IMES DISCUSSION PAPER SERIES

デジタルタイムスタンプ技術の現状と課題

うね まさし　まつうら かんた　たくら あきら
宇根正志・松浦幹太・田倉昭

Discussion Paper No. 99-J-36

IMES

INSTITUTE FOR MONETARY AND ECONOMIC STUDIES
BANK OF JAPAN

日本銀行金融研究所

〒103-8660 日本橋郵便局私書箱 30 号

IMES Discussion Paper Series 99-J-36
1999 年 10 月

デジタルタイムスタンプ技術の現状と課題

うね まさし　　まつうらかんた　　たくら あきら
宇根正志[*1]・**松浦幹太**[*2]・**田倉昭**[*3]

要　旨

デジタルタイムスタンプ技術は、デジタルデータがある特定時刻に存在していたことを証明するとともに、その時刻以降データが変更されていないことを証明する技術である。近年、インターネット上での電子商取引の活発化や、紙ベースの文書を電子媒体に置き換えて管理する電子文書管理の利用拡大に伴い、「誰が、いつ、どんなデータを生成し、交信したか」を第三者が証明する「電子公証」の仕組みが必要とされている。電子公証は、送信受信者の特定、到達確認、時刻情報の付与、改ざんの検知、電子文書保管等の機能を具備するものといわれており、デジタルタイムスタンプ技術は、このうち、時刻情報付与や改ざん検知の機能を実現する技術である。

従来からデジタルタイムスタンプ技術に関する理論研究が行われてきたが、最近では、デジタルタイムスタンプシステムの実装を視野に入れた研究が世界各国で開始されている。日本では、法務省が、電子確定日付サービスを含む電子公証制度の実現に向けて検討を行っているほか、海外では、ベルギーやスペイン等において研究プロジェクトが進められている。また、米国や英国では、既に民間企業がデジタルタイムスタンプサービスを開始している。

一方、デジタルタイムスタンプ技術の標準化も進められている。インターネット上での公開鍵インフラに関する標準化を行う IETF PKIX は、タイムスタンププロトコルの標準規格の策定を行っているほか、情報セキュリティ技術の国際標準化を担当する ISO/IEC JTC1/SC27 においても、デジタルタイムスタンプサービスに関する標準化作業が進められている。

デジタルタイムスタンプ技術は、今後、金融分野をはじめとする幅広い分野において利用されるようになるものとみられる。本稿では、デジタルタイムスタンプ技術の特徴や機能について整理した上で、最近の研究・実装動向、標準化動向に加えて、デジタルタイムスタンプ技術に関連する主要な特許について説明する。

キーワード：　デジタルタイムスタンプ、電子公証、デジタル署名、ハッシュ関数、確定日付制度、国際標準

JEL classification：　L86、L96、Z00

*1 日本銀行金融研究所研究第 2 課（E-mail: masashi.une@boj.or.jp）
*2 東京大学生産技術研究所第三部（E-mail:kanta@iis.u-tokyo.ac.jp）
*3 日本電信電話株式会社情報流通プラットフォーム研究所（E-mail: takura@slab.ntt.co.jp）

本稿を作成するにあたっては、横浜国立大学大学院工学研究科の松本勉助教授から有益なコメントを頂戴した。記して感謝したい。もっとも、本稿においてあり得べき誤りはすべて筆者に帰することは言うまでもない。

目　次

頁

Ⅰ．はじめに

近年、インターネットの急速な拡大等に伴って、オープンなネットワークを利用した電子商取引が活発化してきている。また、イントラネットやグループウェア等を活用し、紙ベースの文書に代わって、電子媒体による文書での回覧・決裁・保管といった電子文書管理に取り組む企業等が増えてきている。

しかし、デジタルデータは紙ベースの文書に比べて内容の改ざんが容易であり、現在利用されている電子商取引や電子文書管理のシステムにおいては、紙ベースの文書が有する「後々の係争や情報公開等に備えて、文書の内容やその取扱履歴を長期間保管することができる」という文書保管機能が実現されているとは言い難い。このため、今後、電子商取引や電子文書管理の利用拡大には、「誰が、いつ、どんなデータを生成し、交信したか」を第三者が証明する「電子公証」の仕組みが必要とされている。電子公証の主な機能としては、①送受信者特定機能、②到達確認機能、③時刻付与機能、④改ざん検知機能、⑤電子保存機能、⑥アクセス記録機能、⑦プロセス記録機能の 7 つが挙げられる（電子商取引実証推進協議会[1998]）[1]。

デジタルタイムスタンプ技術は、デジタルデータがある特定時刻に存在していたことを証明するとともに、それ以降、当該データが変更されていないことを証明する技術である。デジタルタイムスタンプ技術は、上記の電子公証の機能のうち、③時刻付与と④改ざん検知を実現する技術として、その重要性に対する認識が高まっている。

現在、世界各国では、デジタルタイムスタンプ技術について、実用化を視野に入れた実装研究が開始されている。日本では、法務省が、2001 年のサービス開始に向けて電子公証制度の検討を進めているほか、いくつかの電子公証システムの実装実験が開始されている。一方、海外では、ベルギー、スペイン等の国々において、国家プロジェクトの一部としてデジタルタイムスタンプ技術の実装研究が進められているほか、米国等では、Surety 社をはじめとするいくつかの民間企業によって、既にデジタルタイムスタンプの商用サービスが開始されている。

---

[1] これらの機能について、電子商取引実証推進協議会[1998]では次のように定義されている。
①送受信者特定機能：コンピューターシステム、ネットワークシステム等の利用者を特定する機能
②到達確認機能：送信者から受信者へ情報を送信した事実を証明する機能
③時刻付与機能：情報に対して日時データを付与し、情報に付与された日時データが正しいかを検証する機能
④改ざん検知機能：文書（電子データ）が改ざんされたことを利用者が検知する機能
⑤電子保存機能：情報の内容を媒体に記録・保管する機能
⑥アクセス記録機能：利用者がシステムを利用した事実を記録する機能
⑦プロセス記録機能：電子記録が組織や集団の中で変更されたり承認されたりする場合、その更新・承認過程を記録・保持する機能

また、デジタルタイムスタンプ技術の標準化も進められており、幅広い分野においてデジタルタイムスタンプ技術を利用するための土台が整備されつつある。インターネット上での公開鍵インフラに関する標準化を行う IETF PKIX は、タイムスタンプのプロトコルの標準化を進めている。また、汎業界用の情報セキュリティ技術の国際標準化を担当する ISO/IEC JTC1/SC27 は、デジタルタイムスタンプサービスに関する国際標準の策定作業を行っている。

本稿では、デジタルタイムスタンプ技術の機能・要件等について説明するとともに、最新の研究・実装動向や標準化動向について説明する。まず、第Ⅱ章において、デジタルタイムスタンプ技術の機能や要件について説明した上で、タイムスタンプシステムを分類し、各々のシステムの長所・短所や安全性について説明する。第Ⅲ章では、デジタルタイムスタンプ技術の研究プロジェクトや商用サービスを紹介する。第Ⅳ章では、IETF PKIX や ISO/IEC JTC1/SC27 におけるデジタルタイムスタンプ技術の標準化動向について説明し、第Ⅴ章では、タイムスタンプ技術に関する主要な特許の内容について説明する。

なお、第Ⅱ、Ⅲ、Ⅳ、Ⅴ章では技術的に詳細な内容に立ち入っているが、デジタルタイムスタンプ技術の概要に関心のある読者は、各章・各節に記載した表を参照することによって、全体像を理解できるようになっている。表の一覧は以下の通り。

章	節	表番号	タイトル
Ⅱ	2	表 1	デジタルタイムスタンプシステムの分類
		表 2	3 種類のプロトコルの主な特徴点
		表 3	3 種類の Linking Protocol の概要と長所・短所
	3	表 4	デジタルタイムスタンプシステムに対する攻撃の目的・手段
		表 5	Simple Protocol に対する攻撃成功の可能性
		表 6	Linking Protocol に対する攻撃成功の可能性
		表 7	Distributed Protocol に対する攻撃成功の可能性
		表 8	各デジタルタイムスタンプシステムの安全性
Ⅲ	1	表 9	日本における主要な研究・開発プロジェクト
		表 10	海外における主要な研究プロジェクト
	2	表 11	主要なデジタルタイムスタンプシステムの商用サービス
Ⅳ		表 12	デジタルタイムスタンプ技術に関連する主な標準化の動向
Ⅴ		表 13	デジタルタイムスタンプシステム関連特許

Ⅱ. デジタルタイムスタンプ技術の機能とモデル

1. デジタルタイムスタンプ技術の概要

(1)デジタルタイムスタンプ技術の機能と要件

デジタルタイムスタンプ技術は、デジタルデータに時刻情報を付与し、その時刻情報やデータの真正性を証明するタイムスタンプを生成する技術である。デジタルタイムスタンプ技術の主な機能として、以下の 2 つが挙げられる。

・データの存在証明

タイムスタンプによって示される時刻にデータが存在していたことを第三者に証明する。通常タイムスタンプによって示される時刻は、タイムスタンプを生成する組織がタイムスタンプの要求情報を受け付けた時刻とされる。

・データの完全性証明

タイムスタンプが付与された時刻以降、そのデータが改ざんされていないことを第三者に証明する。

これらの機能を達成するためには、少なくとも以下の 4 つの要件がデジタルタイムスタンプ技術において必要になると考えられる[2]。

①デジタルタイムスタンプの生成に利用される時刻情報が、各実装環境において支障が出ない程度に正確であること。
②タイムスタンプの改ざんが困難であること。
③タイムスタンプを付したデータの改ざんが困難であること。
④タイムスタンプを更新する仕組みが完備されており、数十年単位の長期間においてタイムスタンプの有効性を維持することが可能であること[3]。

---

[2] これら以外にも、デジタルタイムスタンプのシステムが満足することが望ましい要件として、「タイムスタンプの生成には秘密情報を利用しない」といった要件を指摘する研究成果もみられる（Haber, Kaliski and Stornetta[1995]他）。この条件がデジタルタイムスタンプシステムの要件となる理由について、「例えば、デジタル署名のように署名鍵に代表される秘密情報に依存するシステムの場合、万一秘密情報が漏洩したり、効率的な解読法が発見された場合には、システム全体の信頼性が大きく低下する。デジタルタイムスタンプシステムは、電子公証において重要な役割を担うものであり、そうしたリスクを考慮すると、秘密情報を必要としないハッシュ関数のような技術のみで構築することが必要である」と説明されている。署名鍵の漏洩は鍵管理の問題であり、①秘密鍵が容易に露見してしまうような弱い公開鍵を利用しないようにする、②十分な鍵長の公開鍵を利用する、あるいは比較的短い公開鍵であっても鍵の変更を頻繁に行う、といった方法によって対応すべき問題であると考えられる。また、解読技術の進歩等による安全性低下の可能性は、ハッシュ関数においても同様に存在する。このため、秘密情報を必要としないことがデジタルタイムスタンプの要件とは必ずしも言えないと考えられる。

[3] これは、デジタルタイムスタンプ技術には、「紙」の持つ機能の一つである「長期間にわたって文書を保管する」という機能を電子的に代替することが想定されているためである。これら 4 つの要件が満足されると、後々係争が発生する可能性がある取引や契約等に関連する文書や、情報開示に備えて長期間の保管が必要となる文書の電子化が可能となる。この結果、民間企業における文書の電子化や、政府における公文書をはじめとする様々な種類の文書の

このうち④の要件は、デジタル署名を利用したシステムにおいて、署名生成・検証に利用される秘密鍵・公開鍵ペアが一定期間後に無効となった場合[4]や、タイムスタンプを生成する際に利用されるハッシュ関数等の安全性が低下し、十分な安全性を確保することができなくなった場合等において、タイムスタンプをどのようにして更新するかという問題である。各アプリケーションでどの程度の有効期間が望ましいか、あるいは、更新に必要なコストは如何程かといった検討事項から総合的に判断して、1 つのタイムスタンプの有効期間が決定される。その有効期間を実現させるため、どの程度の安全性を有するデジタル署名やハッシュ関数を利用すればよいか、また、どのような公開鍵証明書のシステムを利用すればよいかについて検討が行われる。ただし、後述する Surety 社のサービスのように、1 つのタイムスタンプの有効期間を予め設定しない方式も存在する。タイムスタンプの更新方法については、1 つのタイムスタンプの有効期間が切れる前に、そのタイムスタンプおよびタイムスタンプの対象となっているデータに対して、より安全性の高い新しいタイムスタンプを再び生成する、という方法が提案されている（Bayer, Haber and Stornetta [1993], Massias and Quisquater [1997]他）。

(2)デジタルタイムスタンプシステムの構成主体

デジタルタイムスタンプシステムの構成主体として、以下の 4 つが挙げられる（次頁の図 1 参照）。

①タイムスタンプシステムの利用者

タイムスタンプシステムの利用者には、(A)特定のデータに対するタイムスタンプの生成を要求する要求者と、(B)生成されたタイムスタンプの真正性を検証する検証者の 2 種類が存在する。要求者が自分のタイムスタンプの真正性を検証する場合、要求者と検証者が一致する。一方、検証者が他人のタイムスタンプを検証する場合には要求者と検証者は異なる。

②時刻情報生成機関（TA：Time Authority）

タイムスタンプの生成に利用される時刻情報を生成する主体。

電子化が促進されることが期待される。

[4] デジタル署名に利用される公開鍵証明書の有効期間は、利用されるアプリケーションや利用者によってまちまちであるが、通常デジタル署名の安全性を考慮して数ヶ月から数年程度に設定される。デジタル署名を生成するために必要となる計算量は、通常鍵長とアルゴリズムに依存しており、限られた計算能力の下で高速で署名生成を行うために鍵長を短めに設定するケースが多い。このため、デジタル署名の安全性が確保される時間も短くなることから、公開鍵証明書の有効期間も短めに設定されることが多い。

③タイムスタンプ生成機関（TSA：Time Stamp Authority）

利用者から要求されたデータに対するタイムスタンプを生成する主体。利用される時刻情報は TA から入手する。

④Temporal Data 生成機関（TDA：Temporal Data Authority）

「ある時点までは確定せず、一旦確定した後は誰でも容易に知り得る情報」（例えば、気象情報、株価情報等）である Temporal Data（TD）を生成し、必要に応じて TSA に送付する主体。

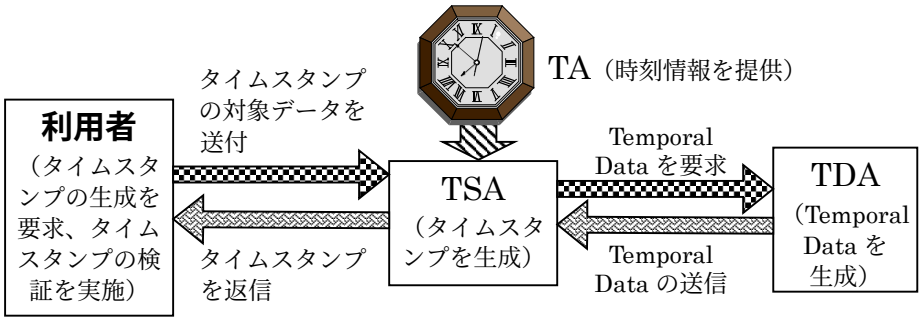


図 1　タイムスタンプシステムの構成主体

ただし、①～③の構成主体はタイムスタンプシステムにおいて必須となるが、④の TDA は必ずしも必要というわけではない。TDA が提供する Temporal Data は、タイムスタンプに含まれる時刻情報を補完する役割を有しており、時刻情報だけで十分と考えられる場合には、Temporal Data は必要ではなくなる。

2. デジタルタイムスタンプ技術のモデル

(1)本稿における検討内容と前提条件

デジタルタイムスタンプシステムの安全性（タイムスタンプの改ざんや偽造がどの程度困難か）や安定性（必要に応じてタイムスタンプサービスを利用することができるか）は、時刻情報生成技術、ネットワーク技術、暗号技術等のタイムシステムに利用される関連技術に依存する。このため、タイムスタンプシステムを検討する場合には、①これらの関連技術に関する検討に加え、②関連技術が十分に整備されているという前提の下で、安全性の高いタイムスタンプを生成するためのプロトコルに関する検討や、③関連技術が十分に整備されていないという前提の下で、安全なタイムスタンプを生成するためのプロトコルに関する検討、が必要となる。

本稿では、上記 3 種類の検討のうち、「②関連技術が十分に整備されているという前提の下で、安全性の高いタイムスタンプを生成するためのプロトコル」について検討を行うこととし、上記①および③に関する検討は今後の課題とする。「関連技術が十分に整備されている」という前提は、具体的には以下の 3 点

が満足されていることを指す。

(A)タイムスタンプに利用される時刻情報の精度はアプリケーションごとに異なるが、TSA は各アプリケーションに応じて必要な精度で時刻情報を TA から入手することができる。すなわち、TA が生成する時刻情報が正確であるとともに、TSA と TA との間では時刻情報のやり取りが安全に行われる（ネットワークのトラフィック状況等による時刻情報の遅延や欠損は発生しない）[5]。

(B)TSA・利用者間および TSA・TDA 間の通信経路は完備されており、ネットワークのトラフィック状況等によって、交信される情報の遅延や欠損は発生しない[6]。

(C)タイムスタンプの生成に利用されるデジタル署名方式やハッシュ関数は、アプリケーションに応じて十分な安全性[7]を有する方式を選択することができるほか、署名生成鍵および検証鍵の鍵管理も安全に行われる。

これらの技術に対する要件はアプリケーションによって異なるが、タイムスタンプシステムのネットワークとしてインターネットを利用する場合、トラフィックの状況等によってデータの遅延や欠損が発生する可能性があり、時限性や安全性の要請が比較的高いアプリケーションに適用する際には問題となる可能性がある。現在、インターネットにおいてデータをより迅速かつ安全に送受信する技術の研究・開発が進められている。また、暗号技術についても、現時点では数十年の長期間安全性を維持できる方式は存在せず、鍵長やハッシュ長を拡大する等より安全性の高いデジタル署名方式やハッシュ関数に関する研究が進められている[8]。

---

[5] 正確な時刻情報を提供する技術について様々な研究開発が進められている。例えば、郵政省通信総合研究所と NTT は、日本標準時を用いたインターネット時刻サービスに向けた共同研究を 1998 年 10 月に開始している（プレスリリースは http://www.nttsl.mfeed. ne.jp/）。

[6] タイムスタンプの要求が増加した場合でも、安定したサービスが可能であること（scalability）が問題となることがある。本稿では、各システムが十分な Scalability を有していることを前提に議論を進めることとする。

[7] 本稿では、ハッシュ関数が安全であるとは、そのハッシュ関数が汎用一方向性ハッシュ関数（universal one-way hash function）であることを指す。汎用一方向性ハッシュ関数とは、「ある与えられた入力値 X に対するハッシュ関数の出力値（ハッシュ値）を H(X)としたとき、ハッシュ関数の出力値が H(X)となるような別の入力値 Y（すなわち、H(X)=H(Y)かつ X≠Y であるような Y）を見つけることが統計的に困難である」という性質をもつハッシュ関数のことである。同じハッシュ関数の出力値を有する異なる入力値のペアは、collision と呼ばれる。ただし、「n bit のサイズのデータをランダムに $2^{n/2}$ 個集めたときに、その中に同じデータが 2 個以上存在する確率が約 0.5 になる」という性質（バースデーパラドックス）を利用することで、例えばハッシュ値のサイズが 160 bit の場合、2^{80} 個のハッシュ値を集めることによって collision を見つけることが可能となる（この攻撃はバースデー攻撃と呼ばれる）。このため、安全なハッシュ関数は、アルゴリズムに欠陥が存在しないだけではなく、そのハッシュ値が十分なサイズであることが必要となる。現時点では、安全なハッシュ関数を実現するためには、ハッシュ値のサイズを 160 bit 以上に設定することが必要といわれている。

[8] デジタル署名方式の研究動向については、宇根・岡本[1999]を参照。

(2)デジタルタイムスタンプシステムの分類方法

デジタルタイムスタンプシステムを分類する際には、①TSA の性質に関する前提条件と、②タイムスタンプの生成方法に着目する方法が一般的である（Massias and Quisquater[1997]、Lipmaa[1999]等）[9]。具体的には、TSA を信頼することを前提とするか否かによって 2 種類に分類され、タイムスタンプの生成方法によって 3 種類に分類される（表 1 参照）。

表 1　デジタルタイムスタンプシステムの分類

分類項目	内容
TSA に対する信頼性	①TSA を信頼できる場合 ②TSA を信頼できない場合
タイムスタンプの生成方法	①Simple Protocol ②Linking Protocol ③Distributed Protocol

①TSA に対する信頼

タイムスタンプシステムの分類方法の 1 つは、TSA が信頼できるか否かである。本稿では、TSA に対する信頼を、「TSA が規定された業務内容に沿って適正に機能し、不正な行為を一切行わないと期待することができるか否かに関する信頼（TSA の業務規律に対する信頼）」とする[10,11]。

(A)TSA を信頼できる場合

TSA を信頼できる場合、タイムスタンプの生成や管理をすべて TSA に任せることが可能となり、TDA 等他の組織の介在をなくすことができる。このため、例えば、TSA がデータもしくはそのハッシュ値[12]に時刻情報を追加し、それらのデータに対するデジタル署名をタイムスタンプとする非常に単純なシステムによってタイムスタンプを生成可能である。

---

[9] こうした分類とは異なる観点からの分類も可能である。例えば、①タイムスタンプの生成に利用される TSA の数による分類や、②タイムスタンプの安全性がデジタル署名の署名生成鍵等の秘密情報に依拠するか、もしくは他の公開された情報（他人のタイムスタンプ）に依拠するかといった分類方法（Haber, Kaliski and Stornetta [1995]）も考えられる。本稿では、これまでの研究成果の中で最も一般的である上記の分類方法を採用した。

[10] 認証業務等、情報セキュリティ関連業務の管理・運営状況に対して他の組織から高い信頼を寄せられている組織は、TTP（Trusted Third Party）と呼ばれる（ISO/IEC 13888-1 の TTP の定義）。本稿では、「TSA が信頼できる」とは、その TSA が TTP であることを意味する。

[11] TSA の信頼度については、正確には、TSA を信頼できるか否かという二者択一の問題ではなく、TSA の業務運営・管理状況等の属性によって何段階ものレベルに分類した上で検討することが必要となる。ただし、本稿では、単純化のために「ある具体的なアプリケーションへのタイムスタンプシステムの実装を前提とし、業務運営状況等から判断した TSA の信頼度が、そのアプリケーションにおける安全性の要件を満足すると判断された場合、その TSA は信頼できる」として、二者択一の TSA の信頼度に関する判断を行うこととする。

[12] タイムスタンプの対象となるデータの機密性を確保すると同時に、TSA の処理負担を少しでも軽減するために、利用者は TSA に対してデータそのものではなくデータのハッシュ値をタイムスタンプの要求情報として送信する仕組みが一般的である。

(B)TSA **を信頼できない場合**

TSA を信頼できない場合、TSA が生成した情報の一部を公表する、あるいはタイムスタンプの生成に利用される TSA の数を増やす等の追加的な対策を講じることによってタイムスタンプの安全性を高め、結果的にシステム全体への信頼を高めることができる。ただし、TSA を信頼できる場合に比べてシステムが複雑化する。

②デジタルタイムスタンプの生成方法

もう 1 つの分類方法は、デジタルタイムスタンプの生成方法による分類であり、Simple Protocol、Linking Protocol、Distributed Protocol の 3 つに分類される。最初に各プロトコルの特徴点について整理すると、以下の表 2 の通り。

表 2　3 種類のプロトコルの主な特徴点

プロトコル	概要	長所	短所
Simple Protocol	・1 つの TSA が、利用者がタイムスタンプを希望するデータのハッシュ値に時刻情報等を添付してデジタル署名（タイムスタンプ）を生成。TSA はデジタル署名を保管。 ・タイムスタンプに時刻情報を含めることで、TSA がタイムスタンプ要求を受信した時刻を特定。	・システム構成が比較的単純	**・TSA を信頼できることが前提条件であり、TSA が攻撃者と結託すると、容易にタイムスタンプの偽造や改ざんが可能。**
Linking Protocol	・TSA が過去のハッシュ値を関連付ける Link 情報を生成し、Link 情報からタイムスタンプを生成。TSA は、タイムスタンプの検証に必要な Link 情報やハッシュ値を保管。Link 情報の一部を新聞等に掲載したり、複数の TSA を利用してタイムスタンプを生成したりする。 ・必ずしも時刻は特定されず、時間帯や前後関係を特定する方式がある。	**・各 TSA を信頼できなくても、システム全体として高い安全性を確保することが可能。**	・Simple Protocol に比べてシステム構成が複雑化し、タイムスタンプの生成・検証に必要な処理量が増加。さらに、Link 情報を保管するための追加的なデータベースが必要。
Distributed Protocol	・複数の TSA が、デジタル署名を利用して、共同で 1 つのタイムスタンプを生成（複数の TSA が各々デジタル署名を生成し、それらを結合してタイムスタンプとする方式や、秘密分散技術によるデジタル署名を利用した方式等が提案されている）。 ・タイムスタンプに時刻情報を含めることで、TSA がタイムスタンプ要求を受け取った時刻を特定。	**・各 TSA を信頼できなくても、システム全体として高い安全性を確保することが可能。**	・複数の TSA の存在が前提条件。 ・Simple Protocol に比べてシステム構成が複雑化し、タイムスタンプの生成・検証の処理量が増加。

