Andreas L. Heuer, Frank Losemann, Christoph Meinel: Signed Preservation of Online References; in Proceedings of WebNet 2000 - World Conference on the WWW and Internet (WebNet 2000), AACE, San Antonio, Texas, USA, pp. 247-251, 10, 2000. ISBN: 1-880094-40-1.

Signed Preservation Of Online References

Andreas L. Heuer, Frank Losemann, Christoph Meinel
Institute fo Telematics
Bahnhofsstr. 30-32
54292 Trier, Germany
E-mail: {heuer,losemann,meinel}@ti.fhg.de

Abstract: Since online-available documents are more and more outclassing paper based documents, the use of online references is constantly growing. Online references suffer strongly from their often only temporarily character. With the slightest modification - not to mention a complete removal of its online references - the scientific value of the referencing document would decline. Therefore, mechanisms are needed that strive to preserve the value of the documents in digital libraries. In this paper, we will discuss the concept and prototype implementation of a service that can help to minimize this problem. The service produces files which contain signed data referring to the online documents. These files will be used as fall back online references. Offering this data-file analogously to the link that points to the orig i-nal online reference gives every user the opportunity to recover the respective online doc u-ment even if the original reference was modified or is not available anymore.

1. Introduction

Anyone is familiar with the serious problems that arise from the use of online documents in scientific work. The trouble with online references is their more or less dynamic character. An online document can get modified over time, can be moved to a different location or can even be removed all together. In either case, the reference pointing to that document becomes, in most cases, worthless. Although this problem is not restricted to the sci-entific field, it becomes very obvious here. The value of scientific work depends to a vital degree on its refer-ences. Only if a comprehensive summary of related work is given, can the author's intention be understood and the value of the work be determined. Furthermore, references help to prevent detailed descriptions of features that have already been discussed somewhere else extensively. The problem described here is recently becoming more urgent since online references become more and more common these days and outbalance paper based references.
In this paper, we will focus on these problems and will offer a quite simple but powerful solution. We will d e-velop a concept and describe a prototype that can be used to evaluate the concept.

2. Concept

In order to solve the missing online reference problem, we tried to find a simple mechanism that can be used commonly. Of course the additional effort for the author of a online document should be minimal. Only simpli c-ity and rapidness of the procedure will result in a broad application on the World Wide Web.
The concept that we developed in order to provide reliable references in online documents is based on a gateway service, that provides after any data transfer it handled a digitally signed [1] log of that transfer. With the signed data transfer log (SDTL) the user is afterwards able to prove the data transfer he did. If he offers those files to any other users, they can in hindsight reconstruct the data transfer. Since the log contains a signature, the cor-rectness of the log can be verified at any time.
A special scenario, where this mechanism can be applied is of course the authoring of online documents. So if an author wants to use a online reference for his online document he only has to use a commonly trusted gateway offering that service. Transferring the content of the online reference via the gateway gives him afterwards a SDTL. This SDTL will contain the request, the reply and the document itself as well as the signature.
Analogous to images linked in his/her online document the author now can provide the SDTL along with his document. Any link to an external online reference should then be followed by a link to the local available SDT of that reference. Now if the external online reference is later removed or modified for any reason, readers of the online document are still able to get the original content of the reference in form of the SDTL in a reliable way. It takes little effort for them to validate the SDTL. If the owner of the certificate given in the SDTL is trust wo r-thy they can now believe in the "copy" of the online referenced document.

2.1 SDTL-Conglomeration

The concept of the SDTL does not determine, how many "documents" with their corresponding transfer logs can be included in one single file. Of course it is possible to put into one single SDTL all components, e.g. images, that belong to the referenced document. Then a document with all its components would be packed together.
The next step is based of the thought, that the document D one wants to reference itself already references online documents X,Y and Z. Guessed the concept of the SDTLs finds broad use, the author of D would also provide along with D SDTLs for X,Y and Z. Now a packaged SDTL of D would also include the SDTLs of X,Y and Z.
Such an assumption implies several impacts. We would like to mention two of them. At first the SDTL's size could grow into bulky regions. At second the higher the depth of the reference chain becomes, the more the probability of duplicate entries rises.
Solutions for SDTL conglomerations can be found either on sensible policies that constitute a sensible depth of SDTL chains or in the concept of commonly accessible persistent services that is described in the employment section of this paper.

2.2 Public Key Infrastructure

Although not directly claimed, only a working PKI [2, 3] can provide the basis for the broadly introduction of SDTLs. The critical point, where a PKI is required is the verification of the Certificate of the trusted third part that runs the gateway. Furthermore a validation service [4] is needed. In both cases there are detours possible, but the elegance of a PKI based solution is not reached.

