2002 International Computer Symposium:

Workshop on Computer Networks

Title: A Time Stamp Service Framework with Application on the Copy Right Protection of World Wide Web Documents

Abstract:

Time stamp service is an important mechanism in the modern electronic information infrastructure. Time stamps in digital signatures, digital certificates, digital contracts, and digital documents provide proofs of the existence for that particular electronic information in that exact form and that particular time. In order to construct an efficient information infrastructure for electronic government, electronic business, and electronic commerce, a convenient time stamp service paradigm and environment is required. The overall time stamp service environment includes the time stamp server, various application softwares, and the network environment. In this paper, inter-operating mechanisms between the implemented time stamp authority and a specific Web Page Time Stamp application software are described. Two application paradigms of this time stamp service are implemented and discussed. The first application provides an trustworthy time information for each Web page. The second one employs time stamp service for the protection of copyright of publicized Web site materials.

keywords: time stamp service, public key infrastructure, copy right protection, World Wide Web

Authors:

Pei-yih Ting	pyting@mail.ntou.edu.tw	Tel: 886-2-24622192 6615
Chun-Yen Wang	b8554029@ind.ntou.edu.tw	Tel: 886-2-24622192 6643
Hsing-Chien Ni	enijmax@cyber.cs.ntou.edu.tw	Tel: 886-2-24622192 6643
Po-Yueh Hung	b88023@cyber.cs.ntou.edu.tw	Tel: 886-2-24622192 6643

Affiliation: Dept. of Computer Science, National Taiwan Ocean University,
Address: No. 2, Bei-Ning Rd. Keelung, 202, TAIWAN
Fax: 8862-24623249

All **correspondences** should be addressed to Pei-yih Ting.

I. Introduction

To establish legality of any electronic document, the application of digital signatures are required and the deployment of the underlying Public Key Infrastructure (PKI)[5] is currently a major issue in many countries.

Time information is an important ingredient of an electronic document. This piece of information is crucial for proper operations of many interactive protocols dealing with electronic documents. For example, for a document which entitles a person of a property transfer to be effective, the document is required to have a time mark earlier than the time mark on a second document which entitles the same property transfer to another person. For a digital signature of a compromized private key to be valid on a certain document, the signing time of the signature is required to be earlier than the revocation time of the private key. When documents are exchanged between different hosts or different applications, the timing sequence information again plays a key role in the operating protocol. For example, a receipt should have a time mark later than the corresponding payment time.

A digital time stamp is a piece of time-related information, denoted by (t, y), which shows a clear and unchangeable relationship between the time instant t and the corresponding electronic document y. Everyone can readily believe that the document y exists before time t. For any two time stamps $(t1, y1)$ and $(t2, y2)$, the

order of them on the time axis should be clear and unchangeable.

A time stamp authority (TSA) in the public key infrastructure plays a role of a trusted third party and provides proofs of the requested time corresponding to some existent documents in the form of time stamps. Users can request for a time stamp corresponding to a particular document through the time stamp protocol (TSP) [1]. Users can also search, verify, or compare time stamps.

Currently, business plans for most Certificate Authorities (CA) include primitive time stamp services. These time stamp services are used primarily for the supporting of digital certificates and for the certificate revocation lists (CRLs).

Many of them are not deployed as an independent service to satisfy the requirements of all sorts of applications that handles digital documents. Let alone some problems pertaining to the timing order comparisons of time stamps from different time stamp servers.

A valid and effective time stamp should satisfy the following three requirements:

1. there exists unalterable relationship to the corresponding document
2. a time stamp contains some reliable, nonmalleable, and suitably synchronized time information
3. the integrity and authenticity of a time stamp can be publicly verified

In order to achieve the first property, a time stamp usually includes the hash

value of the corresponding document, sealed by the digital signature of the time stamp server. By verifying the digital signature sealing the time stamp, property three can also be ensured. The second property is implemented with mainly two types of mechanisms. The first one utilizes a suitably calibrated time source tracking back to the Coordinated Universal Time (UTC). The second one keep records of all time stamp requests in an unalterable manner[6]. Each time stamp therefore contains a unique sequence number to identify its timing order.

