IMES DISCUSSION PAPER SERIES

The Security Evaluation of Time Stamping Schemes: The Present Situation and Studies

Masashi UNE

Discussion Paper No. 2001-E-18

IMES

INSTITUTE FOR MONETARY AND ECONOMIC STUDIES

BANK OF JAPAN

C.P.O BOX 203 TOKYO
100-8630 JAPAN

IMES Discussion Paper Series 2001-E-18
December 2001

The Security Evaluation of Time Stamping Schemes: The Present Situation and Studies

Masashi UNE*

Abstract

Time stamping is a technique used to prove the existence of certain digital data prior to a specific point in time. With the recent development of electronic commerce, time stamping is now widely recognized as an important technique used to ensure the integrity of digital data for a long time period. Various time stamping schemes and services have been proposed.

When one uses a certain time stamping service, he should confirm in advance that its security level sufficiently meets his security requirements. However, time stamping schemes are generally so complicated that it is not easy to evaluate their security levels accurately. It is important for users to have a good grasp of current studies of time stamping schemes and to make use of such studies to select an appropriate time stamping service.

Une and Matsumoto [2000], [2001a], [2001b] and [2002] have proposed a method of classifying time stamping schemes and evaluating their security systematically. Their papers have clarified the objectives, functions and entities involved in time stamping schemes and have discussed the conditions sufficient to detect the alteration of a time stamp in each scheme.

This paper explains existing problems regarding the security evaluation of time stamping schemes and the results of Une and Matsumoto [2000], [2001a], [2001b] and [2002]. It also applies their results to some existing time stamping schemes and indicates possible directions of further research into time stamping schemes.

Key words: time stamping, security evaluation
JEL classification: L86, L96, Z00

* Institute for Monetary and Economic Studies, Bank of Japan
(E-mail: masashi.une@boj.or.jp)

Table of Contents

I. Introduction

The rapid expansion of the Internet has brought about great progress in electronic commerce. In the financial sector, various sorts of services, for example, online banking services and securities trading services, have been provided by many financial institutions. In such services, transactions between interested parties are electronically processed and recorded. At the same time, digital documents have been employed as a medium for the transmission, sharing and storage of information in business processes in place of paper-based documents.

Compared with paper-based documents, digital documents have some advantages. For example, digital documents can be transmitted at high speed and are not deteriorated. However, it is more difficult to detect alteration in digital documents than in paper-based documents. In order to use digital documents as a medium for keeping information and records of transactions as securely as paper-based documents, it is necessary to apply a technique to assure the integrity of digital documents over a long time period.

Digital documents with a digital signature also have the same problem as the one explained above. A digital signature is a cryptographic technique to assure the integrity of digital data and to confirm their originator. The verification of a digital signature is conducted by using a public key certificate corresponding to a private key to sign. However, if the certificate expires or is revoked, it is impossible to confirm whether or not the corresponding digital signature was generated during the validity period of the certificate (Haber et al. [1995]). This is because nobody guarantees appropriate controls of the private key after the certificate expires. The validity period of a certificate is commonly one or two years. Therefore, in order to keep digital data with a digital signature securely for decades (for example, for the purpose of information disclosure), another technique is needed to prove that the digital signature was generated during the validity period for a long time.

Time stamping is a technique to prove the existence of certain digital data prior to a specific point in time. It is considered to be an important tool that can be used to make up for the disadvantage of digital documents explained above (Haber et al. [1995]).

Various time stamping schemes have been proposed. Popular examples of these are as follows: the scheme proposed by Benaloh and de Mare [1994], TIMESEC (Massias and Quisquater [1997], Preneel et al. [1998]), PKITS (Fabrica Nacional de Moneda y Timbre [1998]), the electronic notarization system (The Study Group on the Legal System of Electronic Commerce [1998]), the time signature distributed system (Takura et al. [1999]), Cuculus (Buldas et al. [2000], Cybernetica [2001]), TrueSign (Privador [2000]), Digital Notary (Surety.com [2001]), SecureSeal (NTT Data [2001]) and Notary Service (VeriSign [2001]).

Digital Notary, SecureSeal and Notary Service have already been provided as business products. Moreover, the standardization activities of time stamping services have been promoted in ISO/IEC

JTC1/SC27 (ISO/IEC [2000a], [2000b] and [2000c]) and IETF PKIX (Adams, et al. [2001]). Taking these current situations into consideration, it is naturally expected that many kinds of time stamping services will be provided in the near future.

The next issue in importance is how users of time stamping services should select an appropriate one. It is necessary for users to evaluate the security of each scheme and to confirm in advance that its security level meets their security requirements. However, time stamping schemes are generally so complicated that it is not easy for the users to evaluate their security levels accurately. It is important to have a good grasp of current studies of time stamping schemes and to make use of such studies to select an appropriate time stamping service.

Une and Matsumoto [2000], [2001a], [2001b] and [2002] have proposed a method of systematically evaluating the security of time stamping schemes. Their papers have clarified the objectives, functions and entities involved in time stamping schemes, classified them comprehensively and provided a method of evaluating their security without discussing the details of their specifications. It is recommended that users of any particular time stamping scheme refer to their results when they wish to evaluate its security.

This paper introduces Une and Matsumoto [2000], [2001a], [2001b] and [2002] as current studies of the security evaluation of time stamping schemes and explains how their results can be used for the selection of an appropriate scheme. Section II explains previous studies on the security of time stamping schemes and their conventional classification. Section III introduces an outline of their papers, explains their method of classifying time stamping schemes and then goes on to discuss security against alteration of a time stamp in each scheme. Section IV applies their results to some existing schemes. Finally, Section V summarizes their results and indicates possible directions of further research.

II. Conventional Method of Classifying Time Stamping Schemes

A. Simple, Linking and Distribution Schemes

Time stamping schemes have been generally classified into three: simple, linking and distributed schemes (for example, Massias and Quisquater [1997]). In the simple scheme, a time stamp is generated in such a way that it does not include data included in other time stamps. For example, a time stamp is issued as follows.

(1) An entity that wants a time stamp for certain data *M* (known as the time stamp requester) transmits a request message including a hash value *H* of *M* to an entity issuing a time stamp (known as the time stamp issuer).
(2) The issuer generates a digital signature *S* on data that includes at least *M*, a time parameter *T* and an identifier *ID* of the authority. *T* indicates the point in time at which the issuer received the request message. A time stamp *TS* corresponding to *M* includes at least *H*, *T*, *ID* and *S*.
(3) The issuer sends *TS* to the requester.

The verification of a time stamp is as follows. First, a verifier computes a hash value of *M* and compares it with *H* included in *TS*. Next, the verifier carries out an algorithm to verify *S*.

The main characteristic of the simple scheme is that while its system is relatively simple, its security depends on time stamp issuer's reliability. Haber and Stornetta [1991] have pointed out that if an issuer fraudulently alters the time parameter of a certain time stamp, nobody can detect the alteration.

As countermeasures against the problem, the linking and distributed schemes have been developed. In the linking scheme, the issuer generates a time stamp which includes data included in other time stamps. As a result, a chain of time stamps is constructed, for example by using a one-way hash function. If an issuer is willing to fraudulently alter a certain time stamp, it has to alter all the time stamps relating to that time stamp. This is why it is considered to be more difficult for an issuer to manipulate a time stamp in the linking scheme than in the simple scheme. PKITS and TIMESEC can be listed as examples of popular linking schemes. In TIMESEC, data relating to all time stamps are periodically published on the online site in order to make it difficult for an issuer to manipulate the chain of time stamps. This treatment is considered as improving the security of a time stamp against its alteration in the linking schemes even if the issuer is not always trustworthy. However, a linking scheme system is more complicated than a simple scheme.

The distributed scheme is one in which multiple issuers cooperatively generate a time stamp. One of the main aims of this scheme is to strengthen security against the issuer’s manipulation of a time stamp by sharing the secret data used to generate a time stamp among the issuers. If the number of

collusive issuers is less than a specific predetermined number, they cannot recover the secret data completely and therefore find it hard to manipulate a time stamp. However, just as in the case of the linking scheme, a distributed scheme system is more complicated than a simple scheme. For example, the time signature distributed system (Takura et al. [1998]) belongs to this category, and Ansper et al. [2001] proposed a scheme possessing the characteristics of both the linking and distributed schemes. The main strengths and limitations of these schemes are summarized in Table 1.

Table 1 Main Strengths and Limitations of Three Schemes

Schemes	Strengths	Limitations
simple scheme	The system is relatively simple.	It is necessary to assume that the issuer is the trusted third party.
linking scheme	The assumption that the issuer is the trusted third party is rendered unnecessary, for example, by the periodical publication of a part of a chain of time stamps.	The system is relatively complicated because additional operations for linking all time stamps are needed.
distributed scheme	The assumption that the issuers are the trusted third parties is rendered unnecessary by sharing the secret data among multiple issuers.	The system is relatively complicated because multiple issuers generate a time stamp cooperatively.

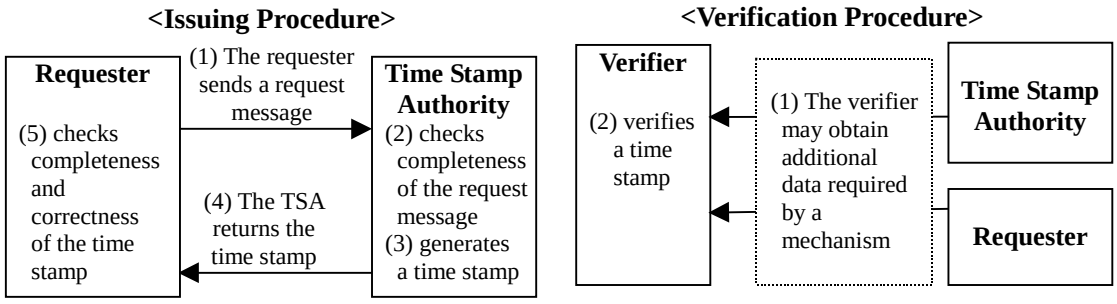
The classification described above is also adopted in standardization activities relating to a time stamping service. A working draft of ISO/IEC 18014 (Time stamping services, ISO/IEC [2001]) includes the following two types of scheme: "mechanisms producing independent tokens" and "mechanisms producing linked tokens." These correspond to the simple and linking schemes, respectively. On the other hand, ISO/IEC 13888 (Non-repudiation, ISO/IEC [1997]) and IETF PKIX TSP (Adams et al. [2000]) employ the simple scheme. The next section briefly explains an outline of ISO/IEC WD 18014.

B. Outline of ISO/IEC WD 18014

ISO/IEC WD 18014 mainly describes a general model on which time stamping services are based. The draft was developed in ISO/IEC JTC1/SC27 and consists of three parts: the framework (Part 1, ISO/IEC [2000a]), mechanisms producing independent tokens (Part 2, ISO/IEC [2000b]) and mechanisms producing linked tokens (Part 3, ISO/IEC [2000c]). The draft defines a time stamp (“time-stamp token”) as “a message consisting of data fields relevant to time-stamping and which contains information that has been transformed using a cryptographic technique.” As entities involving the scheme, it describes a time stamp authority, a requester and a verifier. The time stamp authority (TSA) is defined as a trusted third party. A TSA offers evidence that specific data existed at a certain point in time and guarantees the correctness of the time parameter. The requester is described as an entity possessing data that it wants to be time-stamped. The verifier is described as the entity

confirming validity of a time stamp.

