for widely-distributed users or systems. The X.509 standard constitutes a widely-accepted basis for such an infrastructure, defining data formats and procedures related to distribution of public keys via certificates digitally signed by certification authorities (CAs). [RFC 1422] specified the basis of an X.509-based PKI, targeted primarily at satisfying the needs of Internet Privacy Enhanced Mail (PEM). Since [RFC 1422] was issued, application requirements for an Internet PKI have broadened tremendously, and the capabilities of X.509 have advanced with the development of standards defining the X.509 version 3 certificate and version 2 certificate revocation list (CRL).
>
> The task of the working group will be to develop Internet standards needed to support an X.509-based PKI. The goal of this PKI will be to facilitate the use of X.509 certificates in multiple applications which make use of the Internet and to promote interoperability between different implementations choosing to make use of X.509 certificates. The resulting PKI is intended to provide a framework which will support a range of trust/hierarchy environments and a range of usage environments ([RFC 1422] is an example of one such model).
>
> Candidate applications to be served by this PKI include, but are not limited to, PEM, MOSS, GSS-API mechanisms (e.g., SPKM), IPSEC protocols, Internet payment protocols, and www protocols. This project will not preclude use of non-infrastructural public-key distribution techniques nor of non-X.509 PKIs by such applications. Efforts will be made to coordinate with the IETF White Pages (X.500/WHOIS++) project.
>
> The group will focus on tailoring and profiling the features available in the v3 X.509 certificate to best match the requirements and characteristics of the Internet environment. Other topics to be addressed potentially include:
>
> - Alternatives for CA-to-CA certification links and structures, including guidelines for constraints
> - Revocation alternatives, including profiling of X.509 v2 CRL extensions
> - Certificate and CRL distribution options (X.500-based, non-X.500-based)
> - Guidelines for policy definition and registration
> - Administrative protocols and procedures, including certificate generation, revocation notification, cross-certification, and key-pair updating
> - Naming and name forms (how entities are identified, e.g., email address, URN, DN, misc.)
> - Generation of client key pairs by the PKI

The following working drafts have been issued to date:

- <http://www.ietf.org/internet-drafts/draft-ietf-pkix-ipki-part1-07.txt>
  Internet Public Key Infrastructure X.509 Certificate and CRL Profile

- <http://www.ietf.org/internet-drafts/draft-ietf-pkix-ipki3cmp-07.txt>
  Internet X.509 Public Key Infrastructure Certificate Management Protocols

- <http://www.ietf.org/internet-drafts/draft-ietf-pkix-ipki2opp-07.txt>
  Internet X.509 Public Key Infrastructure Operational Protocols - LDAPv2

- <http://www.ietf.org/internet-drafts/draft-ietf-pkix-ipki-part4-02.txt>
  Internet Public Key Infrastructure Certificate Policy and Certification Practices Framework

- <http://www.ietf.org/internet-drafts/draft-ietf-pkix-ipki6np-00.txt>
  Internet Public Key Infrastructure
  (see next section)

- <http://www.ietf.org/internet-drafts/draft-ietf-pkix-ipki5tsp-00.txt">
  Internet Public Key Infrastructure Part V: Time Stamp Protocols
  (see next section)

- <http://www.ietf.org/internet-drafts/draft-ietf-pkix-ipki-kea-01.txt>
  Internet Public Key Infrastructure Representation of Key Exchange Algorithm (KEA) Keys in Internet Public Key Infrastructure Certificates

- <http://www.ietf.org/internet-drafts/draft-ietf-pkix-opp-ftp-http-03.txt>
  Internet X.509 Public Key Infrastructure Operational Protocols: FTP and HTTP

- <http://www.ietf.org/internet-drafts/draft-ietf-pkix-ocsp-03.txt>
  X.509 Internet Public Key Infrastructure Online Certificate Status Protocol - OCSP

- <http://www.ietf.org/internet-drafts/draft-ietf-pkix-ipki-ecdsa-00.txt>
  Internet X.509 Public Key Infrastructure Representation of Elliptic Curve Digital Signature Algorithm (ECDSA) Keys and Signatures in Internet X.509 Public Key Infrastructure Certificates

- <http://www.ietf.org/internet-drafts/draft-ietf-pkix-crmf-00.txt>
  Certificate Request Message Format

- <http://www.ietf.org/internet-drafts/draft-ietf-pkix-cmmf-00.txt>
  Internet X.509 Public Key Infrastructure Certificate Management Message Formats

- <http://www.ietf.org/internet-drafts/draft-ietf-pkix-cmc-00.txt>
  Certificate Management Messages over CMS

- <http://www.ietf.org/internet-drafts/draft-ietf-pkix-ldapv2-schema-00.txt>
  Internet X.509 Public Key Infrastructure LDAPv2 Schema

- <http://www.ietf.org/internet-drafts/draft-ietf-pkix-ocsp-caching-00.txt>
  Internet Public Key Infrastructure Caching the Online Certificate Status Protocol

- <http://www.ietf.org/internet-drafts/draft-ietf-pkix-webcap-00.txt>
  WEB based Certificate Access Protocol-- WebCAP/1.0

- <http://www.ietf.org/internet-drafts/draft-ietf-pkix-ocdp-00.txt>
  Internet X.509 Public Key Infrastructure OPEN CRL DISTRIBUTION PROCESS (OpenCDP)

The impact of these proposals on the provision of a time-stamping service will be further considered in the following deliverables, when dealing with concrete scenarios for service provision.

