デジタルタイムスタンプ技術の現状と課題

宇根正志（うねまさし）／松浦幹太（まつうらかんた）／田倉 昭（たくらあきら）

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の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、タイムスタンプに関する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の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うえで、最近の研究・実装動向、標準化動向や、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

本稿は、1999年11月1日に日本銀行で開催された「第２回情報セキュリティシンポジウム」への提出論文に加筆・修正を施したものである。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たちに属することはいうまでもない。

宇根正志　日本銀行金融研究所研究第２課（E-mail: masashi.une@boj.or.jp）
松浦幹太　東京大学生産技術研究所第三部（E-mail:kanta@iis.u-tokyo.ac.jp）
田倉　昭　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）

1．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いい難い。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。

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。

---

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

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の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２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うえで、デジタルタイムスタンプを分類し、各々の長所・短所や安全性について説明する。第３章では、わが国および海外における研究・開発プロジェクトや商用サービスを紹介する。第４章では、IETF PKIXやISO/IEC JTC1/SC27における標準化動向について説明し、第５章では、わが国における関連特許を紹介する。

なお、第２、３、４、５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。表の一覧は以下のとおり。

章	節	表番号	タイトル
2	(2)	表１	デジタルタイムスタンプの分類方法
		表２	３種類のプロトコルの主な特徴点
		表３	主なLinking Protocolの形態と特徴
	(3)	表４	攻撃の目的と手段
		表５	３種類の攻撃に対する各プロトコルの安全性
3	(1)	表６	わが国における主要な研究・開発プロジェクト
		表７	海外における主要な研究・開発プロジェクト
	(2)	表８	海外における主要な商用サービス
4		表９	デジタルタイムスタンプ技術に関する主な標準化の動向
5		表10	わが国で出願されている関連特許

2．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

イ. 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 このほか、デジタルタイムスタンプ技術の要件として、「タイムスタンプの生成には秘密情報を利用しない」ことを挙げる研究もある（Haber, Kaliski and Stornetta［1995］他）。この理由について、「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信頼性は、万一秘密情報が漏洩した場合等において大きく低下する。こうしたリスクを防止するために、秘密情報を利用しないシステムが必要である」とされている。ただし、こうした問題は鍵管理の問題であり、①安全性の低い公開鍵の利用を回避する、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と考えられる。このため、秘密情報を利用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い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さまざまな種類の文書の電子化が促進されることが期待される。

このうち④の要件は、デジタル署名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、利用されるハッシュ関数等の安全性が低下し、システム全体で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更新するかという問題である。１つのタイムスタンプの有効期間は、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どれほどか等の検討から総合的に判断される。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あらかじ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］他）。

ロ. デジタルタイムスタンプ技術の構成主体

デジタルタイムスタンプ技術の構成主体として、以下の４つが挙げられる（図１参照）。

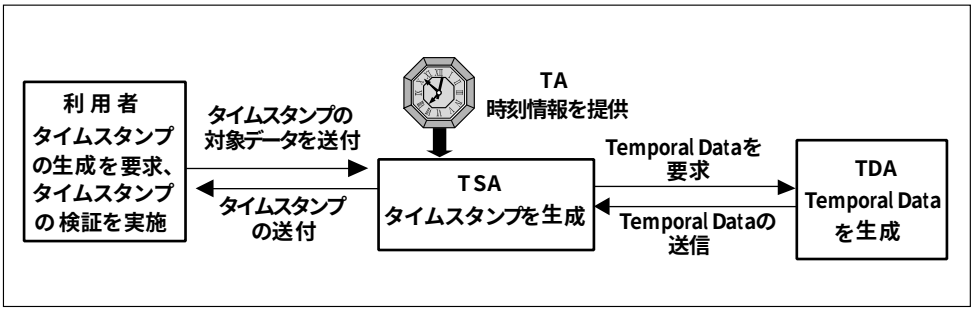
①タイムスタンプの利用者

タイムスタンプの利用者には、（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。

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。

図１　タイムスタンプ技術の構成主体



4 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か月から数年程度に設定される。署名生成の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、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。

④テンポラル・データ生成機関（TDA：Temporal Data Authority）

「ある時点までは確定せず、いったん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）であるテンポラル・データ（TD）を生成し、必要に応じてTSAに送付する主体。

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TD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TD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

イ. 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、暗号技術等に依存する。このため、デジタルタイムスタンプ技術を検討する際には、**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。

本稿では上記のうち②の検討を行い、上記①と③の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[5]。

（B）TSA・利用者間や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]。

5 正確な時刻情報を提供する技術についてさまざま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、安定したサービスの供給が可能であることが問題となる場合がある。本稿では、各システムが安定したサービスを供給可能であることを前提に議論を進めることとする。

（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。

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交信する技術の研究・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]。

ロ. デジタルタイムスタンプの分類方法

デジタルタイムスタンプを分類する際には、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主に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の分類方法

分類の着眼点	分類結果
TSAに対する信頼	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	①単純なタイムスタンプ・プロトコル、②リンキング・プロトコル、③分散プロトコル

7 本稿において、「ハッシュ関数が安全である」とは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をH（X）としたとき、出力値がH（X）となる別の入力値Y（H（X）=H（Y）かつX≠Yとなる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。ただし、「n bitのデータをランダムに$2^{n/2}$個集めると、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）を利用して、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で同じ出力値を有する入力値のペアを入手可能となる（バースデー攻撃）。このため、安全なハッシュ関数は、アルゴリズムに欠陥がないだけではなく、ハッシュ値が十分大きいことが必要となる。現時点では、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［2000］を参照。

9 本稿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。

①TSAに対する信頼

デジタルタイムスタンプの分類方法の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]。

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、比較的単純なシステムによってタイムスタンプを生成することができる。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システムが考えられる。

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、その結果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、単純なタイムスタンプ・プロトコル（Simple Protocol）、リンキング・プロトコル（Linking Protocol）、分散プロトコル（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、表２のとおり。

（A）単純なタイムスタンプ・プロトコル

単純なタイムスタンプ・プロトコルは、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特定される。本プロトコルでは、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。基本的な単純なタイムスタンプ・プロトコルにおけるタイムスタンプの生成手順は以下のとおり（図２参照）。

---

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。

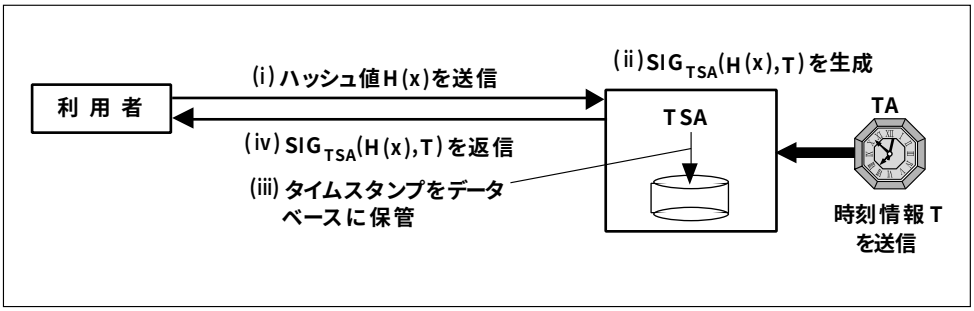
11 TSAの信頼度は、TSAを信頼できるか否かという二者択一の問題ではなく、TSAの業務運営・管理状況等の属性によって何段階ものレベルに分類したうえで検討することが必要となる。ただし、本稿では、単純化のために「ある具体的なアプリケーションへのタイムスタンプシステムの実装を前提とし、業務運営状況等から判断したTSAの信頼度が、そのアプリケーションにおける安全性の要件を満足すると判断された場合、そのTSAは信頼できる」と判断することとする。