(A)Simple Protocol

Simple Protocol は、1 つの TSA が、その利用者のハッシュ値や時刻情報に対するデジタル署名を生成し、そのデジタル署名をタイムスタンプとする

プロトコルである。タイムスタンプにより、TSA がタイムスタンプ要求情報を受信した時刻がピンポイントで特定される。本プロトコルでは、後述する Linking Protocol とは異なり、タイムスタンプを生成する際に他の利用者のハッシュ値は利用されない。基本的な Simple Protocol におけるタイムスタンプの生成手順は以下の通り（図 2 参照）。

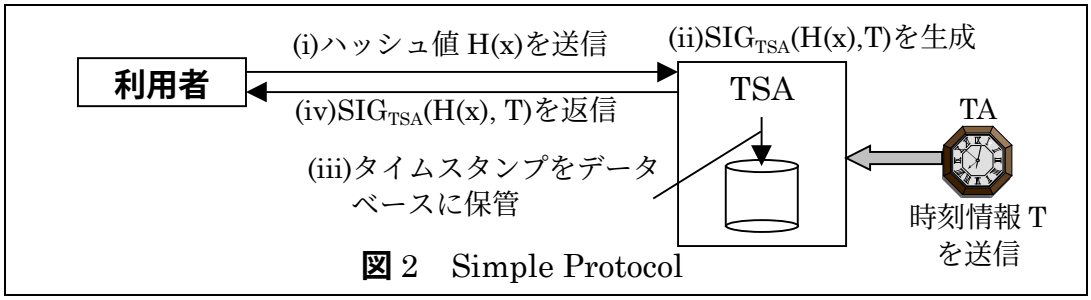


図 2　Simple Protocol

(i)利用者は、タイムスタンプの対象データ x のハッシュ値 H(x)を TSA に送付。
(ii)TSA は、H(x)を受信した時点の時刻情報 T を TA から入手し、[H(x), T]に対する署名 SIG$_{TSA}$(H(x), T)をタイムスタンプとして生成。
(iii)TSA は、タイムスタンプ SIG$_{TSA}$(H(x), T)を自社のデータベースに保管。
(iv)TSA は、タイムスタンプ SIG$_{TSA}$(H(x), T)を利用者に送付。

一方、タイムスタンプの検証は、TSA の公開鍵を入手してデジタル署名であるタイムスタンプの真正性の確認によって行われる。

このように、Simple Protocol は、システム構成が非常に単純であり、実装が比較的容易である。ただし、タイムスタンプの安全性は TSA が生成するデジタル署名の安全性に依存するとともに、TSA の運用・管理状況に全面的に依存していることから､TSA を信頼できることが前提条件となる｡TSA を信頼できない場合、TSA による時刻情報の不正操作等不正行為が容易に成立し、発生した不正行為を外部から追跡・発見することが非常に困難となることから、本プロトコルを利用することはできない。

(B)Linking Protocol

Linking Protocol は、TSA が複数の利用者のハッシュ値を相互に関連づける Link 情報を生成し、各タイムスタンプが、それまでに生成されたすべてのタイムスタンプに依存するように生成されるプロトコルである。これまで提案されている Linking Protocol は、以下のいずれかの方法によってシステム全体の安全性を確保しており、万一 TSA がタイムスタンプや Link 情報を改ざんしたとしても、利用されるハッシュ関数が安全である限り、改ざんの事実を発見することができる仕組みとなっている（詳細は 3.(2)②を参照）。

(i)Link 情報を定期的に新聞に掲載する等の方法で広く公表する。
(ii)複数の TSA を用意し、各 TSA が生成する Link 情報やタイムスタンプに対して他の TSA がタイムスタンプを生成する。

この結果、個々の TSA を信頼できない場合でも、タイムスタンプシステム全体として高い安全性を確保することが可能となる。ただし、Simple Protocol に比べて Link 情報の生成に伴ってシステム構成が複雑化するほか、Link 情報を保管するための追加的なデータベースが必要となる。

本プロトコルにおけるタイムスタンプには様々な形態が存在するが、主な Linking Protocol として、Linear Linking Protocol、Linking Protocol using Tree Structure、複数の TSA による Linking Protocol の 3 つが挙げられる。まず、これらのプロトコルの概要を整理すると、以下の表 3 の通り。

表 3　3 種類の Linking Protocol の概要と長所・短所

プロトコル	概要	長所	短所
Linear Linking Protocol	・TSA は、タイムスタンプ要求情報が寄せられる都度 Link 情報を生成。Link 情報の一部が定期的に公表される。	・TSA に寄せられたタイムスタンプ要求情報の順序を特定可能。 ・1 つの TSA で実装が可能。	・タイムスタンプの検証に必要な計算量は Linking Protocol using Tree Structure よりも多い。
Linking Protocol using Tree Structure	・TSA は、一定時間内に受け付けたタイムスタンプ要求情報をプールし、タイムスタンプの対象となるハッシュ値を「葉」とみなして Link 情報（「幹」に対応）を生成。	・タイムスタンプを検証するために必要な計算量は Linear Linking Protocol に比べて少ない。 ・1 つの TSA で実装が可能。	・一定時間内に受け付けたタイムスタンプの受付時刻の前後関係を特定することは不可能。
複数の TSA **による** Linking Protocol	・複数の TSA が利用可能な環境において、各 TSA の Link 情報やタイムスタンプに対し、他の複数の TSA がタイムスタンプを生成・保管。	・Link 情報を公表しなくても、システムの安全性を確保することが可能。	・複数の TSA が利用できる環境が前提条件。

Linking Protocol の中でも、タイムスタンプを生成する際にデジタル署名を利用せず、ハッシュ関数のみを利用する方式では、そのタイムスタンプは、TSA がハッシュ値を受け付けた時刻をピンポイントで証明するのではなく、TSA が受け付けたハッシュ値の受付時刻の前後関係を証明する[13]。

[13]タイムスタンプに時刻情報が含まれない Linking Protocol の場合、複数の TSA が独自にタイムスタンプを生成する状況では、異なる TSA において生成された複数のタイムスタンプの時間的な前後関係が問題となる可能性がある（なお、タイムスタンプに時刻情報が含まれる Linking Protocol の場合でも、攻撃者と TSA が結託すれば時刻情報の不正な操作が可能となるため、同様の問題が生じる可能性がある）。
例えば、Linear Linking Protocol（詳細は次節）を採用する 2 つの TSA（TSA1 と TSA2

(**ア**)Linear Linking Protocol

Linear Linking Protocol（Chaining Protocol とも呼ばれる）では、ある時刻の Link 情報は、その時刻に送付されたハッシュ値やその直前の Link 情報から生成される。このため、タイムスタンプの要求に応じて Link 情報が生成され、各 Link 情報が時系列的に関連付けられることから、タイムスタンプの要求情報が TSA に到着した順番を特定可能である。Link 情報は定期的に新聞等に公表される[14]。

Linear Linking Protocol には様々なタイプが存在するが、ここでは TSA がデジタル署名を利用するプロトコルを例として説明する。TSA は、以下の手順で Link 情報やタイムスタンプを生成する（次頁の図 3 参照）。

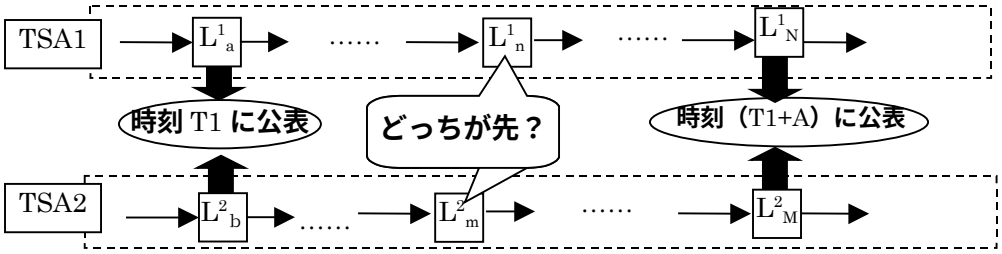
＜前提＞

・利用者が第 n 番目のタイムスタンプの要求情報として、TSA にハッシュ値 H_n を送信する場合を想定。

(i)利用者は、TSA にタイムスタンプの対象データのハッシュ値 H_n を送付。

(ii)TSA は、H_n を受信した時刻の情報 T_n を入手し、Link 情報 $L_n = H(H_n, n, L_{n-1})$ を計算するとともに、H_n に対するタイムスタンプ $SIG_{TSA}(H_n, T_n, n,$

とする）が存在し、いずれも Link 情報を A 分毎に発表するものとする。時刻 T1 において、TSA1 と TSA2 はそれぞれ Link 情報 L^1_a と L^2_b を公表し、時刻（T1+A）では、TSA1 と TSA2 はそれぞれ Link 情報 L^1_N と L^2_M を公表した場合を考える。このとき、TSA1 が L^1_a と L^1_N の間に生成した Link 情報 L^1_n と、TSA2 が L^2_b と L^2_M の間に生成した Link 情報 L^2_m に着目すると、どちらの Link 情報が先に生成されたかを証明することは困難である。この結果、Link 情報 L^1_n を含むタイムスタンプと、Link 情報 L^2_m を含むタイムスタンプとの間の前後関係を証明することが困難となる。



こうした問題を解決する方法として、①各 TSA が生成したすべてのタイムスタンプに対してタイムスタンプを生成する TSA を別途設け、そのタイムスタンプを用いて絶対的な順序付けをするプロトコルや、②各 TSA が生成したタイムスタンプの番号を他のすべての TSA に知らせることによって、タイムスタンプに全体での通し番号が付くようにするプロトコルが提案されている（陣内・櫻井[1999]）。

[14] 通常、すべての Link 情報が公表されるのではなく、一部の Link 情報が公表される。どの程度の Link 情報を公表するかは、アプリケーションに必要とされる安全性と Link 情報の公表に必要なコストとの兼ね合いによって決定される。後述する Linking Protocol using Tree Structure においては SRH が公表の対象となるが、Linear Linking Protocol と同様、一部の SRH が定期的に公表されるケースが多い。

L_n）を計算。

(iii)TSA は、タイムスタンプをデータベースに保管した後、利用者に送付。

(vi)TSA は、一定数（例えば、N 個）の Link 情報が生成される度に、Link 情報 L_N を時刻情報 T_N とともに新聞等に掲載。

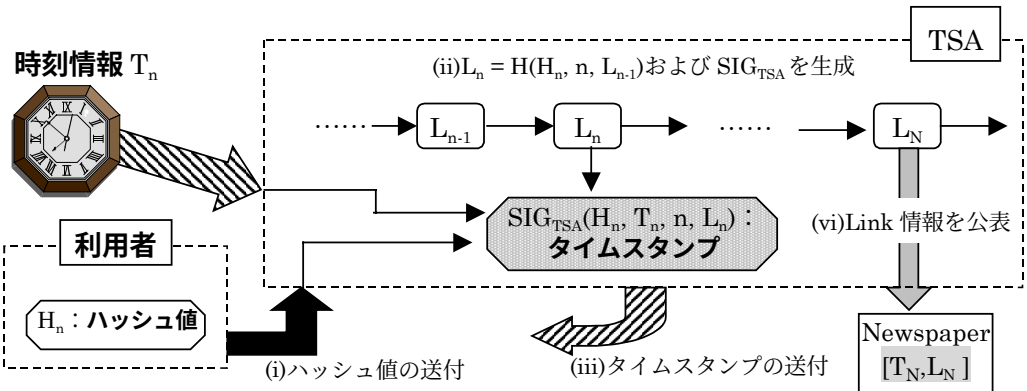


図 3　Link **情報を公開する** Linear Linking Protocol

一方、タイムスタンプの検証手順は以下の通り（図 4 参照）。

(i)検証者は、検証したいタイムスタンプ $SIG_{TSA}(H_n, T_n, n, L_n)$ が生成される前に既に新聞に公表されていた Link 情報の中で、最も新しい Link 情報 L_k を新聞から入手。さらに、ハッシュ値（H_{k+1}, …, H_{n-1}, H_{n+1}, …, H_N）を TSA から入手。ただし、L_N は、検証の対象となっているタイムスタンプが生成された以降最初に新聞に掲載された Link 情報とする。

(ii)検証者は、$L_{k+1} = H(H_{k+1}, k+1, L_k)$、$L_{k+2} = H(H_{k+2}, k+2, L_{k+1})$、…の要領で順々に Link 情報を計算し、検証する対象のデータのハッシュ値 H_n を用いて $L_n = H(H_n, n, L_{n-1})$ を計算。L_n を用いて同様に L_N を生成。

(iii)検証者は、生成した L_N が公表された L_N と一致することを確認。

(iv)検証者は、(H_n, T_n, n, L_n) を利用して、タイムスタンプ $SIG_{TSA}(H_n, t, n, L_n)$ のデジタル署名の真正性を検証。

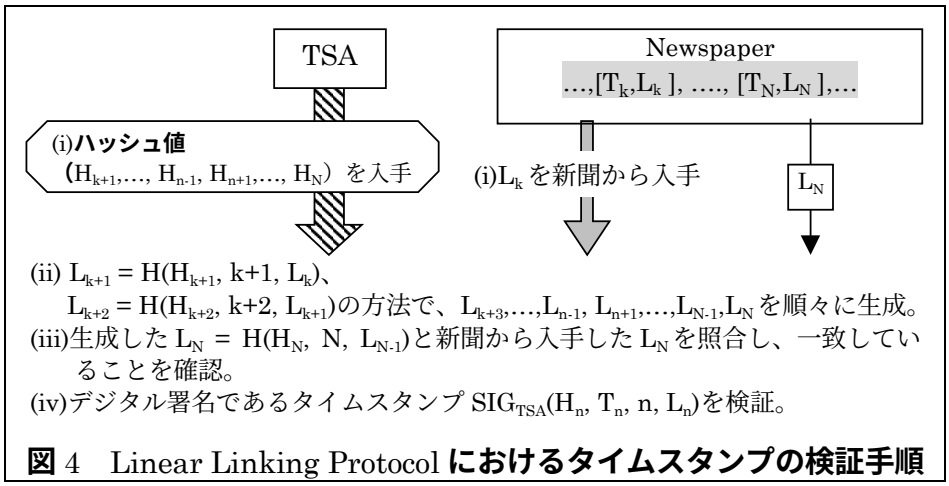


図 4　Linear Linking Protocol **におけるタイムスタンプの検証手順**

このように、検証者は、「TSA から入手したハッシュ値」、「タイムスタンプを確認したいデータのハッシュ値」、「公表されている Link 情報のうち、検証したいタイムスタンプ以前で最も新しいもの」を用いて、それ以降の Link 情報を次々に生成することができる。その結果、タイムスタンプに付された TSA のデジタル署名を検証することができる。また、ハッシュ関数の安全性が十分である限り、「公表されている Link 情報のうち、検証したいタイムスタンプ以降で最も古いもの」を検証することで、攻撃者と TSA の結託による改ざんを防止できる（詳細は本章 3.(2)②を参照）。

ただし、Link 情報の公表頻度を X 回に一度（図 4 の例では X=N-k）とすると、任意のタイムスタンプの検証を行うためには、公表されている Link 情報に対応する Link 情報を生成する必要があることから、最大 X 回のハッシュ関数の計算を実行する必要がある。

(イ) Linking Protocol using Tree Structure

Linking Protocol using Tree Structure では、一定時間（round）内に TSA に送付されたハッシュ値を「葉」とみなし、round 内のハッシュ値を結合・ハッシュ化するプロセス（Tree 構造が形成される）を繰り返して、Root Hash（RH）と呼ばれる Link 情報（「幹」に相当）を生成する（Massias and Quisquater [1997]）。

本プロトコルでは、TSA が利用者のハッシュ値の送付に応じて Link 情報を生成する Linear Linking Protocol とは異なり、1 つの round においてハッシュ値を受け付ける時間と受付可能なハッシュ値の個数が設定される。このため、1 つの round において受け付けたハッシュ値の個数が予め設定された受付可能なハッシュ値の個数を下回る場合、ハッシュ値の代わりとなる値（パディングデータ）を下回る数だけ TSA が生成し、パディングデータを用いて RH を生成する。パディングデータとしては、乱数や定数（例えば 0）が用いられる。RH は、直前の round で生成された Super Root Hash（SRH）と結合されてハッシュ化され、その round の SRH となる。SRH は定期的に新聞等に公表される。

Linking Protocol using Tree Structure では、Link 情報やタイムスタンプの生成方法等によって様々なタイプが存在する。例えば、(a)Tree を再構築する（RH を生成する）ための情報がタイムスタンプとして利用される方式や、(b)時刻情報に対する TSA のデジタル署名がタイムスタンプとして利用される方式もある。前者のタイムスタンプの生成手順は以下の通り（次頁の図 5 参照）。

＜前提＞

・TSA は、予め Link 情報を生成するための round を設定。ここでは、

1 round 中に最大 8 個のハッシュ値を受付可能とする。

・利用者は、TSA にタイムスタンプ要求情報として、タイムスタンプの対象となるデータのハッシュ値 H_1 を第 i round に送付。

(i)TSA は、第 i round が終了した後、第 i round で受け取ったハッシュ値（H_1,...,H_8）を用いて、

$$X_1=\text{hash}(H_1,H_2)$$
$$X_2=\text{hash}(H_3,H_4)$$
$$X_3=\text{hash}(H_5,H_6)$$
$$X_4=\text{hash}(H_7,H_8)$$

を計算（hash はハッシュ関数を表す）。

(ii)TSA は、上記(i)で計算した（X_1, X_2, X_3, X_4）を用いて次の計算を実行。

$$Y_1=\text{hash}(X_1,X_2)$$
$$Y_2=\text{hash}(X_3, X_4)$$

(iii)TSA は、RH_i=hash(Y_1, Y_2)を計算し、第 i round の Root Hash を生成。

(iv)TSA は、第(i-1) round の Super Round Hash（SRH_{i-1}）と RH_i を結合・ハッシュ化し、第 i round の SRH_i を生成。TSA はこれらのデータを保管。

(v)TSA は、ハッシュ値 H_1 に対するタイムスタンプとして（H_2, X_2, Y_2, RH_i）を利用者に送付。

(vi)TSA は、定期的に SRH とその時刻情報を新聞等に発表(例えば、SRH_k を掲載)。

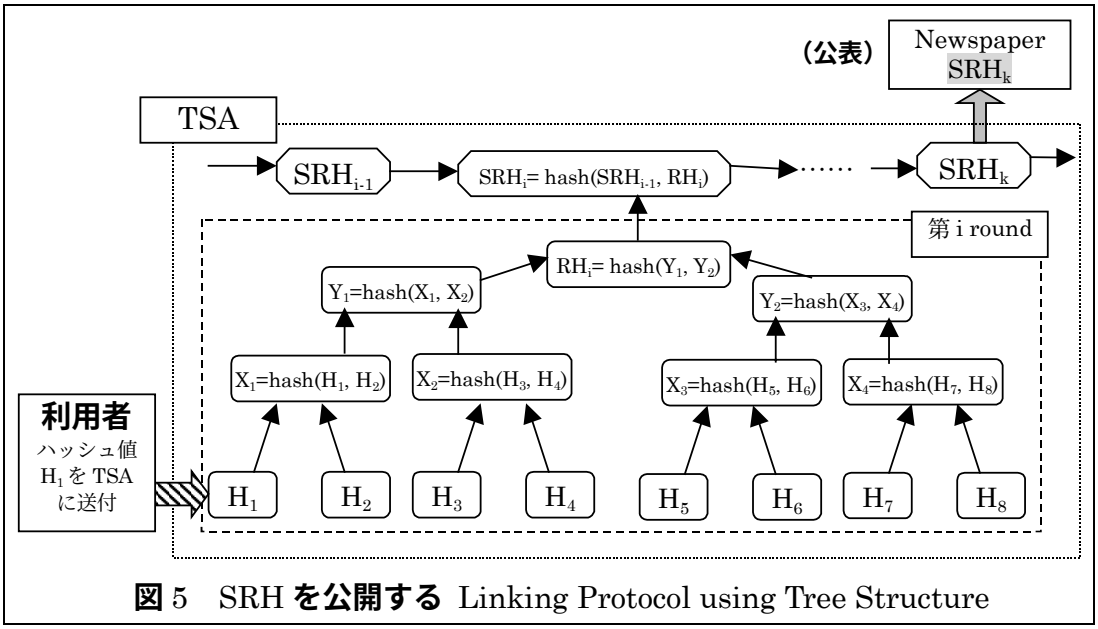


図 5　SRH を公開する Linking Protocol using Tree Structure

上記の方式では、TSA がそのタイムスタンプに含まれるハッシュ値を受け取った時間帯が「その直前の SRH 公表時刻から、直後の SRH 公表時刻までの間」であることが特定される。したがって、SRH 公表時刻をはさんで前後関係にある入力に関しては、その前後関係が証明される。

タイムスタンプの検証方法もタイムスタンプの形態に依存する。図 5 のように、タイムスタンプが Tree を再構築する情報として生成される方式では、検証対象となっているデータのハッシュ値とタイムスタンプを利用して Tree 構造を再構築し、その結果生成される SRH が既に公表されている SRH と一致するかどうかが確認される。

例えば、図 5 のハッシュ値 H_2 に対するタイムスタンプの検証方法は以下の通り。

(i)検証者は、タイムスタンプ（H_1, X_2, Y_2, RH_i）とハッシュ値 H_2 から RH_i を生成し、タイムスタンプの一部である RH_i と比較。

(ii)検証者は、TSA が保管している SRH_{i-1} と生成した RH_i から SRH_i を生成する。生成した SRH_i と、TSA が保管している $(RH_{i+1},\ldots,RH_k)$ から SRH_k を生成。

(iii)検証者は、生成した SRH_k と新聞に掲載されている SRH_k が一致することを確認。

Linking Protocol using Tree Structure においても、Linear Linking Protocol 同様、公表された SRH を利用してタイムスタンプの検証を行うことにより、ハッシュ関数が十分に安全である限り、TSA の不正行為は困難である（詳細は本章 3.(2)②を参照）。一方、タイムスタンプがデジタル署名として生成される場合、デジタル署名の真正性確認によってタイムスタンプの検証が行われる。

また、Linking Protocol using Tree Structure においてタイムスタンプの検証に必要となるハッシュ関数の計算量は、Linear Linking Protocol に比べて少ない。例えば、TSA が 8 個のハッシュ値に対して 1 つの RH および SRH を生成し、64 個のハッシュ値を受け付ける度に SRH を公表する場合を考える。これは、Linear Linking Protocol の場合、64 個のハッシュ値を受け付けて、64 個の Link 情報を生成する度に Link 情報を公表するケースに相当する。Linear Linking Protocol の場合では、任意のタイムスタンプを検証するためには最大 64 回のハッシュ関数計算が必要となる。一方、Linking Protocol using Tree Structure の場合、まず①ハッシュ値とタイムスタンプに含まれるデータから RH を生成するためには 3 回（$=\log_2 8$）のハッシュ関数計算が必要となるほか、②RH から SRH を順々に生成し、公表されている SRH に対応する SRH を生成するためには最大 8 回のハッシュ関数計算が必要となるため、合計で最大 11 回のハッ

シュ関数計算となる。このように、Linking Protocol using Tree Structure では、検証対象となっているタイムスタンプが属する round の RH を計算した後は、公表されている SRH に対応する SRH を順々に生成すればよく、公表されている Link 情報に対応する Link 情報を生成するために、その間の Link 情報をすべて生成する必要がある Linear Linking Protocol に比べてハッシュ関数の計算量を削減できる。

(ウ)複数の TSA **による** Linking Protocol

複数の TSA による Linking Protocol は、複数の TSA が存在する環境を前提としたプロトコルであり、各 TSA は独自に Linear Linking Protocol や Linking Protocol using Tree Structure を利用して Link 情報やタイムスタンプを生成する一方、自分が生成した Link 情報やタイムスタンプをランダムに選択した他の TSA に送信し、それらに対するタイムスタンプを生成してもらうというものである。各 TSA の Link 情報やタイムスタンプに対して他の TSA が適宜タイムスタンプを生成することで、Link 情報を定期的に公表しなくても、TSA による Link 情報の改ざんを困難にする効果がある。

以下では、Linear Linking Protocol をベースとしたプロトコルにおいて、各 TSA が他の TSA の Link 情報に対してタイムスタンプを生成するプロトコルについて説明する（次頁の図 6 参照）。

＜前提＞

- Linear Linking Protocol におけるタイムスタンプの生成方法は、図 3 と同一。
- 3 つの TSA（TSA^1、TSA^2、TSA^3）が存在する場合を想定。
- タイムスタンプの生成の順番を、①TSA^2 が TSA^1 の Link 情報に対するタイムスタンプを生成、②TSA^2 が TSA^3 の Link 情報に対するタイムスタンプを生成、③TSA^3 が TSA^1 の Link 情報に対するタイムスタンプを生成、と設定（本来は各 TSA がランダムに他の TSA を選ぶ）。
- 各 TSA が他の TSA に Link 情報を送信する場合、Link 情報そのものではなく、そのハッシュ値を送付。