9.4 NOTARY AND TIME-STAMP PROTOCOLS

Currently there is some work in progress driven basically by commercial companies, and informally carried out under the PKIX IETF chapter, that has not officially extended its chapter to enclose this work. This work focuses the association of time to documents, and it might have a future influence on the provision of time services by PKI authorities. Therefore, it is summarised below.

A distinction is made between a notary authority (NA), a time-stamp authority (TSA), and a time-data service (TDS). The first one is covered in [Adams98b], and the other two in [Adams98a]. The three of them are planned to work smoothly together. The draft papers go into detail about the format of the data units to be exchanged to request and provide the notarisation and time-stamping services, as well as a the activities to be carried out by the involved actors.

The TSA is devoted to blindly associate a time to a hash of an hypothetical document. It has no liability as respects to the document contents or to the identity of the requester. The NA uses the TSA to add a time token to its answers to notarisation requests. The NA does check requester identity, and may perform a number of checks on the data to be notarised, depending of their class (a signature, a certificate, a receipt, ...). Lastly, the TDS may be used by the TSA to link its own clock to external events, thus providing an unforgeable proof of synchronisation.

The TSA provides a proof-of-existence at a given time. The NA provides a proof-of-contents. And the TDS provides a proof-of-synchronisation.

9.5 SPKI/SDSI - SIMPLE PUBLIC KEY INFRASTRUCTURE / SIMPLE DISTRIBUTED SECURITY INFRASTRUCTURE

SPKI and SDSI originated from the perception that the existing proposals for a public-key infrastructure were excessively complex and incomplete.

The SPKI Internet Engineering Task Force working group [SPKI] developed a series of specifications for mechanisms to support security in a wide range of internet applications, including IPSEC protocols, encrypted e-mail and WWW documents, payment protocols, and any other application which would require the use of public-key certificates. Independently and at the same time, R. Rivest (MIT) and B. Lampson (Microsoft) proposed a simple security infrastructure called SDSI, with similar goals and approach. Both teams finally merged, and they both work as SPKI / SDSI 2.0 working group. The latest (dated February, 1998) SDSI specification, SDSI 2.0, has been implemented in software, in both C and Java.

The basic requirement on the SPKI specifications is that the key certificate format and the associated protocols should be simple to understand, implement and use. For example, SPKI object descriptions do not use the complex ASN.1 notation, and the specification attempts to provide certificates that are easy to read from an application, without the need for complex parsers.

One of the fundamental realisations behind SPKI is that every user of a public key has a global name that is the public key itself. This identifier may not be pronounceable or easy to type and remember, but has the advantage of not requiring a certificate to tie it to the user's public key. The main purpose of an SPKI certificate is to authorise some action, give permission, etc., to or for a key holder, identified by its public key. SPKI assumes that certificates will be distributed directly by the holder to the verifier, without need for a global repository; in SPKI there is no public certificate repository (nevertheless, if the holder wishes to store the certificate in a global repository, nothing prevents it in the specification): each user keeps a "phone book" with the names and public keys of the known people and systems. SPKI assumes this is what is usually needed, since those people and systems are the ones a user talks to most of the time. If a user ever needs the public key of someone he or she doesn't know, what he or she should do is to ask friends or to try to get it via a parallel channel.

Essentially, the ideas behind SPKI are that each user has a public key to sign *name certificates* which attach names to the public keys of known people, or *delegation certificates*, which give others permission to do some of the things that he/she is allowed to do.

There is no hierarchy: each user can make statements and requests on the same basis as each other user. In practice some users will be more important than others, but the architecture treats them all as equal.

10 APPENDIX C: OTHER RELATED WORK

10.1 PAPERS

The most relevant papers in the time-stamping area, all of them authored or co-authored by Stuart A. Haber and Wakefield S. Stornetta, are enumerated below:

- S. Haber, W.-S. Stornetta, "How to time-Stamp a Digital Document", Journal of Cryptology Vol. 3 No. 2, 1991, pages 99-112.

  In this article, the basic reference for time-stamping, the authors discuss the "before than" and "after than" services of a TSS, and present the distributed protocol and the centralised linking protocol for digital documents.

- D. Bayer, S. Haber, W.-S. Stornetta, "Improving the Efficiency and Reliability of Digital Time-Stamping", Sequences 1991: Methods in Communication, Security and Computer Science; Springer-Verlag, 1992, pages 329-334.

  This article introduces the use of a tree structure of hash value to provide a time-stamping service. It avoids as well the use of secret keys to sign certificates digitally. The idea is presented as a variant to the centralised linking protocol.

- S. Haber, B. Kaliski, W. S. Stornetta, "How do Digital Time-Stamps Support Digital Signatures?", RSA's Cryptobytes, Autumn 1995.

  This article explores the close relationship between digital signatures and digital time-stamping. The authors argue that there are many situations in which the need to establish the date at which a document was digitally signed arises, and that digital time-stamping constitutes an obvious solution.

- J. Benaloh, M. de Mare, "Efficient Broadcast Time-Stamping (Extended Abstract)", Clarkson University Department of Mathematics and Computer Science Technical Report number TR-MCS-92-1. April 1992.

  This article analyses the time-stamping of digital documents in the context of broadcast channels. The authors present a time-stamping protocol based on one-way accumulators.

There also exist some reports on digital time-stamping:

- TIMESEC Project. Internet Home Page:
  http://www.dice.ucl.ac.be/crypto/TIMESEC.html

10.2 PATENTS

The following U. S. patents have been issued on the digital time-stamping area. Some of them cover time-stamping schemes, and others devices to time-stamp digital documents.