Figure 1 Issuing and Verification Procedures of a Time Stamp in ISO/IEC WD 18014



The issuing procedure described in Part 1 consists of the following five steps (on the left hand side of Figure 1). The first step is one in which a requester sends a time stamp request message to a TSA. In the second step, the TSA checks the completeness of the received message. The third step is one in which the TSA generates a time stamp which includes at least a time parameter, a hash value of the data to be time-stamped and data to bind the time parameter to the hash value using a cryptographic technique. In cases in which the mechanism produces independent tokens, a time stamp does not include data included in other time stamps. On the other hand, a time stamp includes the data in the case of a mechanism which produces linked tokens. The fourth step is one in which the TSA returns the time stamp to the requester. At the fifth stage, the requester may immediately check the completeness and correctness of the received time stamp. On the other hand, in the verification procedure described in Part 1 (on the right hand side of Figure 1), a verifier obtains additional data required by the mechanism from other entities and verifies the time stamp by using them.

Parts 2 and 3 show different ways of binding a time parameter to a hash value of data to be time-stamped using a cryptographic technique.

Part 2 describes three sorts of protocols as examples of mechanisms producing independent tokens. The first is one in which the TSA adopts a digital signature as a tool to confirm the integrity of a time stamp. The second is one in which the TSA uses the MAC (Message Authentication Code) to confirm integrity instead of a digital signature. In this type, a verifier has to ask the TSA to check the integrity and believe the TSA's response. While the PKI is not needed, private keys to generate the MAC must be securely stored by the TSA. The third is one in which a TSA returns only reference data including an identifier of time-stamped data as a time stamp. In this type, a TSA has to store time-stamped data (or its hash value) and its time parameter securely. A verifier asks a TSA for the time parameter of data to be verified. Part 2 says that a TSA has to be thoroughly trusted because there is no external evidence by which anyone can detect fraud committed by the TSA.

Part 3 defines “a linked token” as a time stamp that is cryptographically linked to other time stamps. Part 3 describes three basic issuing processes: aggregation, linking and publishing. The aggregation process is used to reduce the load on a subsequent linking process and to provide a set of witnesses to a group of time stamps. In the linking process, a hash value representing either a time stamp or data aggregating a set of time stamps is linked to other hash values in order to both express a temporal order of time stamping events and to serve as a witness to all previously-linked events. The publishing process is designed to allow third parties to confirm the consistency of aggregation and linking processes. Part 3 says that the publishing process helps to prevent an attacker or a rogue TSA from tampering with the linkage data itself.

C. Characteristics of the Conventional Classification

The conventional classification explained above is not considered to be enough for a systematic evaluation of the security of time stamping schemes. In this classification, time stamping schemes are categorized only from the viewpoint of the way of generating a time stamp. Other features of time stamping schemes, for example, the verification procedures and data that a verifier obtains from other entities, are ignored. As a result, the classification is so rough that schemes having different features are put into the same category.

For example, Digital Notary/SecureSeal and PKITS are both classified into the linking scheme according to the conventional classification. However, the verification procedure of Digital Notary/SecureSeal is different from that of PKITS. In Digital Notary/SecureSeal, the verification procedure consists of the following two operations: The first is a comparison of the hash value of data to be verified with the hash value included in the time stamp; The second is issuer’s confirmation of the consistency between a time stamp and the issuer’s database. On the other hand, in PKITS, the verification procedure consists of the following four operations: The first is a comparison of the hash value of the data to be verified with the hash value included in the time stamp; The second is the verifier’s confirmation of the integrity of a time stamp by using issuer’s digital signature; The third is the verifier’s confirmation of the consistency between the time stamp and the issuer’s database; The fourth is the verifier’s confirmation of the integrity of the data used in the third operation by using other data.

It is reasonable to consider that Digital Notary/SecureSeal and PKITS have different security levels because their verification procedures are different. However, no information about the difference between them as regards security can be obtained by using the conventional classification. Thus, a method of comprehensively classifying time stamping schemes is needed.

III. Une and Matsumoto's Study of the Security Evaluation of Time Stamping Schemes

This section introduces an outline of Une and Matsumoto's studies (Une and Matsumoto [2000], [2001a], [2001b] and [2002]) of the security evaluation of time stamping schemes. Their results consist of the following two parts: a classification of time stamping schemes and a clarification of security against the alteration of a time stamp in each scheme.

At first, Une and Matsumoto [2000] and [2001a] defined five entities involved in time stamping schemes, a time stamp and the six operations that make up the verification procedures. These are defined in such a way as to cover features of existing schemes. Based on these definitions, time stamping schemes were classified into 108 categories.

Secondly, Une and Matsumoto [2001a], [2001b] and [2002] discussed the conditions sufficient for the detection of the alteration of a time stamp in each category. They focused on security against alteration because alteration is the most basic attack. As a result, they showed that time stamping schemes employing the same verification procedure have identical sufficient conditions and that there are ten variations of these sufficient conditions. Moreover, Une and Matsumoto clarified not only the relationships between these sufficient conditions but also schemes corresponding to each sufficient condition.

A. Entities of Time Stamping Schemes

As entities involved in time stamping schemes, a time stamp issuer, an evidence amplifier, a prover, a verifier and a time stamp requester are defined (Table 2).

A time stamp issuer (known as an issuer) is defined as an entity that issues a time stamp and stores all data relating to its issue and verification. In some cases, an issuer generates E_{TSI} and E_{AMP}. E_{TSI} is defined as data used to confirm the consistency between data included in a time stamp and the corresponding data in the issuer's database. A subscript "*TSI*" of E_{TSI} denotes a "*Time Stamp Issuer*" and means that an issuer keeps E_{TSI}. On the other hand, E_{AMP} is defined as data used to confirm the integrity of E_{TSI} and is sent to an evidence amplifier (known as an amplifier) by an issuer. A subscript "*AMP*" of E_{AMP} denotes "*AMPlifier*" and means that an evidence amplifier keeps E_{AMP}. An issuer is not supposed to be a trusted third party.

An amplifier is defined as an entity that stores E_{AMP} and provides it to a verifier during the verification phase. An amplifier has the function of amplifying the genuineness of E_{TSI} by keeping E_{AMP} secure. In the case of a scheme in which a part of the data used for the verification procedure is published in a newspaper, the medium and published data correspond to the amplifier and E_{AMP}, respectively. An amplifier is also supported to be a non-trusted third party. As an entity having a similar function, Buldas et al. [2000] proposed a publication authority (abbreviated as a PA). A PA

plays the role of a publisher of the data used to verify a time stamp on some authenticated and easily accessible medium.

Table 2 Entities Involved in a Time Stamping Scheme and Data Included in a Time Stamp

		meanings
entities	time stamp issuer	an entity that issues a time stamp and stores all data relating to its issuing and verification processes
	evidence amplifier	an entity that stores E_{TSI} and provides it to a verifier during the verification procedure
	prover	an entity that claims that certain data existed before a specific point in time and proves the fact
	verifier	an entity that confirms whether or not a prover's claim is true
	time stamp requester	an entity that requests that a time stamp issuer issue a time stamp by sending request data *REQ* and obtains the corresponding time stamp
data	T	a time parameter indicating a specific point in time at which a time stamp issuer received *REQ* from a time stamp requester
	H	the hash value of data to be time-stamped
	REQ	request data to issue a time stamp
	E_{TSI}	data used to confirm the consistency between data included in a time stamp and the corresponding data in the time stamp issuer's database
	E_{AMP}	data used to confirm the integrity of E_{TSI} and kept by an evidence amplifier
	E_{ORE}	data used to confirm the integrity of E_{TSI} and kept by time stamp requesters
	$Info_{INT}$	data used to confirm the integrity of data included in a time stamp
	ID_{TSI}	the identifier of a time stamp issuer
	ID_{AMP}	the identifier of an evidence amplifier
	ID_{ORE}	the identifier of time stamp requesters holding E_{ORE}

A prover is defined as an entity claiming that certain data existed before a specific point in time and proving the fact. A prover sends a time stamp and the corresponding data M to a verifier at the beginning of the verification phase.

A verifier is defined as an entity confirming whether or not a prover's claim is true. During the verification phase, a verifier collects various data from other entities.

A time stamp requester (known as a requester) is an entity that requests that an issuer issue a time stamp by sending request data *REQ* and obtains the corresponding time stamp. When a time stamp includes E_{ORE}, that is data used to confirm the integrity of E_{TSI}, certain requesters holding such time stamps send E_{ORE} to the verifier during the verification phase. Requesters holding E_{ORE} are different from requesters holding a time stamp to be verified. A subscript "*ORE*" of E_{ORE} denotes "*Other Requesters.*"

With respect to the entities, the main difference between ISO/IEC WD 18014 and Une and Matsumoto's results is whether or not an amplifier is clearly defined.

B. What Is a Time Stamp?

Generally, a time stamp is defined as digital data that proves the existence of certain data prior to a specific point in time (for example, ISO/IEC [2000a]). ISO/IEC WD 18014-1 specifies that a time stamp includes the following three data: a time parameter (*T*) generated or received from a reliable source, the hash value (*H*) delivered by a requester and data generated by the TSA to bind *T* to *H* cryptographically.

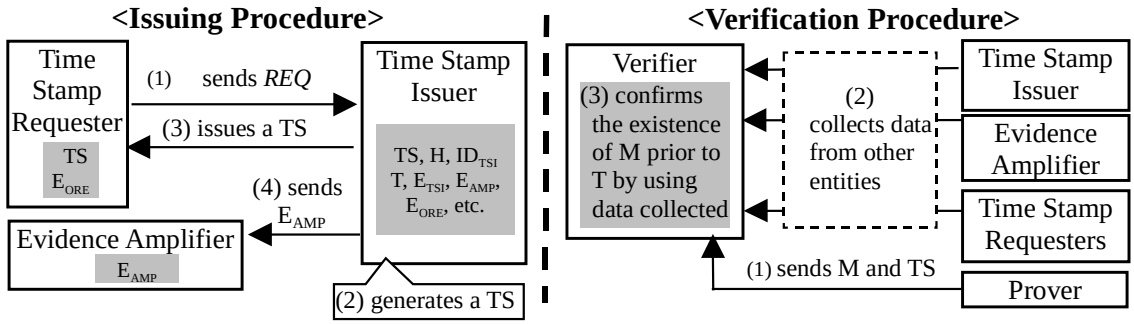
On the other hand, Une and Matsumoto assume that a time stamp includes at least *H* and an identifier of the issuer (ID_{TSI}). It is noted that a time stamp as defined by Une and Matsumoto does not include *T*. This is because the following time stamping scheme is supposed: a time stamp does not include *T*, and a verifier obtains it from the issuer during the verification phase. Therefore, the definition of Une and Matsumoto is considered to be wider than that of ISO/IEC WD 18014.

Une and Matsumoto assume that a time stamp may optionally include the following data: *T*, E_{TSI}, E_{ORE}, $Info_{INT}$, ID_{ORE}, ID_{AMP} and ID_{ORE}. $Info_{INT}$ is defined as data used to confirm the integrity of the data included in a time stamp. A subscript "*INT*" of $Info_{INT}$ indicates "*INTegrity*." A digital signature is one example of $Info_{INT}$. ID_{REQ}, ID_{AMP} and ID_{ORE} are identifiers of the requester holding the time stamp, an amplifier and requesters holding E_{ORE}, respectively.

C. Issuing and Verification Procedures

The issuing and verification procedures defined by Une and Matsumoto are shown in Figure 2. The issuing procedure is nearly identical to that of the simple scheme shown in Figure 1 except for a step in which an issuer sends E_{AMP} to an amplifier. These procedures are both based on the assumption that an issuer can always obtain the time parameter with the necessary accuracy by some method, for example, Network Time Protocol.