(i)TSA^1 は、Link 情報 L^1_n をハッシュ化してハッシュ値 H^1_n を生成し、TSA^2 に対して H^1_n をタイムスタンプ要求情報として送信。

(ii)TSA^2 は、時刻 T^2_m に H^1_n を受信し、Link 情報 $L^2_m=H(H^1_n, m, L^2_{m-1})$ を生成。TSA^2 は、ハッシュ値 H^1_n に対するタイムスタンプ $S^2_m=SIG_{TSA}{}^2(H^1_n, T^2_m, m, L^2_m)$を生成し、$S^2_m$ と H^1_n を自社のデータベースに保管するとともに、S^2_m を TSA^1 に送付。

(iii)TSA^2は、Link 情報 L^2_M をハッシュ化してハッシュ値 H^2_M を生成し、TSA^3 に対して H^2_M をタイムスタンプ要求情報として送信。

(iv)TSA^3は、時刻 T^3_k に H^2_M を受信し、Link 情報 $L^3_k=H(H^2_M, k, L^3_{k-1})$ を生成。TSA^3 は、ハッシュ値 H^2_M に対するタイムスタンプ $S^3_m=SIG_{TSA}{}^3(H^2_M, T^3_k, k, L^3_k)$を生成し、$S^3_m$ と H^2_M を自社のデータベースに保管するとともに、S^3_m を TSA^2 に送付。

(v)TSA^3は、Link 情報 L^3_K をハッシュ化してハッシュ値 H^3_K を生成し、TSA^1 に対して H^3_K をタイムスタンプ要求情報として送信。

(vi) TSA^1は、時刻 T^1_N に H^3_K を受信し、Link 情報 $L^1_N=H(H^3_K, N, L^1_{N-1})$ を生成。TSA^1 は、ハッシュ値 H^3_K に対するタイムスタンプ $S^1_N=SIG_{TSA}{}^1(H^3_K, T^1_N, N, L^1_N)$を生成し、$S^1_N$ と H^3_K を自社のデータベースに保管するとともに、S^1_N を TSA^3 に送付。

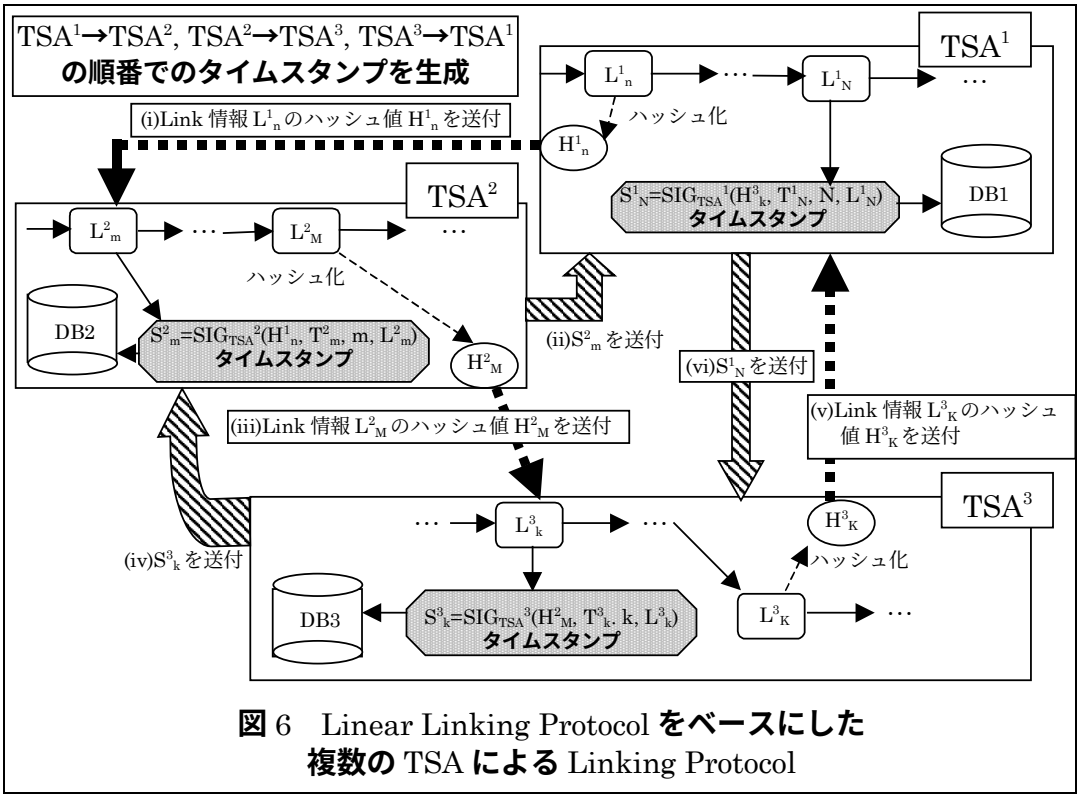


図 6　Linear Linking Protocol をベースにした複数の TSA による Linking Protocol

図 4 における Linear Linking Protocol では、新聞等に公表された Link 情報を基に検証が行われたが、本プロトコルでは、他の TSA によってタイムスタンプが生成された Link 情報を基に検証が行われる。本プロトコルにおけるタイムスタンプの検証方法は以下の通り（次頁の図 7 参照）。

＜前提＞

・検証対象は、TSA^1 が生成したタイムスタンプ $S^1_n=SIG_{TSA}{}^1(H^1_n, T^1_n, n, L^1_n)$ とする。

・TSA^2 が、時刻 T^1_n より前の時刻 T^2_m に TSA^1 から受け付けたハッシュ値 H^1_k（Link 情報 L^1_k をハッシュ化したデータ）に対してタイムスタンプ $S^2_m=SIG_{TSA}{}^2(H^1_k, T^2_m, m, L^2_m)$ を生成したとする。
・TSA^3 が、時刻 T^1_n より後の時刻 T^3_k に TSA^1 から受け付けたハッシュ値 H^1_N（Link 情報 L^1_N をハッシュ化したデータ）に対してタイムスタンプ $S^3_k=SIG_{TSA}{}^3(H^1_N, T^3_k, k, L^3_k)$ を生成したとする。

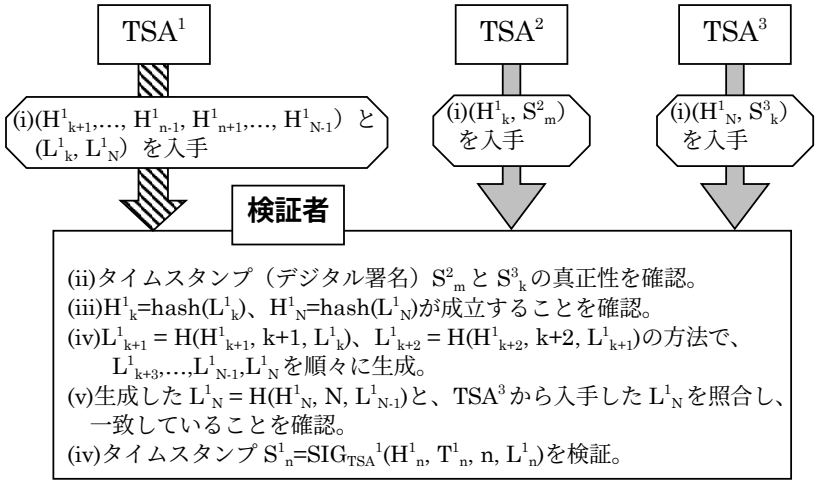


図 7　複数の TSA による Linking Protocol におけるタイムスタンプの検証手順

(i)検証者は、TSA^2 からハッシュ値 H^1_k とタイムスタンプ S^2_m を入手するとともに、TSA^3 から H^1_N と S^3_k を入手する。同時に、TSA^1 からハッシュ値（H^1_{k+1}, …, H^1_{n-1}, H^1_{n+1}, …, H^1_{N-1}）と（L^1_k, L^1_N）を入手。

(ii)検証者は、TSA^2 および TSA^3 からそれぞれ(H^1_k, T^2_m, m, L^2_m)と(H^1_N, T^3_k, k, L^3_k）を入手して、デジタル署名である S^2_m と S^3_k の真正性を確認。

(iii)検証者は、TSA^1 から入手した Link 情報（L^1_k, L^1_N）をそれぞれハッシュ化し、2 つのハッシュ値が TSA^2 および TSA^3 からそれぞれ入手した H^1_k、H^1_N と一致することを確認。

(iv)検証者は、以下の計算を実行し、L^1_N を生成。

$$L^1_{k+1} = H(H^1_{k+1}, k+1, L^1_k)$$
$$L^1_{k+2} = H(H^1_{k+2}, k+2, L^1_{k+1})$$
$$\cdots$$
$$L^1_N = H(H^1_N, N, L^1_{N-1})$$

(v)検証者は、生成した L^1_N と TSA^3 から入手した L^1_N とを照合し、一致することを確認。

(vi)利用者は、(H^1_n, T^1_n, n, L^1_n）を利用して、タイムスタンプ $S^1_n=SIG_{TSA}{}^1(H^1_n, T^1_n, n, L^1_n)$ のデジタル署名の真正性を検証。

このように、他の TSA に保管されている情報を基にタイムスタンプの検証を行う点が重要である。ある TSA が自分の既存のタイムスタンプを改ざんしたいと考えたとしても、自分が保管しているタイムスタンプだけではなく、他の複数の TSA が保管しているタイムスタンプも改ざんしなければならない。このため、TSA によるタイムスタンプの改ざんは Simple Protocol に比べて困難となる。

(C)Distributed Protocol

Distributed Protocol は、複数の TSA が生成したデジタル署名（あるいは部分署名）を利用して、1 つのタイムスタンプを生成するプロトコルである。したがって、複数の TSA が利用可能であることが前提条件となる。

典型的な Distributed Protocol としては、複数の TSA がタイムスタンプの対象となる文書のハッシュ値と時刻情報に対するデジタル署名を生成し、それらの複数のデジタル署名を 1 つに結合してタイムスタンプとする方式である（次頁の図 8 参照)。本方式におけるタイムスタンプの生成手順は以下の通り（Massias and Quisquater[1997])。

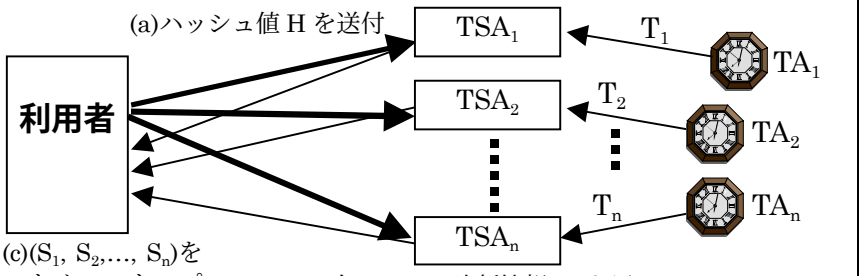


図 8　Distributed Protocol

(a)利用者は、複数存在する TSA の中から n 個の TSA をランダムに選択。選択された TSA（TSA_1,...,TSA_n）に対して、タイムスタンプの対象となる文書のハッシュ値 H をタイムスタンプ要求情報として送付。

(b)各 TSA_i は、タイムスタンプ要求情報を受け付けた時刻の情報 T_i を各自利用している TA_i から入手し[15]、[H, T_i]に対するデジタル署名 S_i を生成、利用者に返信。

(c)利用者は、n 個の TSA から送信されたデジタル署名(S_1,...,S_n)をタイム

[15] 各 TSA_i は、それぞれ別個の TA から時刻情報を入手するケースもあれば、同一の TA から入手するケース等、様々な時刻情報の入手ルートが考えられる。ただし、本稿では、TSA は TA から常に正確な時刻を遅延なく入手できることを前提としているため、こうした時刻情報の入手ルートによる差異については議論しないこととする。ここでは、説明の便宜上、各 TSA_i は各自利用している TA_i から時刻情報を入手している場合を想定して説明する。

スタンプとする[16]。

一方、タイムスタンプの検証は n 個のデジタル署名の検証により行われる。

上記以外の方式としては、秘密分散技術を利用したデジタル署名[17]を用いる方式が存在する。本方式では、利用者が複数の TSA にデータのハッシュ値を送付すると、各 TSA がハッシュ値および時刻情報に対する部分署名を生成し、利用者に返信する。利用者に返信された部分署名が一定数以上となった場合、利用者はハッシュ値に対するデジタル署名を得ることができる。

Distributed Protocol においては、TSA に信頼をおくことを前提にする場合、少数の TSA を利用することによって実現することができる。一方、TSA に信頼をおかないことが前提となっている場合、少数の TSA を利用したのでは、利用者と TSA が結託することによって時刻情報の改ざん等不正行為が発生する可能性がある。こうした不正に対しては、タイムスタンプに必要な TSA のデジタル署名の数を増やし、利用する TSA の数を増やす等の対策が有効となる。つまり、すべての TSA が信頼できるわけではないとしても、攻撃者が、任意に選択された n 個の TSA すべてと結託することは困難であるため、システム全体として高い安全性を確保することが可能となるという考え方である。こうした利点が存在する一方、Distributed Protocol を実装するためには、多数の TSA が利用可能という実装環境が前提となるほか、とりわけ秘密分散技術によるデジタル署名を用いた方式の場合には、部分秘密鍵の生成・廃棄等の追加的な鍵管理が必要となること等から、システム構成が一層複雑化するという問題もある。

---

[16] Massias and Quisquater[1995]には、複数のデジタル署名から一意にタイムスタンプの時刻を特定する方法について記載されていない。各 TSA と TA との間の時刻配送網が完備されている場合、各 TSA が入手する時刻情報 T_i は非常に近い値となると考えられることから、例えば T_i の中央値や平均値がタイムスタンプの時刻として利用されると考えられる。

[17] 秘密分散技術を用いたデジタル署名：秘密分散技術は、秘密鍵を一ヵ所に保管する際のリスク（コンピューターのダウン、パスワードの亡失等）を避けるために、元の秘密鍵からいくつかの部分秘密鍵を生成し、これらを複数人で管理する技術。秘密分散技術を用いたデジタル署名では、署名生成の際に、各部分鍵管理人が部分秘密鍵を用いて部分署名を生成し、一定数（Threshold と呼ばれる）以上の部分署名を集めることができれば、元の秘密鍵によって生成されるデジタル署名を入手することができる。谷口[1999]を参照。

3. 各デジタルタイムスタンプシステムの安全性

(1)攻撃の分類

前節において分類した各タイムスタンプシステムの安全性を検討する際には、まず想定される攻撃の目的・手段を整理し、その上で具体的な攻撃法に対する安全性を評価することが必要である。タイムスタンプシステムに対する主な攻撃の目的と手段は、以下の表 4 のように整理することができる[18]。

表 4　デジタルタイムスタンプシステムに対する攻撃の目的・手段

攻撃の目的	攻撃の手段	
	TSA との結託が不可能なケース	TSA との結託が可能なケース
既存のタイムスタンプの改ざん： 既存のタイムスタンプの内容（ハッシュ値や時刻情報）を変更する。	・利用されているハッシュ関数の collision をみつける。 ・TSA が生成したデジタル署名を偽造する。	・利用されているハッシュ関数やデジタル署名を攻撃する。 ・ハッシュ値や時刻情報を改ざんしたデータに対してデジタル署名を生成するように TSA に依頼する。 ・Linking Protocol では、特定のタイムスタンプを検証する際に、TSA が保管する情報の一部を改ざんして検証者に送付するように TSA に依頼する。
特定の利用者に対するタイムスタンプサービスの妨害： 特定の利用者がタイムスタンプをタイムリーに生成できないようにする。	・特定の利用者の要求情報のみ通信途中で傍受・奪取する。	・特定の利用者からのタイムスタンプの要求に対してのみ、タイムスタンプの生成・送付を意図的に遅らせるようにすることを TSA に依頼する。 ・複数の TSA を利用するプロトコルの場合、いくつかの TSA に対してサービス停止を依頼する。
全面的なタイムスタンプサービスの妨害・停止： 特定の TSA のサービスを妨害する。	・Denial of Service 攻撃（例えば、短時間に膨大な量のタイムスタンプの要求情報を送付し、TSA のタイムスタンプ生成サーバーをダウンさせる）を実行する。	

①攻撃の目的

タイムスタンプシステムに対する攻撃の目的は、(A)既存のタイムスタンプの改ざん、(B)特定の利用者に対するタイムスタンプサービスの利用妨害、(C)特定の TSA におけるタイムスタンプサービスの全面的な妨害、の 3 つに大別することができる。

②攻撃の手段

上記3種類の攻撃の目的を達成するために様々な攻撃の方法が考えられるが、

[18] TSA のタイムスタンプ生成のサーバー自体を物理的に破壊する等の攻撃については、タイムスタンプのプロトコルによって対応できるものではなく、入退室管理等の物理的な手段による対応が必要となるため、ここでは検討の範囲に含めないこととする。

攻撃者と TSA が結託することが可能か否かによって、攻撃の方法は変わってくる。このため、以下では、攻撃者が TSA と結託可能である場合とそうでない場合に場合分けを行った上で、考えられる攻撃の方法を整理する。

(A)既存のタイムスタンプの改ざん

利用者（攻撃者）が、以前自分が TSA から入手したタイムスタンプについて、その対象データの内容や時刻情報を後日変更したいと考えた場合には、以下のような攻撃方法が考えられる。

(a)攻撃者が TSA と結託することが不可能な場合

攻撃者は TSA の内部情報（TSA が管理している Link 情報やハッシュ値等）を利用することができない。このため、攻撃者が、タイムスタンプの生成に利用されているデジタル署名を偽造する、あるいはハッシュ関数の collision をみつけるという攻撃が考えられる。

タイムスタンプとしてデジタル署名を利用している場合、既存のタイムスタンプに含まれる時刻情報やハッシュ値を改ざんし、改ざん後のデータに対するデジタル署名を偽造してタイムスタンプとするという攻撃が考えられる。また、タイムスタンプとしてハッシュ値のみを利用し、Link 情報の一部を公表している Linear Linking Protocol の場合（前掲の図 3 参照）には、公表されている Link 情報 L_n（$=H(H_n, n, L_{n-1})$）と同じハッシュ値を有し、(H_n, n, L_{n-1})とは異なる入力データ（例えば、(H_n, n, L'_{n-1})）をみつけ、公表された Link 情報と整合性のとれた「公表されない不正な Link 情報の系列（例えば、$L'_{n-1}=H(H_{n-1}, n-1, L'_{n-2})$, $L'_{n-2}=H(H_{n-2}, n-2, L'_{n-3})$,….）」を生成するという攻撃が考えられる。

(b)攻撃者が TSA と結託することが可能な場合

ハッシュ関数やデジタル署名に対する攻撃のほかに、攻撃者が、ハッシュ値や時刻情報を改ざんしたデータに対してタイムスタンプを生成するように TSA に依頼するという攻撃が考えられる。また、TSA が管理している情報も攻撃に利用することが可能になるため、Linking Protocol の場合、TSA が生成・保管する Link 情報やハッシュ値を用いた攻撃が考えられるほか、タイムスタンプの生成プロセスにおいて特定のタイムスタンプに対して不正行為を行うという攻撃も考えられる。

(B)特定の利用者に対するタイムスタンプサービスの妨害

攻撃者が、特定の利用者がある TSA のタイムスタンプサービスを受けることができないようにしたい場合、以下のような攻撃が考えられる。

(a)攻撃者が TSA と結託することが不可能な場合

攻撃者が、攻撃対象となる利用者と TSA との間の通信経路上において、

攻撃対象の利用者から送信されたタイムスタンプの要求情報を傍受し、奪取するという攻撃が考えられる。

(b)**攻撃者が** TSA **と結託することが可能な場合**

攻撃対象の利用者から送信されたタイムスタンプの要求情報を傍受・奪取する攻撃に加え、攻撃者が、攻撃対象の利用者からのタイムスタンプの要求に対して意図的にタイムスタンプの生成を遅らせるように TSA に依頼するという攻撃も考えられる。また、タイムスタンプの生成に複数の TSA の協力が必要となるプロトコルでは、攻撃者が一部の TSA に対してタイムスタンプを正しく生成しないように依頼するという攻撃も考えられる。

(C)**タイムスタンプサービスの全面的な妨害・停止**

攻撃者が、特定の TSA におけるタイムスタンプサービスを全面的に妨害したいと考えた場合、短時間に TSA の処理能力を超える膨大な量のタイムスタンプ要求情報を送信し、TSA のタイムスタンプ生成用のサーバーをダウンさせるという攻撃が考えられる。また、TSA が時刻情報を入手している TA との通信経路を利用不可能にするといった攻撃も考えられる。このように、特定のサービス自体を利用できなくなるようにする攻撃は Denial of Service（DoS）攻撃と呼ばれている。

(2)**各攻撃の効果と対策**

①Simple Protocol

(A)**既存のタイムスタンプの改ざん**

まず、攻撃者が TSA と結託することが不可能なケースでは、タイムスタンプの改ざんが成功するか否かは、タイムスタンプの生成に利用されるハッシュ関数やデジタル署名の安全性に依存する。

一方、攻撃者が TSA と結託することが可能なケースでは、タイムスタンプを生成するために必要な署名生成鍵を有している TSA が、タイムスタンプの対象となる時刻情報やハッシュ値を改ざんし、それらのデータに対して改めてタイムスタンプを生成することができる。このため、攻撃者が TSA と結託することができれば、攻撃が成功する可能性が高い。

(B)**特定の利用者に対するタイムスタンプサービスの妨害**

攻撃者が TSA と結託することが不可能な場合、攻撃者は、攻撃対象となる利用者と TSA との間の通信経路上において、攻撃対象の利用者から送信されたタイムスタンプの要求情報を傍受し、奪取するという攻撃が考えられる。このため、攻撃が成功する可能性は、利用する通信ネットワークがどの程度完備されているか（ネットワークの信頼性）に依存する。

攻撃者が TSA と結託することが可能な場合、攻撃が成功する可能性は、攻撃対象の利用者が毎回どの TSA を利用するかを特定できるか否かに依存する。毎回利用する TSA を特定することが可能な場合には、攻撃が成功する可能性が高くなる。一方、例えば、攻撃対象となる利用者が毎回タイムスタンプの生成を要求する TSA をランダムに選択するようなシステムの場合、攻撃者は毎回どの TSA と結託すればよいかがわからない。このため、攻撃を成功させるためには、攻撃対象の利用者が利用する可能性のあるすべての TSA と結託する必要があり、攻撃が成功する可能性は低下する。

(C)**全面的なタイムスタンプサービスの妨害・停止**

DoS 攻撃が成功する可能性は、(a)タイムスタンプ要求を受け付けるサーバーの処理能力や(b)タイムスタンプ要求の受付方法等に依存する。

(a)については、例えば、受付サーバーの処理能力を大きくすることによって、DoS 攻撃が成功する可能性は低下すると考えられる。(b)については、例えば、タイムスタンプ要求の受付方法として、利用者と TSA がデータを交互にやり取りする Challenge and Response 方式を採用することで、攻撃者は TSA に対して短時間で大量のタイムスタンプ要求情報を送信することが困難となる。このため、DoS 攻撃が成功する可能性は低下すると考えられる。

以上の分析結果を整理すると、以下の表 5 の通り。Simple Protocol は、攻撃者が TSA と結託する可能性がある場合には、タイムスタンプを改ざんする攻撃が成立する可能性が高い。

表 5　Simple Protocol **に対する攻撃成功の可能性**

攻撃の目的	**攻撃の手段**	
	TSA **との結託が不可能なケース**	TSA **との結託が可能なケース**
既存のタイムスタンプの改ざん	デジタル署名やハッシュ関数の安全性に依存。	デジタル署名やハッシュ関数が安全であったとしても、攻撃が成立する可能性が高い。
特定の利用者に対するタイムスタンプサービスの妨害	利用される通信ネットワークの信頼性に依存。	攻撃対象者が毎回利用する TSA が特定できるか否かに依存。
全面的なタイムスタンプサービスの妨害・停止	タイムスタンプ要求情報の受付サーバーの処理能力や受付方法等に依存。	

②Linking Protocol

(A)**既存のタイムスタンプの改ざんに対する安全性**

攻撃者が TSA と結託不可能な場合、攻撃が成功する可能性は、タイムスタンプの生成に利用されるハッシュ関数やデジタル署名の安全性に依存する。

攻撃者が TSA と結託可能な場合においても、攻撃が成功する可能性は、ハッシュ関数やデジタル署名の安全性に依存する。このため、十分な安全性を有す

るハッシュ関数やデジタル署名を利用すれば、攻撃が成功する可能性は、Simple Protocol に比べて低いと考えられる。

Link 情報の一部を公開するタイプのプロトコルでは、利用されているハッシュ関数やデジタル署名が十分な安全性を有していると仮定すれば、任意のタイムスタンプを検証する際に、公開されている Link 情報を利用することによって、Link 情報やハッシュ値の改ざんを発見することができる。また、複数の TSA によるプロトコルにおいて、利用されているハッシュ関数やデジタル署名が十分な安全性を有していると仮定した場合、攻撃者が、TSA*が生成したタイムスタンプ X を改ざんするためには、TSA*が生成した Link 情報に対して他の TSA が生成したタイムスタンプも改ざんする必要がある。したがって、攻撃者は、TSA*だけではなく、「TSA*が生成した Link 情報に対してタイムスタンプを生成し、そのタイムスタンプを保管している他の TSA」とも結託する必要があり、攻撃が成功する可能性は Simple Protocol に比べて低下すると考えられる。