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。

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
単純なタイムスタンプ・プロトコル	・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、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。
リンキング・プロトコル	・TSAが過去のハッシュ値を関連付けるリンク情報を生成し、リンク情報からタイムスタンプを生成。TSAはリンク情報やハッシュ値を保管。リンク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。 ・時刻は必ずしも特定されず、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、システム全体で安全性を確保可能。	・単純なタイムスタンプ・プロトコルに比べてシステムが複雑化し、タイムスタンプの生成・検証に必要な処理量が増加。また、リンク情報を保管するための追加的なデータベースが必要。
分散プロトコル	・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によるデジタル署名を結合してタイムスタンプとする方式や、秘密分散技術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場合、TSAのタイムスタンプ要求受信時刻を特定可能。		・複数のTSAの存在が前提条件。 ・単純なタイムスタンプ・プロトコルに比べてシステムが複雑化し、タイムスタンプの生成・検証に必要な処理量が増加。

図２　単純なタイムスタンプ・プロトコル



(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SIG$_{TSA}$(H(x), T)を自社のデータベースに保管。
(iv)TSAは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の真正性の確認によって行われる。

このように、本プロトコルは、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、デジタル署名の安全性およびTSAの運用・管理状況に依存することから、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）リンキング・プロトコル

リンキング・プロトコルは、TSAが複数の利用者のハッシュ値を相互に関連付けるリンク情報を生成し、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リンキング・プロトコルは、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リンク情報を改ざんしたとしても、利用されるデジタル署名やハッシュ関数が安全である限り、改ざんを発見できる仕組みとなっている（詳細は（３）ロ.②を参照）。

(i) リンク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が生成するリンク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。

ただし、単純なタイムスタンプ・プロトコルに比べてシステム構成が複雑化するほか、リンク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。

主なリンキング・プロトコルとして、リニア・リンキング・プロトコル（Linear Linking Protocol）、ツリー構造のリンキング・プロトコル（Linking Protocol using Tree Structure）、複数のTSAによるプロトコルの３つが挙げられる（表３参照）。なお、

表３　主なリンキング・プロトコルの形態と特徴

プロトコル	概要	長所	短所
リニア・リンキング・プロトコル	・TSAは、タイムスタンプ要求情報が到着するつどリンク情報を生成。リンク情報の一部が定期的に公表される。	・TSAに到着し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ツリー構造のリンキング・プロトコルよりも多い
ツリー構造のリンキング・プロトコル	・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保管し、タイムスタンプの対象となるハッシュ値を「葉」とみなしてリンク情報（「幹」に対応）を生成。	・タイムスタンプ検証に必要な計算量はリニア・リンキング・プロトコルに比べて少ない。 ・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によるリンキング・プロトコル	・各TSAのリンク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。	・リンク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。

タイムスタンプを生成する際に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]。

（イ）リニア・リンキング・プロトコル

リニア・リンキング・プロトコルでは、ある時刻のリンク情報はその時刻に送付されたハッシュ値やその直前のリンク情報から生成される。タイムスタンプの要求に応じてリンク情報が生成され、各リンク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。リンク情報は定期的に新聞等に公表される[14]。

本プロトコルにはさまざまなタイプが存在するが、ここではTSAがデジタル署名を生成するプロトコルを例として説明する。TSAは、以下の手順でリンク情報やタイムスタンプを生成する（図３参照）。

〈前提〉

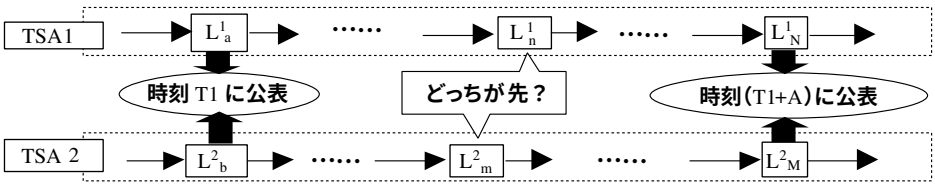
・利用者が第n番目のタイムスタンプの要求情報として、TSAにハッシュ値H_nを送信する場合を想定。

（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、リンク情報$L_n=H(H_n, n, L_{n-1})$を計算。さらに、H_nに対するタイムスタンプ$SIG_{TSA}(H_n, T_n, n, L_n)$を計算。

13 タイムスタンプに時刻情報が含まれないリンキング・プロトコルの場合、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リンキング・プロトコルの場合でも、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）。

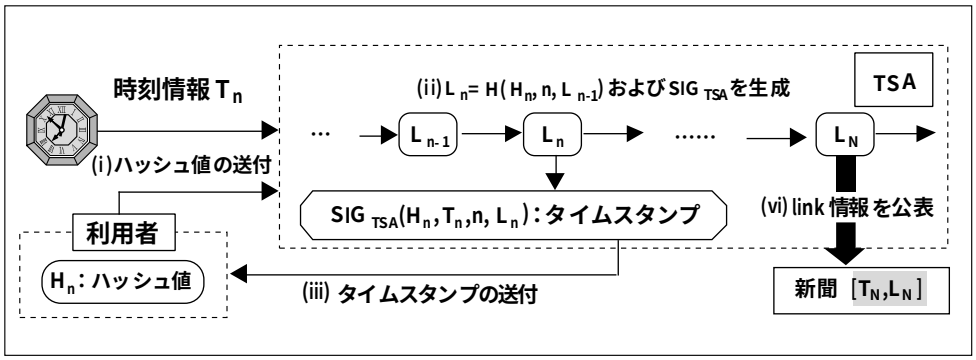
例えば、リニア・リンキング・プロトコル（詳細は次節）を採用する２つのTSA（TSA1とTSA2とする）が存在し、いずれもリンク情報をA分ごとに発表するものとする。時刻T1において、TSA1とTSA2はそれぞれリンク情報L^1_aとL^2_bを公表し、時刻（T1+A）では、TSA1とTSA2はそれぞれリンク情報L^1_NとL^2_Mを公表した場合を考える。このとき、TSA1がL^1_aとL^1_Nの間に生成したリンク情報L^1_nと、TSA2がL^2_bとL^2_Mの間に生成したリンク情報L^2_mに着目すると、どちらのリンク情報が先に生成されたかを証明することは困難である。この結果、L^1_nを含むタイムスタンプと、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方式が提案されている（陣内・櫻井［1999］）。

14 通常、すべてのリンク情報が公表されるのではなく、一部のリンク情報が公表される。どの程度のリンク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リンク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。

図３　リンク情報を公開する リニア・リンキング・プロトコル



（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）のリンク情報が生成されるたびに、リンク情報L_Nを時刻情報T_Nとともに新聞等に掲載。

一方、タイムスタンプ$SIG_{TSA}(H_n, T_n, n, L_n)$の検証手順は以下のとおり。

（i）検証者は、$SIG_{TSA}(H_n, T_n, n, L_n)$が生成される前に既に新聞に公表されていたリンク情報の中で、最も新しいリンク情報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（$L_N$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リンク情報）。
（ii）検証者は、$L_{k+1} = H(H_{k+1}, k+1, L_k)$、$L_{k+2} = H(H_{k+2}, k+2, L_{k+1})$、...の要領で順々にリンク情報を計算し、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、$SIG_{TSA}(H_n, T_n, n, L_n)$のデジタル署名の真正性を検証。

なお、リンク情報の公表頻度をX回に一度（図３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、公表されたリンク情報に対応するリンク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が必要となる。

