A HYBRID APPROACH TO THE PROBLEMS OF TIME-STAMPING

Cristian MARINESCU[1], Nicolae ŢĂPUŞ[2]

Una din problemele actuale cele mai mari ale protocoalelor de securitate o reprezintă stabilirea cu exactitate a momentului de timp la care au loc anumite evenimente. Toate protocoalele şi standardele de ştampile digitale de timp se confruntă cu probleme de securitate. În aceste condiţii, ce modificări şi caracteristici noi sunt necesare pentru realizarea unor servicii de ştampilare temporală sigure? Articolul de faţă prezintă o privire de ansamblu asupra situaţiei actuale din domeniul ştampilelor de timp, subliniând atât avantajele cât şi dezavantajele schemelor existente. Articolul propune totodată o abordare hibridă a problemelor, pentru a soluţiona situaţia actuală şi a facilita realizarea unor scheme sigure de ştampile digitale de timp.

One of the important problems with today's security protocols is to establish the exact time of certain events. Many security services require this capacity, but all known time-stamping protocols and standards encounter problems in delivering reliable and secure time-stamps. If this is in fact true, what changes and new protocols are required to achieve secure and trustworthy time-stamping services? This state-of-the-art paper presents an overview of the current situation in time-stamping and tries to put the advantages and disadvantages of the different available time-stamping schemes in perspective. The paper also aims to propose a hybrid approach to overcome the current dead-end in achieving highly secure time-stamping schemes.

Keywords: digital signatures, PKI, security, time-stamping, TSA

1. Introduction

Security has become a key requirement for private, business and government activities. The need to deliver secure and reliable time-stamping services has increased recently, also because of the fact that information and its digital representation differs fundamentally from paper-based, hand-written documents [1].

Time-stamps are a digital binding between a time parameter and a electronic representation of the data [2]. This is why they are considered to be the

[1] PhD Eng., Faculty of Automatic Control and Computers, University POLITEHNICA of Bucharest, Romania
[2] Prof., Faculty of Automatic Control and Computers, University POLITEHNICA of Bucharest, Romania

electronic proof that certain digital data existed prior to the time indicated by the time-stamp token. The Time-Stamping Authority (TSA) is the trusted third party that offers this digital evidence and guarantees that the time parameter is correct for the specified accuracy [2].

The exact time the electronic data was signed is very important, since the verification process of a digital signature has to check the validity of the certificate and its revocation status at that particular moment. An electronic document has to contain undeniable proof about the time the document was created, in order to be used in a court of law [3]. Time-stamps can indicate the time when a work of art or a scientific document was created or published, helping to enforce the intellectual property rights [2]. They also can be used to indicate the time of submission when a deadline is critical; they can establish a timeline for the existence of digital data. Time-stamping is also the simplest form of notarial action, all other forms being based on the ability to specify the exact time of this action. Time-stamps can be also used to perform electronic commerce on the Internet, indicating the moment of the transaction. They can be used in a large number of security infrastructures and protocols, but the real problem is to deliver secure and trustworthy time-stamping services [2].

2. An overview of the current situation of time-stamping

Time-stamping schemes have been generally classified as follows [4]:

- simple schemes;
- linking schemes;
- distributed schemes.

Simple schemes are typically synchronous one-step protocols. The result generated by the TSA is an independent time-stamp token. Simple schemes are quite easy to implement, the time-stamps generated by different authorities can be compared, using the time and accuracy parameters [2]. The main problem of this type of scheme is that the TSA has to be trusted unconditionally. A malicious or compromised authority can back-date time-stamps, since the TSA is the one that guarantees the correctness of the time parameter [2]. There are at least three approaches to develop a simple time-stamp scheme [5]: digital signatures, archived tokens, and message authentication codes (MACs).