(a)Linear Linking Protocol

Linear Linking Protocol（図 3 参照）を例に、安全なハッシュ関数やデジタル署名を利用している場合には、Link 情報を公開することによって、タイムスタンプ $SIG_{TSA}(H_n, T_n, n, L_n)$の対象となる文書のハッシュ値 H_n を改ざんすることは困難であることを以下の手順で確認できる（図 9 参照）。なお、時刻情報 T_n を改ざんする場合も同様となる。

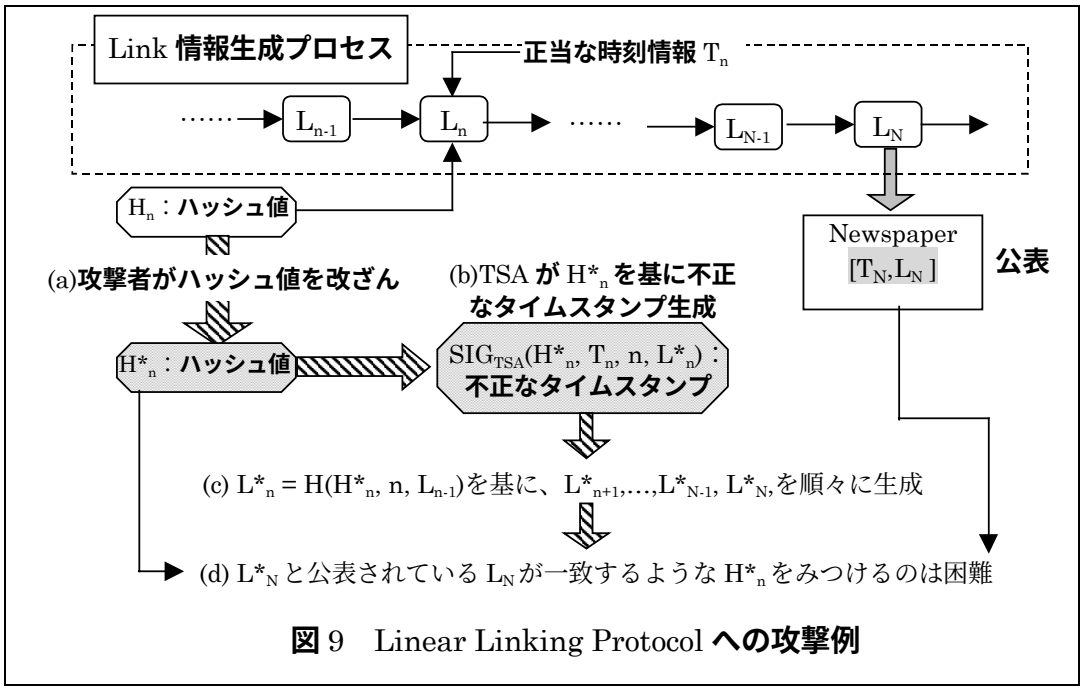


図 9　Linear Linking Protocol への攻撃例

(i)攻撃者は、時刻 T_n のタイムスタンプ $SIG_{TSA}(H_n, T_n, n, L_n)$に対して、ハッシュ値 H_n を別のハッシュ値 H^*_n に置き換えたタイムスタンプ $SIG_{TSA}(H^*_n, T_n, n, L^*_n)$を生成したいとする。攻撃者は、TSA に H^*_n を送付。

(ii)TSA は、ハッシュ値 H^*_n を時刻 T_n に受け付けたものとして、Link 情報 $L^*_n = H(H^*_n, n, L_{n-1})$を不正に生成した上で、不正なタイムスタンプ $SIG_{TSA}(H^*_n, T_n, n, L^*_n)$を生成。

(iii)TSA は、Link 情報 L^*_n を基に Link 情報$[L^*_{n+1},..., L^*_{N-1}, L^*_N]$を生成。

(iv)攻撃者と TSA は、不正なタイムスタンプ $SIG_{TSA}(H^*_n, T_n, n, L^*_n)$の検証が成功するように、公表されている L_N と不正に生成した L^*_N が一致するように、ハッシュ値 H^*_n を設定する必要がある。このようなハッシュ値 H^*_n を設定することが可能か否かは、ハッシュ関数の安全性に依存する。このため、攻撃に必要なハッシュ値 H^*_n をみつけることは困難となるように、十分な安全性を有するハッシュ関数を利用する必要がある。

(b)Linking Protocol using Tree Structure

Linking Protocol using Tree Structure（図 5 参照）の場合も、既存のタイムスタンプの内容を改ざんする攻撃については、利用するハッシュ関数が安全であると仮定すれば、攻撃の成功する可能性は低い。

Linking Protocol using Tree Structure では、1 つの round で受付可能なハッシュ値の数が予め設定されており、実際のハッシュ値の数がそれを下回ったケースでは、TSA がランダムな値もしくは事前に設定された値をパディングする仕組みとなっている。このため、本プロトコルでは、TSA がパディングデータを改ざんするという攻撃が考えられる。しかし、この攻撃も、安全なハッシュ関数を利用する場合、攻撃が成功する可能性は低い。これは以下の手順で確認することができる（次頁の図 10 参照）。

(i)攻撃者は、TSA と結託して、ハッシュ値 H_1 に対するタイムスタンプ(H_2, X_2, Y_2, RH_i)を改ざんし、別のハッシュ値 H^*_1 に対するタイムスタンプ$(H_2, X_2, Y^*_2, RH^*_i)$を生成したいとする。なお、既に他の利用者に送付してしまった$(H_2, X_2, Y^*_2, RH^*_i)$以外のデータは操作できないものとする。

(ii)TSA は、ハッシュ値 H^*_1 を第 i round において受け付けたものとして、パディングデータ P を操作して中間のハッシュ値 Y^*_2 や RH^*_i を生成し、タイムスタンプ$(H_2, X_2, Y^*_2, RH^*_i)$を生成。

(iii)TSA は、パディングデータ P*を、以下の式が成立するように操作する。

$$RH_i = RH^*_i (= H(\, H(\, H(H^*_1, H_2), X_2), H(X_3, H(H_7, P^*))))$$

ただし、H_2、X_2、X_3、H_7 は他の利用者のハッシュ値およびそれらを結合してハッシュ化したデータであることから操作することはできない。TSA が操作可能なのは P*である。

上記の式が成立するような P*の値をみつけることができるか否かは、ハッシュ関数の安全性に依存する。このため、RH_i= RH^*_i を満足するような P*をみつけることが困難となるように、十分な安全性を有するハッシュ関数を利用する必要がある。

(iv)TSA は、RH_i= RH^*_i が成立するような P*をみつけることができない場合であっても、

$$SRH^*_i = H(RH^*_i, SRH_{i-1})$$
$$SRH^*_{i+1} = H(RH_{i+1}, SRH^*_i)$$
$$\cdots$$
$$SRH^*_k = H(RH_k, SRH^*_{k-1})$$

という手順で、SRH の系列を改ざんすることができる。生成される SRH^*_k が既に公表されている SRH_k と一致するように、P*の値、他の round におけるパディングデータ、(SRH_{i-1}, SRH_{i-2},....）および（$SRHi_{i+1}$, SRH_{i+2},...）を操作することができるか否かは、利用されるハッシュ関数の安全性に依存する。このため、SRH^*_k=SRH_k を満足するようなパディングデータや SRH の系列をみつけることは困難となるように、十分な安全性を有するハッシュ関数を利用する必要がある。

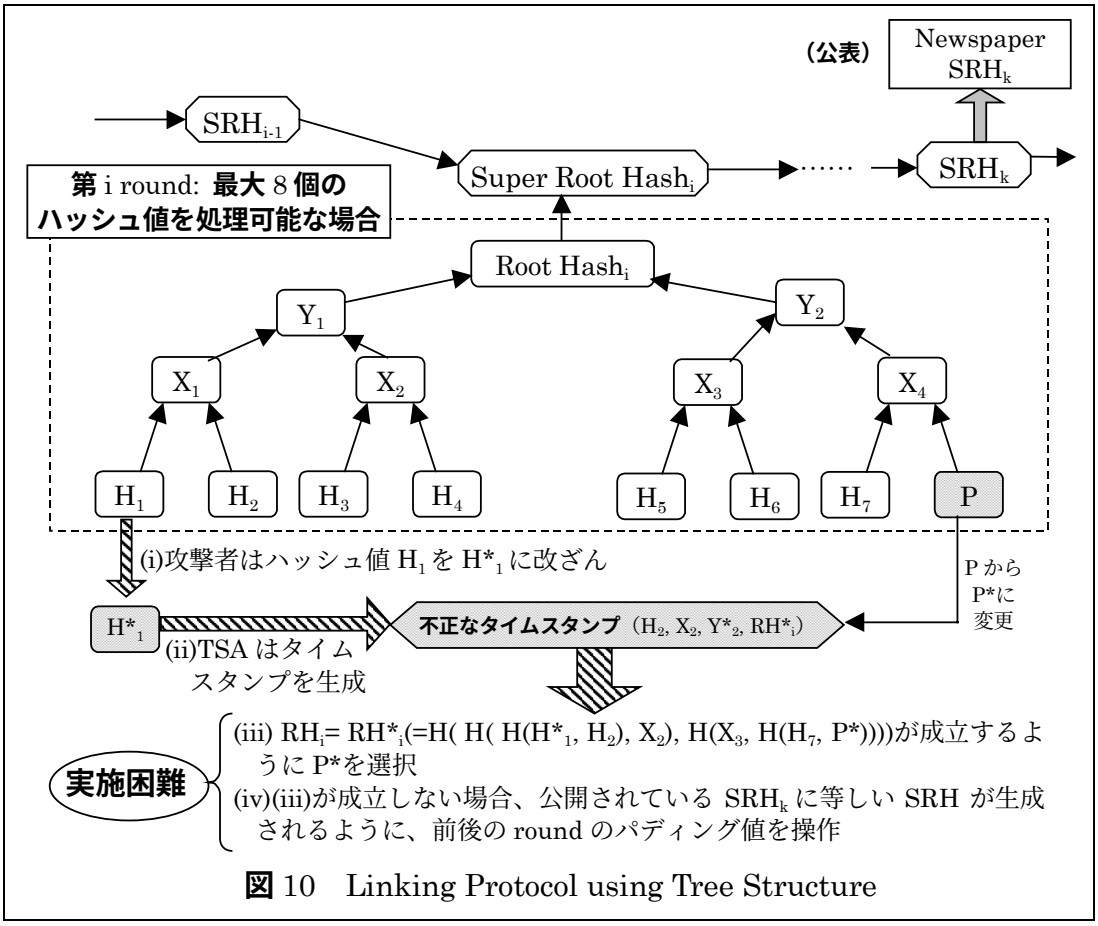


図 10　Linking Protocol using Tree Structure

このように、Linking Protocol using Tree Structure において発生するパディングデータを操作する攻撃においても、Link 情報（Super Root Hash）の一部が公表され、さらに利用されるハッシュ関数が十分な安全性を有している場合には、一旦生成されたタイムスタンプを改ざんすることは困難である。このことは、例えば図 10 において、H_1 以外のすべてのハッシュ値（H_2, ….,H_7）がすべてパディングデータであって、TSA が操作可能であったとしても同様の結果となる。

ただし、SRH をどの程度の頻度で公表するかが重要である。公表の頻度が低いほど、SRH が公表されるインターバルが長くなり、利用者から送付されたハッシュ値を受け付けた時間帯として特定される時間帯の幅が広くなる。すなわち、タイムスタンプの精度が低下する[19]。このため、SRH の公表頻度は、各アプリケーションごとにタイムスタンプに必要とされる精度や公表に伴うコスト等を検討した上で決定する必要がある。

(c)**複数の** TSA **による** Linking Protocol

利用されるハッシュ関数やデジタル署名が安全である場合、攻撃者が、複数の TSA による Linking Protocol（図 6 および図 7 参照）において既存のタイムスタンプを改ざんするためには、自分が管理するタイムスタンプを改ざんするだけではなく、他の TSA が保管している Link 情報のハッシュ値やそれに対するタイムスタンプを改ざんする必要がある。このため、TSA の数が多くなるほど、攻撃者がすべての TSA と結託することはより困難となることから、攻撃が成功する可能性は Simple Protocol に比べて低いと考えられる。

なお、本プロトコルにおいても、1 つの TSA だけを利用する Linking Protocol と同様に、Link 情報の一部（例えば、Linking Protocol using Tree Structure を利用する場合には SRH）を公開することによって、タイムスタンプの改ざんを一層困難にすることができる。Link 情報の公表頻度は、タイムスタンプを利用するアプリケーションに必要とされるタイムスタンプの安全性等の要件から決定されることとなる。

(B)**特定の利用者に対するタイムスタンプサービスの妨害**

攻撃者が TSA と結託することが不可能な場合、攻撃者は、攻撃対象となる利用者と TSA との間の通信経路上において、攻撃対象の利用者から送信され

[19] 例えば、週に一度 SRH が公表されている場合、TSA と利用者が結託したとしても、タイムスタンプの要求情報の受付時間を週を越えて改ざんすることはできない。しかし、SRH の公表頻度が低下し、月に一度公表することになった場合、タイムスタンプの要求情報の受付時間は月を超えて改ざんすることができないものの、その月の中では受付時間が改ざんされる可能性が残ると考えられる。

たタイムスタンプの要求情報を傍受し、奪取するという攻撃が考えられる。このため、攻撃が成功する可能性は、利用する通信ネットワークがどの程度完備されているか（ネットワークの信頼性）に依存する。

攻撃者が TSA と結託することが可能な場合には、攻撃が成功する可能性は、攻撃対象の利用者が毎回どのTSAを利用するかを特定できる否かに依存する。

(C)**全面的なタイムスタンプサービスの妨害・停止**

DoS 攻撃が成功する可能性は、Simple Protocol 同様、(a)タイムスタンプ要求を受け付けるサーバーの処理能力や(b)タイムスタンプ要求の受付方法等に依存する。

以上の分析結果を整理すると、以下の表 6 の通り。Linking Protocol では、利用するハッシュ関数やデジタル署名が安全である場合、Link 情報を公表したり、複数の TSA を利用したりすることによって、攻撃者が TSA と結託する可能性がある場合でも、既存のタイムスタンプを改ざんするという攻撃が成立する可能性は Simple Protocol に比べて低い。その他の攻撃が成功するか否かについては、Simple Protocol と同様である。

表 6　Linking Protocol **に対する攻撃成功の可能性**

攻撃の目的	**攻撃の手段**	
	TSA **との結託が不可能なケース**	TSA **との結託が可能なケース**
既存のタイムスタンプの改ざん	デジタル署名やハッシュ関数の安全性に依存。	デジタル署名やハッシュ関数が安全である場合、攻撃が成立する可能性は Simple Protocol に比べて低い。 ― ①Link 情報の一部を公表する、②ある TSA の Link 情報に対して他の TSA がタイムスタンプを生成するという方法により、TSA の外部に改ざんが困難なデータを確保。 ― 複数の TSA によるプロトコルの場合、タイムスタンプの生成に関わる TSA の数が増えるほど攻撃は困難となる。
特定の利用者に対するタイムスタンプサービスの妨害	利用される通信ネットワークの信頼性に依存。	攻撃対象者が毎回利用する TSA が特定できるか否かに依存。
全面的なタイムスタンプサービスの妨害・停止	タイムスタンプ要求情報の受付サーバーの処理能力や受付方法等に依存。	

③Distributed Protocol

(A)**既存のタイムスタンプの改ざんに対する安全性**

攻撃者が TSA と結託不可能なケースでは、攻撃が成功する可能性は、利用されるハッシュ関数やデジタル署名の安全性に依存する。

攻撃者が TSA と結託可能なケースでは、利用されるハッシュ関数やデジタル署名が安全である場合、攻撃が成功する可能性は Simple Protocol に比べて低い。これは、攻撃者が複数の TSA と結託することが必要であり、TSA の数が増加するにつれて、攻撃者がすべての TSA と結託することはより困難であるとの考え方である。したがって、攻撃を防止するためには、十分な数の TSA を利用することが必要となる。利用する TSA の数を決定するためには、タイムスタンプを利用する対象となるアプリケーションの安全性やコスト等に関する要件を考慮する必要がある。

(B)特定の利用者に対するタイムスタンプサービスの妨害

攻撃者が TSA と結託することが不可能な場合、攻撃者は、攻撃対象となる利用者と TSA との間の通信経路上において、攻撃対象の利用者から送信されたタイムスタンプの要求情報を傍受し、奪取することが必要となる。このため、攻撃が成功する可能性は、利用する通信ネットワークがどの程度完備されているか（ネットワークの信頼性）に依存する。

攻撃者が TSA と結託することが可能な場合、攻撃が成功する可能性は、攻撃対象となる利用者が毎回利用する TSA を特定できるか否かに依存する。

(C)全面的なタイムスタンプサービスの妨害・停止

DoS 攻撃が成功する可能性は、Simple Protocol 同様、(a)タイムスタンプ要求を受け付けるサーバーの処理能力や(b)タイムスタンプ要求の受付方法等に依存する。

以上の分析結果を整理すると、以下の表 7 の通り。

表 7　Distributed Protocol に対する攻撃成功の可能性

攻撃の目的	攻撃の手段	
	TSA との結託が不可能なケース	TSA との結託が可能なケース
既存のタイムスタンプの改ざん	デジタル署名やハッシュ関数の安全性に依存。	安全なデジタル署名やハッシュ関数を利用する場合、攻撃が成立する可能性は Simple Protocol に比べて低い。 ― タイムスタンプは複数の TSA が生成したデジタル署名を基に生成されており、任意のタイムスタンプを改ざんするためには複数の TSA と結託することが必要。 ― TSA の数を増やすことで、攻撃者と TSA の結託はより困難となる。
特定の利用者に対するタイムスタンプサービスの妨害	利用される通信ネットワークの信頼性に依存。	攻撃対象者が毎回利用する TSA が特定できるか否かに依存。
全面的なタイムスタンプサービスの妨害・停止	タイムスタンプ要求情報の受付サーバーの処理能力や受付方法等に依存。	

Distributed Protocol では、攻撃者が既存のタイムスタンプの改ざんに成功するためには、複数の TSA と結託する必要があり、攻撃が成功する可能性は Simple Protocol に比べて低い。その他の攻撃が成功するか否かについては、Simple Protocol や Linking Protocol と同様である。

(3)安全性に対する検討のまとめ

前節における安全性に関する分析結果を整理すると、以下の表 8 の通り。

表 8　各デジタルタイムスタンプシステムの安全性

<table>
<tr><th colspan="2"></th><th>Simple Protocol</th><th>Linking Protocol</th><th>Distributed Protocol</th></tr>
<tr><td rowspan="2">既存のタイムスタンプの改ざん</td><td>攻撃者が TSA と結託可能な場合</td><td>安全なデジタル署名やハッシュ関数を利用したとしても、他のプロトコルに比べて改ざんが成功する可能性が高い。
― TSA がタイムスタンプの生成に関するすべてのデータを管理しているため、TSA が既存のタイムスタンプの内容を改ざんし、再びタイムスタンプ（デジタル署名）を生成することが可能。</td><td>安全なデジタル署名やハッシュ関数を利用する場合、Simple Protocol に比べて、改ざんが成功する可能性は低い。
― ①Link 情報の一部（例えば SRH）を公表する、あるいは、②ある TSA の Link 情報に対して他の TSA がタイムスタンプを生成するという方法により、TSA の外部に改ざんが困難なデータを確保。
― 上記②の方式では、TSA の数を増やすことで、攻撃者と TSA の結託はより困難となる。</td><td>安全なデジタル署名やハッシュ関数を利用する場合、Simple Protocol に比べて、改ざんが成功する可能性は低い。
― タイムスタンプは複数の TSA が生成したデジタル署名を基に生成されており、任意のタイムスタンプを改ざんするためには複数の TSA と結託することが必要。
― TSA の数を増やすことで、攻撃者と TSA の結託はより困難となる。</td></tr>
<tr><td>攻撃者が TSA と結託不可能な場合</td><td colspan="3">利用されているハッシュ関数やデジタル署名が十分な安全性を有している限り、攻撃が成功する可能性は低い。</td></tr>
<tr><td rowspan="2">特定の利用者に対するタイムスタンプサービスの妨害</td><td>攻撃者が TSA と結託可能な場合</td><td colspan="3">攻撃が成功する可能性は、攻撃の対象となる利用者が毎回利用する TSA を特定可能か否かに依存。
― 複数の TSA を利用可能な環境では、利用者は、毎回 TSA をランダムに選択することによって、攻撃成功の可能性を低下させることが可能。</td></tr>
<tr><td>攻撃者が TSA と結託不可能な場合</td><td colspan="3">攻撃が成功する可能性は、タイムスタンプシステムに利用される通信ネットワークの信頼性に依存。
― 攻撃者は、攻撃対象となる利用者と TSA との間の通信経路上において、攻撃対象の利用者から送信されたタイムスタンプの要求情報を傍受し、奪取することが必要。</td></tr>
<tr><td colspan="2">全面的なタイムスタンプサービスの妨害・停止</td><td colspan="3">攻撃が成功する可能性は、タイムスタンプ要求情報の受付サーバーの処理能力や受付方法等に依存。
― サーバーの処理能力の拡張を行うほか、Challenge and Response 方式によってタイムスタンプ要求情報を受付けるといった方法により、本攻撃法の有効性を低下させることが可能。</td></tr>
</table>

TSA が攻撃者と結託する可能性がある場合、既存のタイムスタンプの改ざんを防ぐためには、Link 情報の一部を公開する、もしくは複数の TSA による Linking Protocol を利用することが必要となる。また、十分な数の TSA を利用した Distributed Protocol を利用することも有効な対策となる。いずれのプロトコルにおいても複数の TSA を利用可能な環境であれば、利用者がどの TSA を利用するかをランダムに決めるスキームを導入することが望ましい。ただし、こうした評価は、ハッシュ関数の安全性を前提とするものである点には留意が必要である。

実際に Linking Protocol や Distributed Protocol を利用するためには、Simple Protocol に比べてシステムを構築するためのコストがかかるほか、同一のタイムスタンプサービスを提供する複数の TSA を利用することができる環境が前提となる。このため、どのプロトコルを採用するかを検討する場合には、現在の TSA 等タイムスタンプのインフラ状況を踏まえた上で、タイムスタンプシステムに必要とされる安全性だけではなく、TSA の不正行為が発生するリスク等を含めたコストも十分考慮することが必要である。

対象となるアプリケーションによっては Simple Protocol が利用されるケースは十分存在すると考えられる。TSA を信頼してよい場合はもちろんだが、それ以外にも例えば、それほど安全性を必要とせずコストを重視するアプリケーションでは Simple Protocol は利用価値がある。また、システム構成が非常に単純であることから、TSA にとってサービスを開始しやすいプロトコルであり、実際米国等では Simple Protocol を利用したタイムスタンプの商用サービスが既に開始されている。その詳細は、第Ⅲ章において説明する。

Ⅲ. デジタルタイムスタンプ技術の研究・実装動向

1. デジタルタイムスタンプシステムの研究プロジェクト

(1)日本における主要な研究プロジェクト

日本におけるデジタルタイムスタンプシステムに関する主要な研究・開発プロジェクトとして、①法務省・電子公証制度、②ニューメディア開発協会・電子公証システム実証実験、③NTT・分散時刻署名システムが挙げられる（表 9 参照)。

表 9　日本における主要な研究・開発プロジェクト

	法務省・電子公証制度	ニューメディア開発協会・電子公証システム実証実験	NTT・分散時刻署名システム
検討開始（～終了）時期	1998 年	1997 年（～1998 年）	1999 年
システムのタイプ	Simple Protocol		Distributed Protocol
利用される暗号技術	ハッシュ関数、デジタル署名		
タイムスタンプの時刻情報	TSA がハッシュ値を受け取った時刻を特定。		
システムの概要	利用者は公証人役場（あるいは電子公証センター）にデータを送付。公証人はデータに日付を添付し、署名を生成。署名はタイムスタンプとなり、利用者に送信されると同時に、公証人役場に保管。	電子公証機能の 1 つとして、1 つの TSA が生成するデジタル署名をタイムスタンプとするタイプのシステムが採用されている。	受付サーバーは、受け付けたハッシュ値を複数の分散時刻署名装置に送付。分散時刻署名装置はハッシュ値に対する分散時刻署名を生成して受付サーバーに送付。受付サーバーは一定数以上の分散時刻署名から時刻署名を生成し、タイムスタンプとする。

①法務省・電子公証制度

従来から、書面ベースの取引等については、公証人制度の下で、公証人役場において、書面に対する確定日付の付与や公正証書 [18] の作成といった公証サービスが提供されてきた。法務省では、「公証人制度を基礎として、現在提供されている公証サービスを電子文書についても利用可能なものとし、電子取引の安全を図るための手段を提供することが考えられる」として、電子公証制度の検討を進めている（法務省［1998］)。電子公証制度では、①電子確定日付の付