Figure 2 Verification Procedures of a Time Stamp in Une and Matsumoto



[Issuing Procedure (Left of Figure 2)]

(1) A requester sends a time stamp request data (*REQ*) to the issuer.

(2) The issuer generates a time stamp by using data included in *REQ* and *T*. In the case of a scheme in which an amplifier and requesters are used during the verification phase, the issuer includes ID_{AMP} and ID_{ORE} in the time stamp.
(3) The issuer sends the time stamp to the requester that originated *REQ*.
(4) If E_{AMP} is used during the verification phase, the issuer may send E_{AMP} to the amplifier.

[Verification Procedure (Right of Figure 2)]
(1) A prover sends at least data to be verified (*M*) and the corresponding time stamp to a verifier.
(2) The verifier collects data relating to the verification procedure from other entities.
(3) The verifier carries out a predetermined verification procedure with collected data. By using the results, the verifier decides whether or not *M* has existed prior to *T*.

D. Verification Operations

The verification procedures employed in some existing schemes are decomposed, and the following six verification operations: **a**, **b**, **c**, **d**, **e** and **f** are defined.

Operation **a** is defined as one in which a verifier compares a hash value of *M* with *H* included in a time stamp to be verified.

Operation **b** is one in which a verifier confirms the integrity of data included in a time stamp by using $Info_{INT}$. For example, in a case in which $Info_{INT}$ is the digital signature of an issuer, the operation is to verify a digital signature by using the issuer's public key. This operation includes not only carrying out a verification algorithm on a signature but also checking the validity of the public key.

Operation **c** is one in which a verifier asks an issuer to confirm the consistency between a time stamp to be verified with the corresponding data in the issuer's database. First, a verifier sends a time stamp to the issuer. Then, the issuer confirms the consistency and returns its result to the verifier. Therefore, the verifier has to trust the result sent by the issuer. In the case of a scheme in which a time stamp does not include *T*, an issuer sends *T* with the result. In order to use this operation, an issuer must be sufficiently trustworthy.

Operation **d** is one in which a verifier confirms the consistency between data included in a time stamp and the corresponding data in the issuer's database by E_{TSI}. A verifier has to obtain E_{TSI} from the issuer during the verification phase in the case of schemes in which a time stamp does not include E_{TSI}. In the case of schemes in which a time stamp includes it, a verifier can also obtain E_{TSI} from the issuer during the verification phase.

Operation **e** is one in which a verifier obtains E_{AMP} from an amplifier and confirms the integrity of E_{TSI} by using E_{AMP}. This operation can be used in schemes in which a time stamp includes E_{TSI} or in which a verifier can obtain E_{TSI} from the issuer during the verification phase.

Finally, operation **f** is one in which a verifier obtains E_{ORE} from predetermined requesters and confirms the integrity of E_{TSI} by using E_{ORE}. As well as operation **e**, this operation can also be used in schemes in which a time stamp includes E_{TSI} or in which a verifier can obtain E_{TSI} from an issuer during the verification phase.

In any time stamping scheme, checking the format of a time stamp to be verified is commonly carried out at the beginning of the verification procedure. Therefore, Une and Matsumoto do not make use of it for the classification which is explained below.

From these operations, 32 patterns of verification procedures are supposed by combining operations **b**, **c**, **d**, **e** and **f**. This is because Une and Matsumoto set operation **a** as a mandatory one in all verification procedures. Each verification procedure is described as a combination of operations. For example, **ade** denotes a verification procedure consisting of three operations **a**, **d** and **e**.

Une and Matsumoto define a "type" as a category of schemes having the same verification procedure. The number of types is 32. For example, type **ade** denotes schemes adopting **ade** as the verification procedure.

E. Classification of Time Stamping Schemes

Time stamping schemes are classified into ten groups from the following three viewpoints: ways of generating a time stamp, data included in a time stamp and a verifier's availability of E_{TSI}.

First, schemes are divided into the following three: one in which a time stamp includes neither T nor E_{TSI} (NN), one in which although a time stamp includes T, it does not include E_{TSI} (TN) and one in which a time stamp includes both T and E_{TSI} (TE).

Next, schemes are divided into the following two: "evidence-Available schemes (A)" in which a verifier can obtain E_{TSI} and "evidence-Unavailable schemes (U)" in which a verifier cannot do so.

Finally, focusing on a way to generate a time stamp, schemes are divided into "Linked time stamp schemes (L)" and "Isolated time stamp schemes (I)." In schemes of linked time stamp schemes, an issuer is supposed to generate a time stamp with data included in certain other time stamps. In isolated time stamp schemes, it is supposed to do so without the data.

Combining these classifications, Une and Matsumoto defined 10 groups of time stamping schemes: NN-U-I, NN-U-L, NN-A-I, NN-A-L, TN-U-I, TN-U-L, TN-A-I, TN-A-L, TE-A-I and TE-A-L (Table 3). For example, NN-U-L means a group of schemes having the three features of NN, U and L. The combination of TE and U does not exist because a time stamp of TE always includes E_{TSI}.

In addition, each of these ten groups is divided with respect to applicable verification procedures. Operation **a** is mandatory for all groups. Operation **b** is applicable to all groups. Operation **c** is mandatory for NN because its time stamp doesn't include T. Operation **c** is applicable to the other groups. Operations **d** and **e** are applicable to all groups except for NN-U and TN-U. This is because a

verifier cannot obtain E_{TSI} in these groups. Operation **f** is applicable only to L.

The total number of the schemes classified is 108 as shown in Table 3. For example, schemes of NN-U-L can be classified into NN-U-L with **ac** and **abc** because NN-U-L can adopt either of the two verification procedures **ac** and **abc**.

The characteristic of Une and Matsumoto's method of classifying time stamping schemes is that the method covers not only the issuing procedure of a time stamp but also both the verification procedure and the contents of a time stamp. It is considered that time stamping schemes can be more closely classified by the proposed method than by the conventional method.

Table 3 Une and Matsumoto's Classification of Time Stamping Schemes

Groups of the schemes	Viewpoints of Classification			Applicable verification procedures
	Classification by data included in a time stamp	Classification by the verifier's availability of E_{TSI}	Classification by how to generate a time stamp	
NN-U-I	NN (No time data and No evidence schemes)	U (evidence-Unavailable schemes)	I (Isolated time stamp schemes)	**ac, abc**
NN-U-L			L (Linked time stamp schemes)	**ac, abc**
NN-A-I		A (evidence-Available schemes)	I	**ac, abc, acd, abcd, acde, abcde**
NN-A-L			L	**ac, abc, acd, abcd, acde, acdf, abcde, abcdf, acdef, abcdef**
TN-U-I	TN (Time data and No evidence schemes)	U	I	**a, ab, ac, abc**
TN-U-L			L	**a, ab, ac, abc**
TN-A-I		A	I	**a, ab, ac, ad, abc, abd, acd, ade, abcd, abde, acde, abcde**
TN-A-L			L	**a, ab, ac, ad, abc, abd, acd, ade, adf, abcd, abde, abdf, acde, acdf, adef, abcde, abcdf, abdef, acdef, abcdef**
TE-A-I	TE (Time data and Evidence schemes)	A	I	**a, ab, ac, ad, ae, abc, abd, abe, acd, ace, ade, abcd, abce, abde, acde, abcde**
TE-A-L			L	**a, ab, ac, ad, ae, af, abc, abd, abe, abf, acd, ace, acf, ade, adf, aef, abcd, abce, abcf, abde, abdf, abef, acde, acdf, acef, adef, abcde, abcdf, abcef, abdef, acdef, abcdef**

F. Security Analysis

Using the classification explained above, Une and Matsumoto [2001a], [2001b] and [2002] discussed the security of each scheme. They selected the alteration of a time stamp as an attack to be discussed. This is because the alteration of a time stamp is considered to be the most basic attack. Under some assumptions, Une and Matsumoto [2001a] discussed conditions sufficient to detect the alteration of a time stamp in each scheme. In other words, it clarified the following relationship in each scheme: if a certain condition is satisfied, the scheme is secure against the alteration of a time stamp. Une and

Matsumoto [2001a] found that the sufficient condition of a certain scheme depends on its verification procedure. By using the result of Une and Matsumoto [2001a], Une and Matsumoto [2001b] and [2002] showed that there are just ten variations of the sufficient conditions. It also clarified types corresponding to each sufficient condition and the relationships between the sufficient conditions.

1. Six Assumptions

The following six assumptions are set. The first is that all information relating to time stamping schemes, except for secret data needed for cryptographic operations, are open.

The second is that an attacker attempts to alter H included in a time stamp TS into H'. H' denotes a hash value of M' that the attacker is willing to replace with M as data to be time-stamped. TS' denotes the altered time stamp that includes H'.

The third is that while an attacker obtains all the public information about security of a cryptographic technique needed to generate $Info_{INT}$, the attacker does not obtain enough of the issuer's secret data to generate $Info_{INT}$ without colluding with the issuer.

The fourth is that the hash function used to generate H is second pre-image resistant. This means that it is computationally infeasible for an attacker to find any second input that has the same hash value as any specified input. In accordance with this assumption, an attacker cannot make TS correspond to M' without altering H.

The fifth is that the cryptographic techniques employed in operations **c**, **d**, **e** and **f** (for example, a one-way hash function) are supposed to have no flaw in security.

The sixth is that all data transmitted among entities during issuing and verification phases assure confidentiality and integrity.

2. Three Conditions

The following three conditions are set. The first is whether or not a cryptographic technique to generate $Info_{INT}$ becomes weak at the time of the attack. “The technique becomes weak” means that the technique has a security flaw serious enough to allow someone to forge $Info_{INT}$ without the issuer's secret data and only the attacker is aware of this fact. On the other hand, “the technique does not become weak” means not only a situation in which the critical flaw does not exist but also one in which an attacker does not notice the flaw even though it exists. For example, in the case of a digital signature, one of the situations in which the technique becomes weak corresponds to one in which only the attacker finds a method of efficiently forging an issuer’s digital signature without his private key.

The second is whether or not an attacker colludes with the issuer, an amplifier or requesters holding E_{ORE}. Even though an insider fraudulently cooperates with the attacker, the situation is called collusion. It is assumed that if an attacker colludes with other entities, the attacker can make them do

whatever it wants, such as forgery of $Info_{INT}$, E_{TSI}, E_{AMP} and E_{ORE}. It is natural to consider that no matter how trustworthy those entities are, it is difficult to affirm that they do not behave fraudulently. Therefore, it is reasonable to take the possibility of collusion into consideration.

The third is whether or not an attacker impersonates the issuer, an amplifier or requesters holding E_{ORE}. It is also supposed that if an attacker impersonates them, it can carry out the same functions as they can and make a verifier obtain forged data suitable for the attacker.

3. Analysis of Conditions Sufficient to Detect Alteration in Each Scheme

The sufficient conditions are analyzed in the following two steps (Une and Matsumoto [2001b]). At first, the sufficient conditions are discussed in time stamping schemes employing relatively simple verification procedures. Those procedures are as follows: **a**, **ab**, **ac**, **ad**, **ae**, **af**, **ade** and **adf**. Next, combining the results of the first step, the sufficient conditions are discussed in the context of schemes employing more complicated verification procedures.

a. Type a

Type **a** consists of schemes adopting operation **a** as the verification procedure. A verifier compares *H'* included in a time stamp with a hash value of *M'*. A verifier cannot detect an attacker's alteration of a time stamp by operation **a** under any conditions because *H'* is a hash value of *M'*. Therefore, the sufficient condition in type **a** is empty.

b. Type ab

The verification procedure of type **ab** consists of operations **a** and **b**. Because it is known that a verifier cannot detect alteration by operation **a**, a sufficient condition to detect the alteration by operation **b** is discussed here.