Linking schemes are more sophisticated and complicated to implement; the model makes back-dating time-stamps more difficult [2]. The authority generates a time-stamp which contains a hash of the preceeding time-stamp tokens. As a result, a chain of time-stamps is built, linking all time-stamps ever produced by the TSA. The basic idea of the time-stamping chain is to lower the trust required of the TSA. The time-stamps are generated by using data from previously issued time-stamp tokens, using a hash function. If the issuer would try

to alter a certain time-stamp, the whole chain of time-stamps would have to be changed, too. This is a major advantage compared to the simple schemes, but this fact also complicates the verification procedure, since the cooperation of the TSA is needed in order to verify the time-stamp. Linking schemes typically contain three phases [6]: aggregation, linking, and publishing. The most popular scheme of this kind is the one published by Haber and Stornetta [7].

Distributed schemes consist of multiple authorities belonging to one of the other two categories presented, responsible together for the generation of the time-stamps. The focus is set on strengthening the security against manipulating time-stamps by sharing the generation process and the private key between multiple servers. In order to back-date time-stamps, all involved authorities have to become part of the malicious attack. Distributed schemes decrease the dependence on the TSA and also increase the availability and resistance to *Denial of Service* attacks, but on the other hand they suffer from almost the same problems as simple and linking schemes, being also based on these types of schemes. Examples of distributed schemes are presented in [8] and [9].

It is important to notice that, while all simple schemes assume that the TSA is a trusted third party (TTP), the linking and the distributed scheme render this assumption unnecessary [4], since they lower the trust necessary towards the TSA.

3. The problems of time-stamping

Simple time-stamp schemes cannot guarantee that the time parameter attached to the time-stamp token is correct. Since the user has to trust the TSA, there is no way to detect a fraud of a malevolent or compromised TSA. It is also impossible to check whether the time parameter is correct or not [2]. In case the TSA is using a private key, an undetected leak of this key could compromise the security of the whole time-stamping scheme. It is practically impossible to verify in all these cases if the time-stamp token was generated at the stated time or not. Since there is typically just one server responsible for time-stamping, these schemes are vulnerable to *Denial of Service* attacks [2].

A notable example of a simple time-stamping scheme is the PKIX TSP protocol (RFC3161) [10]. The protocol is based on digital signatures and presents a typical client-server architecture. The RFC3161 specification introduces, next to the usual issues of simple schemes, also other problems [2]. The supported underlying transport protocols (raw sockets, HTTP, FTP, and e-mail) present some differences, raw socket-based solutions offering more capabilities than the others (e.g., a delayed retry mechanism, etc.). This is the reason why interoperability between software solutions implementing different transport mechanisms becomes difficult to achieve [11]. The PKIX TSP protocol also

contains some questionable *features*, like the *ordering* field or the *policy* information, which can cause problems if implemented as the protocol suggests [11]. Finding out which policies are supported by the TSA is also not part of the protocol, which complicates the implementation of any client. An interesting ideea would be to add some possibilities in the next version of the protocol in order to be able to find out the implemented policies [12].

Even though linking schemes present some undeniable advantages, we also have to consider their weaknesses. After the publication step, forging the time-stamps becomes impossible, but between two publication steps, the server is able to alter the order of the time-stamps generated since the last publication [12]. This is the reason why the TSA has to be trusted not to fake tokens between two publication steps. Another problem is the complexity of the verification procedure, the entire chain of time-stamps has to be preserved in order to allow their verification [2]. This problem was addressed by using Merkle or binary trees, but this does not increase the efficiency of these schemes, which is in any case quite low. The time-stamping server is also in this case vulnerable to *Denial of Service* attacks, and last but not least, the publication step may be quite expensive and inconvenient [8]. Several problems of linking schemes are analyzed by Just in his article [13].

There are also other problems which concern time-stamping schemes in general. Interoperability is unfortunately not regarded as an important goal to be achieved. Time-stamps also cannot entirely solve the problems of PKI's, since they change the working model of such infrastructures [2]. There has been a great effort invested lately in producing time-stamping standards based on simple and linking schemes. The results are several standards concerned with the generation of time-stamps: ISO/IEC 18014 [14] [15], RFC3161 [10], or ANSI X9.95-2005 [16]. Unfortunately none of these standards satisfactorily solve all the problems of time-stamping.

