Proof of Authorship for Copyright Protection in OPELIX*

Markus Schneider and Thomas Keinz

German National Research Center for Information Technology
Dolivostraße 15, 64293 Darmstadt, Germany
{markus.schneider|keinz}@darmstadt.gmd.de

ABSTRACT

Producers that create and sell digital information have a high interest to prevent malicious parties selling copies of their work or claiming that they are the true creator. In order to provide the authors of some digital information with undeniable evidences that prove the authorship we propose a timestamp mechanism. This mechanism can be applied even if the good to be protected is simple text in ASCII where no watermarking can be applied. Furthermore we show what can be done to prevent the timestamps from expiration even if certificates have expired, public keys have been revoked or cryptographic primitives are broken. This is necessary if the timestamps should remain valid evidences over a long period, e.g., many years.

1. INTRODUCTION

Information goods are dramatically exposed to dangers such as copyright infringement and piracy. Goods can be copied at nearly no costs without any loss of quality. Thus, there is a need for the creators of goods who invested high amounts of money and work to protect their rights.

The focus of OPELIX[1] is on information commerce. Information commerce is widely assumed as a new and promising area in the whole sector of electronic commerce. While electronic commerce is more general comprising all relationships in which business interactions are done electronically, information commerce focuses exclusively on those businesses which deal with information products or information services. Information commerce has the big advantage that all phases in the trading process can be performed by using exclusively computers and networks in contrast to other types of electronic commerce in which some phases are performed offline, e.g., delivery in case of physical goods. Thus, via the Internet, information products and information services can be offered, ordered, and delivered instantaneously with global reach with 24-hour per day availability at low transaction costs.

Since the investment for the generation of high quality products can be very high, their producers have a high interest to prevent other parties from copying or even reselling. Therefore, lots of work has been done in order to develop techniques to protect the rights and interests of the creators. To avoid illegal copying and redistribution, there are different approaches: strategies for prevention of misuse, such as tamper-resistant devices or copy protection, and on the other hand, strategies for detection, such as watermarking. In general, these approaches when applied to information products are also dependant on their type of media, e.g., a movie obtained via Internet or CD, music, or images. Beside this, dependant on the media to be protected, the protection goals can also be different. While some mechanisms focus on the prevention and detection of illegal copying others go further and focus on the generation of undeniable evidences to prove the authorship. Such an approach is necessary in a case in which the producer of an information good finds a party that owns or sells an illegal copy and who claims maliciously to be the real producer of the information good. Here, the honest party requires some kind of proof to convince a judge about who is cheating and who is not. Dependent on the information product, such malicious behaviour is likely to be successful. If the good consists only of text information both parties could possibly

* This work was supported in part by the European Commission under contract IST-1999-10288, project OPELIX.
[1] In order to get more information about OPELIX, we refer to www.opelix.org.

claim to have written the text. In such a case, watermarking techniques will not be sufficient for the provision of undeniable evidences that prove the real authorship because they can be deleted or circumvented. A description telling the story of plagiarism in the academic area can be found in [Kock99].

Since we are dealing with text information that is sold and delivered over the Internet in the OPELIX information commerce scenarios, the authors have to be provided with an evidence that proves their authorship for each product in case of a conflict. In order to do this, we propose the usage of undeniable timestamps that guarantee that the author possessed the text at a specific time. Then, the party that has the oldest timestamp is assumed to be the real author. We require that the timestamps should keep their validity as evidences over many years, (e.g., more than 10 years).

The timestamp generation is based on standard security technology, e.g., well-known cryptographic primitives, public keys and accompanying certificates based on a public key infrastructure. Since cryptographic primitives can become weak because of short key lengths or can even be broken and certificates can be revoked or expire naturally, we need some additional mechanisms that guarantee the validity of timestamps over many years. Thereby, the philosophy of our approach is that the validity of the timestamp should be controlled exclusively —or at least as far as possible— by the producer and not by the timestamp authority (TSA). Thus, we can use a simple TSA that is just issuing timestamps on requests and nothing else to guarantee the validity of the timestamps over a longer period.

In this paper we describe a timestamp system that is used for the proof of authorship. We explain briefly the basic underlying security technologies and present the principles of the timestamp protocol. Afterwards, we show how the client side in the timestamp scenario can achieve long term validity of timestamps. Because of the restriction on the number of pages we will mainly sketch our ideas.