In a case in which the technique to generate $Info_{INT}$ does not become weak, if an attacker does not collude with an issuer, the attacker does not forge $Info'_{INT}$ consistently with *TS'*. Therefore, a verifier detects the alteration during the verification procedure. Otherwise, an attacker forges it, and a verifier cannot detect the alteration. On the other hand, in a case in which the technique used to generate $Info_{INT}$ becomes weak, an attacker forges $Info'_{INT}$ consistently with *TS'*. A verifier cannot detect the alteration. As a result, the sufficient condition is that the technique to generate $Info'_{INT}$ does not become weak and an attacker does not collude with the issuer.

c. Type ac

In a case in which an attacker colludes with the issuer, the attacker makes the issuer send a result of the confirmation that *TS'* is consistent with the issuer's database. Therefore, a verifier cannot detect the alteration made during the verification procedure. In a case in which an attacker impersonates an

issuer, the attacker informs a verifier of the same result, and the verifier cannot detect the alteration. Otherwise, the verifier obtains an appropriate result of operation **c** from a proper issuer and detects the alteration. As a result, the sufficient condition is that an attacker neither colludes with nor impersonates an issuer.

d. Type ad

In a case in which an attacker colludes with the issuer, a verifier cannot detect the alteration because it obtains E'_{TSI} forged consistently with *TS'*. In a case in which an attacker impersonates the issuer, a verifier cannot also detect the alteration as well as a case of collusion. Otherwise, the verifier obtains E_{TSI} from an issuer and detects the alteration. As a result, the sufficient condition is that an attacker neither colludes with nor impersonates the issuer.

e. Types ae and af

Although an attacker replaces *H* with *H'*, the attacker does not forge E_{TSI} because operation **d** is not carried out by a verifier. Therefore, a verifier cannot detect the alteration under any conditions. The sufficient conditions in types **ae** and **af** are empty as well as that of type **a**.

f. Types ade and adf

From the results explained above, it seems that operations **e** and **f** have no effect in detecting an alteration. However, it is to be expected that they will have such an effect when operation **d** is carried out. This is because an attacker must forge E_{AMP} and E_{ORE} in a way to assure consistency with E'_{TSI} forged in order to make operation **d** ineffective.

In case of operation **e**, if an attacker neither colludes with nor impersonates an issuer, a verifier detects the alteration when operation **d** is carried out. If an attacker neither colludes with nor impersonates an amplifier, a verifier detects the alteration when operation **e** is carried out. Otherwise, a verifier cannot detect the alteration because an attacker makes a verifier obtain E'_{TSI} and E'_{AMP} forged consistently with *TS'*. As a result, the sufficient condition is that an attacker neither colludes with nor impersonates an issuer or that an attacker neither colludes with nor impersonates an amplifier.

In case of operation **f**, as well as operation **e**, if an attacker neither colludes with nor impersonates an issuer, a verifier detects the alteration when operation **d** is carried out. In a case in which an attacker neither colludes with nor impersonates requesters holding E_{ORE}, a verifier detects the alteration when operation **f** is carried out. This is because, in order to assure consistency with operation **d**, an attacker has to forge E'_{TSI}. Otherwise, a verifier cannot detect the alteration because an attacker makes a verifier obtain E'_{TSI} and E'_{ORE} forged consistently with *TS'*. As a result, the sufficient condition is that an attacker neither colludes with nor impersonates an issuer or that an attacker neither colludes with nor impersonates requesters holding E_{ORE}.

g. Types employing More Complicated Verification Procedures

The sufficient conditions in the other types can be discussed by using the results explained above. In the case of type **abc**, the sufficient condition is the union of types **ab** and **ac**, for example. In other words, the sufficient condition is that the technique which is used to generate $Info'_{INT}$ does not become weak and an attacker does not collude with an issuer or that an attacker neither colludes with nor impersonates an issuer.

Before describing the sufficient conditions in all types, symbols indicating conditions are defined as below.

- J: A technique to generate $Info_{INT}$ does not become weak.
- K: An attacker does not collude with an issuer.
- N: An attacker does not impersonate an issuer.
- O: An attacker does not collude with an amplifier.
- P: An attacker does not impersonate an amplifier.
- Q: An attacker does not collude with requesters holding E_{ORE}.
- R: An attacker does not impersonate requesters holding E_{ORE}.

In addition, in order to enhance the reader's understanding, JK, KN, OP and QR are denoted by "J and K," "K and N," "O and P" and "Q and R," respectively.

Using the symbols defined above, the relationship between each type and the corresponding sufficient condition are shown in Table 4. Table 4 indicates that there are ten variations in the sufficient conditions.

Table 4 Each Type and its Corresponding Sufficient Condition

Types	Sufficient Conditions
a, ae, af, aef	empty (The verifier cannot detect the alteration under any conditions.)
ab, abe, abf, abef	JK
ac, ad, acd, ace, acf, acef	KN
abc, abd, abcd, abce, abcf, abcef	JK or KN
ade, acde	KN or OP
abde, abcde	JK or KN or OP
adf, acdf	KN or QR
abdf, abcdf	JK or KN or QR
adef, acdef	KN or OP or QR
abdef, abcdef	JK or KN or OP or QR

h. Ten Classes and the Relationships between their Corresponding Sufficient Conditions

Based on the relationships explained above, a "class" is defined as a category of schemes having the same sufficient conditions. There are just ten variations of classes, and the definition of each class is shown in Table 5.

Table 5 Definition of Each Class

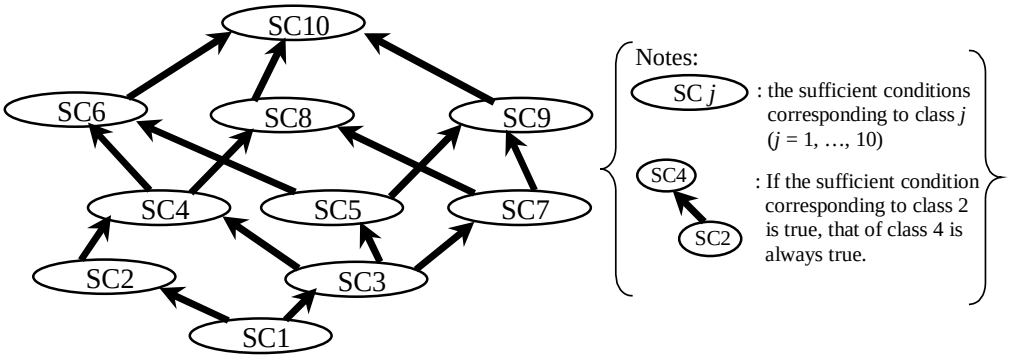
Classes	Definitions
1	all schemes (including ones in which the verifier cannot detect the alteration under any conditions)
2	schemes in which the verifier detects the alteration under the condition JK
3	schemes in which the verifier detects the alteration under the condition KN
4	schemes in which the verifier detects the alteration under the condition JK or KN (the intersection of classes 2 and 3)
5	schemes in which the verifier detects the alteration under the condition KN or OP
6	schemes in which the verifier detects the alteration under the condition JK or KN or OP (the intersection of classes 4 and 5)
7	schemes in which the verifier detects the alteration under the condition KN or QR
8	schemes in which the verifier detects the alteration under the condition JK or KN or QR (the intersection of classes 4 and 7)
9	schemes in which the verifier detects the alteration under the condition KN or OP or QR (the intersection of classes 5 and 7)
10	schemes in which the verifier detects the alteration under the condition JK or KN or OP or QR (the intersection of classes 6, 8 and 9)

Next, the relationships between the sufficient conditions are discussed. In the following, the relationship between certain sufficient conditions A and B is denoted by "A $\Rightarrow$ B" when the following relationship is satisfied: if a certain condition A is true, another condition B is always true. In other words, the relationship is termed "B is weaker than A." The sufficient condition corresponding to class *i* denotes SC*i*. SC1 is defined as null. The relationships between the sufficient conditions are described below (Figure 3).

- SC1 $\Rightarrow$ SC2, SC3, SC4, SC5, SC6, SC7, SC8, SC9 and SC10
- SC2 $\Rightarrow$ SC4, SC6, SC8 and SC10
- SC3 $\Rightarrow$ SC4, SC5, SC6, SC7, SC8, SC9 and SC10
- SC4 $\Rightarrow$ SC6, SC8 and SC10
- SC5 $\Rightarrow$ SC6, SC9 and SC10
- SC6 $\Rightarrow$ SC10
- SC7 $\Rightarrow$ SC8, SC9 and SC10
- SC8 $\Rightarrow$ SC10
- SC9 $\Rightarrow$ SC10

Figure 3 indicates that SC10 is the weakest. Therefore, schemes belonging to class 10 are considered to be the most secure against the alteration of a time stamp.

Figure 3 The Relationships Between the Sufficient Conditions



i. Time Stamping Schemes belonging to Each Class

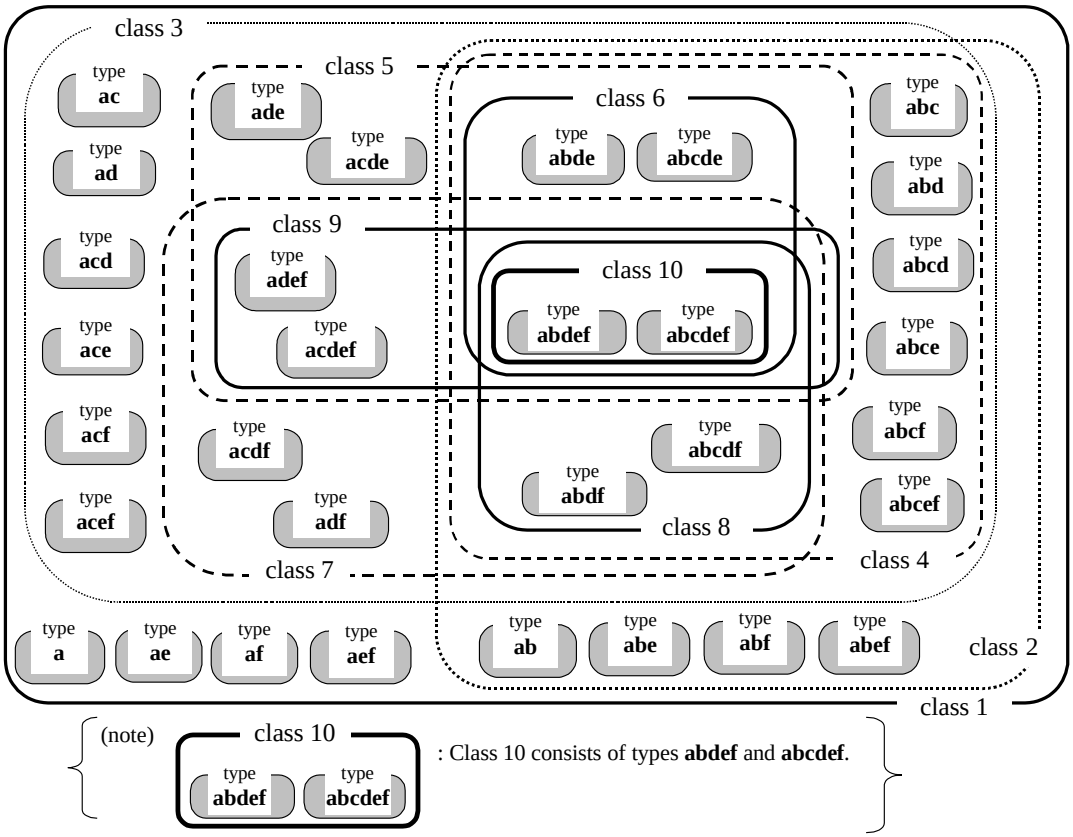
Using the relationships between the sufficient conditions, schemes belonging to each class are clarified.