II. Time Stamp Authority and Time Stamp Client Application

As shown in figure 1, there are two basic components connected through the

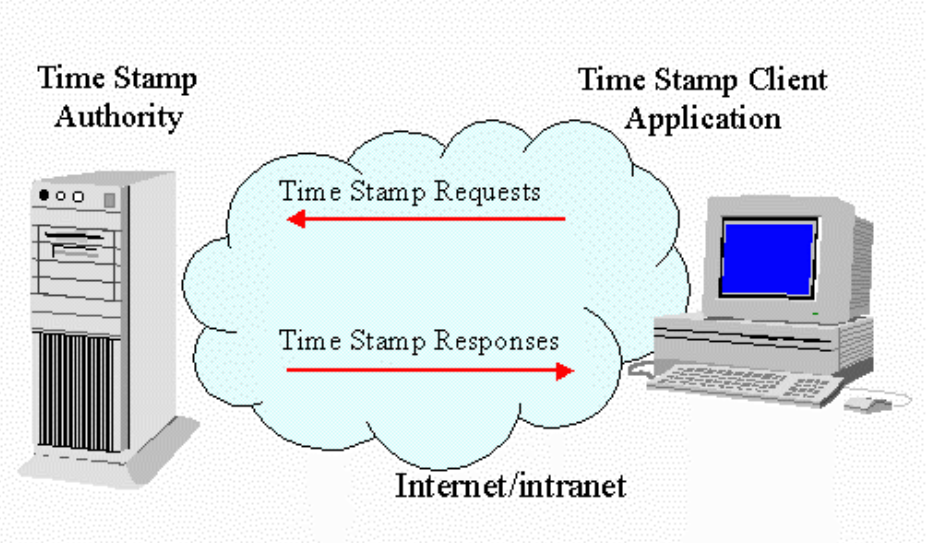

Figure 1: Time Stamp Service Framework

internet in a general time stamp service framework, the time stamp authority and the

time stamp client application. On the server side, TSA maintains the complete time stamp logging data and provides a set of services including time stamp request, time stamp verification, time stamp searching, time stamp reissuing, and timing order comparison, etc. On the client side, application softwares usually exhibit features such as key management, time stamp request and verification, time stamp storage and management. To obtain the time stamp of a particular document, we:

1. calculate the message digest of a document,
2. send the digest to the TSA server,
3. obtain the public key of the TSA server,
4. verify the integrity and authenticity of the time stamp by verifying the signature of the TSA server, and
5. save the time stamp with the document.

To verify the time stamp of a document, we

1. calculate the message digest of a document,
2. obtain the public key of the TSA server,
3. verify TSA's signature on the time stamp,
4. compare the stored message digest with the digest calculated in step 1, and
5. in addition to the above local check, client side software can request for a time stamp verification service, which invokes a mechanism to verify the

trackability of a time stamp.

There are several issues required to be elaborated in designing a time stamp service application framework. The first one is the physical distribution of each components. TSA server is always on a remote machine, it should use plenty of disk space and computation power to keep all traces that are required for ensuring the nonmalleability and uncorruptibility of all issued time stamps. TSA server should be unbiased in processing all time stamp requests. All responses exceeding some specified maximum delay from the requested time should be discarded. The client side service requesting software can reside on a client desktop, a notebook, PDA, or a handset. However, due to the limited computation capabilities of various IA products, the client side service requesting software can also be mounted on a proxy – a centralized time stamp requesting and time stamp managing utility. In the WWW environment, the centralized service requesting software can be implemented with the combination of a server-side CGI and a supporting database system.