Patent Number	**Date**	**Inventor(s)**	**Name**
5,001,752	1991-03-19	A. M. Fischer	Public-Key Date-Time Notary Facility
5,022,080	1991-06-04	R. T. Durst K. D. Hunter	Electronic Notary

5,136,646	1992-08-04	S. A. Haber W. S. Stornetta	Digital Document Time-Stamping with Catenate Certificate
5,136,647	1992-08-04	S. A. Haber W. S. Stornetta	Method for Secure Time-Stamping of Digital Documents
5,189,700	1993-02-23	R. R. Blandford	Devices to (1) Supply Authenticated Time and (2) Time-Stamp and Authenticate Digital Documents
5,373,561	1994-12-13	S. A. Haber W. S. Stornetta	Method for Extending the Validity of a Cryptographic Certificate
Re.34, 954	1995-05-30	S. A. Haber W. S. Stornetta	Method for Secure Time-Stamping of Digital Documents
5,422,953	1995-06-06	A. M. Fischer	Personal Date-Time Notary Device

5,001,752

This patent presents the design of a device that includes a digital clock, a secure storage medium, and a data processing engine. With these sub-systems the device is capable of time-stamping digital documents that are going to be transferred electronically.

5,022,080

This patent describes a device that can be used to time-stamp digital documents. Although the patent mentions the use of a simple CRC generator, without non-invertibility and collision free properties, the proposed scheme is very similar to the centralised time-stamping protocol, without linking information.

5,136,646

This patent introduces the centralised time-stamping protocol with linking. This patent does not describe any kind of device to perform the time-stamping of digital documents, but claims the method.

5,136,647

This patent introduces the centralised time-stamping protocol without linking information.

Re. 34,954

Reissue of patent 5,136,647, including additional claims.

5,189,700

This patent discloses a device that is able to receive digital information (either a complete document or a hash), combine it with the data or the time when it was received, and encrypt the whole receipt to produce a time-stamping certificate.

5,373,561

This patent discloses a method to extend the validity of time-stamping certificates. The method is described in the renewal section.

5,422,953

This patent describes a small device, intended for implementation in smart cards, to provide digital-time notarisation of digital signatures.

10.3 IMPLEMENTATIONS

In this section we will briefly comment the currently offered time-stamping services.

10.3.1 Surety Technologies Inc.

This company offers a digital notary service that commercially provides time-stamping services to a wide community of customers in the U.S. (http://www.surety.com)

It improves the linking protocol described above in several ways:

- it uses a tree of digests of individual documents to time-stamp in order to speed up the hash evaluation process, and establish a harder to fake relationships between the different documents
- a super-hash is evaluated every second, incorporating all the digests of all the documents received during each period, and it is linked to the previous super-hash
- the trees of hashes, with the identity of the requesters is published in a CD_ROM to affiliates; a database is available on-line for verification, and every week, the last super-hash is published in a well-known newspaper

10.3.2 Stamper E-Mail Time-Stamping Service

In the UK there is a consultancy firm (http://www.itconsult.co.uk/stamper.htm) that provides a free time-stamping and proof-of-posting service via the Internet. It uses email and the PGP (Pretty Good Privacy) cryptography application. This service works as follows:

1. The user sends a document via e-mail to the service robot address (post@stamper.itconsult.co.uk), indicating at the top of the message body the addresses of the final message recipients (the syntax is described in the indicated Internet page).
2. The robot attaches a text stating the time when the message was received, a monotonically increasing (in steps of one!) increasing serial number, and the final intended recipient of the document.
3. The resulting message is signed using the PGP public key of the time-stamping service.
4. The signed message is sent to the originator and to the final recipient.

The serial numbers of the last document signed each day are listed on a public database. Since the serial numbers increase monotonically and in steps of one, no undetectable document insertions can be performed in the future.

10.3.3 TicTac

This experimental service, offered by the cryptography laboratory of the Computer Science Faculty, Polytechnic Madrid University (Spain) in the Spanish language, enables the user to time-stamp any desired document via a WWW interface with a wide variety of hash and digital signature options. It can be accessed as http://tictac.fi.upm.es.

11 APPENDIX D: SECURE TIME

11.1 GENERAL CONSIDERATIONS

The design and operation of a Time Stamping Service must previously address all issues, amongst many others, relating to the selection and maintenance of the time bases used by the server or servers who are providing the service. Some of the most important points to be considered are:

a) Characteristics of the TSA server's internal clock.
b) Criteria to be followed for the selection of the master clock which will provide the time reference for the TSA server's internal clock.
c) Procedures and protocols for the synchronisation and broadcasting of time signals between time sources and slave clocks.
d) Criteria to be followed in token stamping with regard to formats and the time reference.
e) Problems deriving from the coordination of TSA in supranational geographic areas, that is to say, the E.U.
f) Secure conditions for the TSA server's internal clock.

Each of the issues mentioned in the previous paragraphs may have different solutions and approaches that will need to be analysed in order to identify those best suited to the requirements of the different usage scenarios described elsewhere in this document. In this regard, for each of the five issues mentioned, an analysis has been made of the most significant elements that would have to be considered.

11.2 SERVER'S INTERNAL CLOCK

Conventional commercial computers use two types of internal clocks: those based on software and those based on hardware. Extensive use is made of the latter which contain cheap, inaccurate and unstable oscillators. These are known to run either fast or slow up to five or six seconds a day, which rules them out for use in Time Stamping services. In order to avoid this problem it will be necessary to replace the server clock with a more accurate and stable clock. Many solutions have been developed and are on the market. These are often high precision oscillators built into cards with easy "plug in" to the server's hardware. These clocks can give the server quite a high degree of accuracy and stability depending on the cost of the card. However, in addition to having an accurate internal clock, it will be necessary to have an external, reliable time reference source to update this clock periodically. In other words, a master clock.