（ロ）ツリー構造のリンキング・プロトコル

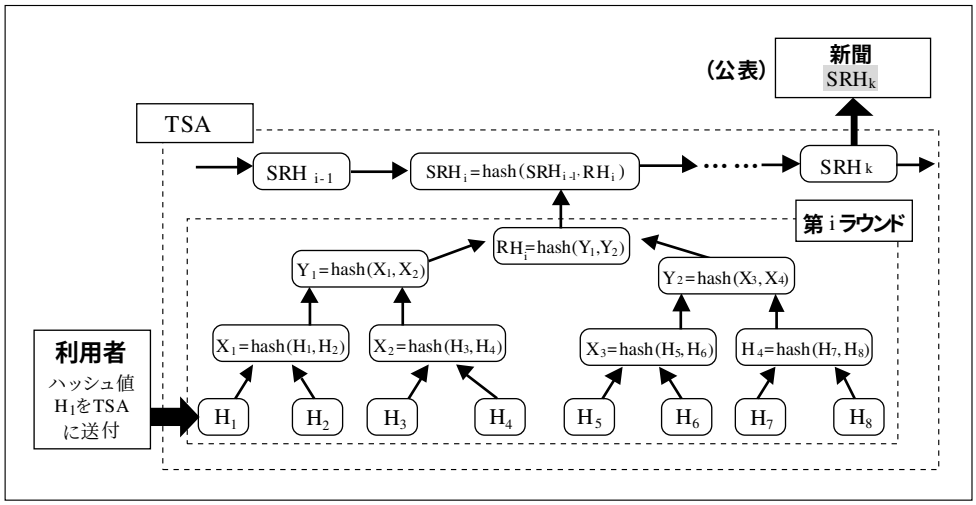
ツリー構造のリンキング・プロトコルでは、一定時間（ラウンド）内にTSAに送付されたハッシュ値を「葉」とみなし、ラウンド内のハッシュ値を結合・ハッシュ化するプロセス（ツリー構造が形成される）を繰り返して、ルート・ハッシュ（RH:Root Hash）と呼ばれるリンク情報（「幹」に相当）が生成される（Massias and Quisquater［1997］）。

本プロトコルでは、１つのラウンドで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。１つのラウンドにおいて受け付けたハッシュ

値の個数があらかじ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、TSAがハッシュ値の代わりとなる値（パディングデータ）を利用してRHを生成する。パディングデータには乱数や定数（例えば0）が用いられる。RHは、直前のラウンドで生成されたスーパー・ルート・ハッシュ（SRH:Super Root Hash）と結合されてハッシュ化され、そのラウンドのSRHとなる。SRHは定期的に新聞等に公表される。

本プロトコルでは、さまざまなタイプのシステムが存在しており、例えば（a）ツリーを再構築する（RHを生成する）ための情報がタイムスタンプとなる方式や、（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なる方式もある。ここでは、例として前者のタイムスタンプの生成手順を説明すると以下のとおり（図４参照）。

図４　SRHを公開するツリー構造の リンキング・プロトコル



〈前提〉

・TSAは、あらかじめリンク情報を生成するためのラウンドを設定。ここでは、1ラウンド中に最大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ラウンドに送付。

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

X_1=hash（H_1, H_2）、X_2=hash（H_3, H_4）

X_3=hash（H_5, H_6）、X_4=hash（H_7, H_8）

を計算（hashはハッシュ関数を表す）。

（ii）TSAは（X_1, X_2, X_3, X_4）を用いて次の計算を実行。

Y_1=hash（X_1, X_2）、Y_2=hash（X_3, X_4）

（iii）TSAはRH_i=hash（Y_1, Y_2）を計算。
（iv）TSAは、第（i-1）ラウンドのSRH_{i-1}とRH_iを結合、ハッシュ化し、第iラウンドの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掲載。

本方式では、TSAが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ハッシュ値に関しては、TSAへの到着時刻の前後関係が証明される。

一方、タイムスタンプの検証方法では、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と一致するかが確認される。図４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とおり。

（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。生成したSRH_iと、TSAが保管している（RH_{i+1},...,RH_k）からSRH_kを生成。
（iii）検証者は、生成したSRH_kと新聞掲載のSRH_kが一致することを確認。

また、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、リニア・リンキング・プロトコルに比べて少ない。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たびにSRHを公表するケースを考える。リニア・リンキング・プロトコルの場合、64個のリンク情報を生成するたびにリンク情報を公表するケースに相当し、最大64回のハッシュ関数計算が必要となる。一方、ツリー構造のリンキング・プロトコルでは、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を生成する際に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を生成する際に最大８回のハッシュ関数計算が必要となり、合計で最大11回のハッシュ関数計算が必要となる。

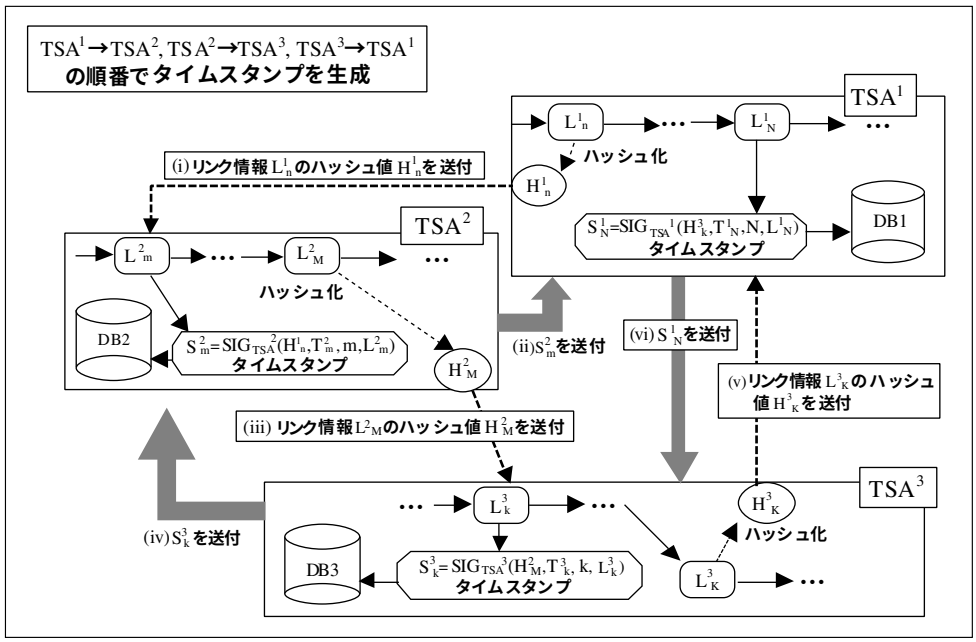
（ハ）複数のTSAによるリンキング・プロトコル

複数のTSAによるリンキング・プロトコルは、各TSAが独自にタイムスタンプを生成する一方、自分が生成したリンク情報やタイムスタンプを他のTSA（ランダムに選択）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プロトコルである。各TSAのリンク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とで、リンク情報を公表しなくても、TSAによるリン

ク情報の改ざんを困難にする。

以下では、リニア・リンキング・プロトコルをベースとしたプロトコルにおいて、タイムスタンプの生成手順を説明する（図５参照）。

図５　リニア・リンキング・プロトコルをベースにした複数のTSAによるリンキング・プロトコル



〈前提〉

・タイムスタンプの生成方法は図３と同一とする。

・３つのTSA（TSA^1、TSA^2、TSA^3）が存在する。

タイムスタンプの生成の順番を、①TSA^2がTSA^1のリンク情報に対するタイムスタンプを生成、②TSA^2がTSA^3のリンク情報に対するタイムスタンプを生成、③TSA^3がTSA^1のリンク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にリンク情報のハッシュ値を送付。

（i）TSA^1は、リンク情報L^1_nのハッシュ値H^1_nを生成し、H^1_nをタイムスタンプ要求情報としてTSA^2に送信。

（ii）TSA^2は、時刻T^2_mにH^1_nを受信し、リンク情報$L^2_m=H(H^1_n, m, L^2_{m-1})$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、リンク情報L^2_Mのハッシュ値H^2_Mを生成し、H^2_Mをタイムスタンプ要求情報としてTSA^3に送信。