[18] 公正証書とは、本来は、公務員がその権限内において適法に作成する一切の文書を指すが、通常は、公証人が公証人法等の関係法令に従って、法律行為その他私権に関する事実について作成した証書を指すことが多い。なお、公正証書以外の文書は、私署証書と呼ばれる（有斐閣『法律学小事典[新版]』1996 年, p.319 および p.455)。

与、②電子私署証書の認証、③電子公正証書の作成、④電子文書の保管および存在・内容証明のサービスが検討されており、このうち①の電子確定日付の付与に対しては、特に強いニーズが存在するとされている。

(A)既存の確定日付制度

現在の法制度の下では、書面の作成の日付につき、法律上、完全な証拠力[19]が付与される場合があり、その場合の日付を確定日付という（民法施行法第４条）。また、民法等においては、確定日付が対抗要件の要素の一つとされている場合がある。例えば、指名債権の譲渡を債務者以外の第三者に対抗するためには、確定日付のある証書によって、債務者への通知を行い、または債務者の承諾を得ることが必要とされている（民法第 467 条）。

確定日付の形態としては、いくつかのものが認められており、その一つが公証人による確定日付の付与（日付のある印章の押捺）である[20]。文書に公証人による確定日付の付与を受けようとする者は、当該文書を公証人役場に持参する。公証人は、確定日付付与の請求を受けて、(a)確定日付簿に当該文書の署名者の氏名、件名および登録番号を記載し、(b)当該文書に登録番号を記入した上で、確定日付簿および当該文書に日付のある印章[21]を押捺し、(c)さらにその印章で確定日付簿と当該文書に割印を行う。これによって、当該印章の日付が確定日付となる。

(B)電子確定日付のシステム

電子確定日付のシステムは Simple Protocol に分類され、公証人が生成するデジタル署名としてタイムスタンプが生成される。このため、Linking Protocol や Distributed Protocol と比較すると、比較的単純なシステム構成

---

[19] 民事訴訟においては、裁判所は判決をするに当たり、口頭弁論の全趣旨および証拠調べの結果をしん酌して、自由な心証により、事実についての主張を真実と認めるべきか否かを判断することとされている（自由心証主義。民事訴訟法第 247 条）。従って、文書の証拠力（証拠価値の程度）の評価も、裁判所の自由心証に委ねられるのが原則であるが、文書の作成日付に関する確定日付の効果は、その例外をなすことになる。

[20] 確定日付の形態については、民法施行法第 5 条において、次の 5 つが規定されている。公証人による確定日付の付与は、(b)に該当し、内容証明郵便の日付は、(e)に該当する。

(a)公正証書である場合にはその日付。
(b)登記所又は公証人役場において私署証書に日付のある印章を押捺したときはその印章の日付。
(c)私署証書の署名者の中に死亡した者があるときは、その死亡の日。
(d)確定日付のある証書の中に引用されている私署証書の場合は、その確定日付の日付。
(e)官庁又は公署において私署証書にある事項を記入し、これに日付を記載したときは、その日付。

[21] 確定日付簿および日付のある印章の様式等については、確定日附簿及び日附印章調製規則（昭和 24 年 6 月 1 日法務府令第 11 号）に定められている。

によってタイムスタンプシステムを構築可能となっている。

具体的なタイムスタンプの生成手順は以下の通り（図 11 参照）。

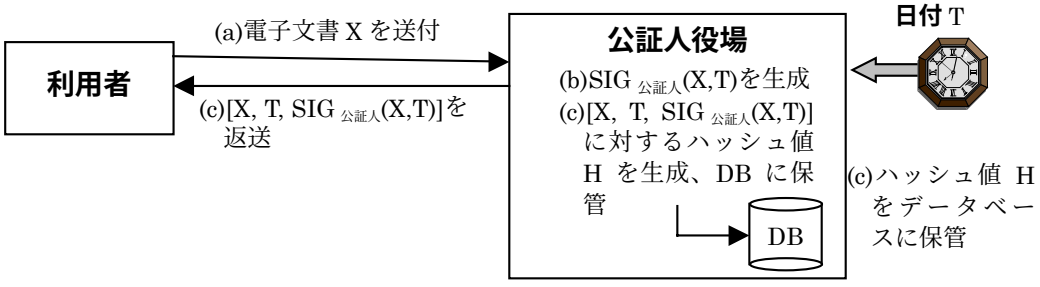


図 11 法務省・電子公証制度のタイムスタンプシステム

(a)利用者は、公証人役場（あるいは電子公証センター）に電子文書 X を送付。

(b)公証人は、電子文書 X に日付 T を加え、[X, T]のデジタル署名（タイムスタンプに相当）SIG 公証人(X,T)を生成。

(c)公証人は、電子文書 X に日付 T とデジタル署名 SIG 公証人(X,T)を加えたデータ[X, T, SIG 公証人(X,T)]のハッシュ値 H を生成してデータベース保管するとともに、データ[X, T, SIG 公証人(X,T)]を利用者に送付。

タイムスタンプの検証は、公証人が生成したデジタル署名の正当性を確認することによって行われる。

②ニューメディア開発協会・電子公証システム実証実験

ニューメディア開発協会[20]は、1997 年 10 月から 1998 年 2 月までの間、情報処理振興事業協会が推進する「エレクトロニック・コマース推進事業」[21]の一環として、日本で初めての電子公証システムの実証実験を行った。本電子公証システムの概略は、(i)まず利用者が電子メールを使って電子公証センターに電子文書を送付する、(ii)電子公証センターが電子文書に利用者情報やタイムスタンプ等の属性情報を添付し、デジタル署名を生成する、(iii)電子公証センター

[20] ニューメディア開発協会：ニューメディアについての調査、開発、啓発、普及に関する活動を行う通商産業省の認可団体であり、1972 年に設立された映像情報システム開発協会を母体としている。現在、電子公証システム等のインターネット利用環境整備のためのシステム開発のほか、電子保存システム、インターネット電子申請システム、次世代 IC カードシステム等の開発を行っている。ニューメディア開発協会および本実証実験については、http://www.nmda.or.jp/nmda/about-nmda.html を参照。

[21] エレクトロニック･コマース推進事業の一環として開始されている電子公証システムに関する実証実験には、本実証実験のほかに、富士総合研究所、日立製作所、三菱電機、日本電気、富士通等が電子公証関連のプロジェクトが存在する。これらのプロジェクトにおけるデジタルタイムスタンプシステムも Simple Protocol となっている。

が指定された受信者に電子文書を転送するとともに、CD-R 等の追記型記録媒体を用いて電子文書を保管する、というものである[22]。本システムにおけるタイムスタンプ機能に着目すると、時刻情報が添付された電子文書に対するデジタル署名がタイムスタンプとなっており、Simple Protocol に分類される。

本電子公証システムにおける電子文書の登録・照会サービス（タイムスタンプの生成を含む）の概要は以下の通り（図 12 参照）。

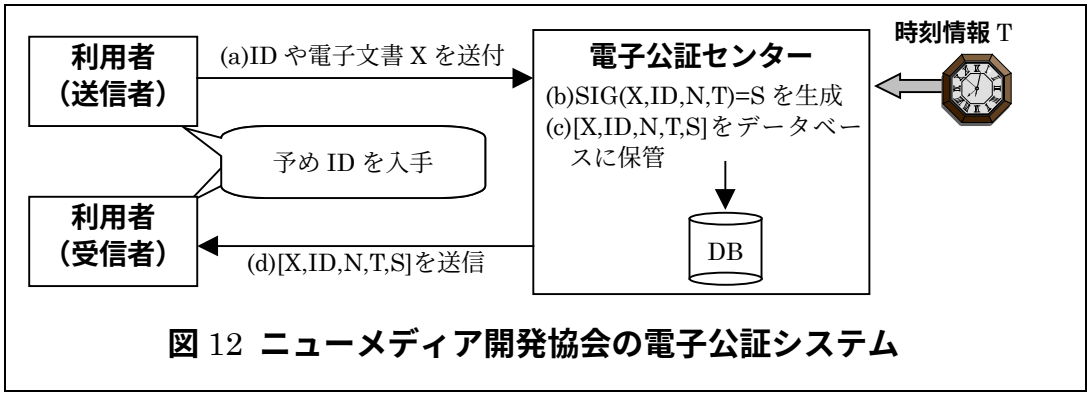


図 12 ニューメディア開発協会の電子公証システム

(A)利用者（送信者）は、予め利用者自身の ID、利用者自身の公開鍵、利用者自身の秘密鍵、利用者自身の公開鍵証明書、電子公証センターの公開鍵証明書を配布される。利用者は、電子文書 X を暗号化電子メール PEM[23]によって電子公証センターに送付。具体的には、(i)自分の秘密鍵で X に対するデジタル署名を生成、(ii)X をセッション鍵で暗号化（共通鍵暗号方式）、(iii)セッション鍵自身を電子公証センターの公開鍵で暗号化した後、これらのデータを自分の ID と公開鍵証明書とともに電子公証センターに送付。

(B)電子公証センターは、利用者の ID を確認し、自分の秘密鍵でセッション鍵を復号した後、X をセッション鍵で復号。次に、送信者の公開鍵証明書を用いて公開鍵の有効性を確認し、利用者の公開鍵を用いてデジタル署名を検証。電子公証センターは、電子文書 X に時刻情報 T、ID、シリアル番号 N 等を添付して、デジタル署名 S を生成。電子文書の真正性等を証明するデータとして、ID、N、T、S から構成される電子証明書を生成。

(C)電子公証センターは、電子文書 X に電子証明書を加えて CD-R 等追記型記録媒体に書き込み、保管。

---

[22] 本実証実験の対象分野は、情報処理技術者試験およびパソコン利用技術認定試験のインターネットを利用した受験申請受付業務や、インターネット経由での行政書士文書届出業務をはじめとする 5 つの業務であり、約 3500 人のモニターが参加。これらの業務では、インターネット経由で送信されたデータが確実送信先に到着したことを証明するサービスや送信されたデータを記録するサービス等が実施された（丹波・国分[1998]）。

[23] PEM（Privacy Enhanced Mail）：RFC 1421, 1422, 1423, 1424 に規定されている暗号化電子メールのインターネット標準。暗号化によるデータ守秘機能のほか、公開鍵証明書を用いたユーザー認証機能やデジタル署名によるメッセージ認証機能を有している。

(D)電子公証センターは、送信者が指定した受信者に電子文書と電子証明書を送付するほか、送信者本人の求めに応じてデータベースから電子文書を検索、返信。

このように、電子公証センターが生成する電子証明書がタイムスタンプの役割も果たしており、その真正性は電子公証センターのデジタル署名によって検証する。

③NTT・分散時刻署名システム

分散時刻署名システムは NTT によって研究が進められているタイムスタンプシステムであり、1999 年 2 月に開催された IWS '99[24]において発表されている（Takura, Ono and Naito [1999]）。本システムでは、分散時刻署名を生成する分散時刻署名装置と利用者との間に受付サーバーが介在し、受付サーバーが利用者のタイムスタンプ要求情報を分散時刻署名装置に配信するとともに、各分散時刻署名装置から寄せられた分散時刻署名を用いてタイムスタンプを生成する役割を有している。本システムは、Distributed Protocol に分類される。

(A)タイムスタンプシステムの形態

タイムスタンプの生成手順の概要は以下の通り（図 13 参照）。

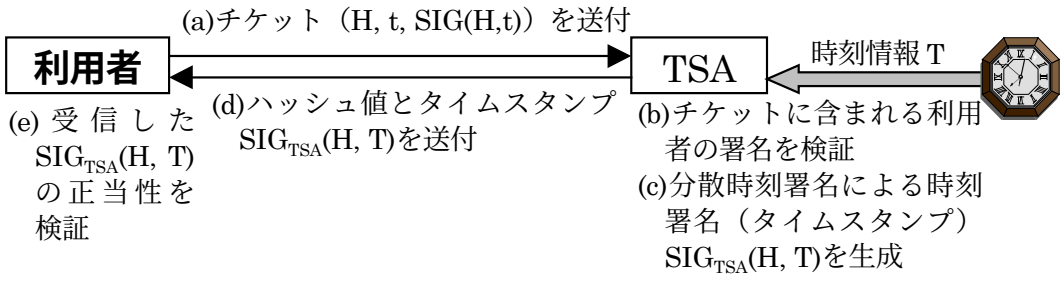


図 13 分散時刻署名システムの概要

(a)利用者は、TSA にタイムスタンプ要求情報（チケットと呼ばれる）として文書のハッシュ値 H、チケットの有効期間 t、これらのデータに対する利用者の署名 SIG(H, t)を送付。チケットを利用することのメリットは、(i)TSA が、チケットに添付されている署名を検証することによって、利用者の本人確認を行うことができる、(ii)チケットに含まれる署名の有効期間を短くすることによって、署名に必要とされる安全性のレベルを低く設定することが可能となり、デジタル署名方式の鍵長を短くして署名生成・検

[24] IWS（Internet Workshop）：電子情報通信学会の時限研究会であるインターネット研究会が主催するワークショップであり、ネットワークセキュリティ、マルチメディア通信、トラフィック管理等、インターネット関連技術が対象分野となっている。

証処理の高速化を図ることができる、という 2 点。

(b)TSA は、チケットの有効期間と利用者の署名の正当性を検証。

(c)TSA は、チケットを受け付けた時刻をタイムスタンプの時刻とし、ハッシュ値に時刻情報 T を添付して、それらのデータに対する署名 $SIG_{TSA}(H, T)$（タイムスタンプに相当）を生成する。時刻署名の生成には、「分散時刻署名」と呼ばれる方式（後述）が利用される。

(d)TSA は、タイムスタンプ $SIG_{TSA}(H, T)$を利用者に送付。

(e)利用者は、受信したタイムスタンプの真正性を検証。

(B)分散時刻署名の生成手順

分散時刻署名は、各分散時刻署名装置が部分秘密鍵によって生成した部分時刻署名を用いて生成される。各分散時刻署名装置は、1 つの秘密鍵から生成された部分秘密鍵を所有しており、その部分秘密鍵を利用して分散時刻署名を生成する。分散時刻署名装置が 3 個であり、署名を生成するためには 2 つ以上の部分署名が必要な場合、タイムスタンプの生成手順は以下の通り（図 14 参照）。

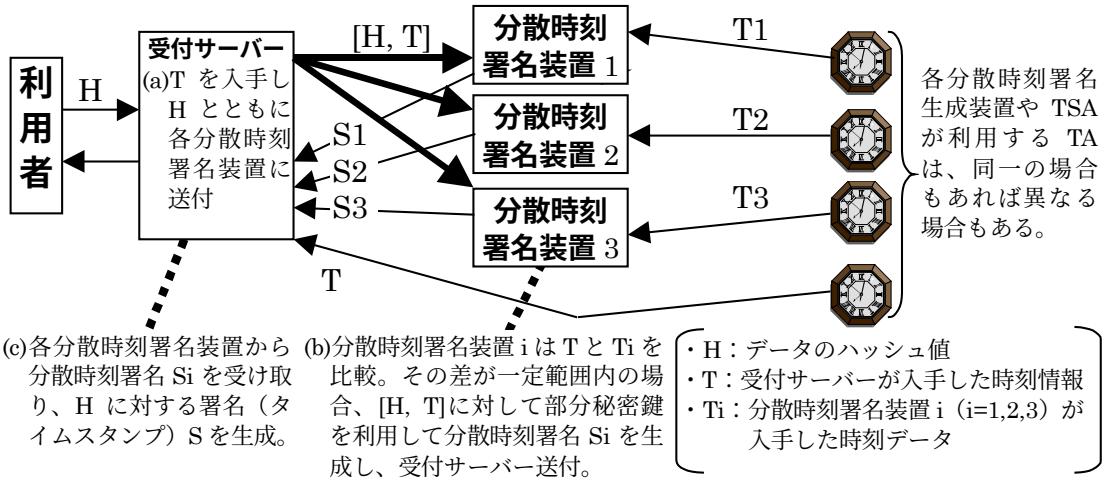


図 14 分散時刻署名装置が 3 個の場合の分散時刻署名の生成手順

(a)受付サーバーは、利用者から受け取ったハッシュ値 H に時刻情報 T を添付し、各分散時刻署名装置にデータ[H, T]を送付。

(b)分散時刻署名装置は、受付サーバーから[H, T]を受信した時刻 Ti（i=1,2,3）と T の差を計算し、その差が一定範囲内に収まっている場合には分散時刻署名 Si を生成した上で受付サーバーに送付。

(c)受付サーバーは、3 つのうち、例えば 2 つ以上の分散時刻署名を入手すれば、タイムスタンプ S を生成することができる。ただし、1 つの部分時刻署名しか入手できなかった場合にはタイムスタンプは生成できない。この場合、受付サーバーは再び T を生成して、各分散時刻署名装置に[H, T]を送付。

本システムでは、Threshold が存在する秘密分散技術を用いたデジタル署名を利用することによって、分散時刻署名装置間の結託を困難にするほか、一部の装置に故障が発生した場合でも対応できる仕組みになっている。ただし、独立した複数の分散時刻署名装置が必要である等、実装するためには一定のインフラ整備が必要となる。

(2)海外における研究プロジェクト

海外における主要な研究プロジェクトとしては、①ベルギーの TIMESEC、②エストニアの Cuculus、③スペインの PKITS が挙げられる（表 10 参照）。

表 10 海外における主要な研究プロジェクト

	TIMESEC （ベルギー）	Cuculus (エストニア)	PKITS (スペイン)
検討開始（～終了）時期	1996 年（～1998 年）	1997 年	1997 年（～1998 年）
システムのタイプ	Linking Protocol using Tree Structure	Linear Linking Protocol	複数の TSA による Linking Protocol
利用される暗号技術	ハッシュ関数（SHA-1, RIPEMD-160 を利用）	ハッシュ関数、デジタル署名	ハッシュ関数
タイムスタンプの時刻情報	TSA が各ハッシュ値を受け取った時刻の前後関係を特定。	TSA がハッシュ値を受け取った時刻を特定。	
システムの概要	TSA は、一定時間内に受信したハッシュ値を結合して Round Value(RV)を生成。RH は、直前の Round の Root RV (RRV)と結合・ハッシュ化されて RRV となる。RRV は、一定期間ごとに時刻情報とともに新聞等に掲載される。タイムスタンプは、RV を計算するために必要な情報から構成される。	TSA は、直前に受信したハッシュ値等を利用して Link 情報を生成。Link 情報は、ハッシュ値を時系列的に連結するデータ。TSA は、ハッシュ値、ID 情報、時刻情報、Link 情報等を含むタイムスタンプを生成して、利用者に送信。 ― Cuculus ではデジタル署名がタイムスタンプとなる一方、PKITS ではデジタル署名を利用しない。	

①TIMESEC（ベルギー）

TIMESEC は 1996 年 8 月から 2 年間実施されたベルギーの研究プロジェクトであり、ルーベン・カトリック大学（Katholieke Universiteit Leuven, KUL）やルーバン・カトリック大学（Université Catholique de Louvain, UCL）の暗号研究チームを中心とする検討グループによって進められた。研究に必要な資金はベルギー連邦科学技術文化局（Federal Office for Scientific, Technical and Cultural Affairs）から援助されており、TIMESEC は国家プロジェクトの 1 つとして位置付けられている。本プロジェクトの研究成果は、将来ベルギーにおいてデジタルタイムスタンプサービスを公的に提供する場合の参考情報として

利用されることとなっている[25]。

(A)**タイムスタンプシステムの形態**

TIMESEC では、現時点で最も実装に適したタイムスタンプシステムの形態として Linking Protocol を挙げている。これは、①Simple Protocol を実装する場合、TSA が攻撃者と結託不可能であることが前提条件となるが、現時点では、業務運営に関して信頼性の高い TSA を利用することが困難である、②Distributed Protocol を実現するためには数多くの TSA が必要となる、等の問題点が存在するためであると指摘されている（Massias and Quisquater [1997]）。このため、TIMESEC では Linking Protocol using Tree Structure を採用している。

(B)TIMESEC の Linking Protocol using Tree Structure

TIMESEC の Linking Protocol は Tree 構造を利用するものであり、タイムスタンプの生成手順は以下の通り（図 15 参照）。

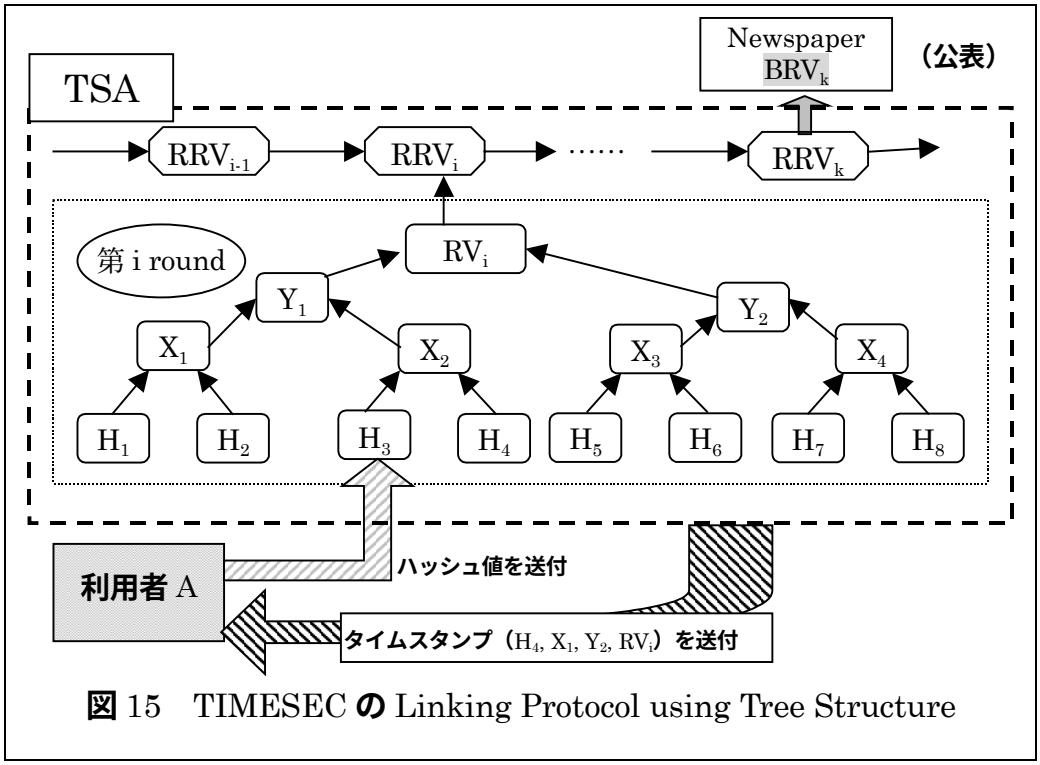


図 15　TIMESEC **の** Linking Protocol using Tree Structure

(a)TSA は予め round を設定。round は一定時間ごとに区切られており、TSA は 1 round の時間と 1 round 内で受付可能なハッシュ値の数の最大値を設定（図 15 では 8 個）。

[25] TIMESEC に関する情報は、http://www.dice.ucl.ac.be/crypto/TIMESEC/TIMESEC.html から入手可能。

(b)利用者（例えば、図 15 の A）は、TSA にタイムスタンプ要求情報としてタイムスタンプの対象となる文書のハッシュ値 H_3 を送付（第 i round に送付したものとする）。

(c)TSA は、第 i round が終了した後、受け取ったハッシュ値 H_1～H_8 を用いて Link 情報である Round Value（RV_i）を生成（手順については前掲の図 5 参照）。第(i-1) round の Root Round Value（RRV_{i-1}）と RV_i を結合・ハッシュ化して、第 i round の RRV_i を生成。RRV は TSA が保管。

(d)TSA は、ハッシュ値 H_3 に対するタイムスタンプとして（H_4, X_1, Y_2, RV_i）を利用者 A に送付。

(e)TSA は、一定期間ごとに RRV を時刻情報とともに新聞に発表。発表される RRV は Big Round Value（BRV）と呼ばれる。

タイムスタンプの検証は以下の手順で行われる（詳細は図 6 参照）。

(a)利用者は、タイムスタンプ（H_4, X_1, Y_2, RV_i）とハッシュ値 H_3 から RV_i を生成し、タイムスタンプの一部である RV_i と比較。

(b)利用者は、TSA が保管している RRV_{i-1} と生成した RV_i から RRV_i を生成する。生成した RRV_i と、TSA が保管している(RV_i, RV_{i+1},…,RV_k)から RRV_k を生成。

(c)利用者は、生成した RRV_k と新聞に掲載されている RRV_k が一致することを確認。

このように、本システムでは、タイムスタンプを生成する際にデジタル署名を利用していない。これは、デジタル署名における署名生成鍵のような「秘密情報」に依存したシステムでは、万一秘密情報が露見した場合にシステム全体の信頼が大きく低下するおそれがあることから、そうした秘密情報に依存しないシステムが望ましいとの考え方によるものである。

また、本システムで利用されているハッシュ関数は SHA-1[26]と RIPEMD-160[27]の 2 種類であり、それぞれのハッシュ関数を利用したタイムスタンプや Link 情報を生成するスキームを採用している。これは、どちらか一方のハッシュ関数に有効な攻撃法が発表されたとしても、もう 1 つのハッシュ関数を利

[26] SHA-1：1995 年に米国政府のハッシュ関数標準となった SHA（Secure Hash Algorithm）の改良方式であり、ハッシュ値のサイズは 160 bit である（NIST[1995]）。現時点では、SHA-1 に対して、安全性上の問題点は指摘されていない。