The second issue is the storage and management of requested time stamps and the original documents. Although it is favorable to keep the time stamps together with the documents, in most current computer environments, the file system and most document processing application software does not directly support an additional time stamp information block. If one try to store the time stamp together with the original

document, the document processing application software normally will reject the compound document simply because of its illegal format. If one keep the time stamp separately from the document, it is very likely to have inconsistent time stamp later when one retrieves the document. A separate time stamp database indexed through the message digest and the time value is a reasonable solution at the cost of one additional database system.

A third design issue in a time stamp service framework is the format of the time sources. Basically the three types of time sources are absolute time sources, relative time sources, and hybrid time sources. A TSA server using absolute time sources incorporates a time value with suitable resolution into each time stamp. The time sources need to be synchronized to some standard time references through ACTS, NTP, GPS, or low frequency radio. The maintained clock values need to be strictly increasing. However, due to the limited resolution of the maintained clock values, two time stamps issued in a short period will be considered issued at the same moment.

The time stamps issued by a TSA with relative time source have a sequence number in them. This sequence number represents the time order of the issued time stamp. Sometimes, the time stamps also contains published well known nounce data such as the closing indexes of stock markets. In this way, the sequence number can

be correlated to actual time events. The comparison of the order of two time stamps in this case is trivially accomplished through the comparison of the sequence numbers. Currently, the above two kinds of time sources are usually mixed together to obtain advantages of both types of systems.

III. Application of the time-stamp service: the Web Page Time Stamp(WPTS) system

World Wide Web (WWW) has been a major information exchanging media since its first introduction in 1992. With various kinds of multimedia presentation enhancements, multitudes of interaction technique improvements and tons of tons of information contents, the WWW environment has proven itself to be the first real killer application on the internet. The essence of this successfulness lies in the convenience of content production and providing, and the effectiveness of the multimedia presentation. However, there are several inherent intellectual property right (IPR) concerns which keep bothering all content providers and the web surfing majorities. The first one is the copyright of the produced Web materials. Is there a convenient method to establish proof of the copyright of an originally produced electronic multimedia document? If a part of this document were found inappropriately on some other Web Sites or some other communication media, is there a convenient way to provide effective evidences of the original document's

ownership.

The second issue is the time effectiveness and the validity of the contents of a web document. For some years, most web surfing population are well trained not to trust the unconfirmed information from the web. Are there handy ways to restore the confidence of the web contents both on the instants of their appearances and on the sources of liabilities? At the same time, one is willing to publish and share information or knowledge with others as soon and far as possible, however, one is also afraid of being illegally copied or even being accused of replication after being copied. Over years, there have never been a simple and popular solution for this dilemma.

With the time stamp authority we implemented[2] and the corresponding services provided [2], a feasible and convenient solution to the above situation is proposed and evaluated as follows.

As shown in Figure 2a, the format of a web page is text-based HTML. Without backing up by any cryptographic means, the contents of this document are not legally endorsed by anybody. The instant of this document is not known precisely even if it were explicitly specified in plain text in the documents. This document is virtually free to copy and nothing seems to be copyrighted. Shown in Figure 2b, a block of time stamp information together with a verification JAVA applet are inserted in the