（iv）TSA^3は、時刻T^3_kにH^2_Mを受信し、リンク情報$L^3_k=H(H^2_M, k, L^3_{k-1})$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、リンク情報L^3_Kのハッシュ値H^3_Kを生成し、H^3_Kをタイムスタンプ要求情報としてTSA^1に送信。

(vi) TSA^1は、時刻T^1_NにH^3_Kを受信し、リンク情報$L^1_N = H(H^3_K, N, L^1_{N-1})$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$H^3_K$をデータベースに保管するとともに、$S^1_N$を$TSA^3$に送付。

本プロトコルにおけるタイムスタンプの検証では、他のTSAに保管されているタイムスタンプやリンク情報のハッシュ値を基に実行される。その手順は以下のとおり。

〈前提〉

・検証対象はタイムスタンプ$S^1_n = SIG_{TSA}{}^1 (H^1_n, T^1_n, n, L^1_n)$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（リンク情報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（リンク情報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$S^3_k = SIG_{TSA}{}^3 (H^1_N, T^3_k, k, L^3_k)$を生成したとする。

(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から入手したリンク情報（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を照合。

(vi) 利用者は$S^1_n = SIG_{TSA}{}^1 (H^1_n, T^1_n, n, L^1_n)$のデジタル署名の真正性を検証。

TSAが自分の既存のタイムスタンプを改ざんするためには、他の複数のTSAが保管するタイムスタンプも改ざんする必要がある。このため、TSAによるタイムスタンプの改ざんは単純なタイムスタンプ・プロトコルに比べて困難となる。

（C）分散プロトコル

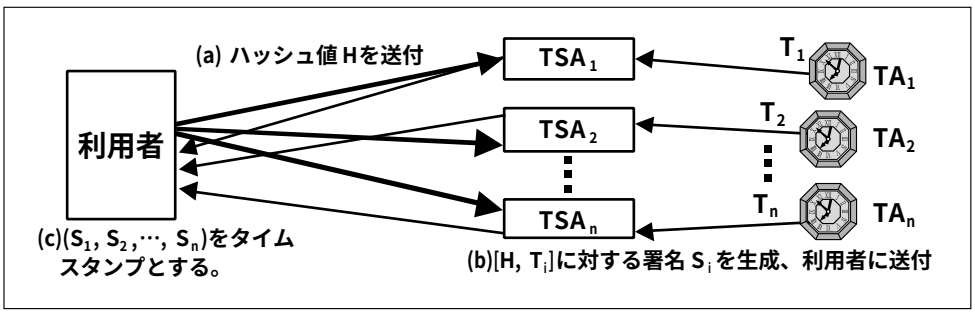
分散プロトコルは、複数のTSAが生成したデジタル署名（あるいは部分署名）を利用して、１つのタイムスタンプを生成するプロトコルであり、複数のTSAが利用可能であることが前提条件となる。典型的な分散プロトコルとしては、複数のTSAがタイムスタンプの対象となる文書のハッシュ値と時刻情報に対するデジタル署名を生成し、それらの複数のデジタル署名を１つに結合してタイムスタンプとする方式である（図６参照）。タイムスタンプの生成手順は以下の通り（Massias and Quisquater［1997］）。

（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、ハッシュ値Hを送付。
（b）各TSA_iは、ハッシュ値を受け付けた時刻の情報T_iを各自入手し[15]、［H, T_i］に対するデジタル署名 S_iを生成したうえで、S_iを利用者に返信。
（c）利用者は、n個のデジタル署名（S_1,...,S_i）をタイムスタンプとする[16]。

一方、タイムスタンプの検証は各デジタル署名の検証によって行われる。

また、秘密分散技術を利用したデジタル署名[17]を用いる方式も提案されている。

図６　分散プロトコル



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、さまざま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TSA_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。

16 Massias and Quisquater［1997］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１か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（閾値と呼ばれる）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［2000］を参照。

この方式では、利用者が複数のTSAにハッシュ値を送付すると、各TSAがハッシュ値と時刻情報に対する部分署名を生成し、利用者に返信する。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。

このように、本プロトコルでは、複数のTSAを用いてタイムスタンプを生成することによって攻撃者とTSAによる結託を困難にし、個々のTSAが信頼できない場合でも、システム全体としての信頼を高めることが可能となる。ただし、システムが複雑化するほか、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になるといった問題もある。

（3）各デジタルタイムスタンプの安全性

イ. 攻撃の分類

各デジタルタイムスタンプ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、そのうえ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。主な攻撃の目的と手段を整理すると、表４のとおり[18]。

表４　攻撃の目的と手段

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	・ハッシュ関数において、出力値が同一となる入力値のペアを探索 ・TSAが生成したデジタル署名を偽造。	・利用されているハッシュ関数やデジタル署名を攻撃。 ・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。 ・リンキング・プロトコルでは、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。
特定の利用者に対するタイムスタンプサービスの妨害	・特定利用者の要求情報のみ通信途中で傍受・奪取。	・特定利用者からのタイムスタンプ要求に対して、タイムスタンプの生成・送付を意図的に遅らせるようにTSAに依頼。 ・複数のTSAを利用するプロトコルでは、複数のTSAに対してサービス停止を依頼。
全面的なタイムスタンプサービスの妨害・停止	・ディナイアル・オブ・サービス（DoS: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

想定される攻撃の目的として、（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つが挙げられる。

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対策が必要である。このため、本稿は検討の範囲に含めないこととする。

②攻撃の手段

想定される攻撃の手段は、攻撃者とTSAが結託することが可能か否かによって変わってくる。以下では、攻撃者がTSAと結託可能な場合と不可能な場合に分けて、想定される攻撃方法を整理する。

（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と結託不可能な場合

攻撃者は、TSAが管理するリンク情報やハッシュ値等を利用できないため、タイムスタンプの生成にデジタル署名を利用する場合、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、タイムスタンプの生成にハッシュ値のみを利用するリニア・リンキング・プロトコルの場合（図３参照）、公表されたリンク情報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})と異なるデータ（例えば、(H_n, n, L'_{n-1})）を見つけ、公表されたリンク情報と整合性のとれた「公表されない不正なリンク情報の系列（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と結託可能な場合

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が保管する情報も攻撃に利用可能なため、リンキング・プロトコルの場合、TSAが保管するリンク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と結託不可能な場合

攻撃者が、攻撃対象者とTSAとの間の通信経路上で、攻撃対象者から送られたタイムスタンプ要求情報を傍受し、奪取するという攻撃が考えられる。

（b）攻撃者がTSAと結託可能な場合

攻撃対象者から送ら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の協力が必要なプロトコルでは、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の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、DoS攻撃のように、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とTAとのネットワークを利用不可能にするといった攻撃も考えられる。

ロ. 各攻撃の効果と対策

①単純なタイムスタンプ・プロトコル

（A）既存のタイムスタンプの改ざん

攻撃者がTSAと結託不可能な場合、タイムスタンプの改ざん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と結託可能な場合、タイムスタンプ生成用の署名生成鍵を有するTSAが、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できる。このように、攻撃者がTSAと結託可能な場合、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と結託不可能な場合、攻撃者は、攻撃対象者とTSAとの間の通信経路上で、攻撃対象者から送られたタイムスタンプ要求情報を傍受し、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、通信ネットワークがどの程度完備されているかに依存する。

攻撃者がTSAと結託可能な場合、攻撃が成功する可能性は、攻撃対象者が利用するTSAを特定できるか否かに依存する。TSAを特定できれば、攻撃が成功する可能性が高くなる。これに対し、攻撃対象者が毎回TSAをランダムに選択するシステムの場合、攻撃者はどのTSAと結託すればよいかがわからない。このため、攻撃者は、攻撃対象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高めることで、DoS攻撃が成功する可能性は低下する。(b) については、タイムスタンプ要求情報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チャレンジ／レスポンズ方式を採用することで、攻撃者が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り、DoS攻撃が成功する可能性は低下する。