Tables 4 and 5 show that types **abdef** and **abcdef** belong to class 10. SC10 is the weakest sufficient condition. Therefore, class 10 consists of types **abdef** and **abcdef**. With respect to class 9, Tables 4 and 5 show that types **adef** and **acdef** belong to class 9. From Figure 3, only SC10 is weaker than SC9. Therefore, class 9 is the union of types **abdf** and **abcdf** and class 10. By applying the same classification method to classes from 2 to 8, types belonging to each class and types are clarified as shown in Figure 4.

From the viewpoint of the cost of carrying out the verification procedure, the scheme in which the number of the verification operations is the smallest is considered to be the most desirable. Such schemes are types **a** and **ab** in classes 1 and 2, respectively. In classes 3 and 4, types **ac** and **ad** and types **abc** and **abd** are the most desirable, respectively. In classes 5, 6, 7, 8, 9 and 10, types **ade**, **abde**, **adf**, **abdf**, **adef** and **abdef** are the most desirable, respectively.

This result implies that the linked time stamp schemes can achieve higher security level against alteration than the isolated time stamp schemes. Although both the linked time stamp schemes and the isolated time stamp schemes can be used for applications whose security requirement corresponds to one of classes from 1 to 9, only the linked time stamp schemes can be employed in applications whose security requirement corresponds to class 10.

Figure 4 Relationships between Classes and Types



4. Summary of Analysis

Une and Matsumoto clarified the following three facts. The first is that security against the alteration of a time stamp depends on the verification procedure employed in each scheme. The second is that there are ten variations in the sufficient conditions for detecting the alteration. The third is that there are clear relationships between the sufficient conditions and that class 10, consisting of types **abdef** and **abcdef**, are the most desirable from the viewpoint of security against alteration.

IV. Analysis of Security of Time Stamping Schemes Using Une and Matsumoto's Results

Users of time stamping schemes can make use of Une and Matsumoto's results when selecting a time stamping service. This section applies Une and Matsumoto's results to seven existing time stamping schemes.

The procedure of applying the results consists of the following three steps. The first is to identify a type corresponding to the scheme to be evaluated by focusing on its verification operations. The second is to check which class includes the type and then obtain the sufficient condition for detecting the alteration. The third is to confirm whether or not the sufficient condition is satisfied. Time stamping schemes to be analyzed in this section are shown in Table 6.

Table 6 Main Features of Seven Time Stamping Schemes to be Analyzed

Schemes	Data included in a time stamp	Entities	Verification procedures
The electronic notarization system	H, ID_{TSI}, T, $Info_{INT}$ etc.	a requester, an issuer	- comparison of hash values (operation **a**) - confirmation of the integrity of a time stamp by checking a digital signature provided by the issuer (operation **b**)
The time signature distributed system	H, T, $Info_{INT}$, etc.	a requester, an issuer	- comparison of hash values (operation **a**) - confirmation of the integrity of a time stamp by using a digital signature provided by the issuer (operation **b**)
Digital Notary /SecureSeal	H, ID_{TSI}, ID_{REQ}, T, etc.	a requester, an issuer, an amplifier (newspaper)	- comparison of hash values (operation **a**) - confirmation of the consistency between a time stamp and the issuer's database (operation **c**)
PKITS	H, ID_{TSI}, ID_{REQ}, ID_{AMP}, T, $Info_{INT}$, serial number, etc.	a requester, an issuers, an amplifier (other time stamp issuers)	- comparison of hash values (operation **a**) - confirmation of the integrity of a time stamp by a digital signature (operation **b**) - confirmation of the consistency between a time stamp and linking information stored by the issuer (operation **d**) - comparison between linking information regenerated and obtained from other issuers (operation **e**)
TIMESEC	H, ID_{TSI}, ID_{REQ}, ID_{AMP}, T, $Info_{INT}$, serial number, etc.	a requester, an issuer, an amplifier (the online site of the issuer)	- comparison of hash values (operation **a**) - confirmation of the integrity of a time stamp by a digital signature by the issuer (operation **b**) - confirmation of the consistency between a time stamp and linking information stored by the issuer (operation **d**) - comparison between linking information regenerated and obtained from an online site (operation **e**)
The scheme proposed by Benaloh and de Mare	H, T, E_{TSI}, etc.	a requester, an issuer	- comparison of hash values (operation **a**) - confirmation of the consistency between y_i, z_i and z (operation **d**)
The scheme proposed by Buldas et al.	H, D_{TSI}, ID_{AMP}, E_{TSI}, $Info_{INT}$, serial number, etc.	a requester, an issuer, an amplifier (newspaper)	- comparison of hash values (operation **a**) - confirmation of the integrity of a time stamp by a digital signature (operation **b**) - confirmation of the consistency between a time stamp and linking information stored by the issuer (operation **d**) - comparison between linking information regenerated and obtained from a newspaper (operation **e**)

In the seven schemes listed in Table 6, Une and Matsumoto [2001b] and [2002] have already analyzed the following five schemes: the electronic notarization system, the time signature distributed system, Digital Notary/SecureSeal, PKITS and TIMESEC. The other schemes, the schemes proposed by Benaloh and de Mare [1994] and by Buldas et al. [2000], will be newly analyzed in the following subsection.

A. The Application of Seven Time Stamping Schemes

1. The Electronic Notarization System

The electronic notarization system (The Study Group on the Legal System of Electronic Commerce [1998]) has been promoted by the Japanese Ministry of Justice. This system consists of the following four services: a time stamping service; notarization of an electronic private document; preparation of an electronic notarial document; and maintenance of electronic documents and certification of the existence and the contents of the electronic documents.

In the time stamping service of the electronic notarization system, a notary uses a digital signature as a tool for assuring the integrity of a time stamp. A requester sends data X to be time-stamped to the electronic notarization center or to a specific notary office. After receiving X, a notary concatenates X with the time parameter T and generates a digital signature S on a data set of X and T (shown as $[X, T]$). The notary hashes $[X, T, S]$ into H, stores H and sends $[X, T, S]$ as a time stamp corresponding to X. In the verification procedure, a verifier carries out the off-line verification of S by using the public key of the notary. Thus, a scheme of the time stamping service is considered to correspond to a group of TN-U-I.

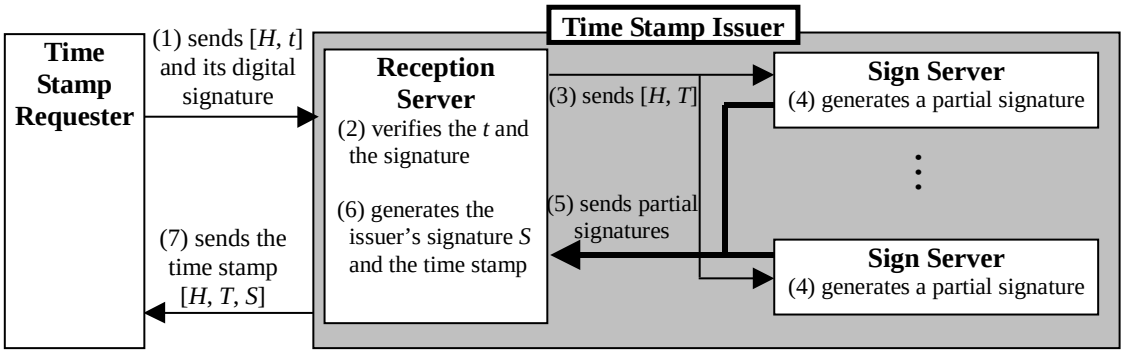
The operation of verifying a digital signature corresponds to operation **b**. In addition, it is naturally considered that a verifier first checks the correspondence between data to be verified and X included in the time stamp. Therefore, the verification procedure is set as **ab** and is classified into type **ab**. Figure 4 indicates that type **ab** belongs to class 2. The sufficient condition corresponding to class 2 is that a technique to generate $Info_{INT}$ does not become weak and that an attacker does not collude with an issuer.

This result suggests the following two points that arise when evaluating the security of schemes similar to the electronic notarization system. First, it is necessary to check whether or not it is infeasible for an attacker to forge a digital signature employed in the scheme without the issuer's private key. Secondly, it is necessary to check whether or not the issuer is trustworthy enough to believe that the issuer does not carry out any fraudulent manipulation in the course of operations relating to its service.

2. The Time Signature Distributed System

The main feature of the time signature distributed system proposed by Takura et al. [1999] is that an issuer consists of two kind of servers: a reception server and multiple sign servers.

Figure 5 Issuing Procedure of the Time Signature Distributed System



In the issuing procedure (Figure 5), a requester sends an issuer request data consisting of a hash value H of data to be time-stamped, a valid period data t and a digital signature on $[H, t]$. The reception server receives the data, verifies the signature and confirms that the validity period does not expire. Next, the reception server attaches H with the time parameter T and sends $[H, T]$ to sign servers at the same time. Each sign server contains part of a private key used to generate a digital signature. Each sign server generates a partial signature on $[H, T]$ when the current time is close to T and returns the signature to the reception server. If the number of partial signatures received by the reception server is more than a predetermined number (a threshold), the reception server can generate a digital signature S and sends the time stamp to the requester. The time stamp contains H, T, S and so on. Although multiple independent sign servers must be prepared in implementing this system, the use of the partial digital signature together with the threshold makes it more difficult for an issuer to fraudulently manipulate the operations involved in generating a time stamp. This system is classified into a group of TN-U-I.

In the verification procedure, a verifier carries out the verification of a digital signature by the issuer. Therefore, a verification operation is to verify the digital signature by the issuer and corresponds to operation **b**. Takura et al. [1999] does not clearly show that a verifier confirms the correspondence between the data to be verified and the hash value included in the time stamp. However, it is considered that this operation is implicitly included in the verification procedure. As a result, the verification procedure is set as **ab**, and the scheme is classified into type **ab**. Figure 4 indicates that type **ab** belongs to class 2. The sufficient condition corresponding to class 2 is that the technique to generate $Info_{INT}$ does not become weak and that an attacker does not collude with an

issuer.

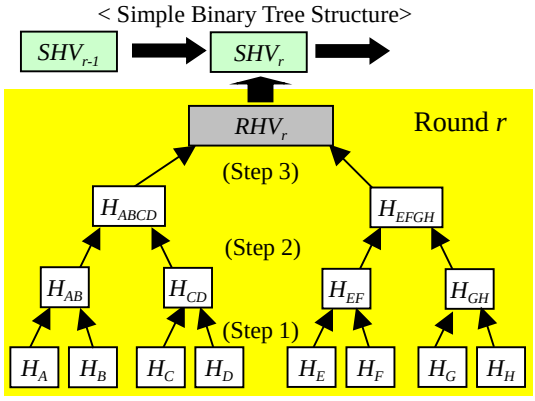
This result suggests the following points when evaluating the security of schemes similar to the time signature distributed system. First, it is necessary to check whether or not it is infeasible for an attacker to forge the digital signature employed in the scheme without the partial private keys of the sign servers. Secondly, it is necessary to check whether or not both the reception server and the sign servers are trustworthy enough to believe that they do not carry out any fraudulent manipulation of any of the operations relating to the service.