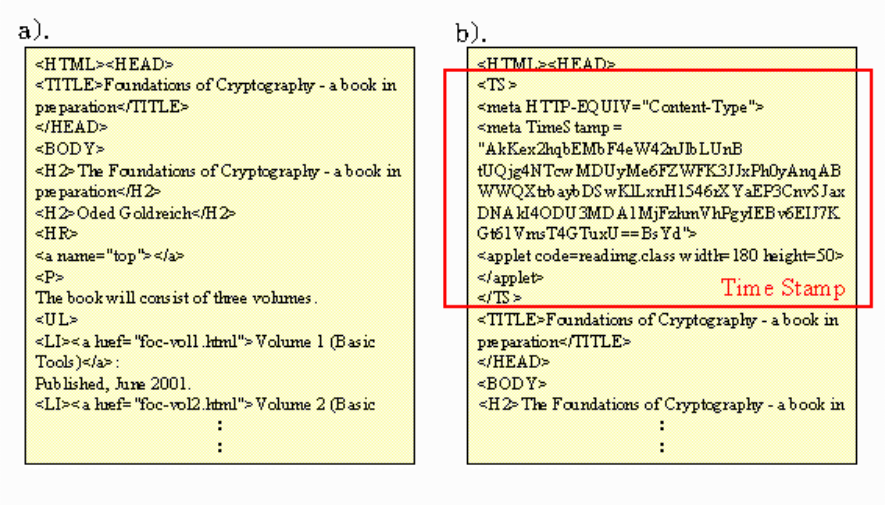

Figure 2: a) original HTML document. b) HTML document with time stamp embedded

header of the original HTML document by the proposed WPTS system. This time stamp proofs that this document was publicly known on the specified time with the exact content shown. This time stamp can be verified by the browser with a locally executed JAVA applet which is downloaded together with the web page. With the time stamp services provided by a TSA [2], this time stamp can also be online verified with the TSA. In this way, the exact instant of the appearance and the integrity of this web page are guaranteed. If the original document contains information of its authors, the owner ship of this document can be claimed. If the original document is digitally signed by someone with his legally effective private key, the validity of the content of the document is also endorsed by the person signing on the moment of the time stamp. The two steps in the WPTS system are as follows:

Time stamp requesting

The flow chart of the overall time stamp requesting client application is shown in

Figure 3. The client application first fetches the requesting web page using the http protocol, parses the HTML content of the requesting web page, fetches all the page's first level embedded figures and other media, and concatenates all downloaded materials to be a single compound document. The message digest of this document is then calculated with a cryptographic hash function. This digest, together with other identifying information [2] from the time stamp request which is sent to the TSA over the internet. TSA keeps record of the request and issues a time stamp back to the client. This time stamp is first verified with the TSA's public key and then matched with the message digest previously calculated. The Base64 encoded

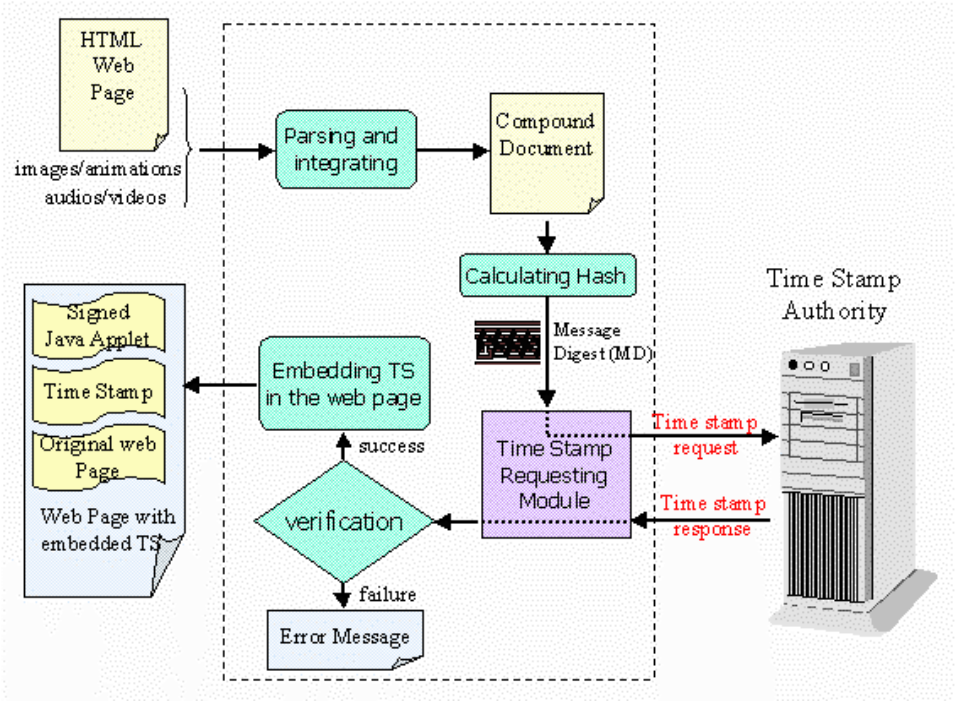

Figure 3: Time stamp requesting

time stamp and the signed verification JAVA applet are then embedded into the requesting web page.