②リンキング・プロトコル

（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、単純なタイムスタンプ・プロトコル同様、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。

攻撃者がTSAと結託可能な場合、ハッシュ関数やデジタル署名の安全性が十分な場合、攻撃が成功する可能性は単純なタイムスタンプ・プロトコルに比べて低い。これは、リンク情報の一部を公開するプロトコルの場合、任意のタイムスタンプを検証する際に公開されたリンク情報を利用することで、リンク情報やハッシュ値の改ざんを発見可能なためである。また、複数のTSAによるプロトコルでも、ある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*が生成したリンク情報に対して他のTSAが生成したタイムスタンプも改ざんする必要があり、攻撃者がすべてのTSAと結託しなければならない。

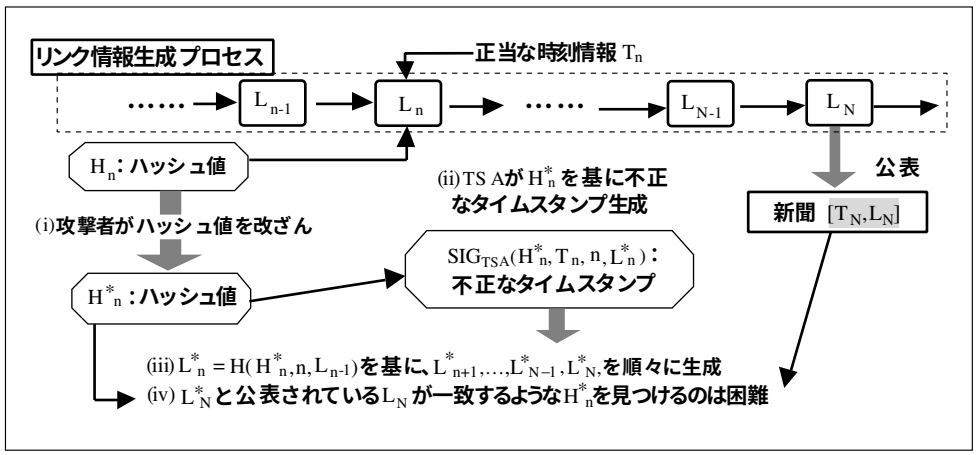
（a）リニア・リンキング・プロトコル

リニア・リンキング・プロトコル（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する場合、リンク情報の公開によってハッシュ値H_nの改ざんが困難になることを以下の手順で確認できる（図７参照）。なお、時刻情報T_nを改ざんする場合も同様となる。

（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、H^*_nを時刻T_nに受け付けたものとして、リンク情報$L^*_n=H(H^*_n, n, L_{n-1})$を生成したうえで、$SIG_{TSA}(H^*_n, T_n, n, L^*_n)$を生成。

図７　リニア・リンキング・プロトコルへの攻撃例



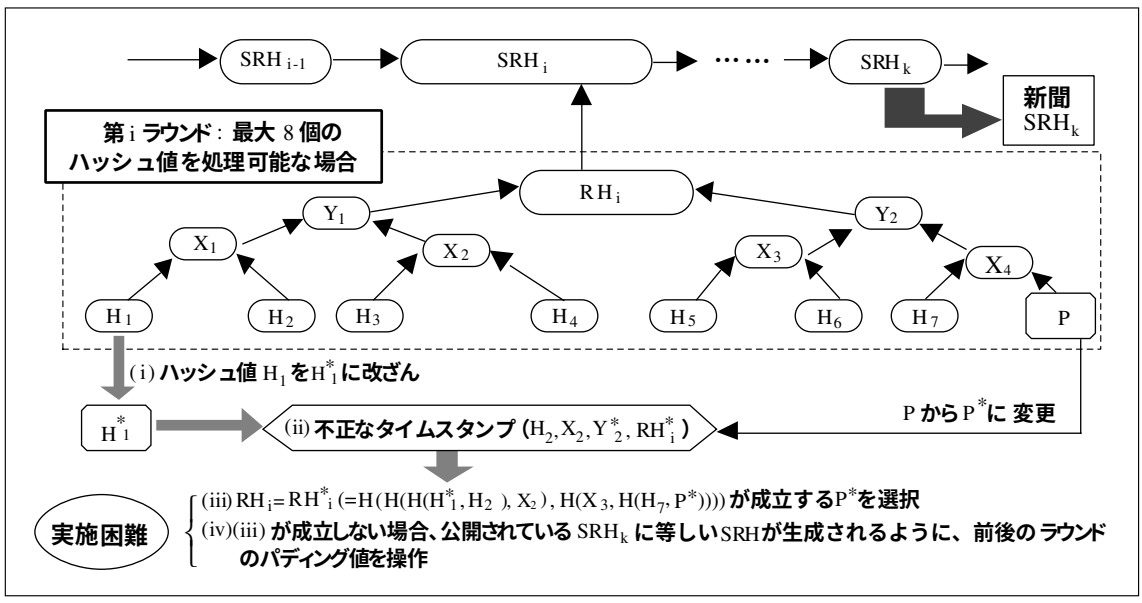
（iii）TSAは、リンク情報L^*_nを基にリンク情報［L^*_{n+1},..., L^*_{N-1}, L^*_N］を生成。
（iv）攻撃者とTSAは、SIG_{TSA}（H^*_n, T_n, n, L^*_n）の検証が成功するように、公表済みのL_NとL^*_Nが一致するH^*_nを設定する必要がある。このようなH^*_nを設定できるかはハッシュ関数の安全性に依存する。このため、H^*_nを発見不可能とする十分な安全性を有するハッシュ関数が必要である。

（b）ツリー構造のリンキング・プロトコル

ツリー構造のリンキング・プロトコル（図４参照）の場合も、タイムスタンプ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れば、攻撃が成功する可能性は低い。ただし、１つのラウンドで受付可能なハッシュ値の数があらかじめ設定されており、実際のハッシュ値の数がそれを下回ったケースではTSAが一定の値をパディングする仕組みとなっている。このため、TSAがパディングデータを操作するという攻撃も考えられるが、安全なハッシュ関数を利用する場合、本攻撃が成功する可能性も低い。これは以下の手順で確認することができる（図８参照）。

（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。（H_2, X_2, Y^*_2, RH^*_i）以外のデータは操作不可能とする。
（ii）TSAは、H^*_iを第iラウンド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を生成して（H_2, X_2, Y^*_2,

図８　ツリー構造のリンキング・プロトコルへの攻撃



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設定する。

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

ただし、H_2、X_2、X_3、H_7は他の利用者のハッシュ値から生成されていることから操作することはできない。上記の式が成立するP^*の値を発見可能か否かは、ハッシュ関数の安全性に依存する。$RH_i = RH^*_i$を満たすP^*を見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P^*を発見不可能な場合でも、

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

$$\cdots$$

$$SRH^*_k = H(RH_k, SRH^*_{k-1})$$

という手順でSRHの系列を改ざんできる可能性もある。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、他のラウンドにおけるパディングデータ、$(..., SRH_{i-2}, SRH_{i-1})$、$(SRHi_{i+1}, SRH_{i+2},...)$を設定できるか否かは、ハッシュ関数の安全性に依存する。P^*やSRHの系列を発見不可能なように、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。

上記の内容は、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た場合でも変わらない。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、ハッシュ値を受け付けた時間帯の幅が広くなる[19]。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うえで決定する必要がある。

（c）複数のTSAによるリンキング・プロトコル

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によるリンキング・プロトコル（図５参照）において既存のタイムスタンプを改ざんするためには、他のTSAが保管しているリンク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。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になることから、攻撃が成功する可能性は単純なタイムスタンプ・プロトコル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、リンク情報の一部（例えば、ツリー構造のリンキング・プロトコル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。

19 例えば、週に一度SRHが公表される場合、TSAと利用者が結託しても、タイムスタンプの要求情報の受付時間を週を超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。