11.3 CRITERIA FOR THE SELECTION OF A MASTER CLOCK

The basic criteria to be used in selecting a master clock which will serve as a reference for the TSA server's internal clock, are the following:

- its accuracy and stability
- its accessibility

As far as accuracy and stability si concerned, the state of the art recognises that it is the atomic clocks which best satisfy both of these requirements. To a great extent they have replaced the traditionally used precise time sources based on astronomical observations and on the movement of the Earth. These clocks provide the reference which serves as the basis for different institutions world-wide - generally Meteorological institutes and other private and public organisations - to offer the scientific community and the business sector access to the so-called Primary Time Standard signals. Atomic clocks are based on the extraordinary stability of the radiation frequencies of atoms such as caesium or hydrogen. These properties have enabled

the elemental unit of the universal system of time measurements to be redefined. For example, the second is now defined as the duration of 9.192631.770 periods of the frequency, which matches the transition between two hyperfine levels of the ground state of a Cesium-123 atom. As an interesting point, the accuracy of these sources is as much as one billionth of a day, almost a hundred times greater than that of the best astronomical clock. Although precision such as this is only required in exceptional circumstances, such as in high-level research environments (Particle Physics, etc.), it should, on account of its stability and accessibility, be the most widely used in TS services. Nevertheless, other options will also have to be considered for certain cases, such as certain applications which might require the astronomic time reference based on observations of the movement of the stars.

The policy for the use of these sources will be related to factors such as the accuracy and stability of the server's internal clock, and other questions such as the policy on the use of sources located in the country of residence of the TSA, or, as in the case of the countries belonging to the EU, a Master time source which will be unique to the Community and which will duly be determined as a compulsory reference. For information on what these sources might be, the attachments include several references to the major time sources and master clocks available world-wide.

11.4 PROCEDURES USED TO DISTRIBUTE MASTER CLOCK TIME AND SYNCHRONISE THIS WITH THE TSA'S INTERNAL CLOCK.

Synchronisation between the master clock and the server's clock can be done using different procedures, each of which utilises very different resources and protocols. These, in turn, will mean highly variable costs and results as regards the security and accuracy of the adjustment obtained.

The synchronisation procedures most widely used are:

- Synchronisation by connecting to the source via Modem
- Synchronisation by connecting to the source via Radio
- Synchronisation by connecting to the source via Satellite
- Synchronisation by connecting to the source via Internet

11.4.1 Synchronisation Via Modem Through A Telephone Line

There are many institutions and organisations offering this service. The internal time of a computer or PC can easily be updated and synchronised with the time of a precise master clock by using a modem and some low-cost and simple to use software. When the telephone link is made with the source, the source sends its reference time encoded in ASCII characters. The synchronisation program takes into account the time lag produced between the information output from the master clock and the arrival of this information at the slave clock. This time is in turn made up of two parts: one which corresponds to the transmission time and one which refers to the time that the modem takes to process this information and input it to the computer. This time lag is calculated statistically and is added to the master clock time before it is sent so as to make up for the lag that would otherwise be produced. A typical time lag could be around 45 milliseconds, although if a satellite is used as a means of transmission this could be as much as 300 milliseconds. As we have said there are many sources offering this type of service and many software programs that may be used to perform synchronisation with ease. The accuracy that can be attained by this procedure can be as great as only 1 millisecond off the source time. Such accuracy is normally quite a bit higher than the maximum given by a PC's internal clock.

11.4.2 Synchronisation via radio signals

Another option that is widely utilised to synchronise a computer clock with time from a master source is provided by radio clocks, which are capable of receiving the time signals that are

broadcast by different stations specialising in this type of service. These clocks, depending on their accuracy, may cost between $500 and $5000 or more. They are connectable to different types of interface such as RS-232, IEEE-488 and others, and they are also available as plug-in card extensions connectable to the bus. The radio stations can be located by means of an antenna that should be put up outside. These stations broadcast voice signals, generally at one-minute intervals, and also binary encoded time signals that are captured and decoded by the computer's radio clock. The accuracy of this procedure varies depending on the type of station selected and its primary source of reference, as well as the broadcasting frequency used. For example, using stations such as WWW, and WWWVH, which are very inexpensive solutions, the synchronisation accuracy is in the range of 1 to 50 milliseconds. If you use signals from a low frequency WWVB-type station the price of the radio clock would increase, but the accuracy of the synchronisation would be as high as 0.1 milliseconds.

A further method of obtaining master time via radio is by means of the GPS (Global Positioning System) satellite system which will be mentioned below.

Among the broadcasting stations offering time signals and which can be used in Europe, we have:

- TDF in France
- MSF in the United Kingdom
- DCF77 in Germany
- The BBC also provides radio signals with time signals.

All of these provide different types of services generally referring to UCT and generated by atomic clocks.

11.4.3 Synchronisation Via The Use Of Satellites

This is a system that is widely used due to features which make it specially interesting. The most widespread is currently the G.P.S. (Global Positioning System). This is a system which, as opposed to others, can be used on a world-wide scale since it uses a constellation of 24 satellites which operate in 12-hour orbits and at an altitude of 20,183 Km. Among them there are at least five which are always visible to any user from any point on the planet. These satellites broadcast different types of signals which carry information for different uses: basically to estimate the position of a moveable object (a ship, a land vehicle or an aeroplane) and also to broadcast accurate time signals. These signals may be picked up easily by a small, simple antenna anywhere in the world. The time reference is UTC, and as master signals they use several caesium atomic clocks. The accuracy of the time signal broadcast is around 340 nanoseconds for uses of the SPS-type (Standard Positioning System), and 100 nanoseconds for specially authorised users who use the PPS (Precise Positioning Service).