[27] RIPEMD-160：European Community における情報通信技術に関する研究プログラム RIPE（The Research and Development in Advaced Communication Technologies in Europe）の成果の 1 つとして 1996 年に発表されたハッシュ関数であり、ハッシュ値は 160 bit（Dobbertin et al.[1996]）。現時点では、RIPEMD-160 に対して、安全性上の問題点は指摘されていない。

用することによって、タイムスタンプシステムの信頼性を維持することが可能となるためである。

②Cuculus（エストニア）

Cuculus は、電子文書管理やデジタル署名に関するエストニアの研究プロジェクト Timestamping and electronic document（E-Doc）の一部として 1997 年から開始されたプロジェクトである。現在エストニア政府は、電子文書の法的効力に関する立法措置について検討を進めており、Cuculus はそうした検討を進める上での基礎研究として位置付けられている。本プロジェクトは、エストニアの民間研究開発機関 Cybernetica を中心に進められている[28]。

(A)タイムスタンプシステムの形態

Cuculus では、信頼できる TSA を利用することは現時点では困難であり、Link 情報を発表することで TSA によるタイムスタンプの改ざんや利用者との結託を防止することが可能な Linking Protocol が現実的な選択肢であるとしている（Lipmaa [1999]）。本プロジェクトで提案されているプロトコルは、Linear Linking Protocol に分類され、タイムスタンプの生成にハッシュ関数とデジタル署名が利用される。また、Link 情報の一部が定期的に公開される仕組みが採用されている。

(B)Cuculus の Linear Linking Protocol

本システムにおけるタイムスタンプの生成手順は以下の通り（図 16 参照）。

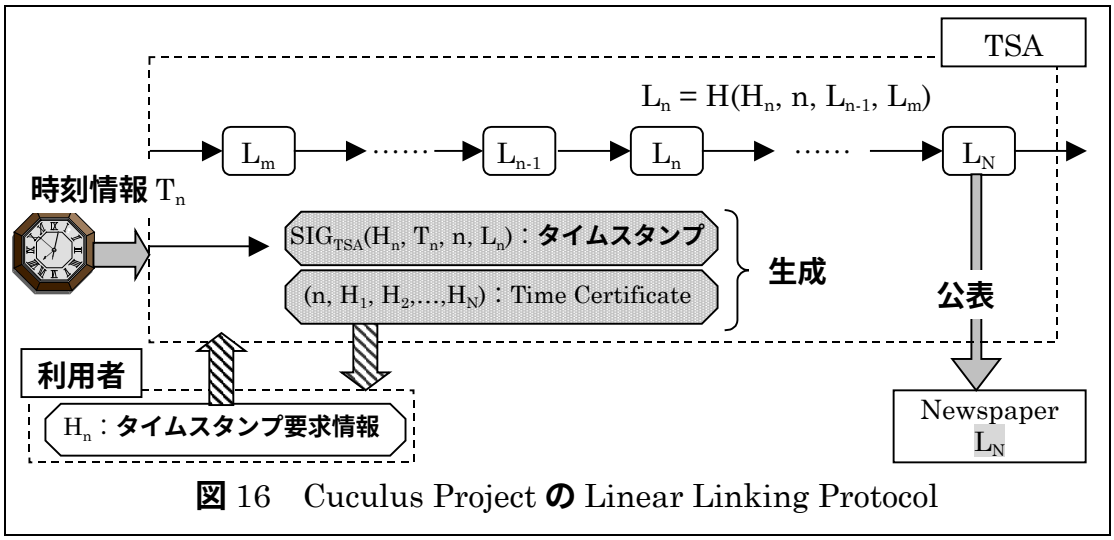


図 16　Cuculus Project **の** Linear Linking Protocol

[28] Cuculus や Cybernetica については、http://www.cyber.ee/company /index.html を参照。

(a)TSA は、予め round を設定し、1 round 内で受け付けることが可能なハッシュ値の数 N を設定。
(b)利用者は、TSA に対してタイムスタンプ要求情報としてハッシュ値 H_n を送付。
(c)TSA は、受信した H_n を用いて Link 情報 $L_n = H(H_n, n, L_{n-1}, L_m)$を生成。ただし、n はシリアル番号を表し、$L_m$ は予め設定された順番の Link 情報（例えば、直前の round における m 番目の Link 情報）を表す。
(d)TSA は、タイムスタンプとして(H_n, T_n, n, L_n)に対するデジタル署名 $SIG_{TSA}(H_n, T_n, n, L_n)$を生成するほか（ただし、$T_n$ は時刻情報）、その round が終了した後に Time Certificate と呼ばれる情報$(n, H_1, H_2, \ldots, H_N)$を生成し、利用者に送付する（同時に、自分のデータベースに保管）。TSA は、1 round 内の最後の Link 情報 L_N を新聞に掲載する。

一方、タイムスタンプの検証は以下の手順で実行される。

(a)検証者は、検証の対象となる文書のハッシュ値とタイムスタンプに含まれるハッシュ値が一致することを確認。
(b)検証者は、タイムスタンプと Time Certificate を利用して round 内の Link 情報（$L_1,\ldots,L_N$）を生成し、公表されている L_N と一致することを確認。

本システムでは、タイムスタンプを生成する際にデジタル署名が利用されており、タイムスタンプの検証はデジタル署名の検証と Link 情報の検証の 2 段階となっている点が特徴である。ただし、署名生成鍵の有効期間が切れる度にタイムスタンプの更新が必要となる。

③PKITS（スペイン）

PKITS（Public Key Infrastructure with Time Stamping Authority）は、European Commission の情報セキュリティ技術に関する調査・研究計画 ETS（European Trusted Service）[29]を構成するプロジェクトの 1 つであり、タイムスタンプ技術の理論・実装研究を行うものである。PKITS は、1997 年にスペインの FNMT[30]や郵政省を中心としたスペインの実務家や技術者によって開

[29] ETS は、欧州を中心としたグローバルな情報インフラの整備の一環として、主にデジタル署名や TTP の技術的要件・サービス内容について調査・研究を行うものであり、1992 年から開始されている。ETS は、①ETS Studies（既存の研究成果や実装例等について検討を実施）、②ETS Ⅰ Project（ETS Studies の成果を踏まえて、望ましい技術要件やサービス内容等について検討を行い、パイロット実験を実施）、③ETS Ⅱ Project（パイロット実験を実施するとともに、実装に向けた具体的な技術要件や業務要件等について検討を実施）の 3 つのフェーズから構成されている。このうち、PKITS は ETS Ⅱ Project に含まれている。
[30] FNMT（Fablica Nacional de Moneda y Timbre）：スペインにおける貨幣鋳造業務および

始され、1998 年に報告書が公表されている（FNMT[1998]）。

(A)**タイムスタンプシステムの形態**

PKITS における検討では、デジタル署名の署名生成鍵のような秘密情報に依存しないシステムが望ましいとの結論に至ったほか、Simple Protocol の前提となっている信頼できる TSA を前提とすることは現時点では時期尚早との見方から、デジタル署名は利用しないで、Linear Linking Protocol をベースとした複数の TSA による Linking Protocol が採用されている。

PKITS のプロトコルでは、Link 情報の一部が公開される仕組みにはなっていない。その代わりに、Synchronization Cycle と呼ばれるスキームが採用されている。これは、同じ Linear Linking Protocol を採用している TSA が複数存在することを前提とし、各 TSA が生成した Link 情報に対して他の TSA が適宜タイムスタンプを生成するというものである。相互に Link 情報に対するタイムスタンプを生成することによって、各 TSA の Link 情報の真正性を確保し、Link 情報の一部を公開するスキームと同様の効果をもたらすと考えられる。

(B)PKITS における複数の TSA による Linear Linking Protocol

PKITS におけるタイムスタンプの生成手順は以下の通り（図 17 参照）。

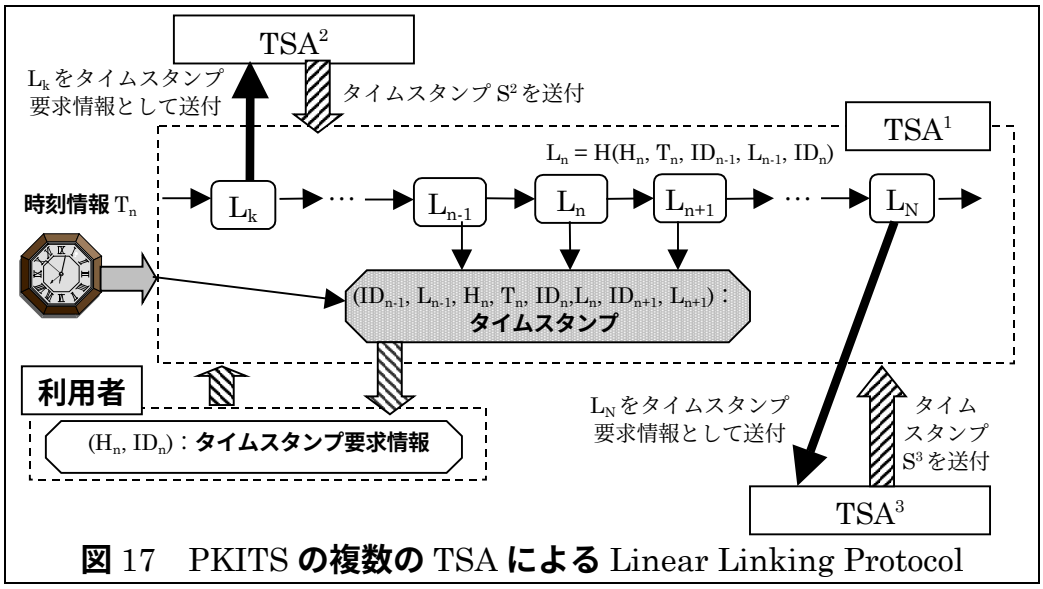


図 17　PKITS **の複数の** TSA **による** Linear Linking Protocol

(a)利用者は、タイムスタンプ要求情報として、文書のハッシュ値 H_n と自分の ID 情報 ID_n を TSA^1 に送付。

政府関連印刷業務等を営む公営企業であり、大蔵省のほか、スペイン銀行や郵便電報局等から構成される理事会によって統轄されている。紙幣印刷、貨幣鋳造、切手やパスポートの印刷のほか、IC カードの研究・開発も行っている（http://www.fnmt.es/index.htm）。

(b) TSA^1 は、Link 情報 $L_n = H(H_n, T_n, ID_{n-1}, L_{n-1}, ID_n)$を生成。ただし、$T_n$ は TSA^1 が H_n を受け付けた時刻情報。

(c) TSA^1 は、次のハッシュ値 H_{n+1} を受け取って Link 情報 L_{n+1} を生成した後、利用者にタイムスタンプとして（ID_{n-1}, L_{n-1}, H_n, T_n, ID_n, L_n, ID_{n+1}, L_{n+1}）を送付。同時に、TSA^1 は自社のデータベースにタイムスタンプを保管。

(d)TSA^1 は、ランダムに TSA^2 を選択して L_k に対するタイムスタンプ S^2 を入手するとともに、L_N に対するタイムスタンプ S^3 を TSA^3 から入手。TSA^2 および TSA^3 は自分が生成したタイムスタンプと L_k、L_N をそれぞれ保管。

タイムスタンプを検証する場合には、以下の手順となる。

(a)検証者は、文書のハッシュ値がタイムスタンプに含まれている H_n に一致することを確認。

(b)検証者は、TSA^1 からハッシュ値の系列（H_k,…, H_N）、時刻情報の系列（T_k,…, T_N）、ID 情報の系列（ID_k,…, ID_N）を入手。さらに、TSA^2 と TSA^3 からそれぞれ L_k と L_N を入手するとともに、それぞれの Link 情報に対応するタイムスタンプ S^2、S^3 を入手。

(c)検証者は、以下の計算を実行し、L_N を生成。

$$L_{k+1} = H(H_{k+1}, T_{k+1}, ID_k, L_k, ID_{k+1})$$
$$L_{k+2} = H(H_{k+2}, T_{k+2}, ID_{k+1}, L_{k+1}, ID_{k+2})$$
$$\cdots$$
$$L_N = H(H_N, T_N, ID_{N-1}, L_{N-1}, ID_N)$$

(d)検証者は、生成した L_N が TSA^3 から入手した L_N と一致することを確認。

(e)検証者は、タイムスタンプ S^2、S^3 の真正性を確認。

このように、Link 情報の真正性を他の TSA が確保することによって Link 情報を公開した場合と類似の状況を生み出し、攻撃者と TSA の結託によるタイムスタンプの改ざんを困難にしている。ただし、TSA の数が少ない場合には、攻撃者はすべての TSA と結託する可能性もある。したがって、TSA の数が多いほど安全性の観点からは望ましいが、システムを構築・管理するためのコストが嵩むと考えられる。タイムスタンプシステムの対象となるアプリケーションにおける安全性および効率性の要件を考慮した上で、最適な TSA の数を検討する必要がある。

2. デジタルタイムスタンプシステムの商用サービス

既に実施されているデジタルタイムスタンプシステムの主な商用サービスとして、米国の①Surety 社・Digital Notary Service、②Firstuse.com 社・Firstuse.com、③DigiStamp 社・e-TimeStamp、④英国の I.T. Consultancy 社・Stamper が挙げられる（表 11 参照）。既存の主な商用システムの大半は、タイムスタンプの生成方法として Simple Protocol を採用しており、Linking Protocol を採用しているのは Surety 社の Digital Notary Service のみである。

表 11 主要なデジタルタイムスタンプシステムの商用サービス

	Surety 社（米国）	Firstuse.com 社（米国）	DigiStamp 社（米国）	I.T. Consultancy 社（英国）
サービス名	Digital Notary Service	Firstuse.com	e-TimeStamp	Stamper
サービス開始年	1992 年	1998 年	1998 年	1995 年
タイムスタンプの生成方法	Linking Protocol using Tree Structure	Simple Protocol		
利用される暗号技術	ハッシュ値	ハッシュ関数		デジタル署名
タイムスタンプの時刻情報	TSA がハッシュ値を受け取った時刻を特定。			
システムの概要	TSA は 1 秒間に受信したハッシュ値を結合して Root Hash(RH)を生成（1 つの round のインターバルは 1 秒）。RH は 1 秒前の Super Hash(SH)と結合・ハッシュ化されて SH となり、NY Times 紙に掲載。タイムスタンプは時刻情報、RH の計算に必要な情報等から構成される。	TSA は、ハッシュ値に ID 情報や時刻情報を結合し、そのデータに対してタイムスタンプを生成。TSA は、タイムスタンプを利用者に送信するとともに、自社内の記憶媒体で管理。		TSA は、文書に時刻情報を加え、デジタル署名を生成してタイムスタンプとする。TSA は、タイムスタンプを利用者に送付するとともに、自社の Website 上に毎日掲載。

(1)Digital Notary Service（Surety 社）

Digital Notary Service は、Linking Protocol using Tree Structure を利用したタイムスタンプサービスであり、1992 年に開始された。Digital Notary Service では、タイムスタンプとしてハッシュ値に対する Certificate が作成されるほか、Link 情報の一部が New York Times 紙に毎週日曜日に公開される。Certificate には、タイムスタンプの対象となっている文書のハッシュ値、時刻情報、Certificate の正当性を証明するために必要な情報（Link 情報や他の文書のハッシュ値）が含まれている[31]。

①認証書の生成手順

Digital Notary Service のタイムスタンプシステムは、TIMESEC のシステムと類似している。タイムスタンプに相当する Certificate の生成手順は以下の通り（図 18 参照）。

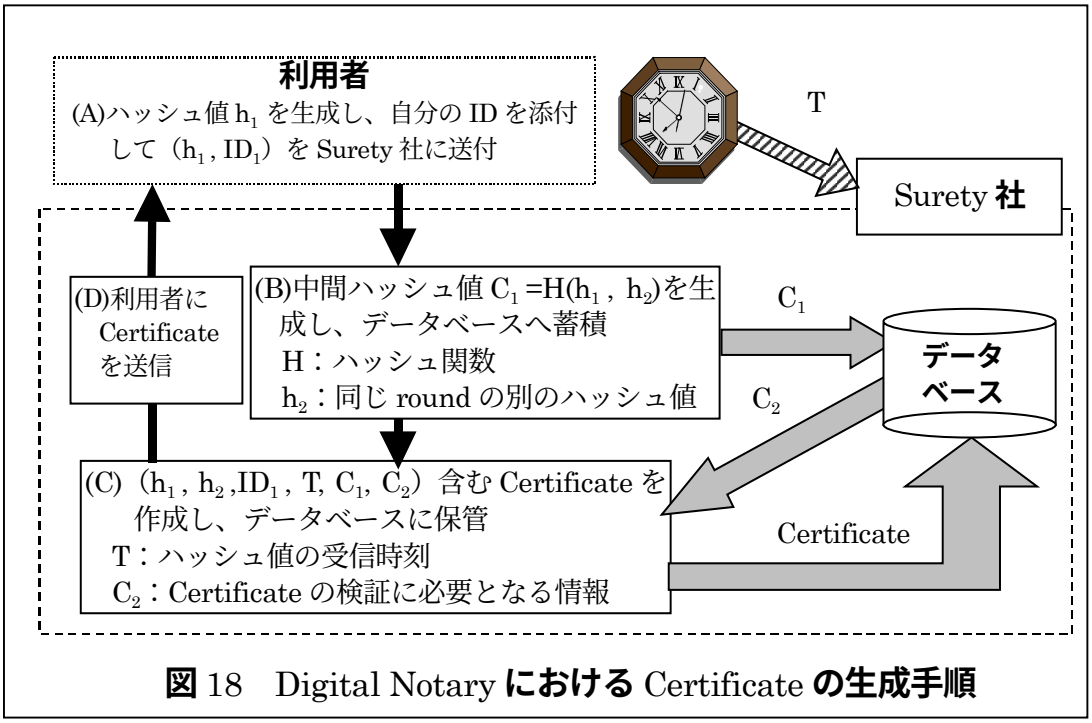


図 18　Digital Notary **における** Certificate **の生成手順**

(A)利用者は、専用ソフト（オフラインで入手）を利用してタイムスタンプの対象となる文書のハッシュ値 h_1（ハッシュ長は 288 bit）を生成した上で、h_1 に利用者固有の ID_1（SureID と呼ばれる）を結合し、(h_1, ID_1) を Surety 社に送付。

(B)Surety 社は、h_1 と同じ round（round は 1 秒ごとに更新される）で受信した別のハッシュ値 h_2 を利用して、中間ハッシュ値 C_1（ハッシュ長は 288 bit）を生成し、CD-ROM に保管。

(C)Surety 社は、(h_1, h_2, ID_1, T, C_1, C_2) を内容とする Certificae を作成し、自社のデータベースに保管。なお、T は受付時刻、C_2 は Certificate の検証を可能にするための情報（時刻 T における他の中間ハッシュ値や、後述するルートハッシュ値等が含まれる）。

(D)Surety 社は、Certificate を各利用者に送付。

②Link 情報の生成手順と Certificate **の検証手順**

Link 情報として、ルートハッシュ値やスーパーハッシュ値が生成される。これらのハッシュ値の生成手順は以下の通り（図 19 参照）。

---

[31] Surety 社のサービスについては、http://www.surety.com を参照。

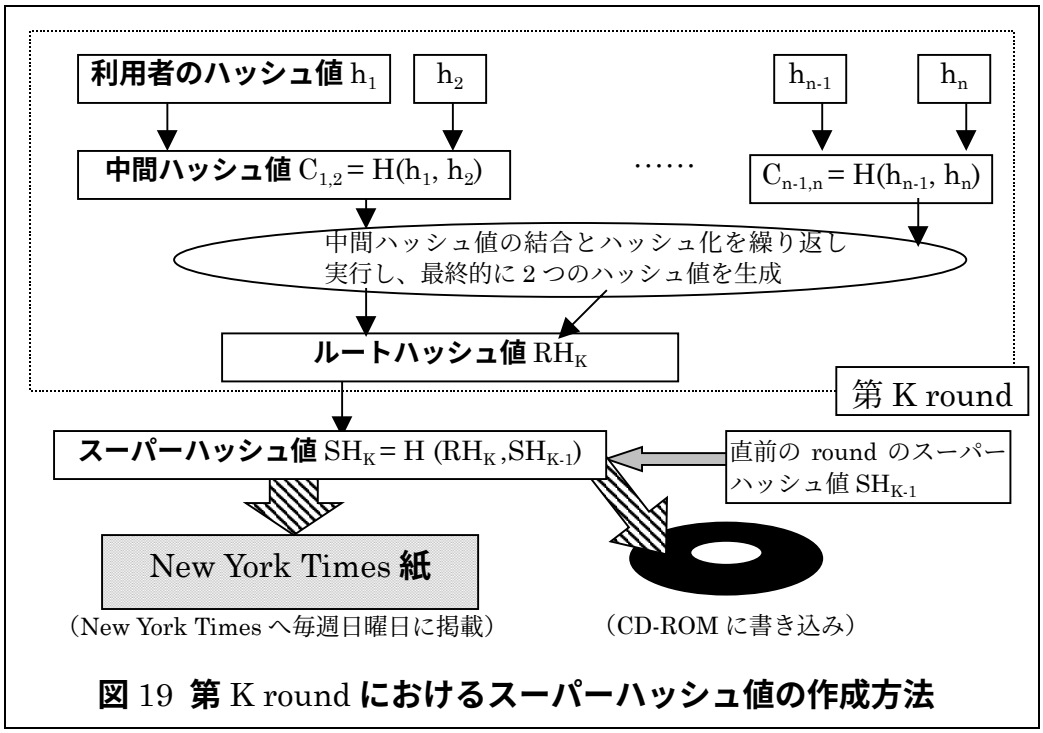


図 19 第 K round におけるスーパーハッシュ値の作成方法

(A)同一 round（図 17 では第 K round）において受信したハッシュ値（$h_1,\ldots,h_n$）（288 bit）を 2 組ずつ結合して 576 bit のデータを生成し、そのデータを再びハッシュ化して 288 bit の中間ハッシュ値（$C_{1,2},\ldots,C_{n-1,n}$）を生成。

(B)複数の中間ハッシュ値を再び結合・ハッシュ化し、最終的に 288 bit のルートハッシュ値 RH_K を生成した上で、CD-ROM に記録。ただし、ルートハッシュ値の生成に利用されるハッシュ関数は、中間ハッシュ値の生成に利用されるハッシュ関数とは異なる。

(C)第 K round のルートハッシュ値 RH_K と、第 K-1 round のスーパーハッシュ値 SH_{K-1} を結合し、再びハッシュ化して第 K round のスーパーハッシュ値 SH_K を作成。

(D)スーパーハッシュ値は、Surety 社の CD-ROM に書き込み・保管されるほか、その一部は毎週日曜日の New York Times 紙に掲載。

このように、Link 情報を生成する際にすべて同一のハッシュ関数を利用するのではなく、中間ハッシュ値とルートハッシュ値のハッシュ関数が異なっている点が特徴である。

一方、Certificate の検証は以下の手順で実行される。

(A)検証者は、対象となる文書のハッシュ値と Certificate のハッシュ値が同一であることを確認。

(B)検証者は、Certificate の情報から当該 round のルートハッシュ値を生成し、

Surety 社の CD-ROM に記録されているルートハッシュ値やスーパーハッシュ値を用いてスーパーハッシュ値の系列を生成。生成したスーパーハッシュ値の系列のうち、New York Times 紙に掲載されているスーパーハッシュ値に対応するものをみつけ、両者を比較。

③利用状況

Digital Notary Service を利用する主要な分野としては、CAD・音響実験データ（設計会社）、顧客と代理店の取引記録（保険会社、証券会社）、特許等の知的財産関連情報が中心となっており、1 日の利用件数は 100 件～10 万件であると報告されている（電子商取引実証推進協議会 [1998]）。

また、Digital Notary Service を利用したデータ仲介サービスを提供する企業も存在する。米国の NetDox 社は、インターネット上で送受信されるデータの仲介を行い、異なる認証機関から公開鍵証明書の発行を受けている利用者間の暗号化通信を可能にするとともに、データの送受信者確認、改ざん防止、送受信時刻確認、記録保管等のサービスを実施している[32]。また、米国の ZANTAZ 社は、顧客から送信された電子メール等のデータを保管し、必要に応じてデータを検索・提供する「データ保管サービス」を実施している。ZANTAZ 社に送信されたデータには、送信者、タイトル、受信時刻等の情報が添付され、CD-R や磁気ディスクに記録される仕組みとなっており、時刻情報を管理する手段として Digital Notary Service が利用されている[33]。

(2)Firstuse.com（Firstuse.com 社）と e-TimeStamp（DigiStamp 社）

Firstuse.com 社の Firstuse.com と DigiStamp 社の e-TimeStamp は、いずれもインターネットを利用した 24 時間利用可能なタイムスタンプサービスであり、1998 年にサービスが開始されている[34]。これらのサービスでは、タイムスタンプの生成方法として Simple Protocol が採用されている。なお、Firstuse.com と e-TimeStamp のシステムの内容はほぼ同一であるため、ここでは Firstuse.com について説明する。