Two of the properties listed by [12] are by far the most important for enabling secure time-stamping: enabling the verification of the provided time parameter and empowering the verifier to fully audit and verify the behavior of the time-stamping authority. We must also point out that no time-stamping scheme available today fulfills these two important requirements. This is also the reason why we consider that it is necessary to have a new, *hybrid* time-stamping scheme which combines the presented properties and design requirements in one single solution [17].

4. A new hybrid time-stamping scheme

But how can secure time-stamping services be delivered, based just on the protocols known today? In practice, each of the mentioned schemes presents very

useful properties but also contains some disadvantages. This leads us to the idea that combining the properties of the simple, linking, and distributed schemes would create a new, *hybrid* time-stamping scheme that could fulfil all the requirements in one single solution. One of the most complicated aspects of this approach is maintaining the properties of a simple time-stamp scheme (independent tokens, direct and easy modality of checking the time-stamps), but lowering the necessary trust in the time-stamping authority.

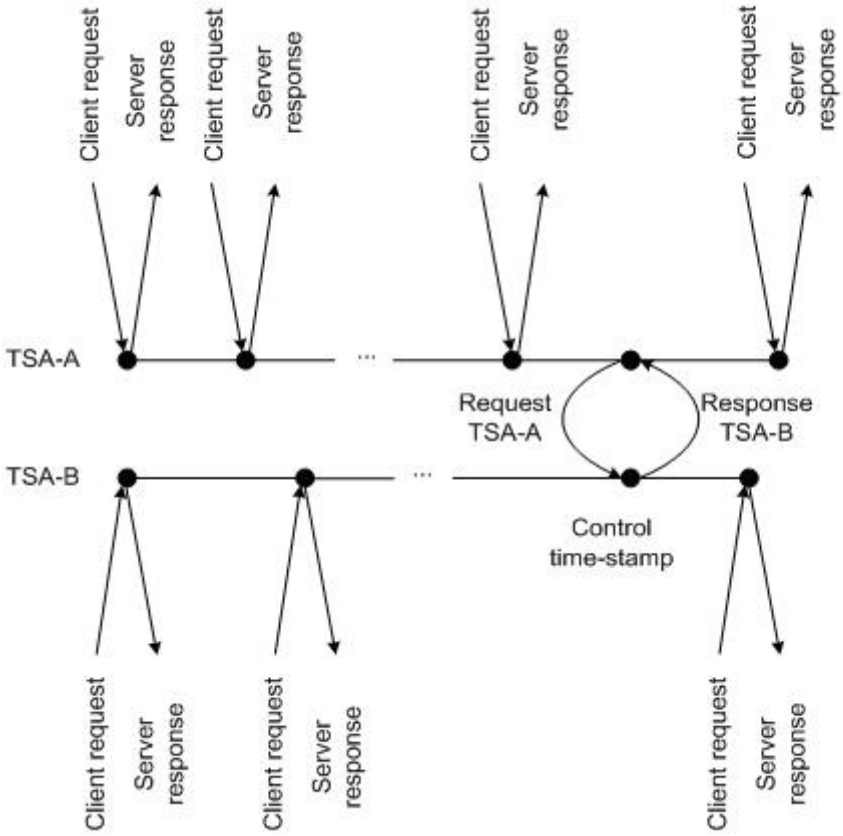

Fig. 1. Representation of the hybrid time-stamping scheme

The idea of linking schemes was exactly to lower the trust in the TSA. This was achieved by linking the time-stamp tokens and making a change to these tokens impossible at a later moment. But as we could see, this was not enough in some cases. This suggests that it would be even better if the time-stamp scheme would also distribute the responsibility and, based on this, the secret that is used to generate the tokens. Even though this would seem to be the solution to the problem, we have to consider the complexity of the solution and the fact that such a solution would probably lose the property of being able to easily verify time-stamps without the help of the TSA. The solution to this problem would be to

create different levels of trust and different possibilities of verifying a generated time-stamp token.