Although this is a project originated by the military and sponsored by the U.S. Department of Defence (DoD), designed and operated by the military, it is used by many thousands of civilian users all over the world.

As alternatives we should mention GLONASS SYSTEM TIME, a satellite navigation system controlled by the government of the Russian Federation. Its master time sources are cesium and hydrogen clocks. It broadcasts information on time with an accuracy of one microsecond, and these atomic clocks get their seconds updated every time the U.T.C. does. The European Union is also developing a new navigation system which will be valid for the broadcasting of precise time signals: the GNSS1. The GSS2, which will contain significant technological improvements, is also being developed.

11.4.4 Synchronisation Via Internet.

The Internet can also sometimes be a suitable means of synchronizing a PC or other computer clock from a master source, and for this there are many institutions offering solutions easy to install and operate. However, this solution offers less security than the use of a connection through a telephone line and a modem, since it is common knowledge that it is much more difficult to estimate the delays produced on the network than those which arise in a telephone link, and it is therefore more difficult to introduce a correcting element with precision.

For these reasons the maximum accuracy that can be guaranteed by this method has a limit of one second.

11.5 PROTOCOLS FOR CLOCK SYNCHRONISATION IN A NETWORK

11.5.1 Functional layers for secure time

The availability of a secure time source, which delivers secure time information, is essential for a TSA to generate reliable timestamps. The aim of this chapter is to study the requirements of a secure time source, i.e. of secure time generation and of secure time distribution. Here the terms “secure” and “security” are used in a broad sense. Time may be unreliable no only because of malicious replay attacks but also because a lack of stability of the source.

Generally speaking, the clocks of today’s computer systems are not accurate enough. External time generators are used to provide accurate time services. The transmission of time information from the generator to the computer system requires special means to minimise delays and synchronisation errors. The time information can be replicated to other systems once it is available in a computer network (either a LAN or a WAN) but erratic delays have to be avoided and the accuracy and the integrity of the information have to assured.

TABLE 11.5.1

Functional layered model for secure time

Functional layer	Description	Some security aspects
Time generation	Master clocks, primary reference sources	- Stability of frequency, accuracy of time signal and availability are critical aspects
Time dissemination	Telecommunication systems connecting the master clock and time servers (upper level)	- Security aspects of public telecommunication systems and networks
Time servers (upper level)	Primary servers which are directly synchronised to master clock time signal	- Computer system security aspects
Synchronisation and diffusion	LAN or WAN connecting primary servers and servers in middle layers	- Local network security aspects - Internet security aspects - Security aspects of public telecommunication networks
Time servers (middle levels)	Servers which are synchronised to clocks in higher level servers	- Computer system security aspects
Synchronisation and diffusion	LAN or WAN connecting servers in middle levels to servers in middle or lower layers	- Local network security aspects - Internet security aspects - Security aspects of public telecommunication networks
Time servers (lower level)	Server synchronised to clocks in intermediate servers	- Computer system security aspects
Time stamp	Primary TSA function	

Table 11.5.1 summarises the functional layers of a typical time source of today from generation to usage of time. Basically an oscillator and a counter (the clock) generate the time information. This information is then transferred to the first level of time servers which are directly synchronised to the time generator. The time stamping function can be directly attached to a first level time server. This function can also be associated to a time server of an intermediate or lower level. Time information is passed on along the chain of time servers. Synchronisation among time servers is necessary because of the lack of clock stability as well as of the variable delays and network latency.

In table 11.5.1 the layers of intermediate and lower level time servers as well as the corresponding synchronisation layers, are are optional. The brief description of each functional layer refers only to the main function of the layer (for instance, it is not mentioned that a time server may conduct time format conversion as well). The security aspects which are most relevant to each layer are also highlighted on the last column of the table.

In the following paragraphs the security requirements of a secure time source are checked for each functional layer of table 11.5.1.

11.5.2 Primary reference sources

The time source for a TSA to gain the highest legal recognition, has to be the national time authority or another recognised national or international time institution. These bodies and institutions are national standard laboratories and are responsible for provision of official time. There are some 60 time standard laboratories and about 100 atomic clocks in the world. They provide information for the BIMP (Bureau International de Poids et Mesures) to calculate the TAI (International Atomic Time). The International Earth Rotation Service (IERS) adjust the TAI to the earth rotation by adding a second (called the leap second) when necessary. The modified TAI is known as the UTC (Co-ordinated Universal Time) which has been accepted world wide as a time reference since 1972.

Examples of national time standard laboratories are the following:

- CIS Main Metrological Center of Russian Time and Frequency Service (VNIITRI)
- France Laboratoire Primaire du Temps et des Fréquences (LPTF)
- Germany Physikalisch Technische Bundesanstalt (PTB)
- Italy Instituto Elettrotecnico Nazionale “G.Ferraris”
- Spain Real Instituto y Observatorio de la Armada (ROA)
- U.S.A. Directorate of Time, U.S. Naval Observatory

Security requirements are a major concern in the time standard laboratories. They are at the highest level of security. Through redundancy in oscillators and systems they achieve a high level of availability. Accuracy and stability (less than 100 picoseconds of drift in a day) are the highest of today’s technology and industry standards (see standard industry classification of clocks according to its stability, accuracy, precision, etc. [ALL74])

For certain time services a less secure and reliable time source may be sufficient. However, a TSA using a primary time reference different from those mentioned above (i.e., using a local clock or its own clock) should check that the availability, the accuracy and the stability meet the requirements of the time services to be provided by the TSA.