2. RELATED WORK

There is a variety of research that was done in the area of copyright protection. Most of the work done so far focuses on the protection of multimedia data. This type of data allows the embedding of some hidden additional information that can used for the identification of either the real creator / seller of these data, or the buyer, or even both of them. It is assumed that this hidden information can not be deleted without reducing the quality of the information product. An overview concerning information hiding is given in [PeAn99].

There are some proposals for embedding some additional information into text documents, e.g., by varying the distances of words and lines [BrLo99], [MaLo97], [Maxe94]. Even if this way of protection resists photocopying, there are possibilities for circumvention. Thus, there is a need for the prevention of this potential circumvention as it is done by the proposed use of timestamps.

When we talk about timestamps then we think implicitly of them as unforgeable timestamps as they are dealt with in the cryptographer community. There, timestamps are considered to be protected in a way that any modification of the time value can be detected and will destroy the validity of the timestamp. Some work done in this area can be found in [HaSt91], [Lipm99]. Currently, there are some activities for proposing a standard for a timestamp protocol which is available as a draft [PiKa00]. However, our ideas are not in contradiction to these activities whatever their result will be.

3. SECURITY TECHNOLOGY

Before presenting the timestamp protocol, we introduce briefly the underlying security techniques. For more details we refer to [MevO97]. The prerequisites for timestamps can be classified in the availability of some specific infrastructures and of secure cryptographic primitives.

The functional components of the infrastructure are the certification authorities (CA) for public keys, (e.g., see [X.509]), and the TSA. Both authorities are trusted to behave honestly and are assumed to have enough competence and technical background. In detail, the CA is trusted to verify the identity of the certificate requesting party accurately. This assures that a malicious party

cannot request certificates for a different identity. Furthermore, the TSA is trusted to issue only correct timestamps. In our context, this means that a TSA will never issue a timestamp carrying an older time value than the actual time of its generation.

As cryptographic primitives, we require hash functions and digital signatures. In general, the concept of timestamp systems makes only sense under the assumption that there exist collision free one-way hash functions. Such a function h maps an input x of arbitrary but finite length to an output $h(x)$ of fixed length. Given h and the input x, $h(x)$ is easy to compute. In order to be used in a timestamp system, for all pre-specified outputs it is computationally infeasible to find any input that is mapped to that output. This property is closely related to the requirement that for a hash function it should also be computationally infeasible to find an input x' given an input x, with $x' \neq x$, that $h(x) = h(x')$. The hardest requirement for a hash function is that it is computationally infeasible to find any two distinct inputs x, x' which are hashed to the same output. Even if it is not clear if such functions ever exist, there are some functions available for which no counter-examples to these requirements are known, e.g., SHA-1 or MD5.

The second basic cryptographic primitive of a timestamp service is given by digital signatures. Loosely spoken, digital signatures can be understood as the electronic equivalent of handwritten signatures. They were first sketched in [DiHe76], and meanwhile, there exist several standards for digital signatures, e.g., DSS [DSS00]. An extensive overview on digital signatures can be found in [MevO97]. In the electronic world, digital signatures bind pieces of information to identities. Thereby, no other party should be able to create a digital signature binding a statement to the person's identity. Therefore, this person uses a secret —a secret cryptographic key— to calculate the digital signature. Since no other party knows this secret and by the assumption that the underlying signature algorithm prevents forging, a digital signature can be used as a proof to convince any other party —e.g., a judge— of the signer's identity. Beside the means for signature creation, the concept of digital signature involves also means for signature verification. In order to do this, the verifier requires a public key corresponding to the signer's secret key. Since a malicious party is able to create a public key and claim a faked identity, this concept requires a method to support the authenticity of public keys. This is achieved by the certification of public keys which will be done by a CA.

4. THE SIMPLIFIED TIMESTAMP PROTOCOL

In the following, we sketch the basic ideas of the timestamp protocol. It has to be emphasized that the single application of the timestamp protocol is not sufficient for long term validity of timestamps. This problem will be addressed in the next section.