3. Prototype Implementation

In order to test the concept we implemented a prototype gateway program that offers the service described above. The gateway program is a multi-threaded server implemented in Java [5]. On the one hand Java was chosen, because of its platform independence. Coded in Java the program can be executed on most of the available platforms. On the other hand Java provides a powerful securit api [6] along with several tools [7] handling security relevant tasks.
Any client that wants to use the service must change the preferences and configure the browser to use the gat eway as a proxy. Therefore the gateway program listens on a given port for HTTP-requests [8] from clients. Those requests are forwarded to the destination web-servers. The corresponding replies of the servers are tran sferred back to the clients. Each combination of request and the respective reply, building a request-reply-pair, is logged by the program.
Of course the program has to distinguish between requests that came from different clients. Therefore an identification of the clients is required. The prototype identifies at the moment each client by its ip-address. This mechanism of course only works, if not several clients share the same ip-address. Further improvements ma result in a more sophisticated version of our prototype where implementations could for example provide a dedicated port for each client after registration for the service.
After a browsing session, e.g. consisting of the download of a document with the appropriate images, the client connects directly to the gateway. Since the gateway itself is addressed in this request, it will be handled in a different way. The server looks in its logs, temporarily stored in memory or in future implementations in a database, for all requests that came from that certain ip-address. It transfers to the client a list with all those requests from the last session. Since no registration was made, the session itself is no real session. At the moment equests of a convenient time period of some minutes are stored in the memory. The list displayed to the client in a HTML-page contains for each request a link that triggers the creation and delivery of a SDTL. Creation of a SDTL means that the request and the reply are signed and stored in a file that is then downloaded by the client.
After the download of a SDTL the user can provide it himself along with his/her online document that references the online version of the document contained in the SDTL. Although the prototype does not support that feature, it is planned to offer some kind of bundling of requests in a SDTL. Bundling in this case means, that if a online document consists of several parts, e.g. HTML-page plus some images, those requests all will be packed in a single SDTL. Then anybody that has access to the bundled SDTL is able to reconstruct the whole document in focus.
Of course the SDTL-files that can be downloaded from the gateway have a certain format. For the ease of i mplementation we choose the Jar-file format [9] as appropriate to be used in a prototype. The format offers two features. At first the Jar files are compressed, which is no problem, since the Jar format is based on the zip-format [10]. This is quite important, since it minimizes the place required to store the data. Furthermore it fasten the transfer of SDTLs through the network. At second Jar files were designed to encapsulate data and signatures in a single file. This offers two further advantages. At first there is no new specification to be developed that determines how any signature is to be stored in the file. At second there are tools [11] broadly available that allow the validation of the Jar file. This is very important insofar that in principle everybody has the ability to verify a SDTL.

Our prototype gateway creates the SDTLs in the following manner. It creates a jar file that contains originally five files. One file for the request header, one file for the request data, one file for the reply header, one file that is the requested document itself and finally the manifest [12]. Where the manifest is a kind of directory listing of the Jar file's content. After the creation of that Jar file the gateway uses its private key to sign the archive. The signing action results in two more files that are stored in the Jar file. Those files are the signature file [12] and the signature block [12].
As mentioned above in principle validation of the SDTL is possible for anybody, since the accordant software i distributed in a bundle with the JDK freely by SUN. The verification of the signed Jar file can be done with the jarsigner tool [11]. This is a command line based tool and therefore not as easy to handle as a Graphical Use Interface (GUI) based tool. Nevertheless one can with a simple command find out, if the signatures are correct.
The validation process consists of two task. The first task is the import of the certificate of the party that did the signing. This certificate has to be accepted as trusted by the user. In the terms of Java this means the user has to import the certificate with the keytool [13] into his/he keystore [13]. The second task is the actual verification of the signatures contained in the Jar file. For each stored file the jarsigner verifies the signature and finally makes a statement that tells if the complete Jar file was successfully verified. In short the user has to check the correct-ness of the certificate and the correctness of the signatures of the Jar file to be able to trust in the SDTL.
Although the prototype uses the Jar file format as an concrete implementation of a SDTL there is no reason, why not other formats can be developed and employed. Possible candidates could be derived from the XML-Signatur initiative [14] or the Signed Document Markup Language [14]. Depending on the requirements other information can be added to SDTLs, probably signatures given by the document authors themselves.