A special case of time source is the network of clocks used in modern isochronous telecommunications networks. In these networks each oscillator is phase-locked to a single frequency standard. The networks based on the ITU-T recommendations on the Synchronous Digital Hierarchy (SDH) transmit bits streams at speeds over a Gigabit per second. The time sources and synchronisation mechanisms of the SDH ([ITU97], [ITU96], [ITU93]) can certainly fulfil the requirements of a time source for a TSA. It is not clear, however, that a telecom operator will accept an external TSA to access and to use the time synchronisation network and its time information.

The positioning satellite systems constitute presently the most accurate time source available everywhere on the earth. These systems can be considered as time sources as well as and time dissemination channels. There are two working systems: GPS ([KAP96]) controlled by the Department of Defence (DoD) of the USA and GLONASS controlled by Russian Space Forces of the Russian Federation Government. A regulated quasi-civil satellite navigation system (GNSS1) is currently being developed in Europe and fully civil-controlled system (GNSS2) has been proposed ([EC98]).

GPS consists of 21 satellites (+ 3 spares) in 12 hour MEOs (Medium Earth Orbit, i.e., at around 20,000 km from earth) distributed in six orbits. Satellites carry atomic clocks onboard which are synchronised with the Master Clock at the US Naval Observatory (USNO).

At least four GPS satellites are always in sight. The distances of any point to three of the satellites determine the co-ordinates of the point. The user receiver computes the distance to the satellites by measuring the transmission delay of a pseudo-random code (prn) sent by the satellite. The user receiver does not need an accurate clock because time data sent from the satellite keep the clock on track. The fourth satellite is used for 3D measurements or, if the altitude is known, to reduce errors. When a more precise co-ordinates are needed, differential positioning techniques can be used. This is basically to correct bias errors of the measurements in location L with known errors at a known location M. Of course L and M are using the same satellites.

GPS provides two level of service: Standard Positioning Service (SPS) and Precision Positioning Service (PPS). The PPS is a restricted service. The satellite's prn and navigation data are encrypted (anti-spoofing) to avoid fake transmission and masquerades of GPS satellites. A position accuracy of about 20 meter can be achieved using the PPS (between 1 mm to 1 cm when using differential techniques).

The SPS is similar to the PPS but the navigation data and clock frequency of the satellite are manipulated in order to reduce the general accuracy of the system. This is called Selective Availability, i.e. the denial of full accuracy, and has been imposed by the US DoD. The accuracy of this service is of about 100 m.

The GPS time was started on 17th June, 1990 at 00.00 h. The time transfer accuracy of SPS is of about 340 ns and can be of about 10 ns when differential techniques are used. Leap seconds are not added to GPS time which is now ahead of UTC by twelve seconds.

GLONASS works under similar principles. The 24 satellites are distributed in 3 orbits instead of six and they are higher in the sky (about 25.000 km). There is no Selective Availability. Time accuracy is of the order of 1 microsecond. Leap seconds are added automatically (no need for leap corrections as in GPS). There is a difference of three hours between GLONASS time and UTC.

11.5.3 Time dissemination

The time information reaches the first level servers via a telecommunication systems. Table 11.5.3 presents the various systems used today for time dissemination and some security remarks.

TABLE 11.5.3
Telecommunication Systems for Time Dissemination

Type	Security issues	Systems
Centrally controlled systems and networks	- Routing of time information is known and controlled by a central network management authority - Low protection against risks of tampering, modification, replay - Constant system delays and network latency - A variety of protocols are used: from a simple time scale and format definition to NTP - An accuracy of about 10 ns can be achieved at time server clock	Radio systems - Broadcast (VLF, LF, HF, UHF) - Satellite (VSAT, etc.) - Positioning and Navigation systems (GPS, GLONASS, LORAN-C, OMEGA, etc.)
		Leased line + modem
		PSTN (Public Switched Telephone Network) + modem
Unmanaged open networks	- Routing of time information is unknown and unpredictable - Totally unprotected against the risks of tampering, modification, replay - Unpredictable system delays and network latency - NTP is the most common protocol in use - An accuracy of some 1 ms can be achieved at time server clock	Internet

As could be expected, the accuracy of clocks is degraded from some 100 picoseconds to 10 ns (1 ms if Internet is used and 1 s if radio broadcasts systems are used). This a general rule in Table 11.5.1, the lower the layer of the time server the lower the accuracy of its clock.

In general, time information is transmitted in an open format (or as plain text) and there are no authentication mechanisms of the time generator. Some of the centrally controlled dissemination systems that are mentioned in Table 11.5.3 can be considered secure enough for most of the usual time services provided by a TSA. However, the possibility of modification of time information through tampering exists even when there is a central authority controlling the telecommunication system/network that is used for time dissemination.

The standard time laboratories can also provide the frequency of the oscillator used to generate the time. At first look this could be a solution because tampering and replaying frequency signals is difficult and would only affect the time information and the synchronisation in the long run. But in fact this solution only transfers the security problem of a reliable time generation and dissemination to the time server, i.e. to the TSA.

A better solution is to connect the time server to two (or more) time information sources from two (or more) time generators through two (or more) dissemination paths. The combination of the time information from the various sources could use the algorithms of the NTP (see 11.5.5.2 below) for synchronisation of local clock.