Note that before requesting time stamps for all specified web documents, the web server need to be set up in its normal service mode. All the web pages and included multimedia files need to be fetched through the http port. This allows all the server side scripts (e.g. ASP VBScript, PHP, Server-side include, or Java servelet) and CGI included in the web page being executed normally. Also, in the parsing and integrating step, the web page authors are allowed to specify explicitly the parts of documents that are not owned by themselves such that they are excluded in the time stamped compound document. The client side application processes the requests of time stamps for a whole set of documents in the specified directory. The public key of the TSA can be online requested from a Certificate Authority (CA) using LDAP [6] protocol and verified through its certification path. In order to retain the appearance of the original document, the embedded time stamp is added to the header part of the HTML so that the browser can skip it. The java applet time stamp verifier is put into an overlaid transparent layer which is not formatted together with the original document.

Time Stamp Verification

As shown in Figure 4, everyone on the internet who browse the contents of this web

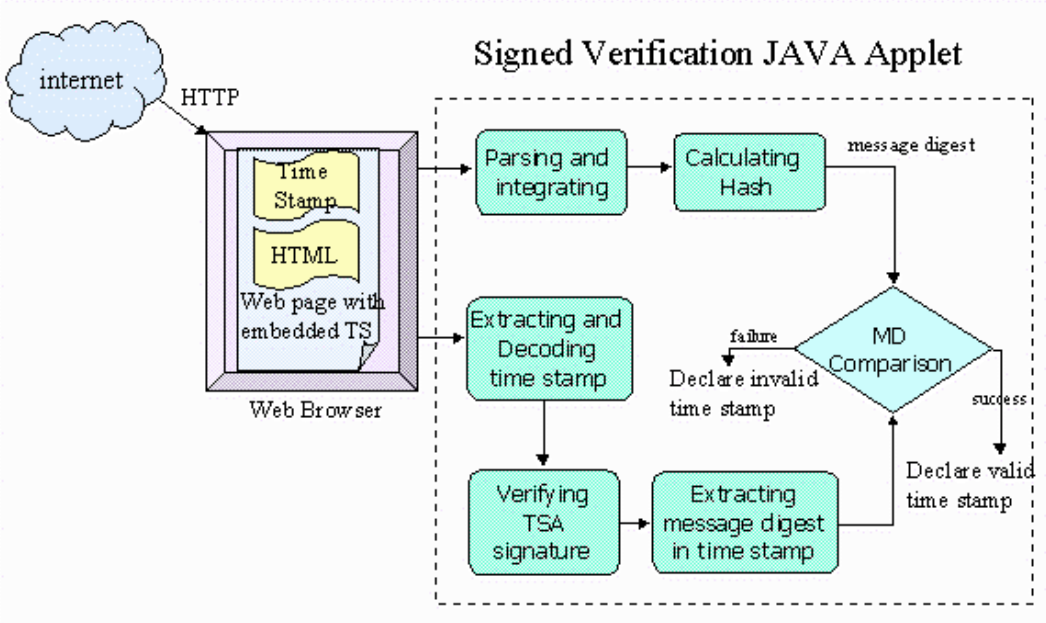

Figure 4: Time stamp verification

page can verify the embedded time stamp through the time stamp verification JAVA applet. It is a TSA signed JAVA applet. The browser environment will verify the validity of the code. The verification flow chart is described as follows: An internet surfer comes across the web document with time stamp. He notices the time stamp verification logo image shown by the JAVA applet. He double clicked this logo image and start the verification procedure. The web page is first parsed and all first level embedded images and media files are fetched and integrated into a compound document. A message digest of this compound document is then calculated. On the lower branch of the figure, the signature of the TSA is verified, the message digest of the document is extracted from the time stamp. The public key of the TSA can be requested from a CA and kept in the key ring of the web

browser. The digest is compared to the digest of the compound document to validate the time stamp. The above procedure is accomplished locally on the client machine, inside the browser, through a JAVA applet program.