One possible way is to create a hybrid system that is based on the principles of simple schemes, but also borrows some properties from linking and distributed schemes. The system should contain two kinds of time-stamps: the normal ones generated using a private key, with the possibility of validating them by using a corresponding public certificate; and control time-stamps, used to offer better control possibilities over the TSA. In the following we define the formal definition between the the time-stamps of the hybrid system:

> Let n be an odd number, and TSA_A a hybrid time-stamping authority. We define $TS_k = F(sn_k, t_k, acc_k, D_k, H(ŞD_{k-1}))$ the time-stamp generated by this authority, where $0 \le k \le n$, sn_k represents the unique identifier of the stamp, t_k the included time parameter, acc_k the accuracy of the time source, D_k the data to be time-stamped, and $H(ŞD_{k-1})$ the hash of the preceeding time-stamp. TSA_A can generate n time-stamps TS_k, after this it has to request a control time-stamp from another authority of the same hybrid time-stamping group. The data stamped for the control time-stamp is built upon the hashes of the time-stamps generated after the last control time-stamp:
>
> $TS_{n+1} = F(sn_B, t_B, acc_B, D_{n+1}, H(TS_B))$, where D_{n+1} are the data to be time-stamped in the control time-stamp is defined as a concatenation of hashes: $D_{n+1} = F(H(ŞD_0), H(ŞD_1), ..., H(ŞD_k))$.

In general, one such time-stamping service would consist of at least two time-stamping servers. Fig. 1 presents a possible approach of how to build a hybrid time-stamping scheme, based on two TSAs. Both of them can serve requests from different clients independently from each other. At a defined moment, one TSA has to time-stamp its own generated time-stamps using the second TSA and vice-versa. This creates the special control time-stamps and makes it impossible to change the generated time-stamps, especially if these control time-stamp tokens are to be published using an out-of-band method. In order to also ensure the sequential timeline between two such special time-stamps, it would be enough to link the time-stamps through three different methods. This could be achieved by including a sequence number that would be incremented with every produced time-stamp and by adding the hash of the latest time-stamp to the next generated one.

Another important characteristic of the hybrid approach is the definition of several methods of how to verify the time-stamps and how to audit the behavior of the TSA. Based on the methods described in [17] and [18], it is possible to create a verification procedure for time-stamps for different levels of trust. Combined with a fast method of verification for the control time-stamps, this creates also the possibility to audit the behavior of the TSA [17], [18].

Since the scheme consists of at least two TSA's that can act independently, it is possible to create two time-stamps at two different authorities. This enables the user to also compare and verify to some extent the time parameter of the one time-stamp through the other. This unique feature is very important since it creates the possibility to verify the time parameter provided by the TSA [18]. If we postulate that the two different authorities that make up the hybrid scheme have also to be synchronized with at least the same accuracy that is used for the time parameter, it becomes feasible to check the time parameter by comparison. We also have to assume that there is no difference between the normal time-stamp and the second one that is used in the verification process.

5. Conclusions

Unfortunately, the problems of time-stamping schemes are not easy to solve. Evaluation of the existing solutions is very important in order to recognize all open issues. Most security evaluations concentrate on just the alteration of time-stamps, leaving the collision issues or any other problems out of the discussion. The whole array of problems should be addressed carefully in order to find appropiate solutions.

The present paper outlines a wide range of time-stamping issues, with respect to the existing time-stamp schemes. We focused on what we believe to be real problems with today's time-stamping, since there is a strong need to address them. In our opinion it is difficult to deliver secure and trustworthy time-stamping services based only on the technologies known today. Without solving these problems, time-stamping will fail to become a widespread technology.

Our suggestion is to combine the properties of simple, linking, and distributed schemes in order to solve the issues that create the mentioned problems. A new hybrid scheme based on digital signatures and linking the hashes of consecutive time-stamps would combine the necessary properties, creating a more secure and trustworthy time-stamp scheme, that could fulfill the presented goals. We argue that even if this would not solve the problems of public key infrastructures, this would be a great step forward, since time-stamp services are needed in many software applications. Time-stamping will gain its true importance if and only if secure time-stamping services become possible.