This solution can be applied in cases where the highest level of security is required. In the EU, for example, for certain scenarios of high security it could be requested that least two national time standard laboratories be involved in the time stamps generated by a TSA.

The security problems in disseminating time information through Internet are the same as the synchronisation and diffusion of time among time servers and is dealt with in paragraph 11.5.5.

11.5.4 Primary Time Servers

These servers synchronise their local clocks directly from the time information received from the standard time laboratory, they diffuse and synchronise other time servers, they can generate the time tokens for the time stamp (including changing time scale and formats as required) and can be part of the TSA. Basically, they are computers systems with particularly accurate clocks.

The accuracy of the local clock should correspond to the type of time services to be provided by the TSA and should be in any case lower then the maximum accuracy allowed by the mechanism of synchronisation with the time generator.

The computer system of a primary time server should meet the security criteria corresponding to the services provided by the TSAs using this time server. Levels of security evaluation of ITSEC E3 and E4 should be considered for highly reliable TSAs.

In the context of the EU the levels of accuracy and security should allow cross synchronisation and diffusion of time among countries.

11.5.5 Sychronisation and diffusion.

11.5.5.1 Time protocols

This paragraph and the following paragraphs deal with protocols to diffuse time and synchronise clocks in an open network without a central administration authority, i.e. in Internet, although the protocols can be used in other type of environment (LAN, WAN, telecoms time dissemination systems, etc.)

The timestamp in the Internet Protocol (IP, [SU81]) and the timestamp message of the Internet Control Message Protocol (ICMP, [DAR81]) are some of the predecessors of today's time protocols.

The Time Protocol ([POS83a]) and the Daytime Protocol ([POS83b]) were developed before the NTP. Both protocols allow automatic synchronisation. They are only appropriate for LAN environments because they do not support any kind of compensation for the transmission delay between the server and the client. As a consequence the accuracy achieved with these protocols in an Internet environment is too low.

The Network Time Protocol (NTP) is widely use today for synchronisation and diffusion in the Internet world. RFC for versions 1 and 2 ([MIL85], [MIL89]) were released as earlier as 1985 and 1989. Today's version (version 3, [MIL92]) was released in 1992.

NTP version 3 has many points in common with the Digital Time Service Protocol (DTS, [DEC89]). DTS is more appropriate for LAN environment where it achieves a high degree of accuracy and synchronisation due to a centralised network management. The robustness and recovery capabilities of NTP make it more appropriate for an unmanaged environment like Internet where this protocol can also achieve high levels of accuracy and stability.

11.5.5.2 The Network Time Protocol (NTP)

11.5.5.2.1 Overview

The structures and models proposed by the NTP fit well in the general schema of functional layers in Table 11.5.1. The primary reference sources, i.e. the primary time servers, are synchronised (via the time diffusion system and network) to the national time standard laboratory. Secondary time servers are synchronised to the primary time servers an so on. NTP allow up to 256 levels (called strata in NTP) of time servers although practical implementations reduce this number to stratum 3 or stratum 4 (in general less than 16 strata).

We quote the RFC 1305 "*The purpose of NTP is to convey timekeeping information from these servers to other time servers via the Internet and also to cross-check clocks and mitigate errors due to equipment or propagation failures.*"

The NTP timestamp can achieve a precision of 200 picoseconds but the practical implementations of the protocol reduce the accuracy of the synchronised clocks to about 100 ms.

NTP counts the time since 00.00h of 1st of January 1900 and its timescale arrives up to some 136 years. This means that sometime in the year 2036 the counter will re-start at zero.

Synchronisation is achieved through the exchange of time packets. NTP procedures and statistical algorithms compute clocks offset (differences between local clock and reference clock), roundtrip delay (message delay between local and reference servers) and dispersion (maximum error of local clock relative to reference clock). Other NTP procedures and algorithms use the offset, roundtrip delays and dispersions to separate truechimers (an accurate and trusted clock) from falseticker (inaccurate and untrusted clock) and the to (smoothly) adjust the local clock.

Two or more time servers must first establish an association in order to exchange time packets. The type of association is controlled by the mode of operation of each server. The correspondent server in an association is called the peer. The following association exists in NTP:

1. Symmetric. The server will synchronise the peer and accept being synchronised by the peer.
2. Client. The time server accept being synchronised by the peer and will not synchronise the peer.
3. Server. The time server will synchronise the peer y will not be synchronised by the peer.
4. Broadcast. The server will synchronised all their peers and will not be synchronised by any of the peers.

The modes of operation allow the construction of a synchronisation subnet of any topology. The only restriction is that no synchronisation loops can be formed. An appropriate synchronisation network will reduce overhead and facilitate the cross-

cheking of clocks.

11.5.5.2.2 NTP Security Issues

The NPT uses two mechanisms against attacks such as malicious modification (tampering) or destruction (jamming) of time information.

The first mechanism is the introduction of redundancy in the time source (as it was also suggested for the time dissemination). A time server can (and shall) have several associations with other servers and cross-check the time information received from all of them. Some NTP statistical algorithms filter and select the appropriate time servers and detect the falseticker or erroneous server.

The purpose of the second security mechanism of NTP is to avoid the masquerade of the time server, i.e. to supplant its identity and to provide false time information. An optional authentication schema is proposed in Appendix C of the RFC 1305. The time packet is signed by a crypto-checksum using DES and one key of a set of keys previously distributed among the time servers. The key distribution procedure is out of the scope of NTP. The keys are 64 bytes long. Only the DES encryption algorithm (chipher-block chaining method) is used.