Consider a producer *P* of digital content who needs an evidence that he possessed this content at a specific time. Before selling or publishing his information good he has to take care about its protection. Thus, *P* requests a timestamp ts for his content. In our protocol, the requesting party *P* does not send the whole content to be timestamped to the TSA. Let us assume this content is contained in a document doc. This document can be clearly identified by a fingerprint obtained by applying a cryptographic collision free hash function h on the document. Thus, it is sufficient to include the hash function output $h(doc)$ in the timestamp request. Thereby, the producer does not have to reveal his newly created content to any other party before he possesses the corresponding ts. Furthermore, the producer does not have to send his content through any open network. Even if (1) the connection over this network is secured, and (2) the receiving *TSA* is trusted, it seems to be more secure to send just a hash value that cannot be inverted to obtain doc. If *P* desires to bind the timestamp to its own identity id then the data exchanged in the timestamp request also should depend on id as well. This is advantageous in a case in which both doc and the corresponding ts are stolen. As long as ts depends somehow on the requester's identity, no other party that holds a copy of ts can claim the creatorship. Thus, after having produced the content, *P* computes $h(doc \mid id)$ to be sent in the timestamp request where '|' stands for the concatenation operator.

Upon receipt of the request, TSA creates ts. This ts consists of some data and the TSA's signature on these data: $ts = h(doc \mid id) \mid t \mid pk_{TSA} \mid sig_{TPA}\big(h(doc \mid id) \mid t\big)$. Here, t stands for the actual time at which the TSA processes the request, pk_{TSA} represents TSA's public key with

accompanying certificate that is necessary to check ts. Finally, $sig_{TPA}()$ describes *TPA*'s signature on the given arguments. After creation of ts, TSA replies it to *P* who stores it. The timestamp will be used in case of a conflict.

Assume now that *P* executed the timestamp protocol for the content contained in doc at time t_1 and sells or publishes doc for the first time at $\tau > t_1$. If later a malicious party *M* intends to resell it and claims to be the author, he can also request a timestamp for doc. This timestamp will be generated at time $t_2 > \tau$. But then, with undeniable timestamps, *P* can prove that he possessed doc before *M* possessed it, since $t_1 < t_2$. Thus, a judge will decide that *P* is the true author of doc.

5. THE PROBLEM WITH LONG PERIOD EVIDENCES

A single timestamp obtained via the execution of the above protocol remains valid as long as all the cryptographic primitives (e.g., hash functions and digital signature schemes) and parameters (e.g., key material and certificates) remain valid. If one of these components becomes invalid, then the timestamp would loose its property as evidence. As a consequence, the producer would loose his proof of authorship. Thus, it is necessary to introduce some mechanisms to prevent this. In this context, it was the goal that these mechanisms can be executed and controlled by the user side as far as possible.

Whenever a specific hash function turns out to be weak, i.e. there are known collisions, it should not be used anymore. Timestamps which are based exclusively on a weak hash function become invalid. This is obvious, since a party that can construct meaningful collisions $h(doc \mid id) = h(doc' \mid id)$ could use the timestamp for doc and also for a doc' created afterwards. In order to cope with this problem, we propose to apply n different hash functions $h_1, \ldots, h_n$ in parallel that should be included in each timestamp request. Hereby, we also require that the security of these hash functions is based on different assumptions. Then, if one hash function turns out to be broken, we still have $n-1$ secure hash values to fall back on. But even if more hash functions will be broken, the problem for an attacker is not only to construct a collision for each weak hash function. In such a case, he would have to find a collision which is fulfilled simultaneously, i.e. $h_1(doc) = h_1(doc'), \ldots, h_n(doc) = h_n(doc')$. The problem for finding simultaneous collisions is harder than the construction of collisions for a single hash function.

The hashing part is the first step in the timestamp protocol. Therefore, the user has some degrees of freedom to choose or modify their input. Thus, we solved this problem by simultaneous application of several hash functions. For the remaining primitives to be applied, the inputs are given by the outputs of the previous steps in the protocol. Thus, modifications of these inputs would affect the protocol and lead to invalid results. Nevertheless, we cannot guarantee that the signature turns out to be forgeable. Another problem stems from the fact that the public key —or its certificate— is only valid for a period of one or a few years. In some cases, it can be revoked earlier. But, this key is necessary for the verification of the timestamp and it is only trusted as long as its certficate is valid.

We propose to solve these problems by the introduction of timestamp chains. This means, that upon receipt of a timestamp issued by TSA_1 the user creates a new request based on the message obtained from TSA_1 and sends it to TSA_2, with $TSA_1 \neq TSA_2$. The second timestamp guarantees that the user possessed the first timestamp at the time included in the second timestamp. TSA_1 and TSA_2 should be selected in such a way that they should use signature schemes that are based on different cryptographic assumptions. Following this idea, a user can create a timestamp chain over the years that consists of several timestamps.