REFERENCES

[1] *U. Maurer*, New approaches to digital evidence, Proceedings of the IEEE, **vol. 92**, Issue 6, 2004, pp. 933-947

[2] *C. Marinescu, N. Ţăpuş*, A Survey of the Problems of Time-Stamping or Why It Is Necessary to Have Another Time-Stamping Scheme, Proceedings of the IASTED International Conference on Software Engineering, Acta Press, SE2007, Innsbruck, Austria, 2007

[3] *C. R. Merrill*, Time is of the Essence: Electronic Documents Will Stand Up in Court Only If the Who, What and When They Represent are Unassailable, CIO Magazine, March 15, 2000, http://www.cio.com/archive/031500_fine.html

[4] *M. Une*, The Security Evaluation of Time Stamping Schemes: The Present Situation and Studies, Discussion Paper No. 2001-E-18

[5] *K. Wouters*, Time-Stamping: a survey, Seminars on COmputer Security and Industrial Cryptography (COSIC), K. U. Leuven, Department of Electrical Engineering, 2004, http://www.esat.kuleuven.ac.be/cosic/seminars/slides/Time-Stamping.pdf

[6] *K. Wouters, B. Preneel, A. I. Gonzalez-Tablas, A. Ribagorda*, Towards an XML Format for TimeStamps, Proceedings of the ACM workshop on XML Security 2002, ACM Press, 2003

[7] *S. Haber, W.S. Stornetta*, How to Time-Stamp a Digital Document, Journal of Cryptology, **vol. 3**, No. 2, 1991, p. 99-111

[8] *A. Bonnecaze, P. Liardet, A. Gabillon, K. Blibech*, A Distributed Time Stamping Scheme, Proc. of the IEEE conference on Signal and Image Technology and Internet Based Systems, Cameroon, 2005

[9] *A. Takura, S. Ono, S. Naito*, A Secure and Trusted Time Stamping Authority, Proceedings of the IEEE Internet Workshop, IWS 99, 1999, pp. 88-93

[10]. *C. Adams, P. Cain, D. Pinkas, R. Zuccherato*, Internet X.509 Public Key Infrastructure Time-Stamp Protocol (TSP), 2002, ftp://ftp.rfc-editor.org/in-notes/rfc3161.txt

[11] *C. Marinescu, N. Ţăpuş*, Some Critical Aspects of the PKIX TSP, Lecture Notes in Computer Science, **vol. 3677**, CMS2005, Salzburg, AUSTRIA, 2005

[12] *C. Marinescu*, Design Requirements For A Secure Time-Stamping Scheme, Proceedings of the IASTED International Conference on Internet and Multimedia Systems and Applications, **vol. 612**, Acta Press, EuroIMSA2008, Innsbruck, Austria, 2008

[13] *M. Just*, Some Timestamping Protocol Failures, Proceedings of the Symposium on Network and Distributed Security, NDSS98, San Diego, CA, USA, 1998, pp. 89 - 96

[14] ***, ISO/IEC FDIS 18014-2 – Information technology. Security techniques. Time-stamping services. Part 2: Mechanisms producing independent tokens. http://oberon.postech.ac.kr/kiisc-sis/timestamp, 2002.

[15] ***, ISO/IEC WD 18014-3 – Information technology. Security techniques. Time Stamping Services. Part 3: Mechanisms producing linked tokens. http://oberon.postech.ac.kr/kiisc-sis/timestamp, 2002

[16] ***, ANSI X9.95-2005, Trusted Time Stamp Management and Security, 2005, http://x9.org/

[17] *C. Marinescu*, Ein hybrides Zeitstempelsystem, Lecture Notes in Informatics, **vol. 121**, Proceedings of Software Enginering 2008, SE2008, München, Germany, 2008

[18] *C. Marinescu*, Semnarea electronică a datelor. Schemă hibrid de realizare a ştampilelor digitale de timp, Teză de doctorat, Facultatea de Automatică şi Calculatoare, Universitatea Politehnica Bucureşti, 2008