The authentication schema proposed by NTP is not sufficient for the requirements of highly reliable TSA. The NTP needs to be complemented with other more powerful security mechanisms.

11.5.6 Time servers intermediate levels and lower level

These two types of time servers synchronise their local clock from the time information received from other time servers and they may diffuse time and synchronise other time servers, they can generate the time tokens for the time stamp (including changing time scale and formats as required) and can be part of the TSA.

The discussion and the conclusions for Primary Time Servers (paragraph 11.5.4) apply also to the intermediate and lower level time servers. The accuracy of local clock and the ITSEC level of security of the systems should correspond to the type of time services to be provided by the TSA.

In the context of the EU the levels of accuracy and security should allow cross synchronisation and diffusion of time among time servers and TSAs of different countries.

11.5.7 Enhancements of time synchronisation and distribution

Time accuracy seems to be increasing through improvements and co-ordination of primary source systems (see [RAY98] for sub-nanosecond time transfer through satellite positioning systems). Accuracy of time of secondary sources is also increasing through improvements of clocks and synchronisation protocols. It can be said that enough time accuracy can be achieved today for the most demanding time stamping services.

However, for a TSA to relay on a secure source of time several security mechanism must be introduced at different levels:

- Generation and dissemination of time information: to have available and to use multiple time sources and master clocks, to guarantee the integrity of time data and the authenticity of times sources.

- Time servers systems: to establish appropriate systems security mechanism (ITSEC levels of security E3 or E4) and to assure local clock accuracy and stability as required by the time stamping services.

- Synchronisation and diffusion: to guarantee the integrity of time data and the authenticity of time servers.

It is then necessary to develop a new time dissemination-synchronisation-diffusion protocol and/or to enhance version 3 of NTP in order to guarantee the time source authentication and time data integrity.

This can be regarded as an opportunity for the EU to establish technical options and agreements to achieve full cross acceptance of time sources (national standard laboratories, time diffusion systems, time servers of) among countries.

11.6 TIME REFERENCES AND FORMATS FOR TIME STAMPING ON TOKENS

11.6.1 Time References

There are many different ways to access a time measure reference, and although conversion from one to another is almost always possible by means of a more or less complex formula, it can pose problems.

The master references most widely used for time are the following

TU	Universal Time, also called Zulu Time
IAT	International Atomic Time
GMT	Greenwich Mean Time
UTC	Universal Coordinated Time
LT	Local Time
TT	Terrestrial Time
GCT	Geocentric Coordinated Time
BCT	Barycentric Coordinated Time
ST	Sidereal Time

Although all these references have application among specific groups of potential users of Time Stamping services, the immense majority of these users utilise the UTC or the LT references, which are the most widely used in civil matters. Since in many cases it is likely that the user of these services will require his document to show either of the two references, the TSA should include a conversion table from the UTC obtained by the master clock, to his server's local time, and possibly also in some cases to the client's local time.

Neither must it be forgotten to take into account in the conversion process to local time, the advances and delays that many countries introduce in their local time in order to bring about an energy saving, and which generally takes place twice a year (daylight saving time). A further factor that in some cases would have to be taken into account by the TSA on issuing the stamped token, is that which would be posed by certain users operating in countries using calendars other than the Gregorian calendar customarily used in western countries. Such would be the case of the Islamic countries, for example.

11.6.2 Time Stamping Formats Used On The Tokens

The proliferation of different formats to represent the time and the date has been and still is a problem in communications among different applications and among different countries via the telematic networks. Among the efforts made to solve this problem, mention should be made of the international standard ISO 8601 which has obvious advantages over other notations used to represent the time. In addition, this standard has been adopted as a European standard, EN 28601 and it is therefore in force in all the EU countries. A further advantage is that it enables local time to be expressed easily from UTC time. Notwithstanding the above, and even though ISO 8601 is currently the most widely accepted time representation standard, additional efforts are being made to improve some of its aspects. For example, a better adaptation to the needs of Internet users, in order to minimise the possibility of confusion among users, and better interoperability among the devices which exchange time data on the Net. In this regard the IETF is drawing up a Draft which proposes a profile for the ISO 8601 standard containing interesting subtleties on the representation of local date & time, in addition to other aspects associated to time representation in E-mail messages on the Internet, and also with the NTC protocol. This profile, which would be a subset of the ISO standard, will also attempt to create a formal grammar so as to prevent, as we said, certain ambiguities present in the standard, and

which would simplify and facilitate its use. See ref. draft.newman-datetime-o1.texen-ietf.org (FTP).

11.7 SECURE CONDITIONS FOR THE TSA INTERNAL CLOCK AND SERVER

Both Time and Date must be obtained from a clock. The clock values, as it is placed inside a tampering proof facility, are impossible to modify or reset without authorisation. With the exception of its setting in synchronism with other master clocks.
It has to be explained that the above-mentioned facility at which the clock is placed, may perfectly be an armoured safe box, with the physical characteristics to resist any criminal attempt, and two keys to unlocked it.

It is also advisable that the time values are hidden to the human eye. Thus, in case anybody might attempt to change the time value for fraudulent purposes, as a document time stamping, it would add more difficulty.

Usually, time and date reading, is permitted in stamping operations at which signature algorithms are involved. Other operations are restricted to other people except those officers joining the time stamping authority (TSA). These operations should be performed (as in the case of Acs secret keys protection) by using data keys in which is possible to store digital data. Also, as they use secret sharing technology, the previous agreement of several people is required to make a number of privileged operations in the clock. Thus, any particular officer attempt to switch the clock values under not authorised circumstances is avoided.