3. Digital Notary/SecureSeal

Digital Notary is a time stamping service provided by Surety.com. SecureSeal is the same service as Digital Notary and is provided by NTT Data as an agent of Surety.com in Japan.

Entities relating to Digital Notary are a requester and an issuer (Surety.com [2001]). A time stamp consists of T_n, ID_n, H_n and L_n, which are a time parameter, an identifier of the time stamp, a hash value to be time-stamped and a data set of other hash values received by the issuer in the same round, respectively. A subscript "*n*" of the parameters shows the serial number of the time stamps. Hash values are generated with two hash functions SHA-1 and MD5, and their length is set as 288 bits.

Figure 6 Example: Generating *RHV* Using a Simple Binary Tree Structure



< Procedures of Generating *RHV* >

- Assumption: In round *r*, there are eight requesters. They send a set of hash values to be time-stamped, [H_A, H_B, …, H_H], to an issuer.
- Step 1: The issuer makes four pairs (H_A, H_B), (H_C, H_D), (H_E, H_F) and (H_G, H_H), concatenates each pair and hashes them into new four hash values H_{AB}, H_{CD}, H_{EF} and H_{GH}, respectively.
- Step 2: The issuer concatenates H_{AB} and H_{EF} with H_{CD} and H_{GH}, respectively, and hashes them into H_{ABCD} and H_{EFGH}, respectively.
- Step 3: The issuer concatenates H_{ABCD} with H_{EFGH} and hashes them into *RHV* of round *r*.
- Step 4: The issuer concatenates RHV_r with SHV_{r-1} and hashes them into SHV_r of round *r*.

In the issuing procedure, a requester generates H_n and first sends it to the issuer. It is assumed that the issuer receives H_n in round *k*. The round is updated every second. After receiving H_n, the issuer generates a hash value RHV_k (Root Hash Value of round *k*) with all the hash values received during round *k*. *RHV* is generated by a simple binary tree structure (Figure 6). The data used to generate RHV_k except for H_n are assigned to L_n. This means that RHV_k can be generated by using H_n and L_n. The issuer generates a time stamp TS_n (including T_n, ID_n, H_n and L_n) and sends it to the requester.

The issuer generates another hash value SHV_k (Super Hash Value) in round *k*. SHV_k is generated

in a manner that SHV_{k-1} and RHV_k are concatenated and hashed. As a result, *SHV* links *RHV*, and all time stamps are also linked with each other. The issuer concatenates all *SHV* that have been generated during a week and hashes it. The hash value is called a "Weekly Hash Value." As corroboration that the appropriate operations of the issuer have taken place, the Weekly Hash Value is published in the New York Times every Sunday.

The verification procedure consists of the following two steps. The first is that a verifier compares a hash value of data to be verified with H_n included in a time stamp. The second is that a verifier asks the issuer to confirm the consistency between SHV'_k generated from a time stamp and SHV_k stored in the issuer's database. In the second step, the issuer first generates RHV'_k from H_n and L_n included in the received time stamp and generates SHV'_k from RHV'_k and SHV_{k-1}. Next, the issuer compares SHV'_k with SHV_k stored in the database and informs the verifier of the result. Although Weekly Hash Values are periodically published, they are not used for verification because data sufficient to regenerate the Weekly Hash Value are not ordinarily available.

These verification operations correspond to operations **a** and **c**, respectively. Therefore, the verification procedure is set as **ac**, and Digital Notary is classified into type **ac**. Type **ac** belongs to class 3, and its sufficient condition is that an attacker neither colludes with nor impersonates an issuer. In addition, Digital Notary is classified as a group of TN-U-L. If the data sufficient to regenerate the Weekly Hash Value were available for a verifier, the verification procedure would be set as **acde**, and the scheme would be classified into class 5.

The result suggests the following two points when it comes to evaluating the security of schemes similar to Digital Notary. First, it is necessary to check whether or not the issuer is trustworthy enough for one to believe that he is not carrying out any fraudulent operations relating to his service. Secondly, it is necessary to check whether or not it is infeasible for an attacker to impersonate the issuer.

4. PKITS

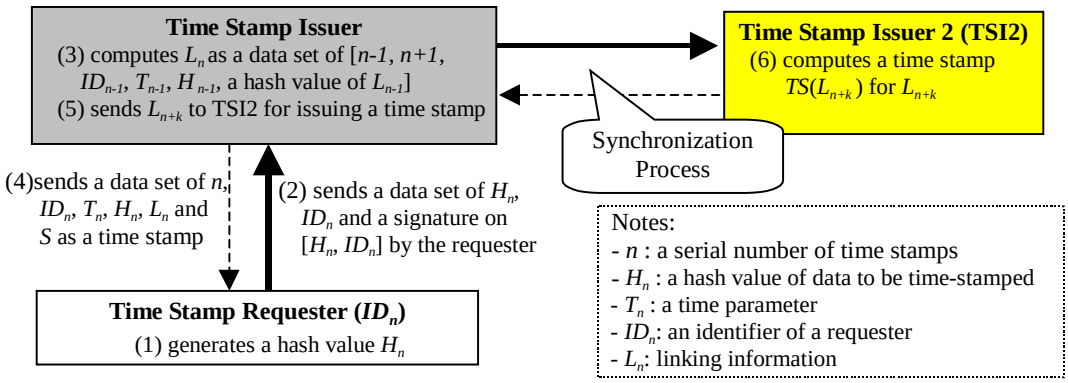
PKITS (Public Key Infrastructure with Time Stamping Authority) is one of the projects of ETS (European Trusted Services), which is a research initiative on information security sponsored by the European Commission (Fabrica Nacional de Moneda y Timbre [1998]). In PKITS, theoretical and practical studies have been carried out, and some time stamping schemes were proposed in 1998.

In the proposed schemes, the linked time stamp scheme with a new mechanism called a "synchronization process" has in particular attracted much attention. In the synchronization process, under the assumption that multiple issuers provide the same service, each of them periodically sends its linking information to one of the other issuers randomly selected in order to obtain a time stamp of the linking information. It is considered that this communication between the issuers makes it more

difficult for each issuer to fraudulently manipulate the linking information. Thus, the synchronization process allows the linking scheme to be more trustworthy from the viewpoint of security.

In the issuing procedure of the scheme with the synchronization process (Figure 7), a requester sends request data consisting of a hash value H_n of data to be time-stamped, an identifier ID_n of a requester and a digital signature on $[H_n, ID_n]$. The subscript "n" denotes the serial number of the time stamps. After receiving the request data and verifying the requester's digital signature, the issuer generates a time stamp TS_n consisting of six data items: n, ID_n, T_n (a time parameter), H_n, L_n (linking information) and S (a digital signature on $[n, ID_n, T_n, H_n, L_n]$). The linking information L_n is a data set of n-1, n+1, ID_{n-1}, T_{n-1}, H_{n-1} and a hash value of L_{n-1}. L_n links all time stamps. The issuer sends TS_n to the requester.

Figure 7 Issuing Procedure of PKITS



In the synchronization process, the issuer requests a time stamp of certain linking information (denotes L_{n+k}) to one of the other issuers (denotes TSI2) and obtains a time stamp $TS(L_{n+k})$ corresponding to L_{n+k}.

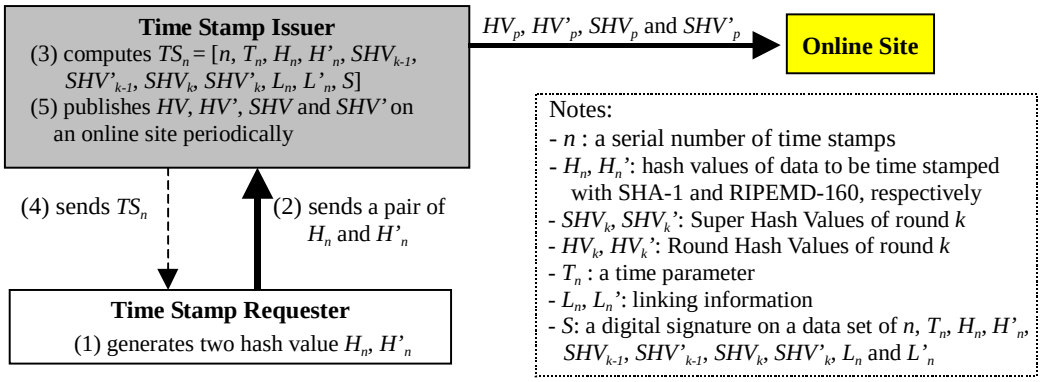
In the verification procedure, a verifier first compares a hash value of data to be verified with H_n and carries out the verification of a digital signature S. These operations correspond to operations **a** and **b**, respectively. Next, the verifier obtains a series of time stamps $[TS_{n+1}, TS_{n+2}, \ldots, TS_{n+k-1}]$, regenerates a series of linking information $[L_{n+1}, L_{n+2}, \ldots, L_{n+k}]$ and confirms a positive correspondence between the linking information regenerated and the information included in a series of the time stamps. The data set of $[TS_{n+1}, TS_{n+2}, \ldots, TS_{n+k-1}]$ corresponds to E_{TSI}, and the operation corresponds to operation **d**. Finally, the verifier obtains $TS(L_{n+k})$ from TSI2 and compares between L_{n+k} included in $TS(L_{n+k})$ and the regenerated one. This operation corresponds to operation **e** because $TS(L_{n+k})$ and TSI2 correspond to E_{AMP} and an amplifier, respectively. Thus, the scheme of PKITS and the verification procedure correspond to TN-A-L and **abde**, respectively.

The scheme of PKITS belongs to type **abde** and class 6. The corresponding sufficient condition is the union of the following three conditions. The first is that the technique to generate $Info_{INT}$ does not become weak and an attacker does not collude with an issuer. The second is that an attacker neither colludes with nor impersonates an issuer. The third is that an attacker neither colludes with nor impersonates an amplifier. Therefore, when evaluating security of schemes similar to the scheme of PKITS, it is necessary to check whether or not at least one of the three sufficient conditions is satisfied.

5. TIMESEC

TIMESEC is a project for studying time stamping and was funded by the Federal Office for Scientific, Technical and Cultural Affairs in Belgium from 1996 to 1998 (Preneel et al. [1998]). Belgian cryptographic researchers have mainly promoted the TIMESEC project and proposed one linked time stamp scheme. It has the following two features. One is that, just like Digital Notary, an issuer aggregates hash values sent from requesters in each round by using a simple binary tree structure. The other is that two hash functions are employed in order to keep the scheme useful even if it is discovered that one of them has a critical security flaw.

Figure 8 Issuing Procedure of TIMESEC



In the issuing procedure (Figure 8), a requester hashes data X to be time-stamped into two hash values H_n and H'_n with hash functions SHA-1 and RIPEMD-160, respectively. In the following, parameters generated with RIPEMD-160 are described as dashed ones. A subscript "n" denotes a serial number of time stamps. A requester sends an issuer H_n and H'_n. It is assumed that the issuer receives the hash values in round k. The issuer generates hash values HV_k and HV'_k (Round Hash values of round k) by using all of hash values received during the round. As with RHV of Digital Notary, HV_k and HV'_k are generated by using a simple binary tree structure. The data used to generate HV_k and HV'_k except for H_n and H'_n are assigned to L_n and L'_n, respectively. The issuer generates a

time stamp TS_n corresponding to X and sends it to the requester. TS_n consists of n, T_n, H_n, H'_n, SHV_{k-1}, SHV'_{k-1}, SHV_k, SHV'_k, L_n, L'_n and S. T_n is the time parameter, and SHV_k is a Super Hash Value in round k. SHV_k is a hash value of data concatenating SHV_{k-1} and HV_k. Finally, S is a digital signature of data [n, T_n, H_n, H'_n, SHV_{k-1}, SHV'_{k-1}, SHV_k, SHV'_k, L_n, L'_n] provided by the issuer.