（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、単純なタイムスタンプ・プロトコル同様、（a）TSAの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。

③分散プロトコル

（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と結託可能な場合、攻撃が成功する可能性は、単純なタイムスタンプ・プロトコルに比べて低い。これは、攻撃者が複数のTSAと結託する必要があり、TSAの数が増えるにつれてすべてのTSAと結託することはより困難になるためである。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。TSAの数を決定する際には、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と結託不可能な場合、攻撃者は、攻撃対象者とTSAとの間の通信経路上で攻撃者から送られたタイムスタンプ要求情報を傍受し、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に依存する。

攻撃者がTSAと結託可能な場合、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、単純なタイムスタンプ・プロトコル同様、（a）TSAの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。

ハ. 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５のとおり。

表５　３種類の攻撃に対する各プロトコルの安全性

		単純なタイムスタンプ・プロトコル	リンキング・プロトコル	分散プロトコル
既存のタイムスタンプの改ざん	攻撃者がTSAと結託可能な場合	安全なデジタル署名やハッシュ関数を利用しても、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データを管理するため、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。	安全なデジタル署名やハッシュ関数を利用する場合、単純なタイムスタンプ・プロトコルに比べ、改ざんが成功する可能性は低い。 ― ①リンク情報の一部を公表する、あるいは②あるTSAのリンク情報に対して他の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。	安全なデジタル署名やハッシュ関数を利用する場合、単純なタイムスタンプ・プロトコルに比べ、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の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の改ざんには複数のTSAとの結託が必要。 ― 攻撃者とTSAの結託はTSAの数を増やすことでより困難となる。
	攻撃者がTSAと結託不可能な場合	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。		
サービスの妨害	攻撃者がTSAと結託可能な場合	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が利用できる環境では、利用者は、毎回TSAをランダムに選択することで攻撃成功の可能性を低下させることが可能。		
	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・奪取することが必要。		
全面的なタイムスタンプサービスの妨害・停止		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、チャレンジ／レスポンス方式によってタイムスタンプ要求情報を受け付ける等の方法により、本攻撃法の有効性を低下させることが可能。		

単純なタイムスタンプ・プロトコルでは、攻撃者がTSAと結託可能な場合、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。リンキング・プロトコルでは、リンク情報を公表したり、複数のTSAを利用したりすることによって、攻撃者がTSAと結託可能な場合でも、既存のタイムスタンプの改ざんが成立する可能性は単純なタイムスタンプ・プロトコルに比べて低い。その他の攻撃が成功する可能性は単純なタイムスタンプ・プロトコルと同様である。また、分散プロトコルにおいても、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単純なタイムスタンプ・プロトコルに比べて低い。その他の攻撃が成功するか否かについては、単純なタイムスタンプ・プロトコルやリンキング・プロトコルと同様である。

このように、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、リンク情報の一部を公開する、もしくは複数のTSAによるリンキング・プロトコルを利用することが必要となる。また、十分な数のTSAを

利用した分散プロトコル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、リンキング・プロトコルや分散プロトコルは単純なタイムスタンプ・プロトコルに比べてシステムが複雑であるほか、同一の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、TSA等のインフラ状況を踏まえたうえで、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単純なタイムスタンプ・プロトコル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、単純なタイムスタンプ・プロトコルは、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、実際米国等では単純なタイムスタンプ・プロトコル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３章において説明する。

３．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に関する研究・開発プロジェクト

イ. わが国における主要な研究・開発プロジェクト

わが国におけるデジタルタイムスタンプ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６参照）。

表６　わが国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	単純なタイムスタンプ・プロトコル		分散プロトコル
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[20]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、とく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[21]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１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、そのひと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[22]。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うえで、確定日付簿および当該文書に日付のある印章[23]を押捺し、(c) さらにその印章で確

20 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）。

21 民事訴訟においては、裁判所は判決をするにあたり、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）。したがって、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。

22 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。

23 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。

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。

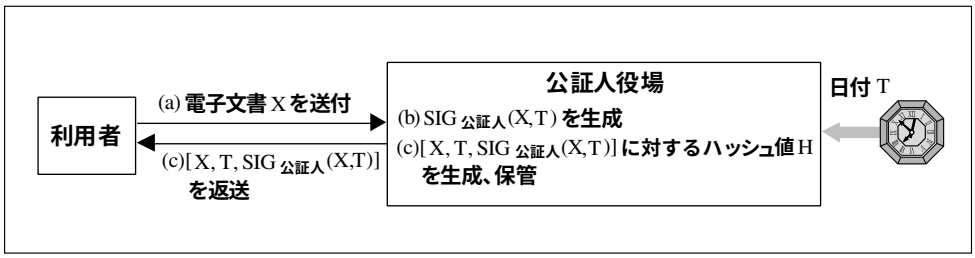
（B）電子確定日付のシステム

電子確定日付のシステムは単純なタイムスタンプ・プロトコルに分類され、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、リンキング・プロトコルや分散プロトコルと比較すると、比較的単純なシステム構成となっている。具体的なタイムスタンプの生成手順は以下のとおり（図９参照）。

（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、［X, T］のデジタル署名$SIG_{公証人}(X,T)$をタイムスタンプとして生成。
（c）公証人は、［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、［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。

図９　法務省・電子公証制度のタイムスタンプシステム



②ニューメディア開発協会・電子公証システム実証実験

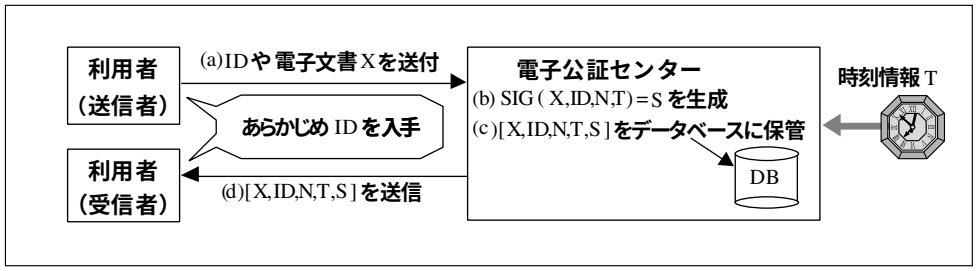
ニューメディア開発協会[24]は、1997年10月から1998年2月までの間、情報処理振興事業協会が推進する「エレクトロニック・コマース推進事業」[25]の一環として、

24 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する。現在、電子公証システム等インターネット利用環境整備のためのシステム開発や、電子保存システム、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や本実証実験については、http://www.nmda.or.jp/nmda/about-nmda.htmlを参照。

25 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単純なタイムスタンプ・プロトコルとなっている。

日本で初めて電子公証システムの実証実験を行った[26]。本システムでは、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されており、単純なタイムスタンプ・プロトコルに分類される。本システムの電子文書の登録・照会サービスの概要は以下のとおり（図10参照）。

図10　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、あらかじめ利用者自身の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入手する。利用者は、電子文書Xを暗号化電子メールPEM[27]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）、（iii）セッション鍵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、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、利用者の公開鍵でデジタル署名を検証。電子公証センターは、Xに時刻情報T、ID、シリアル番号N等を添付し、デジタル署名Sを生成。電子文書の真正性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、Xに電子証明書を加えてCD-R等に書き込み、保管。

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

26 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に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］）。

27 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。

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、その真正性を電子公証センターのデジタル署名によって検証する。

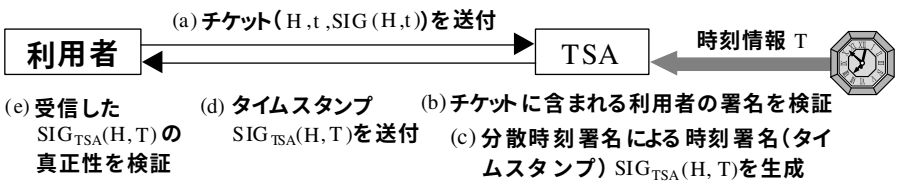
③NTT・分散時刻署名システム

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月のIWS '99[28]で発表されている（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、分散時刻署名を用いてタイムスタンプを生成する。本システムは分散プロトコルに分類される。