Let us demonstrate how the timestamp chain works. Consider a timestamp chain that contains —beside others— the timestamps ts_i with time t_i and ts_{i-1} with time $t_{i-1} < t_i$. Here, we can distinguish different cases. In the first case, we assume that ts_{i-1} becomes invalid —e.g., because of broken signatures or invalid certificate— at time τ with $t_{i-1} < t_i < \tau$. But then, ts_i ensures that ts_{i-1} existed before τ, and therefore, is not forged. In case of $t_{i-1} < \tau < t_i$, ts_i cannot be used to proof that ts_{i-1} existed before τ. Then, one has to use an earlier ts_j which must be valid —if one such timestamp still exists— and all later timestamps $ts_{j+1}, ts_{j+2}, \ldots$ can be deleted from the chain and be replaced by new ones. If no such valid timestamp is available, then the timestamp chain would

loose its quality of evidence. Thus, depending on the value of the good to be protected it is recommended to collect some (2-4) timestamps that are valid for the same period to reduce the probability for loss of evidence. For the possibility of verification of the timestamp chain, all TSAs have to provide their old certificates with clear indication of their true validity end date.

6. CONCLUSION

In this paper we presented a timestamp mechanism that can be used for the proof of authorship. This mechanism can be used for the protection of text documents independently of their format where other strategies such as watermarking can not be applied. However, a simple timestamp mechanism is not sufficient for the creation of evidences that are valid over a long period, since certificates can expire or can be revoked and used cryptographic primitives can turn out to be weak. In order to cope with these problems, we propose what a user can do to keep his timestamp valid over periods of many years.

The large scale usage of timestamp protocols requires the availability of some trustworthy TSAs which do not exist currently. Beside the demonstration of technical feasibility it has to be analyzed if the provision of timestamps is a feasible Internet business model from the economic point of view.

7. ACKNOWLEDGEMENT

We are grateful to Matthias Enzmann and Michael Herfert for lots of helpful discussion.

8. REFERENCES

[BrLo99] J.T. Brassil, S. Low, N.F. Maxemchuk: Copyright Protection for the Electronic Distribution of Text Documents. Proceedings of the IEEE, Vol. 87, No. 7, July 1999.

[DiHe76] Whitfield Diffie, Martin E. Hellman: New Directions in Cryptography. IEEE Transactions on Information Theory, Vol. IT-22, No. 6, November 1976.

[DSS00] U.S. National Institute of Standards and Technology (NIST): The Digital Signature Standard DSS. FIPS PUB 186-2, January, 2000.

[HaSt91] Stuart Haber, W. Scott Stornetta: How to Time-Stamp a Digital Document. Journal of Cryptology, Vol. 3, 1991.

[Kock99] Ned Kock: A Case of Academic Plagiarism. In Intellectual Property in the Age of Universal Access, ACM, (ISBN 1-58113-169-0), 1999.

[Lipm99] Helger Lipmaa: Secure and Efficient Time-Stamping Systems. PhD Thesis, University of Tartu (Estonia), 1999.

[MaLo97] N.F. Maxemchuk, S.H. Low: Marking Text Documents. Proceedings of the 1997 International Conference on Image Processing (ICIP ’97), 1997.

[Maxe94] N.F. Maxemchuk: Electronic Document Distribution. AT&T Technical Journal, Vol. 73, No. 5, 1994.

[MevO97] Alfred J. Menezes, Paul C. van Oorschot, Scott A. Vanstone: Handbook of Applied Cryptography. CRC Press, 1997.

[PeAn99] Fabien Petitcolas, Ross Anderson, Markus Kuhn: Information Hiding -A Survey. Proceedings of the IEEE, Vol. 87, No. 7, July 1999.

[PiKa00] C. Adams, P. Cain, D. Pinkas, R. Zuccherato: Internet X.509 Public Key Infrastructure Time Stamp Protocol (TSP). Internet Draft, November 2000.

[RiSh78] Ron L. Rivest, Adi Shamir, Len Adleman: A Method for Obtaining Digital Signatures and Public-Key Cryptosystems. Communications of the ACM, Vol.21, No.2, February 1978.

[X.509] ISO/IEC: Information Technology -- Open Systems Interconnection -- The Directory: Authentication Framework, 1995.