The issuer periodically publishes the signed values of a certain round (for example, HV_p, HV'_p, SHV_p and SHV'_p of round p) on the online site. As a result, these values are widely witnessed.

In the verification procedure, a verifier first compares hash values of the data to be verified with H_n and H'_n and carries out the verification of S. These operations correspond to operations **a** and **b**, respectively. Next, the verifier regenerates SHV_k and SHV'_k by using data included in a time stamp and compares them with SHV_k and SHV'_k included in a time stamp, respectively. Then the verifier obtains T_n, SHV_k and SHV'_k from the issuer and compares them with the corresponding data included the time stamp. This operation corresponds to operation **d**. Finally, the verifier obtains from the issuer sequential data series of HV and HV' corresponding to rounds from l to p. It is assumed that rounds l and p are before and after round k, respectively, and that SHV_l, SHV'_l, SHV_p and SHV'_p have already been published. The verifier regenerates sequential data series of [(SHV_{l+1}, SHV'_{l+1}), (SHV_{l+2}, SHV'_{l+2}), …, (SHV_k, SHV'_k), …, (SHV_p, SHV'_p)]. Then the verifier compares (SHV_k, SHV'_k) with data included in the time stamp and compares (SHV_p, SHV'_p) with published ones. These operations correspond to operation **e**, and published SHV_p and SHV'_p correspond to E_{AMP}. Thus, just as with the PKITS scheme, the TIMESEC scheme and its verification procedure correspond to TN-A-L and **abde**, respectively.

The TIMESEC scheme belongs to type **abde** and class 6. The sufficient condition is the union of the following three conditions. The first is that the technique to generate $Info_{INT}$ does not become weak and an attacker does not collude with the issuer. The second is that an attacker neither colludes with nor impersonates the issuer. The third is that an attacker neither colludes with nor impersonates an amplifier. Therefore, when evaluating the security of schemes similar to the TIMESEC scheme, it is necessary to check whether or not at least one of the three sufficient conditions is satisfied.

6. The Linked Time Stamp Scheme Proposed by Benaloh and de Mare

Benaloh and de Mare [1994] have proposed a linked time stamp scheme consisting of a requester and an issuer. The proposed scheme employs a technique called a “one-way accumulator.” In briefly, a one-way accumulator is defined as a sort of one-way hash function $f : X \times Y \rightarrow X$ possessing the following feature: $f(f(x, y_1), y_2) = f(f(x, y_2), y_1)$ for all $x \in X$ and for all $y_1, y_2 \in Y$. In the scheme, the RSA encryption function is employed as a one-way accumulator.

At the beginning of the scheme, an issuer prepares a secret integer x and a public integer n. n is a product of two secret primes p and q (Figure 9). Then, the issuer computes $x_0 = x^2 \bmod n$. It is assumed that there are the m time stamp requesters in a certain round and that a requester j (j = 1, …,

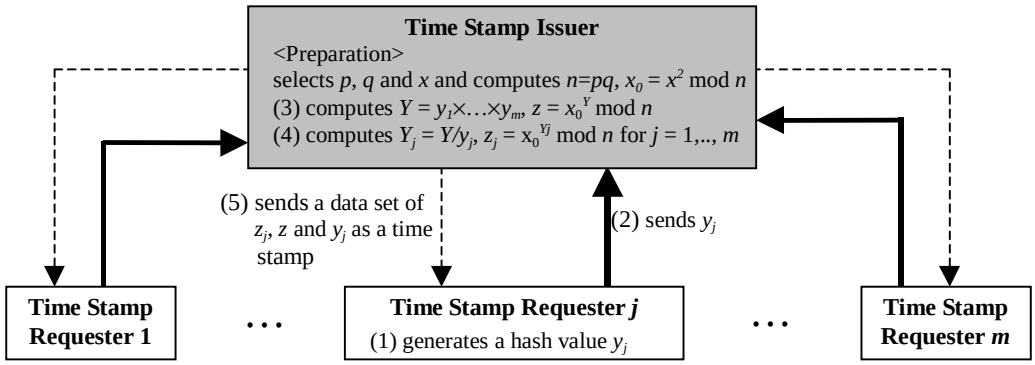
m) takes a hash value y_j to be time-stamped.

In the issuing procedure, requester j sends a hash value y_j, and an issuer receives data [y_1, …., y_m]. Next, the issuer computes $Y = y_1 \times y_2 \times \ldots \times y_m$, $z = x_0^{\,Y} \bmod n$, $Y_j = Y/y_j$ and $z_j = x_0^{\,Yj} \bmod n$ (j = 1, …, m). Finally, the issuer sends requester j a time stamp consisting of a partial accumulated hash value z_j, z, y_j and so on. A pair of z_j and z is considered to correspond to E_{TSI}. A time stamp is considered to include a time parameter because the value of z can identify the round in which the time stamp is issued. In addition, because hash values of the other data to be time-stamped in the same round are used to generate z, the scheme is classified into the linked time stamp scheme.

In the verification procedure, a verifier first compares a hash value of the data to be verified with y_j. Next, the verifier confirms that an equation of $z = z_j^{yj} \bmod n$ holds. Therefore, the scheme and the verification procedure are classified into TE-A-L and **ad**, respectively. Type **ad** belongs to class 3, and its sufficient condition is that an attacker neither colludes with nor impersonates an issuer.

The above result suggests the following two points which arise when evaluating the security of a scheme similar to the scheme proposed by Benaloh and de Mare. First, it is necessary to check whether or not an issuer is trustworthy enough to believe that he does not carry out any fraudulent operations. Secondly, it is necessary to check whether or not it is infeasible for an attacker to impersonate an issuer.

Figure 9 Issuing Procedure of the Scheme Proposed by Benaloh and de Mare



7. The Linked Time Stamp Scheme Proposed by Buldas et al.

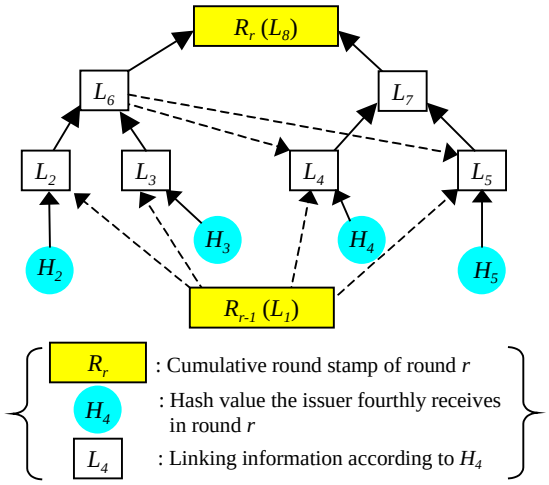
The linked time stamp scheme proposed by Buldas et al. [2000] has the following four main characteristics. The first is that it employs a tree structure called a “threaded authentication tree” when an issuer generates a time stamp (Figure 10). In the threaded authentication tree, the issuing procedure becomes more sophisticated than in a simple binary tree. The second is that the scheme proposed by Buldas et al. is designed to provide the evidence that a certain time stamp is generated before another

specific time stamp (relative temporal authentication). Therefore, the time stamp does not include an absolute time parameter. In order to achieve this function, a "time certificate" is prepared. The time certificate contains data enough to show in which round and in which order a certain time stamp is generated. The third is that the publication authority (PA) publishes a "cumulative round stamp" at the end of each round as evidence for relative temporal authentication. The cumulative round stamp is computed from hash values to be time-stamped in the corresponding round and the previous cumulative round stamp. The fourth is that a verifier can carry out the verification procedure without communicating with the issuer.

In the issuing procedure (Figure 11), a requester sends a hash value H_n of data to be time-stamped to the issuer in round r. The subscript "n" denotes the order of an operation in a certain round. An issuer computes L_n, linking information corresponding to H_n. In the case of H_4 in Figure 11, L_4 is generated from H_4, L_6 and R_{r-1} (a cumulative round stamp of round r-1), and these data link L_4 with the previous round. After adding L_n to the database, the issuer sends the requester a data set of [n, L_n, $S_{TSI}(n, L_n)$]. $S_{TSI}(n, L_n)$ is a digital signature on [n, L_n] by the issuer.

Figure 10 Example of Threaded Authentication Tree

< Example: Threaded Authentication Tree>

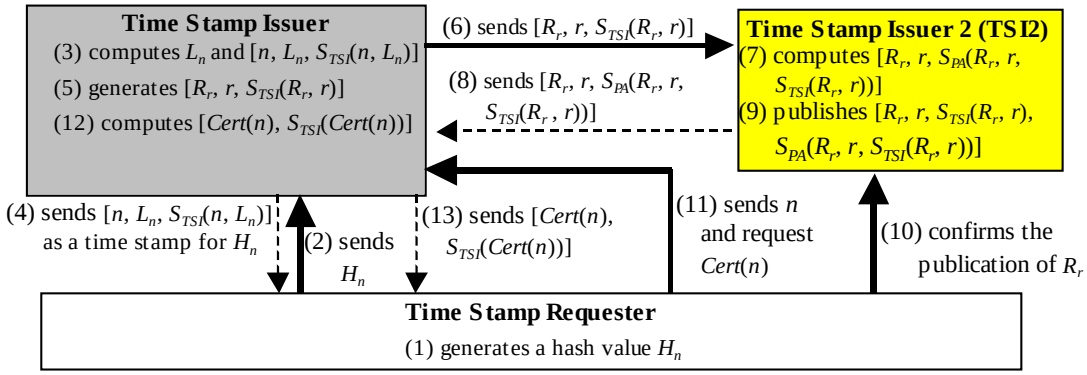


< Procedures of Generating R_r >

- **Assumption**: In round r, there are four requesters. They send four hash values to be time-stamped, [H_2, H_3, H_4, H_5], to an issuer.
- Step 1: concatenate H_2 and R_{r-1} and compute their hash value L_2.
- Step 2: concatenate H_3 and R_{r-1} and compute their hash value L_3.
- Step 3: concatenate L_2 and L_3 and compute their hash value L_6.
- Step 4: concatenate H_4, L_6 and R_{r-1} and compute their hash value L_4.
- Step 5: concatenate H_5, L_6 and R_{r-1} and compute their hash value L_5.
- Step 6: concatenate L_4 and L_5 and compute their hash value L_7.
- Step 7: concatenate L_6 and L_7 and compute their hash value R_r.
- Step 8: send R_r to the Publication Authority.

At the end of round r, the issuer computes a data set of [R_r, r, $S_{TSI}(R_r, r)$] and sends it to the PA. Then, the PA generates and returns [R_r, r, $S_{PA}(R_r, r, S_{TSI}(R_r, r))$] to the issuer. $S_{PA}(R_r, r, S_{TSI}(R_r, r))$ is a digital signature on (R_r, r, $S_{TSI}(R_r, r)$) by the PA. The PA generates [R_r, r, $S_{TSI}(R_r, r)$, $S_{PA}(R_r, r, S_{TSI}(R_r, r))$] and publishes it in the newspaper.

Figure 11 Issuing and Publication Procedures of the Scheme Proposed by Buldas et al.