（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とおり（図11参照）。

図11　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。TSAは、チケットに添付されている署名の検証によって利用者の本人確認を行うことができる。また、チケットの署名の有効期間を短くすることで、署名に必要な安全性レベルを低く設定でき、署名生成鍵を短くして署名生成・検証処理の高速化を図ることができる。

（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。

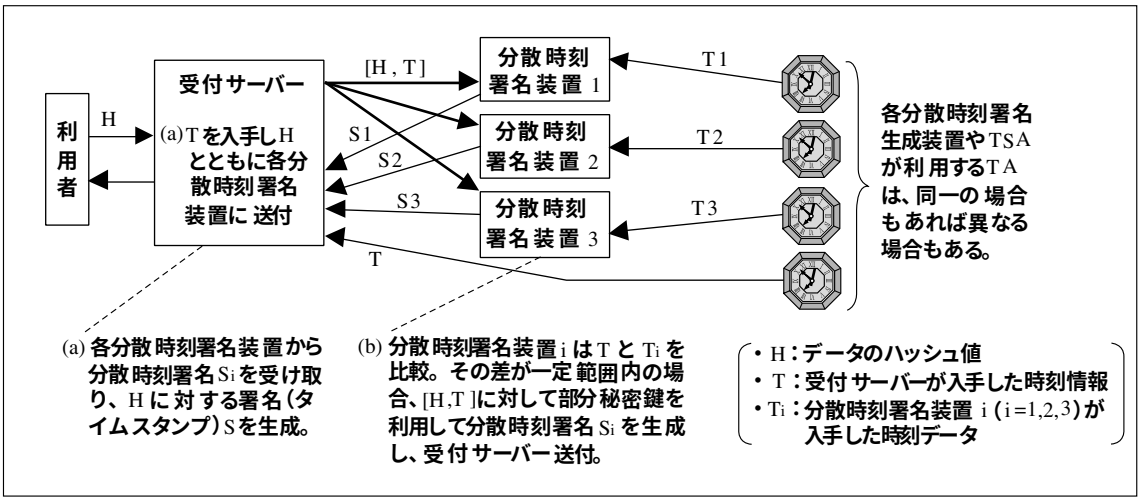
28 IWS（Internet Workshop）：電子情報通信学会の時限研究会であるインターネット研究会が主催するワークショップであり、ネットワークセキュリティ、マルチメディア通信、トラフィック管理等、インターネット関連技術が対象分野となっている。

（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生成した部分時刻署名から生成される。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。以下では、分散時刻署名装置が３個、署名を生成するために２つ以上の部分署名が必要な場合を例に、タイムスタンプの生成手順を説明する（図12参照）。

図12　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、利用者から受信したハッシュ値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T_i（i=1,2,3）とTの差を計算し、その差が一定範囲内である場合には分散時刻署名S_iを生成して受付サーバーに送付。
（c）受付サーバーは、２つ以上の分散時刻署名を入手して、タイムスタンプSを生成する（署名生成の閾値は２）。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ず、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、閾値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で、分散時刻署名装置間の結託を困難にし、一部の装置が故障した場合でも対応可能な仕組みとなっている。ただし、複数の独立な分散時刻署名装置が必要である等、実装には一定のインフラ整備が必要となる。

ロ．海外における主要な研究・開発プロジェクト

海外の主要な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７参照）。

表７　海外における主要な研究・開発プロジェクト

	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	ツリー構造のリンキング・プロトコル	リニア・リンキング・プロトコル	複数のTSAによるリンキング・プロトコル
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RHを生成。RHは、直前のラウンドのSRHと結合・ハッシュ化されてSRHとなる。SRH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、RH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リンク情報を生成。リンク情報は、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、リンク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）の暗号研究者を中心とするグループによって進められた。研究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として位置付けられている[29]。

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リンキング・プロトコルを挙げている。これは、①単純なタイムスタンプ・プロトコルを実装する際に必要な「信頼できるTSA」が利用困難である、②分散プロトコルに必要な多くのTSAが利用不可能といった問題点が存在するためと指摘されている（Massias and Quisquater［1997］）。

TIMESECでは、ツリー構造のリンキング・プロトコルが検討対象となっており、タイムスタンプの生成・検証手順は２.（２）ロ.（B）（ロ）で例示されているものと同一である。タイムスタンプの生成にデジタル署名を利用していないが、これにつ

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

いては、「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恐れがあり、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、SHA-1[30]とRIPEMD-160[31]の２種類のハッシュ関数が利用されており、１つの文書に対して２種類のタイムスタンプとリンク情報が生成される。これは、どちらか一方のハッシュ関数の安全性が低下しても、もう１つのハッシュ関数を利用することでシステム全体の信頼性を維持できるとの考え方によるものである。

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E-Docの一部として1997年から開始された研究プログラムである。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[32]。

Cuculusでは、TIMESECと同様の理由からリンキング・プロトコルが検討対象となっており、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リニア・リンキング・プロトコルが採用されている（Lipmaa［1999］）。タイムスタンプの生成手順は以下のとおり（図13参照）。

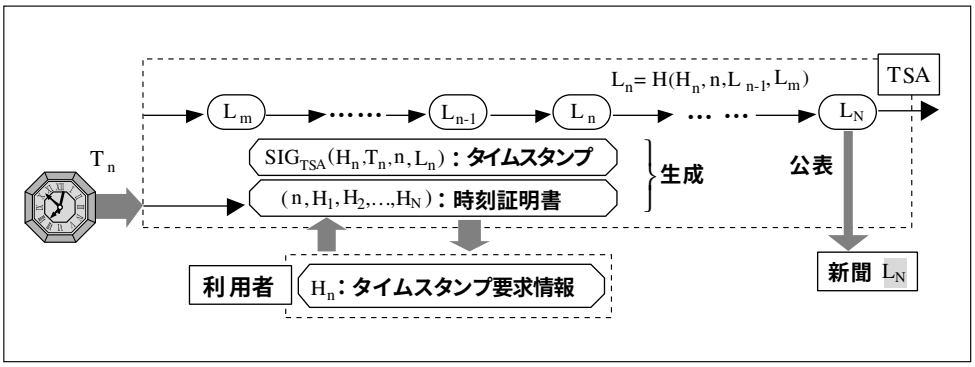
（a）TSAは、あらかじめラウンドを設定し、1ラウンド内で受け付けることが可能なハッシュ値の数Nを設定。
（b）利用者はTSAに対してハッシュ値H_nを送付。
（c）TSAは、受信したH_nを用いてリンク情報$L_n = H(H_n, n, L_{n-1}, L_m)$を生成。ただし、nはシリアル番号を表し、$L_m$はあらかじめ設定された順番のリンク情報（例えば、直前のラウンドにおけるm番目のリンク情報）を表す。
（d）TSAは、タイムスタンプとして（H_n, T_n, n, L_n）に対するデジタル署名$SIG_{TSA}(H_n, T_n, n, L_n)$を生成するほか（ただし、$T_n$は時刻情報）、そのラウンドが終了した後に時刻証明書（Time Certificate）と呼ばれる情報（n, H_1, H_2, ..., 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ラウンド内の最後のリンク情報L_Nを新聞に掲載する。

30 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。

31 RIPEMD-160：欧州議会における情報通信技術の研究プログラム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。

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

図13　Cuculus のリニア・リンキング・プロトコル



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、タイムスタンプと時刻証明書を利用してラウンド内のリンク情報（$L_1, \ldots, L_N$）を生成し、公表されているL_Nと一致することを確認。