4. Employment Scenarios

At the moment we have two different employment scenarios in the field of digital libraries and therefore the provision of online documents on our mind. Both scenarios have in common, that the worth of a online doc u-ment is preserved by the provision of reliable, persistent copies of the online references used in that document. The scenarios differ with regard to the storage location of the SDTLs. On the one hand one can imagine a solu-tion where the author of a online document stores the required SDTLs himself. He/she places the SDTLs of all online references used in his/her document like images somewhere in the document-root near the document. Thi enables anybody the has access to the document to download the required references in the form of the SDTLs as well.
On the other hand it seems feasible to provide a commonly available persistence service fo SDTLs. This service could provide commonl SDTLs with a unique URI for each SDTL. In that case the SDTL is still generated as described above. Instead of a download by the client, a form could be presented that offers the user a 'contract'. In this 'contract' the service provider could commit itself to keep the newly generated SDTL at a given URI on-line available for a certain time. This time is equivalent to a guarantied lifetime of the reference. Therefore a expiration date for the online reference is given. About an extension of the period could be negotiated at an time.
For each SDTL at the persistence service a unique URI must be assigned. Any author referencing a online document could then provide along with the actual URI of the online available reference a corresponding link to the URI of the SDTL that is located at the persistence service of the service provider.
Both services, the signing gateway as well as the persistence service, would increase the value of any digital library. Therefore the parties offering the digital libraries could probably have great interest in the provision of those services them selves.

5. Usability

The concept that we suggest in this work is quite easy to handle from both sides, the service providers as well a the users. Most important is the fact, that there are no modification at existing systems required. Authors of o n-line documents have only to download the online available documents that they want to reference via the gat e-way service described in this paper. This enables them to request a SDTL that can be employed to prove the data transfer as well as the content of the downloaded documents. Online references in any newly created document are not eliminated, there is only an endorsement, the SDTL. Problems with a missing or modified online refer-ence can be solved by the employment of the SDTL. This file allows a simple reconstruction of the original document. Furthermore its authenticity is guarantied by the hopefully commonly trusted party that generated the SDTL.
Supposed that there will be tools developed for management of SDTLs the concept will not result in any higher additional effort than the effort required to manage a reference database with bibliographic content. In fact the SDTLs can increase the value of such a reference database dramatically. Next to the bibliographic information

and the URI of a reference there is also the document available. But not only available, its content is provable fo others. Therefore the references stored in this way are even more valuable.
While any private use of the SDTL will not cause any problems, a publication of the concerned documents must take the question of the original author´s copyright into account. Probably consensual agreements with the referenced authors provide a suitable solution for this problem.

6. Related Work

In this section, a short overview about similar concepts will be given. On the one hand, one can concentrate on systems that try to preserve links [16,17,18]. Their problem arises from the modifications that have to be implemented in the server software as well as in the used data transfer protocols. Furthermore, they cannot solve the problem that will occur when a linked document is not available online anymore.
On the other hand, we have to mention electronic notaries [19] in line with time stamping services [20]. The service we describe in this paper which is to be used in the field of digital libraries is a combination of an electronic notary and a time stamping service. In spite of the employment of separate services, we propose a single, integrated service. Furthermore, with the inclusion of the protocol header information into the signed log-file, additional value will be given. Its importance becomes obvious in cases like content negotiation [21] between the browser software and the web-server.
The most crucial feature distinguishing our concept from, for example, bi-directional links, is that no additional work for the author of any referenced document is implied. The concept does not require that any author digitall signs his or her document or maintains it in any other way. It is only the person wanting to preserve an online document - the document being a valuable online reference for him or her - that will have some additional costs. This person has to employ the signing gateway service and has to provide for the provision of the SDTL.

7. Outlook

As mentioned in several sections of this paper, at this stage, our work is still a combination of a concept and a prototype implementation. On the one hand, the prototype helps to evaluate the concept. On the other hand, it will probably result in - hopefully minor - changes of the concept that will encourage its broad application.
Of course, the usability of the concept requires a working PKI. In combination with GUI based tools that handle the management of the SDTLs, the employment of the concept seems possible.
Thus, there is a need for the conception and prototype implementation of a client which offers a slight automation of the SDTL management. Several features are planned for such a client program. In addition to the request of SDTL conglomerations, there should be features that allow simple validation of SDTLs. After the download of a SDTL, the client tool should provide fast and easy access to the included documents as well as the transfer information.
All in all, the SDTL management client can be implemented as a kind of personal reference database that can be enhanced with corresponding bibliographic information. Given the fact that a SDTL also contains the document in question itself, one could speak of a personal digital library. Since the content can be proved with the help of the trusted third party that signed the data transfer, the evolving local library is gaining value.
As described in the prototype section, further development of the actual gateway program will focus on the generation of dedicated sessions. The combination of session-information and the intelligent arrangement of all requests related to a certain online document will increase the usability of the service.