After confirming the publication, the requester obtains $[R_r, r, S_{TSI}(R_r, r), S_{PA}(R_r, r, S_{TSI}(R_r, r))]$ and checks if it is of the correct form. Then, the requester sends n and requests a time certificate $Cert(n)$ for H_n. The issuer returns $[Cert(n), S_{TSI}(Cert(n))]$. $Cert(n)$ includes enough data to confirm the order of the time stamp of serial number n. In the case of H_4 in Figure 11, $Cert(4)$ consists of the following data: 4, H_4, L_5, L_6 and R_{r-1}. Thus, the requester obtains $[n, L_n, S_{TSI}(n, L_n)]$ and $[Cert(n), S_{TSI}(Cert(n))]$ as the time stamp.

The verification procedure is designed to confirm which time stamp has been generated first for any pair of time stamps. The verification procedure consists of four parts. The first step is to compare the hash value of data to be verified with H_n in each time certificate. The second is to verify the digital signature on each time stamp. The third is to regenerate L_n from each time certificate and to compare it with the one included in each time stamp. The fourth is to regenerate R_r from each time certificate and to compare it with the one published in a newspaper. As a result, if the verifier completes these operations successfully, the verifier checks the order of two time stamps by the value of n in each time certificate. If the two time stamps to be verified belong to different rounds, the verifier checks the order by using R_r regenerated in the verification process for each time certificate.

Thus, the scheme proposed by Buldas et al. is classified into TE-A-L. The verification procedure of the scheme corresponds to **abde**, and the scheme belongs to type **abde**. The PA and the time certificate are considered to correspond to an amplifier and E_{TSI}.

The scheme belongs to class 6, and the corresponding sufficient condition is the union of the following three conditions. The first is that the technique used to generate $Info_{INT}$ does not become weak and an attacker does not collude with an issuer. The second is that an attacker neither colludes with nor impersonates an issuer. The third is that an attacker neither colludes with nor impersonates an amplifier. Therefore, when evaluating the security of schemes similar to the scheme proposed by Buldas et al., it is necessary to check whether or not at least one of the three sufficient conditions is

satisfied.

B. Summary of Security Analysis

The results of applying Une and Matsumoto's results to the seven schemes are summarized in Table 7. The electronic notarization system and the time signature distributed system belong to class 2, and the scheme proposed by Benaloh and de Mare and Digital Notary/SecureSeal belong to class 3. PKITS, TIMESEC and the scheme proposed by Buldas et al. belong to class 6. As a result, the three schemes belonging to class 6 are considered to be the most desirable with respect to security against alteration of a time stamp.

Thus, Une and Matsumoto's results showed which aspects of a certain scheme should be paid special attention when carrying out the security evaluation. Although identification of the verification procedure corresponding to a scheme to be evaluated is needed, it is not necessary to scrutinize the details of its specification. Therefore, even if users of a certain scheme are not experts in cryptographic techniques, they can understand how to evaluate the security of the scheme.

Table 7 Classification and Security Evaluation of Existing Eight Schemes

<table>
<tr><th rowspan="2">Schemes</th><th colspan="2">Classification</th><th colspan="2">Security evaluation</th></tr>
<tr><th>Groups</th><th>Types</th><th>Classes</th><th>Evaluation items to be scrutinized</th></tr>
<tr><td>The electronic notarization system</td><td rowspan="2">TN-U-I</td><td rowspan="2">ab</td><td rowspan="2">2</td><td rowspan="2">- security of the digital signature
- possibility of the attacker's collusion with the issuer</td></tr>
<tr><td>The time signature distributed system</td></tr>
<tr><td>The scheme proposed by Benaloh and de Mare</td><td>TE-A-L</td><td>ad</td><td rowspan="2">3</td><td rowspan="2">- possibility of the attacker's collusion with the issuer
- possibility of the attacker's impersonation to the issuer.</td></tr>
<tr><td>Digital Notary/ SecureSeal</td><td>TN-U-L</td><td>ac</td></tr>
<tr><td>PKITS</td><td rowspan="2">TN-A-L</td><td rowspan="3">abde</td><td rowspan="3">6</td><td rowspan="3">- security of the digital signature
- possibility of the attacker's collusion with the issuer and the amplifier
- possibility of the attacker's impersonation to the issuer and the amplifier</td></tr>
<tr><td>TIMESEC</td></tr>
<tr><td>The scheme proposed by Buldas et al.</td><td>TE-A-L</td></tr>
</table>

In addition, under the assumption that same sorts of entities are identical in all schemes, Une and Matsumoto's results enable one to compare security between different schemes. For example, assuming that a time stamp issuer of PKITS is identical to that of the electronic notarization system, Une and Matsumoto's results imply that PKITS is more secure than the electronic notarization system with respect to security against the alteration of a time stamp. When comparing security between different schemes, it is necessary to confirm whether or not the assumption holds.

V. Concluding Remarks

This paper first pointed out the problem of the conventional method of classifying time stamping schemes. Then, it introduced the outline of Une and Matsumoto [2000], [2001a], [2001b] and [2002] as the recent studies of the security evaluation of time stamping schemes. Furthermore, it explained the procedures and outcomes of applying their results to the existing seven schemes.

Their results clarify which aspects of a certain time stamping scheme should be paid attention to in evaluating its security. In general, users of services adopting cryptographic techniques such as time stamping services do not always have sufficient expertise to evaluate their security. Such users can employ their results to select a time stamping service having an appropriate security level.

However, their results are not sufficient to comprehensively evaluate the security of time stamping schemes. For example, their papers took into consideration only security against the alteration of a time stamp. Moreover their papers did not discuss the details of collusion and impersonation of entities involved in a time stamping scheme precisely. It is necessary to develop a method to evaluate the comprehensive security of time stamping schemes in the future.

As possible directions of further research, the following two items can be listed: The first is to extend their results in such a way as to cover other types of attacks upon time stamping schemes, for example, a denial-of-service attack; The second is to develop a method of evaluating the trustworthiness of each entity involved in a time stamping scheme. In many existing schemes, these entities are assumed to be a trusted third party. In order to discuss whether or not this assumption actually holds in each scheme, it is necessary to analyze the attributes of each entity, to identify evaluation items and to discuss how to confirm that the evaluation items are satisfied. These researches may help to discuss and develop a better method for the security evaluation of time stamping schemes.

From now on, it is necessary to continue to pay attention to the results of studies on the security evaluation of time stamping schemes.

References

Adams, Carlisle, Pat Cain, Denis Pinkas and Robert Zuccherato, “Internet X.509 Public Key Infrastructure Time Stamp Protocol (TSP),” August 2001. (http://www.ietf.org/rtc/rtc3161.txt)

Ansper, Arne, Ahto Buldas, Märt Saarepera and Jan Willemson, “Improving the availability of time-stamping services,” *Proceedings of ACISP2001*, LNCS 2119, Springer-Verlag, 2001, pp.360-375.

Benaloh, Josh, and Michael de Mare, “One-way accumulators: A Decentralized Alternative to Digital Signature,” *Proceedings of EUROCRYPT93*, LNCS 765, Springer-Verlag, 1994, pp. 274-285.

Buldas, Ahto, Helger Lipmaa and Berry Schoenmakers, “Optimally efficient accountable time-stamping,” *Proceedings of PKC2000*, LNCS 1751, Springer-Verlag, 2000, pp. 293-305.

Cybernetica, “Cuculus: How does it work?” (http://www.cyber.ee/research/cuc-work.html, access date: January 17, 2001)

Fabrica Nacional de Moneda y Timbre, *PKITS: Deliverable D4a Description and Results of the Unstructured Data Time-Stamping Protocol Implementation*, Revision Number 16, July 30, 1998. (http://www.fnmt.es/pkits/)

Haber, Stuart, Burt Kaliski and Wakefield Scott Stornetta, “How Do Digital Time-Stamps Support Digital Signatures?” *CryptoByte*, 1 (3), 1995, pp.14-15. (http://www.rsa.com/rsalabs/)

Haber, Stuart and Wakefield Scott Stornetta, “How to Time-Stamp a Digital Document,” *Journal of Cryptology*, 3 (2), 1991, pp.99-111.

International Organization for Standardization and International Electrotechnical Commission, *ISO/IEC 13888-1: Information technology - Security techniques -Non-repudiation -Part 1: General*, 1997.

International Organization for Standardization and International Electrotechnical Commission, *ISO/IEC Working Draft 18014-1: Information technology - Security techniques -Time stamping services -Part 1: Framework*, May 30, 2000a. (http://csrc.nist.gov/cc/t4/sc27/post-london-files/27n2595.pdf, access date: August 30, 2001)

International Organization for Standardization and International Electrotechnical Commission, *ISO/IEC Working Draft 18014-2: Information technology - Security techniques -Time stamping services -Part 2: Mechanisms producing independent tokens*, May 31, 2000b. (http://csrc.nist.gov/cc/t4/sc27/post-london-files/27n2596.pdf, access date: August 30, 2001)

International Organization for Standardization and International Electrotechnical Commission, *ISO/IEC Working Draft 18014-3: Information technology - Security techniques -Time stamping services -Part 3: Mechanisms producing linked tokens*, May 31, 2000c. (http://csrc.nist.gov/cc/t4/sc27/post-london-files/27n2597.pdf, access date: August 30, 2001)

Massias, Henri and Jean Jacques Quisquater, “Time and Cryptography,” *TIMESEC Technical Report*, 1997.

NTT Data, *SecureSeal (technical information)*, (in Japanese). (http://210.144.76.11/technical/tech01.html, access date: January 17, 2001)

Preneel, Bart, Bart Van Rompay, Jean Jacques Quisquater, Henri Massias and J. S. Avila, “Design of a timestamping system,” *TIMESEC Technical Report WP3*, 1998. (http://www.dice.ucl.ac.be/crypto/TIMESEC/TR3.ps.gz)

Privador, *Privador TrueSign™ Technology Overview, Draft*, May 25, 2000. (http://gns.privador.com/ts_tech.pdf)

Surety.com, *Secure Time/Data Stamping in a Public Key Infrastructure*, 2001. (http://www.surety.com/home/pki.pdf, access date: Jan. 17, 2001)

Takura, Akira, Satoshi Ono and Shozo Naito, “Secure and Trusted Time Stamping Authority,” *Proceedings of IWS ’99*, 1999, pp.123-128.

The Study Group on the Legal System of Electronic Commerce, *Report on the Legal System of Electronic Commerce*, March 1998. (http://www.moj.go.jp/ENGLISH/CIAB/ciab-17.html)

Une, Masashi, and Tsutomu Matsumoto, “Management of Information Used to Verify Time Stamps in Linking Schemes,” *Proceedings of Computer Security Symposium*, IPSJ Symposium Series Vol. 2000, No. 12, Information Processing Society of Japan, 2000, pp. 25-30 (in Japanese).

Une, Masashi, and Tsutomu Matsumoto, “Relations between Security and Verification Procedures of Time Stamps,” *Proceedings of the 2001 Symposium on Cryptography and Information Security*, The Institute of Electronics, Information and Communication Engineers, 2001a, pp.629-634 (in Japanese).

Une, Masashi, and Tsutomu Matsumoto, “Ten Security Classes of Time Stamping Schemes,” *IEICE Technical Report*, ISEC2001-38, The Institute of Electronics, Information and Communication Engineers, 2001b, pp.141-148 (in Japanese).

Une, Masashi, and Tsutomu Matsumoto, “A Framework to Evaluate Security and Cost of Time Stamping Schemes,” *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, E-85A (1), The Institute of Electronics, Information and Communication Engineers, January 2002 (to appear).