Firstuse.com におけるタイムスタンプの生成手順は以下の通り（次頁の図 20 参照）。

---

[32] NetDox 社およびサービス内容に関する情報については、http://www.netdox.com/を参照。
[33] ZANTAZ 社およびサービス内容に関する情報については、http://www.zantaz.com/を参照。
[34] Firstuse.com 社およびサービス内容に関する情報については、http://www.firstuse.com/を参照。また、DigiStamp 社およびサービス内容に関する情報については、http://www.e-timestamp.com/を参照。なお、これらのサービスの利用状況に関する情報は、これらの Website には掲載されていない。

(A)利用者は、予めパソコンに専用ソフト Firstuse Direct をインストール。サービス利用時には、タイムスタンプの対象データをハッシュ化して digital fingerprint（ハッシュ値）を生成。

(B)利用者は、インターネット経由で Firstuse.com の TSA サーバーに digital fingerprint を送付。

(C)TSA サーバーは、digital fingerprint、時刻情報、利用者名、住所等を含む Registration Certificate（タイムスタンプ）を生成[35]。

(D)TSA サーバーは、Registration Certificate を自社のデータベースに保管するとともに、利用者に送付。

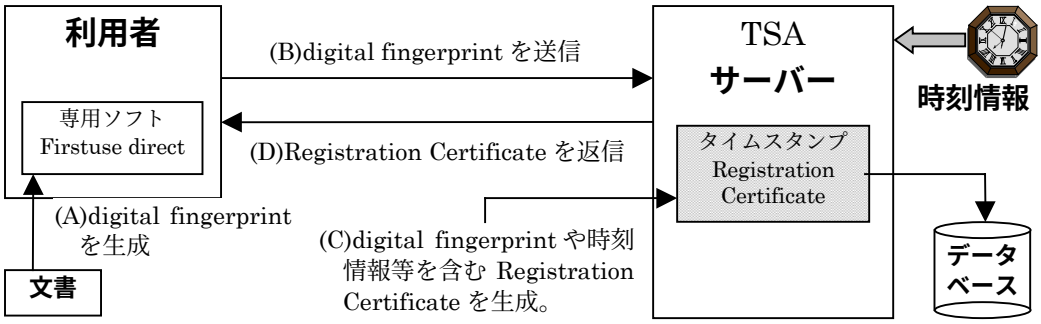


図 20　Firstuse.com のタイムスタンプ生成手順

Registration Certificate の検証は専用ソフト Firstuse direct において実行される。その手順は、(A)Registraton Certificate の真正性を確認、(B)対象データのハッシュ値と Registration Certificate に含まれている digital fingerprint を照合、(C)Registration Certificate の digital fingerprint と Firstuse.com が管理しているデータベースの digital fingerprint を照合、という 3 段階から構成される。

(3)Stamper（I.T. Consultancy 社）

Stamper は、PGP[36]を利用したタイムスタンプサービスであり、英国の I.T. Consultancy 社によって 1995 年に開始された。Stamper では、タイムスタンプの生成方法として Simple Protocol を採用している[37]。生成されたタイムスタ

---

[35] Firstuse.com 社のホームページの情報によると、Registration Certificate には改ざんを検出する仕組みが採用されていると記載されている。ただし、具体的な改ざん検出の方法について記述されていない。

[36] PGP（Pretty Good Privacy）：暗号化によるデータ守秘機能や、デジタル署名によるメッセージ認証機能を有する暗号化電子メールのフリーソフトウェア。PGP のメッセージ形式が RFC 1991 に記載されている。

[37] Stamper の内容については、http://www.itconsult.co.uk/stamper/stampinf.htm を参照。

ンプ（Certificate と呼ばれている）はインターネット上に公開され、既に公開された Certificate を改ざんすることを困難にする仕組みを採用している。

Stamper には、タイムスタンプに関連する 6 種類のサービスタイプが準備されており、通常のタイムスタンプサービスに対応する PGP モードのほか、電子メールの配達記録証明サービスとして POST モードが用意されている。利用者は、Stamper サーバーに送付する電子メールの送付先を変更することによって、これらのサービスを選択することができる。

PGP モードは、デジタル署名が添付された電子メールに対するタイムスタンプを生成するサービスである。具体的なタイムスタンプの生成・検証方法は以下の通り（図 21 参照）。

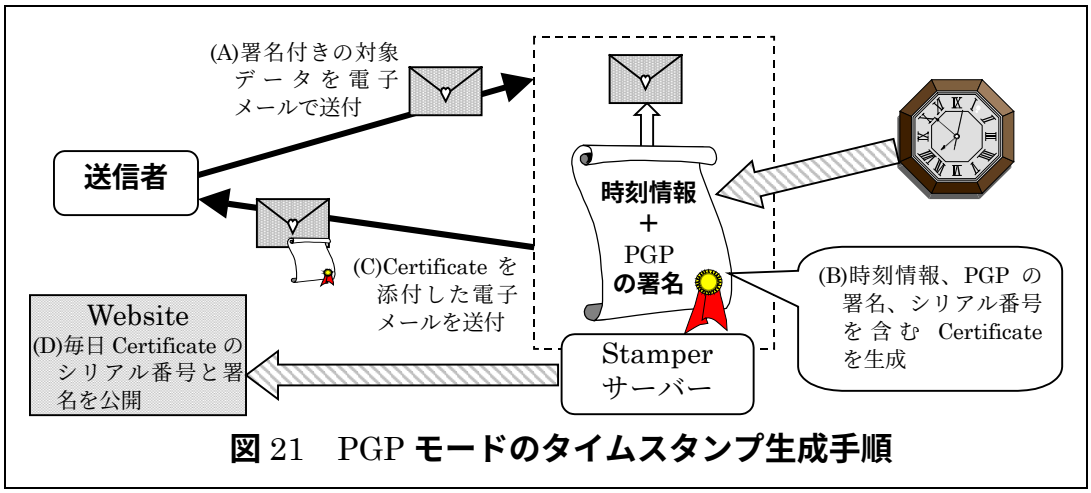


図 21　PGP モードのタイムスタンプ生成手順

(A)利用者は、タイムスタンプの対象となるデータとそれに対するデジタル署名を PGP を利用した電子メールで Stamper サーバーに送付（E-mail アドレスは pgp@ stamper. itconsult.co.uk）。

(B)Stamper サーバーは、受信したデータとデジタル署名に時刻情報、シリアル番号を添付し、これに対するデジタル署名を生成。この署名、シリアル番号、時刻情報を含む Certificate（タイムスタンプに相当）を生成。

(C)Stamper サーバーは、利用者から送付されたデータと Certificate を利用者に送付。同時に、I.T. Consultancy 社の Website 上にシリアル番号と署名を含むファイル（内容には Stamper サーバーの署名が添付される）を掲載し、だれでも自由に閲覧することが可能。ファイルは毎日更新される。

Stamper サーバーによって生成された Certificate の真正性は、そのデータの所有者のデジタル署名と Stamper サーバーのデジタル署名によって確保される。Certificate に含まれるデジタル署名の検証が成功すれば、データの受信時刻の正当性が証明される。

このほか、Stamper のタイムスタンプシステムでは、利用者がデジタル署名付きのデータを電子メールで Stamper サーバーに送信することから、タイムスタンプの要求情報を電子メールで送信した時点以降、要求情報が改ざんされていないことを確認することができる。

Ⅳ. デジタルタイムスタンプ技術の標準化動向

デジタルタイムスタンプ技術は、現在様々な標準化団体によって標準規格の策定作業が進められている。最初に、主な標準化の動向を整理すると以下の表 12 の通り。

表 12　デジタルタイムスタンプ技術に関連する主な標準化の動向

標準化団体	タイトル	ステイタス
IETF PKIX	Time Stamp Protocols	策定中
ISO/IEC JTC1/SC27	ISO/IEC 13888: Non-repudiation	標準化完了
	ISO/IEC WD 18014: Time Stamping Service	策定中
	ISO/IEC PDTR 14516: Guidelines for the use and management of Trusted Third Party	策定中
	ISO/IEC CD 15945: Specification of TTP services to support the application of digital signatures	策定中

1. IETF PKIX におけるタイムスタンププロトコルの標準化

インターネットにおける公開鍵インフラの関連技術の標準化を行う IETF PKIX では、デジタルタイムスタンプのプロトコルに関する Internet Draft[38] として Internet X.509 Public Key Infrastructure Time Stamp Protocols（PKIX-TSP, Adams et al.[1999]）の検討が進められている。PKIX-TSP は、タイムスタンプシステムの利用者であるクライアントと TSA 間のメッセージ形式と、TSA と TDA 間のメッセージ形式を規定するものである。PKIX-TSP が想定するタイムスタンプシステムは、Simple Protocol がベースとなっているとみられるものの、Linking Protocol 等様々な形態のシステムが利用可能となるように、メッセージ形式には機能拡張を担うフィールドが設定されている。

(1)TSA の性質

PKIX-TSP では、TSA に要求される性質が規定されており、主に以下の 8 項目に集約される。

①信頼できる時刻を提供する。
②タイムスタンプの発行先であるクライアントを特定する ID 情報を入手しない。
③クライアントから有効なタイムスタンプの要求情報を受信した場合、早急に

[38] Internet Draft：IETF の working group 等から標準の原案として提案される文書。Internet Draft は、working group 内での議論を経た後に、標準化の運営を担当する IESG（Internet Engineering Steering Group、IETF の下部組織）において審議され、承認されると標準化提案（Proposed Standard）として正式に標準化プロセスに組込まれることとなる。なお、標準化提案となった文書は、様々なテストや審議を経て、標準草案（Draft Standard）、そして標準（Standard）となる。

タイムスタンプを生成する。
④タイムスタンプをデータに直接付与するのではなく、データのハッシュ値に付与する。
⑤利用されるハッシュ関数が十分な安全性を有しているか否かを判断する。
⑥タイムスタンプが付与されるデータを一切吟味しない。
⑦デジタル署名の生成鍵は本プロトコル専用のものを利用し、公開鍵証明書にその旨を記載する。
⑧クライアントから要求があれば、TDA から TD を入手してタイムスタンプに含める。ただし、TD を利用不可能な場合にはエラーメッセージを返信する。

(2)TDA の性質

TDA に要求される性質として、以下の 4 項目が規定されている。

①業務内容は正確な TD を提供することのみである。
②タイムスタンプが付与されるデータを一切吟味しない。
③TSA から有効な TD の要求情報を受信した場合、早急に TD を生成する。
④デジタル署名の生成鍵は本プロトコル専用のものを利用し、公開鍵証明書にその旨を記載する。

(3)PKIX-TSP の概略

PKIX-TSP のベースとなっているのは Simple Protocol である。ただし、オプションとして、TDA が生成する TD をタイムスタンプに含めることが可能となっている（図 22 参照）。

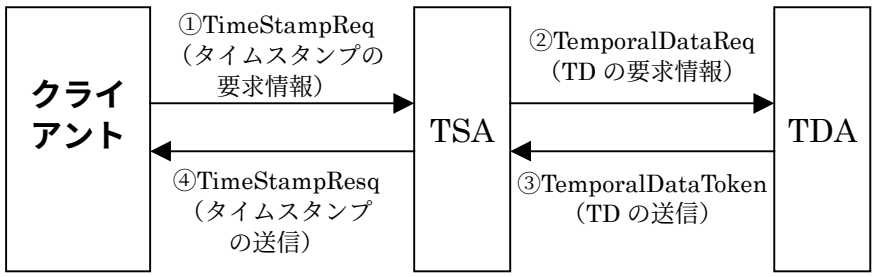


図 22 PKIX-TSP のプロトコルの概略

本プロトコルでは、以下の 4 種類のメッセージが送受信される。

①TimeStampReq

TimeStampReq は、クライアントから TSA に対して、タイムスタンプの生成を要求するためのメッセージである。タイムスタンプの対象となるデータのハッシュ値、利用されるハッシュ関数の識別情報、PKIX-TSP のバージョン

情報[39]、タイムスタンプの生成方法に関するポリシー情報[40]、TD をタイムスタンプに含める場合には TDA を指定する情報等から構成される。

②TemporalDataReq

TemporalDataReq は、TimeStampReq に TDA を指定される情報が含まれていた場合に、TSA から TDA に対して送信されるメッセージである。TemporalDataReq は、ハッシュ値、利用されるハッシュ関数の識別情報、PKIX-TSP のバージョン情報等から構成される。

③TemporalDataToken

TemporalDataToken は、TDA が TSA に対して送信されるメッセージであり、TD のほかに、TDA のデジタル署名、利用されるハッシュ関数の識別情報、TDA の公開鍵証明書（TDA のデジタル署名が含まれる場合に利用される）、シリアル番号等が含まれる。

④TimeStampResp

TimeStampResp は、TSA がクライアントに対して送信するメッセージであり、時刻情報、TSA のデジタル署名、ハッシュ値、利用されるハッシュ関数の識別情報、TSA の公開鍵証明書、公開鍵証明書失効リスト（CRL：Certificate Revocation List）、TD 等から構成される。

さらに、これらのメッセージには上記以外の様々なフィールドがオプションとして設定されており、多様な形態のタイムスタンプシステムへの拡張性が確保されている点が特徴である[41]。タイムスタンプの検証は、TimeStampResp に含まれる TSA のデジタル署名を検証することによって実行される。

(4)標準化と特許問題

現在、PKIX-TSP は Internet Draft となっており、今後の標準化の進め方について PKIX において検討が行われているが、PKIX-TSP に規定されているデジタルタイムスタンプシステムが既存の特許に抵触する可能性があることが指

---

[39]1999 年 7 月 11 日現在の PKIX-TSP の最新バージョンは Version 1 であり、バージョン情報のフィールドには 0 が入力される。

[40] ポリシー情報は、TSA 等におけるログの取り扱いや時刻情報の制度等を表すものであり、どのようなポリシーが選択可能かに関しては RFC 2459 において規定される。

[41]本プロトコルでは 3 者間でやり取りされるメッセージが規定されているのみであり、TSA 内部での処理内容については規定されていない。このため、Simple Protocol 以外の方式を実装する場合、新たに TSA 内部での処理内容を設定した上で、通信オプションを利用する。例えば Linking Protocol の場合には、TSA 内部で Link 情報を生成する仕組みを新たに設定し、TimeStampResp の中に設けられている拡張フィールド TsaFreeData に Link 情報等を入力して、Linking Protocol に必要なデータ交信を行うといった実装が可能となる。

摘されている。PKIX-TSP の Draft には既存のタイムスタンプ関連特許として 7 つの特許が記載されており、「本プロトコルを実装する場合には、自ら関連特許を調査・分析し、抵触する可能性のある特許が存在しないか否かを確認することが望ましい」と記載されている。もっとも、各国の関連特許を検索して各特許の請求項の請求範囲を分析し、具体的な実装方法との関連性について判断を下すことは容易ではない。このため、PKIX-TSP の標準化が完了したとしても、関連特許に抵触する可能性が残されている場合には、標準として利用できない可能性もある。

2. ISO/IEC JTC1/SC27 における国際標準化動向

ISO/IEC JTC1/SC27 では、デジタルタイムスタンプ技術に関連する国際標準および国際標準案がいくつか存在する。既存の国際標準としては、デジタルデータの生成・交信における否認防止サービスに関する国際標準 ISO/IEC 13888 が挙げられる。また、タイムスタンプサービスのプロトコルや TTP のサービス内容を規定した規格案が提案されており、現在検討が進められている。

(1)ISO/IEC 13888 の概要

ISO/IEC 13888（Information technology — Security techniques — Non-repudiation）は、Part 1：概要（General)、Part 2：共通鍵暗号技術を利用した方式（Using symmetric techniques)、Part 3：公開鍵暗号技術を利用した方式（Using asymmetric techniques）から構成されている（ISO/IEC[1997a], [1997b] and [1998])。

①パート 1

パート 1 には、否認防止サービスの目標について、「ある事象や行為が発生したか否かに関する争いを解決するために、そうした事象あるいは行為に関する証拠となるデータを生成、収集、管理、検証することである」と規定されている。その上で、「証拠となるデータには、TSA が生成したタイムスタンプが含まれる必要がある」と規定されており、否認防止サービスを実現する要素技術としてデジタルタイムスタンプ技術が位置付けられている。

さらに、本パートでは、否認防止サービスの種類として、データ中継機関を利用する場合と利用しない場合において、それぞれデータの作成・送信事実の否認防止サービスとデータの受信事実の否認防止サービスが規定されている。また、否認防止サービスの一般的モデルとして、ある事象あるいは行為に関する証拠となるデータの生成・検証の手続きが規定されている。証拠となるデータのフォーマットとしては、Generic Non-repudiation Token と Time Stamping Token の 2 種類が規定されており、いずれも時刻情報とデータに対する TSA

のデジタル署名が組み込まれている。

②パート 2 およびパート 3

パート 2 およびパート 3 には、否認防止サービスにおける 3 種類のプロトコルと、それぞれのプロトコルにおいて交信されるデータの具体的なフォーマットが規定されている。パート 2 では、共通鍵暗号技術によるメッセージ認証コード（MAC： Message Authentication Code）を利用した方式が規定されているほか、パート 3 においては、公開鍵暗号技術によるデジタル署名を利用した方式が規定されている。

(2)現在検討されている標準規格案

ISO/IEC 13888 は否認防止サービスの構成要素としてデジタルタイムスタンプ技術を取り扱っているが、現在 ISO/IEC JTC1/SC27 では、デジタルタイムスタンプサービスに関連する 3 つの標準規格案①ISO/IEC WD 18014（Time Stamping Service）、②ISO/IEC PDTR 14516（Guidelines for the use and management of Trusted Third Party）、③ISO/IEC CD 15945（Specification of TTP services to support the application of digital signatures）が検討されている。

まず ISO/IEC WD 18014 は、タイムスタンプサービスの内容を規定する標準規格案であり、TSA の機能を規定するとともに、タイムスタンプシステムの基本的なプロトコルを定義している。

また、ISO/IEC PDTR 14516 は、TTP の業務内容に関するガイドライン案であり、TTP のサービスの提供形態（on-line や off-line）について解説した上で、タイムスタンプサービス、否認防止サービス、鍵管理サービス、公開鍵証明書管理サービス等の業務内容や業務を行う上での留意点を解説している。

ISO/IEC CD 15945 は、否認防止を目的としてデジタル署名を活用する際に、TTP に求められる役割を規定するものである。具体的には、TTP の役割として公開鍵証明書の管理サービスや鍵管理サービスについて解説した上で、これらのサービスを実施するために必要となる公開鍵証明書廃棄リスト（CRL： Certificate Revocation List）や公開鍵証明書管理メッセージのデータ構造等について規定している。

Ⅴ. デジタルタイムスタンプ技術の主な関連特許

今後、デジタルタイムスタンプ技術を利用する際には、既存の関連特許との兼ね合いをどのように考えるかが重要となる。PKIX-TSP の Draft に記載されているデジタルタイムスタンプ技術関連特許[42]等を参考に、日本で出願されている関連特許を調査した結果、表 13 の 9 件の特許が出願されていることが判明した。

表 13　デジタルタイムスタンプシステム関連特許

特許名	日本における特許 番号・出願日	米国における特許 番号・出願日	出願者
①デジタル時間認証装置	特願平 2-260322 平成 2 年 9 月 27 日	Patent No.5001752 平成 2 年 12 月 20 日	Addison M. Fischer
②電子的公証方法および装置	特願平 3-160135 平成 3 年 6 月 3 日	Patent No. 5022080 平成 2 年 4 月 16 日	Pitney Bowes 社
③数値文書にタイムスタンプを確実に押す方法	特願平 3-516026 平成 3 年 7 月 30 日	・Patent No. 5136647 平成 2 年 8 月 2 日 ・ Patent No. 5136646 平成 3 年 3 月 8 日	Bellcore 社
④暗号証書の有効性延長法	特願平 6-515149 平成 5 年 11 月 17 日	Patent No. 5373561 平成 4 年 12 月 21 日	Bellcore 社
⑤個人用日時認証装置	特願平 6-88526 平成 6 年 4 月 26 日	Patent No. 5422953 平成 5 年 5 月 5 日	Addison M. Fischer
⑥ドキュメントをユニークに特定し認証する認証書を発行するデジタルドキュメント証明システム	特願平 8-514727 平成 7 年 10 月 25 日	Patent No. 5781629 平成 6 年 10 月 28 日	Surety 社
⑦電子情報への確定日付付与法	特願平 8-222994 平成 8 年 7 月 23 日	―――	太田暉人
⑧タイムスタンプサーバシステム	特願平 8-253600 平成 8 年 9 月 25 日	―――	日立ソフトウェアエンジニアリング社
⑨電子文書の存在証明方法	特願平 9-11267 平成 9 年 1 月 24 日	―――	NTT

これらの特許の概要および主な請求項を整理すると以下の通り。

1. 「デジタル時間認証装置」特許出願

出願番号（出願日）	特願平 2-260322（平成 2 年 9 月 27 日）
米国特許番号（出願日）	5,001,752（平成 2 年 12 月 10 日）
出願人	Addison M. Fischer
審査状況[43]	審査中
請求項数	16

[42] PKIX-TSP の Draft には、表 12 の①～⑤の 5 件の米国特許を含め、合計 7 件の米国特許が記載されている。

[43] ここでの審査状況は、日本で出願された特許に関するものであり、各特許が公開された時点での状況であって、必ずしも現時点での審査状況を表すものではない。以下、8 つの特許出願における審査状況も、同様の意味で記載されている

本特許は、デジタル署名を利用したタイムスタンプを生成する装置を特定している。本特許の発明について、「発明の詳細な説明」には以下のように記載されている。

「この発明は総括的には電子的に転送されるデジタル文書をデジタルでタイムスタンプする装置および方法に関する。より特定的には、この発明は公衆キーの日付・時間認証設備としての役割を果たす方法と装置に関する。」

本特許において最も請求範囲が広いとみられるのは請求項 1 であり、その内容は以下の通り（明細書に記載されているデジタル時間認証装置の実施例は、図 23 を参照）。

[請求項 1]

「デジタル時間認証装置であって、コンポーネントを支持するためのプラットホーム手段と、前記プラットホーム手段により支持され、時間を示すクロック信号を用して前記クロック信号および入力値に関して動作しかつ認証タイムスタンプを発生するためのプロセッサ手段とを含む、装置。」

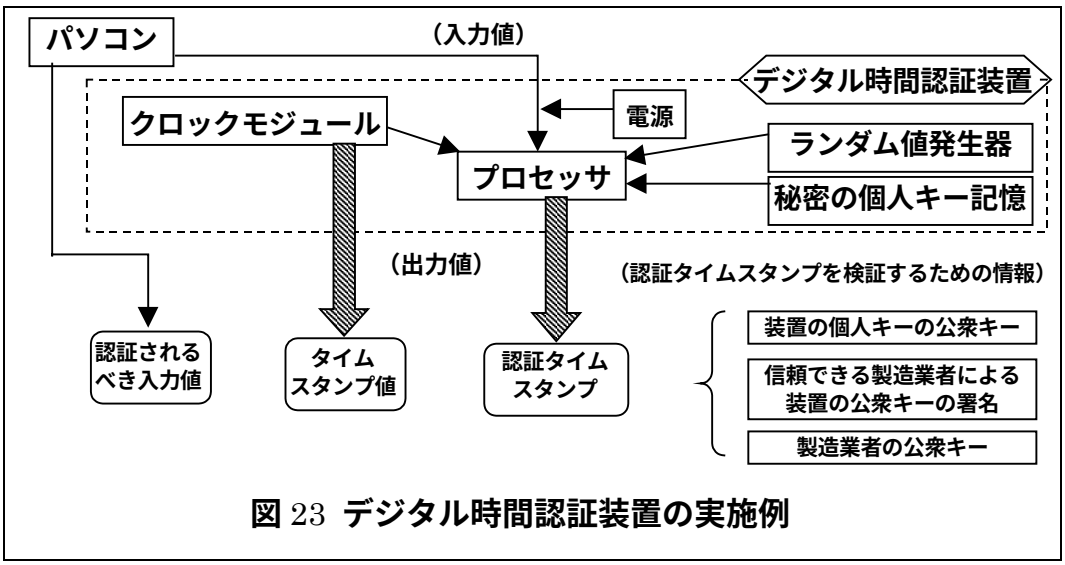


図 23 デジタル時間認証装置の実施例

このように、請求項 1 は、クロック手段を内蔵し、入力（ハッシュ値である場合を含む）に対して公開鍵暗号により内蔵プロセッサで認証タイムスタンプ生成を行うデジタル時間認証装置を請求するものである。

請求項 1 の従属項として請求項 2～16 が記載されている。請求項 2 は、ランダム値を生成する手段を加えたデジタル時間認証装置を特定しているほか、請求項 5 は、クロック手段の信頼性を高めるために複数のクロック手段を含むデジタル時間認証装置を特定している。

2. 「電子的公証方法および装置」特許出願

出願番号（出願日）	特願平 3-160135（平成 3 年 6 月 3 日）
米国特許番号（出願日）	5,022,080（平成 2 年 4 月 16 日）
出願人	Pitney Bowes 社
審査状況	拒絶
請求項数	24

本特許は、タイムスタンプである時刻認証証明書の発行方法および検証方法を特定するとともに、それらを実現する装置を特定している。本特許の発明について、「本特許の詳細な説明」には、まず「本発明の目的は、電子的に記録された文書または電子的に送られた文書およびその他のデータを電子的に公証する装置および方法を提供することである」と記載されており、さらに別の目的として、以下の 2 点が記載されている。