8. Conclusion

Anybody working a lot with online references is familiar with the advantages that are connected to their use. Nevertheless, documents that depend on a high number of online references tend to become worthless within, sometimes, the shortest time. Reasons for this fact are modification, movement or removal of the referenced online documents. Mechanism that try to preserve the links to the original documents are very difficult to create. Furthermore they depend on certain techniques on the server side. The catchword for this is bi-directional links.
In this paper we report a concept and a prototype that is especially interesting in the field of digital libraries. Ou concept has the great advantage that no modifications of any web-server are required. Furthermore, the author of any online resource do not have to modify their documents, by giving, for example, additional provision of digital signatures.
The gateway concept guarantees that the service is available for anybody who trusts in the party that is offering the service. The use of the service is quite easy and the prototype's data-transfer log format (Jar) already offers a practicable solution.

The service we suggest here does not help to fix the broken link problem itself. What it does is to offer the poss i-bility of reading the content of a referenced document no matter if the original document was modified o e-moved. Furthermore, the signature given by the trusted party running the gateway allows a verification of the content. The transfer information that is also included as well as the HTTP-Headers enables the reader to rep o-duce the original request that resulted in the available local copy of the document. Therefore, our service helps to preserve the worth of online references and could solve some of the problems mentioned in [22].

9. References

[1] R.L.Rivest, A.Shamir, L.M.Adleman: “A method for obtaining digital signatures and public-key cryptosys-tems“, Communications of the ACM,21, (1978), 120-126

[2] Public Key Infrastructure (X.509)(pkix); http://www.ietf.org/html.charters/pkix-charter.html

[3] Internet X.509 Public Key Infrastructure Certificate and CRL Profile, http://www.ietf.org/rfc/rfc2459.txt

[4] Internet X.509 Public Key Infrastructure Data Validation and Certification Server Protocols <draft-ietf-pkix-dcs-03.txt, http://www.ietf.org/internet-drafts/draft-ietf-pkix-dcs-03.txt

[5] The Java(TM) 2 SDK, Standard Edition, v 1.3 Beta Release, http://java.sun.com/products/jdk/1.3/

[6] Java security API, http://java.sun.com/products/jdk/1.3/docs/api/java/security/package-summary.html

[7] Summary of Tools for the JavaTM 2 Platform Security, http://java.sun.com/products/jdk/1.3/docs/guide/security/SecurityToolsSummary.html

[8] HTTP, HyperText Transfer Protocol, http://www.w3.org/Protocols/

[9] Package java.util.jar, http://java.sun.com/products/jdk/1.3/docs/api/java/util/jar/package-summary.html

[10] Info-ZIP Application Note 970311, ftp://ftp.uu.net/pub/archiving/zip/doc/appnote-970311-iz.zip

[11] Jarsigner (Windows), http://java.sun.com/products/jdk/1.3/docs/tooldocs/win32/jarsigner.html

[12]Manifest and Signature Specification, http://java.sun.com/products/jdk/1.3/docs/guide/jar/manifest.html

[13] keytool - Key and Certificate Management Tool, http://java.sun.com/products/jdk/1.3/docs/tooldocs/win32/keytool.html

[14] XML-Signatur , http://www.w3.org/Signature/

[15] SDML - Signed Document Markup Language, http://www.w3.org/TR/NOTE-SDML

[16] The PURL Homepage. URL: http://purl.oclc.org/

[17] David Ingham, Steve Caughey, Mark Little, Fixing the "Broken-Link" Problem: The W3Objects Approach, Fifth International World Wide Web Conference(May 6-10, 1996, Paris, France), http://www5conf.inria.fr/fich_html/papers/P32/Overview.html

[18] A. Aimar et al., "WebLinker, A Tool for Managing WWW cross-references," Computer Networks and ISDN Systems, Vol. 28 No. 1&2; Selected Papers from the Second World Wide Web Conference, Decembe 1995

[19] US Patent no. :5,022,080 Electronic Notary; Filing date: April 16, 1989; Inventors: Robert T. Durst, Kevin D. Hunter http://www.clir.org/pubs/reports/graham/intpres.html

[20] Internet X.509 Public Key Infrastructure Time Stamp Protocol (TSP) <draft-ietf-pkix-time-stamp-04.txt, http://www.ietf.org/internet-drafts/draft-ietf-pkix-time-stamp-04.txt

[21] Transparent Content Negotiation in HTTP; http://www.ietf.org/rfc/rfc2295.txt

[22] Peter S. Graham: "Intellectual Preservation: Electronic Preservation of the Third Kind", March 94, available online: http://www.clir.org/pubs/reports/graham/intpres.html