IV. Discussions

In the copy right protection application framework, time stamp of a web page does not need to be verified by all parties who browse the information of the web page. In the current scheme, time stamp and the JAVA verification applet are sent along with the web page automatically to all requesting client browsers. The bandwidth can be saved if these time stamps are kept centrally on the web server, requested and verified on demand in a separate web page. The configurations are still very flexible and depend on the specific application framework.

In the previous section, we mentioned that to claim the ownership of an electronic document, an author is only required to put plain text author declaration in the document and requests for a time stamp on the overall document. In the case when some others make a pirate copies, make a few minor changes and claim for the ownership of the replicated documents, the original owner can easily provide evidences that the original document with the authorship identified is produced earlier that the replicated and partly modified materials. Although the originality of the

work is still left to be judged, a minimal self protection is already set up. The author's digital signature applied on the document has a stronger meaning that the author is legally responsible for all the contents the document contains, at least responsible to the level the policy the private key is entitled to.

In our approach, all multimedia files included by the top level HTML file are concatenated as a whole by default before requesting the time stamp. In some cases, where authors of web pages cite paragraphs, images, or media of others, it might be appropriate if these cited materials are not time-stamped together. In many cases, the advertisements embedded in the web pages are dynamically inserted and also need to be excluded. In the following, we show that the customized attribute 'owner' of the HTML tags 'img' or 'blockquote' can be specified by the author.

```
<img src = ”http://squall.cs.ntou.edu.tw/image/myImage.gif”
        owner = “Pei-yih Ting”>
```

In this way, the parsing module can decide which parts of the web document are to be excluded so that the protected range of copy right claim can be precise.

In the XML Encryption[3] and XML Digital Signature[4] working groups of WWW consortium, cryptographic formats are defined for the XML documents. Because time stamp can be treated as an intrinsic characteristic of an electronic document, it is important to define the basic XML time stamp format and its DTD. In this way, XML documents with time stamps can be processed by various

applications on different platform. Following the time stamp protocol in RFC 3161[1], the XML time stamp element is suggested as follows:

```
<TSTinfo version = 1.0>
    <policy>…….<\policy>
    <messageImport value = "ERfpLKq.....G/UD1cQj=" \>
    <serialNumber value = "LSdKxqG.....HUiJLNbY==" \>
    <genTime value = 20020308105926.56789Z \>
    <accuracy>…….<\accuracy>
    <nounce value = "HuIv……KqSoiD==" \>
    <tsa name = "tsa.cs.ntou.edu.tw" \>
    <extensions>……..<\extensions>
</TSTInfo>
```

An XML document has a single rooted tree structure. Each logical part of the document is modeled by a subtree. It is, therefore, possible to apply the encryption, the digital signature, and the time stamp to a logical subpart of the document. In this way, the granularity of the processing can be smaller, more consistent to the logical meaning of the content. Parts of the time stamped document, therefore, can be individually processed (copied, concatenated, archived, etc).

V. References:

[1] C. Adams, P. Cain, D. Pinkas, R. Zuccherato, “Internet X.509 Public Key Infrastructure: Time Stamp Protocol (TSP),” RFC 3161, August 2001.

[2] Pei-yih Ting, Chun-Yen Wang, “An Implementation of Time Stamp Service Mechanism,” the 6-th Symposium on Information Management Research and Practice, Dec. 2000. (in Mandarin)

[3] XML Encryption WorkGroup, http://www.w3.org/Encryption/2001/

[4] XML Signature WorkGroup, http://www.w3.org/Signature/

[5] Carlisle Adams, Steve Lloyd, “Understanding Public Key Infrastructure,”

Aug 1999, ISBN 157870166X
[6] M. Wahl, T. Howes, S. Kille, “Lightweight Directory Access Protocol (v3),” RFC 2251, Dec. 1997