①本目的の別の目的は、文書またはその他の電子的に記録されたデータまたは電子的に送られたデータがその文書に関連する日時を後で変更されなかったことを確認するために、それらの文書またはデータ真正であることを確認する装置および方法を提供することである。
②本発明の更に別の目的は、電子的様態で記録または送られた文書の作成日と、最後の変更日と、内容との少なくとも 1 つが変更されなかったことを確認するために、そのような文書が真正なものであることを判定する暗号作成方法および装置を提供することである。

本特許における 24 の請求項のうち、請求項 1 および 14 が独立項となっており、それら以外は請求項 1 もしくは 14 の従属項となっている。最も請求範囲が広いとみられる請求項は、電子公証方法を特定する請求項 1 であり、これに対応する装置の請求項が請求項 14 である。請求項 1 は以下の通り。

[請求項 1]

「第 1 のデータユニットを供給するステップと、第 1 のデータユニットの情報内容を表す第 2 のデータユニットを第 1 のデータユニットから発生するステップと、ある時点を指定する第 1 の時刻指示を、第 1 の当事者により変更できない時刻指示を有する時刻発生手段により発生するステップと、第 3 のデータユニットを発生するために第 2 のデータユニットと、発生された第 1 の時刻指示とを暗号化するステップとを有する、指定された時点の後で修正されなかった第 1 の当事者に関連する第 1 のデータユニットを決定する電子的公証方法。」

請求項 1 の従属項である請求項 2、3、4 は、請求項 1 における第 2 のデータユニットを、それぞれ CRC 発生手順、パリティ発生手順、検査合計発生手順によって発生する電子的公証方法を特定している。請求項 5 は、請求項 1 の方法において、第 1 の時刻指示として現在の時刻を発生する方法を特定しているほか、請求項 10 は、請求項 1 における暗号化ステップが、公開キー暗号化、個人キー暗号

化、およびそれらの組み合わせた暗号化手順で行われる方法が特定されている。

3. 「数値文書にタイムスタンプを確実に押す方法」特許出願

出願番号（出願日）	特願平 3-516026（平成 3 年 7 月 30 日）
米国特許番号（出願日）	5,136,647（平成 2 年 8 月 2 日） 5,136,646（生成 3 年 3 月 8 日）
出願人	Bellcore 社
審査状況	審査中
請求項数	15

本特許は、デジタル文書と時刻情報の組み合わせに対して、第三者機関によるデジタル署名を用いたタイムスタンプサービスの方法を特定するものである。本特許の発明について、「発明の概要」には以下のように記載されている。

「この発明は数値文書をタイムスタンプする方法において信頼できるシステムを作り、現在の記録情報の本質的な特徴の二つと同等のものを提供します。
　第一に、文書の内容とその存在のタイムスタンプは、文書の数値データに消えないように組込まれ、これによってできたタイムスタンプされたデータのいかなる部分も、改変が明確とならないように改変することは不可能であります。このように、文書のテキストの状態は、タイムスタンプの瞬間に確定されます。
　第二に、数値文書がスタンプされた時期は、虚偽の時刻の表現を組込むことを防ぐ、数値的に「証人として」署名する手順で確認されます。」

本特許における請求項のうち、最も請求範囲が広いとみられるのが請求項 1 である。請求項 1 の内容は以下の通りであり、タイムスタンプを生成する手順を特定している。

[請求項 1]

「(a)数値文書の数値表示が創作者から外部機関へ送られ、
(b)この外部機関がこの数値文書の数値表示の少なくとも一部分とその時の時刻の数値表示とを包含する受理書を作り、
(c)この受理書がこの外部機関によって証明できる数値暗号署名法によって証明されることを特徴とする数値文書にタイムスタンプを確実に押す方法。」

請求項 1 の従属項である請求項 3 は、請求項 1 の方法において一方向性ハッシュ関数を利用する方法を特定しているほか、請求項 5 は、請求項 1 において外部機関を無作為に複数選ぶ方法を特定している。

また、請求項 9 は、タイムスタンプの対象となるデータを相互に関連付ける情報を利用したタイムスタンプの方法を特定している。その内容は以下の通り。

[請求項 9]

「a 一つのシリーズの文書の特定の一つの数値表示を作り、
b 前記特定文書表示と前記シリーズ中の前記特定文書の直前の文書に対する証明書記載連鎖値表示を包含する連鎖に対して決定関数法を適用して前記特定文書に対する証明書記載連鎖値表示を作る
ことを特徴とする一つのシリーズの数値文書の時間的順序を証明する方法。」

本請求項における連鎖値は Linking Protocol における Link 情報に対応するとも考えることができる。

4. 「暗号証書の有効性延長法」特許出願

出願番号（出願日）	特願平 6-515149（平成 5 年 11 月 19 日）
米国特許番号（出願日）	5,373,561（平成 4 年 12 月 21 日）
出願人	Bellcore 社
審査状況	未請求
請求項数	20

本特許は、過去に作成された暗号証書（タイムスタンプ）の有効期限を、その有効期間内に別の暗号機能を使って延長する方法や、その方法によってディジタル文書に対する暗号証書の作成・検証を行う方法を特定している。本特許の発明について、「発明の分野」では次のように記載されている。

「本発明は、記録された文書または事象の存在または発生を証明または有効化する方法で、特に暗号化の前提に依存してそのような証明または有効性の基盤を確立する方法に関する。
さらに詳しくは、本発明は、オリジナル証書を再確認する方法で、オリジナル証書手順における暗号化の前提またはステップの蓋然的な不履行の点を越える相当長い期間で有効性を維持する方法に関する。」

本特許には 20 の請求項が存在し、請求項 1 は、このうちタイムスタンプの有効期限を延長する方法を特定している。

[請求項 1]

「ディジタル文書に第１暗号機能を応用することによって引き出される第１暗号証書の有効性を延長する方法で、下記のことからなるもの：
(a)該文書のディジタル表現と該証書のディジタル表現を組み合わせること；
および
(b)該第１証書の有効期間中に上述の組み合わせに対して異なる暗号機能を応用し、該第１証書のその時点の有効性を確認する第２証書を生成すること。」

請求項 1 の従属項である請求項 2 は、第 1 暗号機能としてディジタル署名を利用する方法を特定しているほか、請求項 4 は、第 1 暗号機能として一方向性ハッ

シングアルゴリズムを利用する方法を特定している。

一方、請求項 10 は、暗号証書の有効期限を延長して生成される新しい暗号証書の生成方法を特定している。その内容は以下の通り。

[請求項 10]

「文書のディジタル表現を確認する方法で、下記のことからなるもの：
(a)少なくとも第１暗号機能を該ディジタル表現に応用し、第１証書を生成すること；
(b)該第１証書と該ディジタル表現を組み合わせること；および
(c)該第１機能と異なる少なくとも一つの暗号機能を該組み合わせに応用することによって第 2 証書を作成すること。」

5. 「個人用日時認証装置」特許出願

出願番号（出願日）	特願平 6-88526（平成 6 年 4 月 26 日）
米国特許番号（出願日）	5,422,953（平成 5 年 5 月 5 日）
出願人	Addison M. Fischer
審査状況	未請求
請求項数	19

本特許は、デジタル署名を生成する場合に、常にデジタル署名にタイムスタンプを付与することが可能な機能を備えた装置や手段を特定している。本特許明細書における「課題を解決するための手段、作用および発明の効果」には、以下が記載されている。

「本発明は、効果的に、デジタル時刻認証をデジタル署名操作と組み合せ、タイムスタンプがいつでも自動的に確実に生じるようにしている。ユーザは、タイムスタンプが必要かどうかに関して、いかなる追加の意思決定をする必要がない。別個のタイムスタンプ認証装置の必要性をなくすことによって、ユーザは、時間、経費および労力を節約できる。」

本特許では、請求項 1 がタイムスタンプとデジタル署名を生成する装置を特定している。その内容は以下の通り。

[請求項 1]

「ユーザが携行する携帯用媒体上に作成されるトークン装置であって、
前記ユーザに関連するデジタル署名を行うために用いられる機密性を有する個人鍵を少なくとも１つ記憶する記憶装置と、日時表示を提供するためのクロックと、デジタル的に署名されるべき値を受信し、出力するための通信ポートと、前記通信ポートおよび前記クロックと結合され、デジタル的に署名されるべき前記値および前記日時表示を受信し、前記通信ポートに出力するために前記少なくとも１つの個人鍵で少なくとも１つのデジタル署名を行うための処理装置とを備える、トークン装置。」

請求項 1 の従属項である請求項 2 は、ランダム数発生器を備えたトークン装置を特定しているほか、請求項 3 は、追加のクロックを含む装置を特定している。さらに請求項 4 は、ユーザの個人識別パスワード（PIN：Personal Identification Number）を入力し、確認する手段を含む装置を特定している。これらの機能を備えたトークン装置の構成は以下の図 24 の通り。

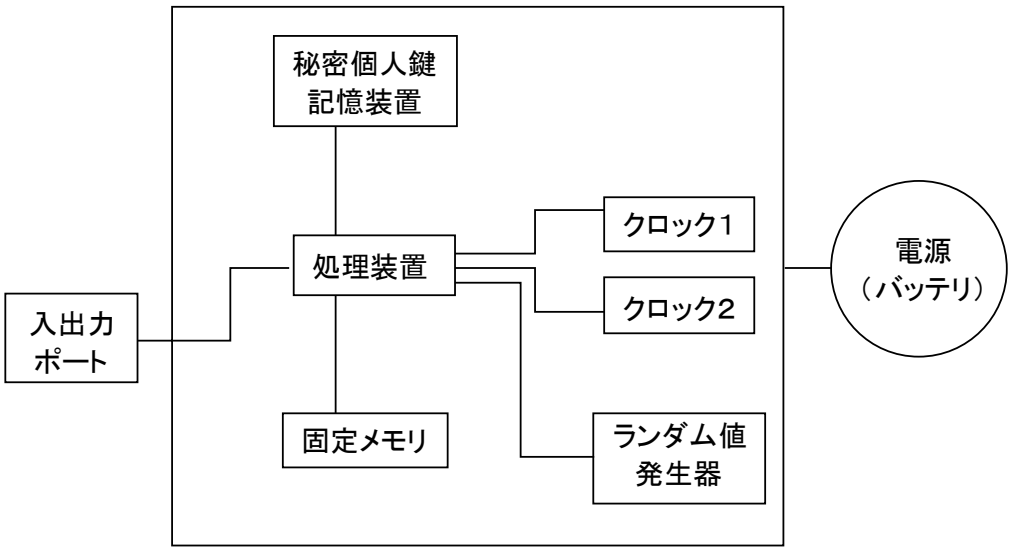


図 24 トークン装置の構成

一方、請求項 7 はトークン装置の操作方法を特定しており、内容は以下の通り。

[請求項 7]

「ユーザトークン装置を操作する方法であって、
(a)デジタル的に署名されるべきデジタル値を受信するステップと、
(b)信頼性のあるクロック源から現在時刻を決定するステップと、
(c)現在時刻と、署名されるべき情報から抽出された値とを含むデジタルデータ構造を作成するステップと、
(d)デジタルデータの前記構造にデジタル的に署名するステップとを備える、操作方法。」

請求項 7 の従属項である請求項 8 は、請求項 7 の操作方法において、個人鍵のうちの少なくとも１つは公開鍵暗号方式における秘密鍵である操作方法を特定しているほか、請求項 10 は、公開鍵証明書が装置内に記憶されている場合における操作方法を特定している。

一方、請求項 14 は、時刻情報を生成する日時認証装置とトークン装置を接続して、タイムスタンプとデジタル署名を生成するシステムと特定している。その内

容は以下の通り。

[請求項 14]

「ユーザトークン装置を基礎にしたシステムであって、前記ユーザに関連するデジタル署名を行うために用いられる機密性を有する個人鍵を少なくとも１つ記憶する記憶手段と、入力デジタル信号を受信し、デジタル出力を発信するための通信手段と、前記入力デジタル信号を受信し、前記の少なくとも１つの個人鍵でデジタル署名を行うための処理手段と信頼性のある日時認証装置によって作成された日時に関する信号と前記処理手段とを結合するための手段とを備える、ユーザトークン装置システム。」

6. 「ドキュメントをユニークに特定し認証する証明書を発行するデジタルドキュメント証明システム」特許出願

出願番号（出願日）	特願平 8-514727（平成 7 年 10 月 25 日）
米国特許番号（出願日）	5,781,629（平成 6 年 10 月 28 日）
出願人	Surety 社
審査状況	未請求
請求項数	60

本特許は、各ドキュメントを保管する際に、そのドキュメントのハッシュ値を生成して各ドキュメントを識別する情報とし、さらに保管される情報の一部を公開することによって保管されているデータの真正性を証明するデジタルタイムスタンプシステムを特定している。本特許の「発明の概要」には、以下のように記載されている。

「本発明は、改良したデジタルタイムスタンプシステムを提供することを目的とする。（中略）また、ドキュメントの名前に関連して情報を提供することができ、本来的に検証することができ、言及しやすく、短く、ユーザフレンドリーな名前を割り当てることができる、名前付けシステムを提供することを目的とする。」

請求項 1 は、タイムスタンプを生成するためにデジタルドキュメントを登録・保管する方法が特定されている。その内容は以下の通り。

[請求項 1]

「認証のために第１のデジタルドキュメントを登録する方法であって、以下のステップを含む、
(a)第１のドキュメントの登録のリクエストを受け取り、
(b)他の複数のリクエストを受け取り、結合して、ハッシュし、複数のリクエストに応じた複数のハッシュ値の保管庫を形成し、
(c)第１ドキュメントに対する、保管庫中のロケーションポインタを生成し、
(d)ロケーションポインタによって前記第１のドキュメントの証明書を生成する。」

請求項 1 の従属項である請求項 7 は、保管庫におけるドキュメントの相互関係の形態が「木」のようになる場合、木の「葉」に対応する部分に登録するデジタルドキュメントが位置し、木の「根」に対応する部分のデータが公開される値となる方法を特定している。本請求項には、Linking Protocol using Tree Structure（本稿のⅡ 2. (2)②(B)(イ)を参照）が該当すると考えることもできる。

請求項 25 は、タイムスタンプとなる証明書を生成・検証する方法を特定しており、葉の位置を表すロケーション値を各ドキュメントの名前とすることによって、各ドキュメントの名前で各ハッシュ値の正当性を検証する方法である。また、請求項 58 は、タイムスタンプの生成方法を特定している。

[請求項 25]

「デジタルドキュメントを認証する方法であって、以下のステップを含む、
(a)第１のドキュメントの自己検証する名前と自己検証ハッシュ値と、前記自己検証ハッシュ値の関連するロケーション値と、を含む証明書を生成することにより、第１のドキュメントを登録し、
(b)第１のドキュメントをハッシュし、第１のドキュメントのハッシュ値を供給し、
(c)第１のドキュメントのハッシュ値を、前記ロケーション値に従って前記自己検証ハッシュ値との、ハッシュと結合を行い、
(d)ステップ（c）の結果得られたハッシュ値が、保証されたハッシュ値に一致するかを決定する。」

[請求項 58]

「認証のために第１のデジタルドキュメントにタイムスタンプを施す方法であって、以下のステップを含む、
(a)第１のドキュメントの登録のリクエストを受け取り、
(b)他の複数のリクエストを受け取り、結合とハッシュを行う、複数のリクエストに応じた複数のハッシュ値の保管庫を形成し、
(c)保管庫の保証に用いる保管庫中の項目を特定し、
(d)前記特定した項目を含む、前記第１のドキュメントの証明書を生成し、リクエスト発行者に送り、
(e)ステップ(d)の後に、前記特定した項目を保証し、本方法によって、登録のリクエストに対するすばやい応答を、対応するすばやい公開時を必要とすることなしに行う。」

7. 「電子情報への確定的日付付与法」特許出願

出願番号（出願日）	特願平 8-222994（平成 8 年 7 月 23 日）
出願人	太田暉人
審査状況	未請求
請求項数	1

本特許は、電子ファイルにタイムスタンプを付与する方法を特定しており、「発明の目的」には、「電子ファイルに確定日付を付与すること」と記載されている。本特許の請求項は１つであり、その内容は以下の通り。

[請求項]

「電子計算機 1 を用いて、対象物である電子ファイル A から所定の方法で同定値 p を算出して、その値を通信システムを介してもう一つ別の計算機 2 に送り、当該計算機を用いて p を鍵にして日付を表す値 t を暗号化し、暗号化された日付 q を再び電子計算機 1 に送り返し、記憶媒体上で A と q とをリンクさせておくことによって、A に確定日付を付与する方法。」

8. 「タイムスタンプサーバシステム」特許出願

出願番号（出願日）	特願平 8-253600（平成 8 年 9 月 25 日）
出願人	日立ソフトエンジニアリング社
審査状況	審査中
請求項数	8

本特許は、デジタル署名を用いてタイムスタンプを生成するタイムスタンプサービスを実現するネットワークシステムを特定している。本特許明細書における「発明が解決しようとする課題」には、以下のような内容となっている。

「本特許の目的は、過去のある時点でコンピュータデータが既に存在したことを立証する証拠として用いることのできる情報の生成および使用することにあり、さらにメッセージの作成者／発信者／受信者、データを機密化し、第三者による情報の漏洩を防ぐことにある。」

本特許の中で、最も請求範囲が広いとみられるのが、タイムスタンプサーバシステムを特定する請求項 1 である。その内容は以下の通り。

[請求項 1]

「複数のクライアントが接続され、特定のサービスを提供するタイムスタンプサーバから成るネットワークシステムにおいて、クライアントのデータ送信に対して、タイムスタンプサーバは、データのメッセージダイジェストを生成するのに使用したアルゴリズムの識別子と付加的にパラメータを要求メッセージと返答メッセージ中のディジタル署名の対象データに含め、クライアントに返信することを特徴とするタイムスタンプサーバシステム。」

請求項 1 の従属項である請求項 2 は、請求項 1 の返信メッセージにメッセージダイジェストを含め、更に、ダイジェスト作成アルゴリズムの識別子またはパラメータ、もしくはクライアントの情報を暗号化するシステムを特定している。また、請求項 3 は、請求項 1 または 2 のシステムにおいて、時刻情報として要求メッセージの受け取り時刻、ディジタル署名作成時刻のいずれかを用いるシステムを特定している。

9. 「電子文書の存在証明方法」特許出願

出願番号（出願日）	特願平 9-11267（平成 9 年 1 月 24 日）
出願人	NTT
審査状況	未請求
請求項数	1

本特許は、TTP を利用したデジタル署名によるデジタルタイムスタンプの生成方法を特定している。本特許の請求項の内容は以下の通り。

[請求項]

「コンピュータネットワーク上での電子文書の送受信において送信者が受信者に対して双信した電子文書が受信者においてある時点で確かに存在したことを証明する電子文書の存在証明方法であって、受信者は、存在証明対象の文書の存在証明依頼申請書を作成し、この作成した申請書に電子署名を付与するとともに該文書の送信元の公開鍵証明証を添付して、信頼のおける第三者機関に送信し、第三者機関は、前記依頼申請書を受け付けると、該依頼申請書の署名検証を行い、公開鍵の認証機関が提供する鍵の無効化リストの更新および第三者機関が提供する文書の無効化リストの更新を一時停止し、前記鍵と文書が無効化リストに存在していないことを確認し、前記文書の存在証明を行うことを特徴とする電子文書の存在証明方法。」

VI. おわりに

デジタルタイムスタンプ技術は、電子公証の実現に不可欠な技術の１つとして、現在日本をはじめとする世界各国で実装に向けた検討が進められている。インターネットの急速な普及が続く中、こうしたデジタルタイムスタンプ技術に関する研究の進展は、今後、安全な電子商取引や電子文書管理の実現に寄与するものとみられる。

ただし、安全なデジタルタイムスタンプシステムを構築するためにクリアすべき課題がいくつか残されている。第一に、第Ⅱ章 2.(1)において整理したように、デジタルタイムスタンプシステムに不可欠な時刻情報生成技術、ネットワーク技術、暗号技術といった関連技術の研究や、これらの技術が十分に整備されていない場合でも、安全なタイムスタンプを生成することができるプロトコルについての研究を行う必要がある。

第二に、信頼できる TSA の実現に向けた研究が必要である。安全なタイムスタンプシステムを構築するためには、高い技術力を有し、システムの適正な運用・管理を実行する TSA が必要である。現在、ISO や IETF 等では、タイムスタンプサービスに関する標準や TTP のサービス・ガイドラインに関する技術文書の策定が進められているが、こうした標準化活動の成果も踏まえ、信頼できる TSA を構築するための技術的、業務的要件が確立され、利用者を含めて正確に理解されることが必要である。

第三に、様々なハイテク技術から構成されるデジタルタイムスタンプシステム全体の安全性評価に関する研究が必要である。タイムスタンプシステムは、暗号関連技術、時刻情報生成技術、ネットワーク技術等、複数の技術によって支えられており、これらのどこか一ヵ所に安全性上の問題が生じる可能性が残されている場合、システム全体として十分な安全性を達成することは困難となる。システム全体の安全性を評価するためには、利用環境に内在するリスクを十分吟味した上で、個々の要素技術を評価するとともに、それらのバランスや相互関係について検討する必要がある。

このように、安全なデジタルタイムスタンプシステムを実現するためには、暗号技術や通信技術をはじめとする関連分野における一層の研究・開発が必要であるが、金融業界にとっても、こうした新しい技術基盤が確立されることは、電子商取引への対応や金融システムの効率化・高度化を進める上で大切なことと考えられる。このため、金融業界としても、デジタルタイムスタンプ技術を巡る研究・実装動向や標準化動向について注目していく必要があろう。

以　上

参考文献

岩下直行・谷田部充子、「金融分野における情報セキュリティ技術の国際標準化動向」、『金融研究』第 18 巻第 2 号、pp.33-56、日本銀行金融研究所、1999 年 4 月（http://www.imes.boj.or.jp/japanese/kinyu.html）

岩本信正、「公証人法」、明文社、1981 年

宇根正志・岡本龍明、「公開鍵暗号の理論研究における最近の動向」、『金融研究』第 18 巻第 2 号、pp.195-251、日本銀行金融研究所、1999 年 4 月（http://www.imes.boj.or.jp/japanese/kinyu.html）

陣内勝・櫻井幸一、「分散機関の相対的時刻印を利用した絶対的時刻印生成プロトコル」、第 2 回コンピュータセキュリティシンポジウム発表論文、1999 年 10 月

谷口文一、「金融業界における PKI・電子認証技術について－ 技術面、標準化に関する最近の動向を中心に」、IMES Discussion Paper Series No. 99-J-30、日本銀行金融研究所、1999 年 8 月

丹波伸行・国分明男、「電子公証システムによるオープンマーケット等の創出のための実装実験」、1998 年（http://www.nmda.or.jp/nmda/ipa/kos/ipa-kos.html）

電子商取引実証推進協議会、「電子公証システムガイドライン（Ver. 1.0）」、1998 年 3 月（http://www.ecom.or.jp）

法務省民事局、「電子取引法制に関する研究会報告書」、1998 年 3 月

郵政省電気通信局、「21 世紀デジタル社会の暗号政策への提言 －暗号通信の在り方に関する研究会報告書－」、1999 年 6 月

Adams, C., P. Cain, D. Pinkas and R. Zuccherato, “Internet X.509 Public key Infrastructure Time Stamp Protocols (TSP),” June 1999. (ftp://ds.internic.net/internet-drafts/draft-ietf-pkix-time-stamp-02.txt)

Bayer, D., S. Haber and W. S. Stornetta, “Improving the efficiency and reliability of digital time-stamping,” R. M. Capocelli et al. Sequences II: Methods in Communication, Security and Computer Science, pp. 329-334, Springer-Verlag, 1993.

Dobbertin, H., A. Bosselaers and B. Preneel, “RIPEMD-160: A strengthened version of RIPEMD,” The Third Workshop of Fast Software Encryption, LNCS 1039, pp.71-82, Springer-Verlag, 1996.

Fabrica Nacional de Moneda y Timbre, “PKITS Overview Final Report,” December 23, 1998. (http://www.cordis.lu/infosec/src/winners.htm)

Haber, S., B. Kaliski and W. S. Stornetta, “How Do Digital Time-Stamps Support Digital Signatures?” CryptoBytes, Vol. 1, No. 3, pp.14-15, 1995. (http://www.rsa.com/rsalabs/)

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

———— and ————, “ISO/ IEC 13888 Information technology — Security techniques — Non-repudiation — Part 2: Using symmetric techniques,” 1998.

———— and ————, “ISO/ IEC 13888 Information technology — Security techniques — Non-repudiation — Part 3: Using asymmetric techniques,” 1997b.

Lipmaa, H., “Digital Signatures and Authentication,” June 28, 1999. (http://www.cyber.ee/infosecurity/resources/auth/)

Massias, H. and J.-J. Quisquater, “Time and cryptography,” TIMESEC Technical Report WP1, March 1997.

National Institute of Standards nad Technology, “Secure hash standard,” Federal Information Processing Standards Publication (FIPS PUB) 180-1, April 17, 1995.

Takura, A., S. Ono and S. Naito, “Secure and Trusted Time Stamping Authority,” Proceedings of IWS’99, pp.123-128.