③PKITS（スペイン）

PKITS（Public Key Infrastructure with Time Stamping Authority）は、欧州議会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）[33]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[34]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始され、1998年に報告書が公表されている（FNMT［1998］）。PKITSでは、①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、②現時点では信頼できるTSAを前提とすることは困難との見方から、デジタル署名を利用しないリニア・リンキング・プロトコルをベースとした複数のTSAによるリンキング・プロトコルが検討対象となっている。

PKITSのシステムでは、シンクロナイゼーション・サイクル（Synchronization Cycle）と呼ばれるスキームが採用されている。これは、同じリニア・リンキング・

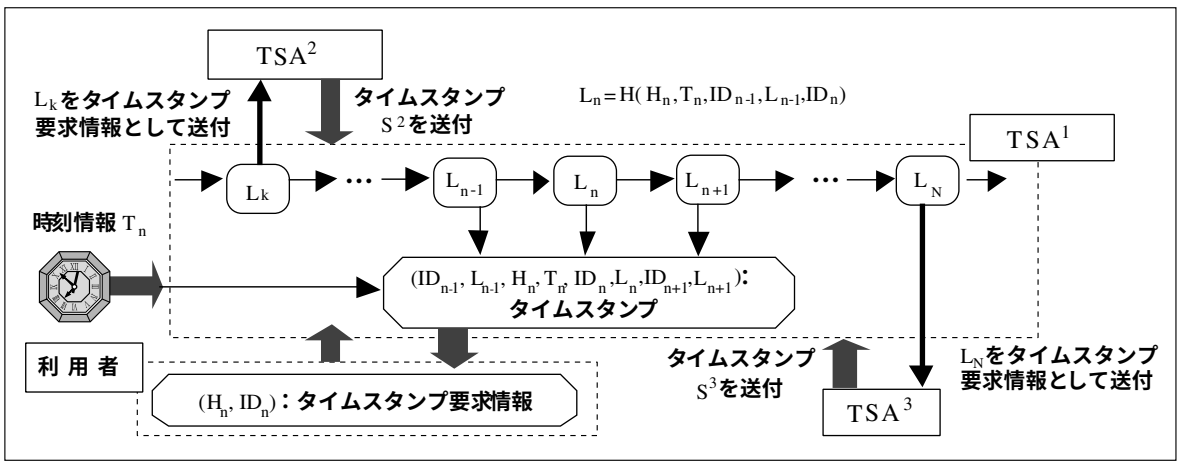
33 ETSは、欧州を中心としたグローバルな情報インフラの整備の一環として、主にデジタル署名やTTPの技術的要件・サービス内容について調査・研究を行うものであり、1992年から開始されている。ETSは、①ETS研究（既存の研究成果や実装例等について検討を実施）、②ETS Ⅰプロジェクト（ETS研究の成果を踏まえて、望ましい技術要件やサービス内容等について検討を行い、パイロット実験を実施）、③ETSⅡプロジェクト（パイロット実験を実施するとともに、実装に向けた具体的な技術要件や業務要件等について検討を実施）の３つのフェーズから構成されている。このうち、PKITSはETSⅡプロジェクトに含まれてい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

34 FNMT（Fablica Nacional de Moneda y Timbre）：スペインにおける貨幣鋳造業務および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）。

プロトコルを採用しているTSAが複数存在することを前提とし、各TSAが生成したリンク情報に対して他のTSAが適宜タイムスタンプを生成するというものである。相互にリンク情報に対するタイムスタンプを生成することで、各TSAのリンク情報の真正性を確保し、リンク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の考え方によるものである。リンク情報は公表されない。PKITSにおけるタイムスタンプの生成手順は以下のとおり（図14参照）。

（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情報であるID_nをTSA^1に送付。
（b）TSA^1は、リンク情報$L_n = H(H_n, T_n, ID_{n-1}, L_{n-1}, ID_n)$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}を受け取ってリンク情報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。

図14　PKITSの複数のTSAによるリニア・リンキング・プロトコル



一方、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、タイムスタンプのH_nが文書のハッシュ値と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を入手し、それぞれのリンク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。

$$L_{k+1} = H(H_{k+1}, T_{k+1}, ID_k, L_k, ID_{k+1})、\ L_{k+2} = H(H_{k+2}, T_{k+2}, ID_{k+1}, L_{k+1}, ID_{k+2})$$

...

$$L_N = H(H_N, T_N, ID_{N-1}, L_{N-1}, ID_N)$$

（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、リンク情報の真正性を他のTSAが確保することによってリンク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ため、対象となるアプリケーションにおける安全性や効率性を考慮しながら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８参照）。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単純なタイムスタンプ・プロトコルを採用しており、リンキング・プロトコルを採用しているのはSurety社のDigital Notary Serviceのみである。

表８　海外における主要な商用サービス

	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	ツリー構造のリンキング・プロトコル	単純なタイムスタンプ・プロトコル		
利用される暗号技術	ハッシュ関数	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ハートハッシュ値（RH:Root Hash）を生成。RHは1秒前のスーパーハッシュ値（SH:Super Hash）と結合・ハッシュ化されてSHとなり、一部がニューヨーク・タイムズ紙に掲載。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。	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、自社のウェブサイト上に毎日掲載。

（1）Digital Notary Service（Surety社）

Digital Notary Serviceは、ツリー構造のリンキング・プロトコルを利用したタイム

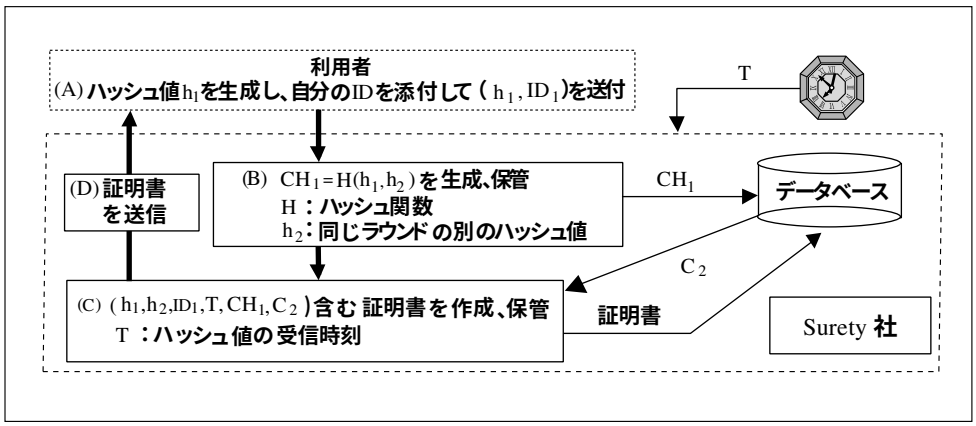
スタンプサービスであり、1992年に開始された[35]。Digital Notary Serviceでは、タイムスタンプとしてハッシュ値に対する証明書（Certificate）が作成されるほか、リンク情報の一部がニューヨーク・タイムズ紙に毎週日曜日に公開される。証明書には、タイムスタンプの対象となっている文書のハッシュ値、時刻情報、証明書の正当性を証明するために必要な情報（リンク情報や他の文書のハッシュ値）が含まれている。

①証明書の生成手順

Digital Notary Serviceのシステムは、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証明書の生成手順は以下のとおり（図15参照）。

（A）利用者は、専用ソフト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）を生成したうえで、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と同じラウンド（ラウンドは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（CH: Combined Hash）CH_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, CH_1, C_2）を内容とする証明書を作成し、自社のデータベースに保管。なお、Tは受付時刻、C_2 は証明書の検証を可能にするための情報（時刻T における他のCHや、RH等が含まれる）。
（D）Surety社は、証明書を各利用者に送付。

図15　Digital Notaryにおける証明書の生成手順



②リンク情報の生成手順と証明書の検証手順

リンク情報として、RHやSHが生成される。これらのハッシュ値の生成手順は以

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

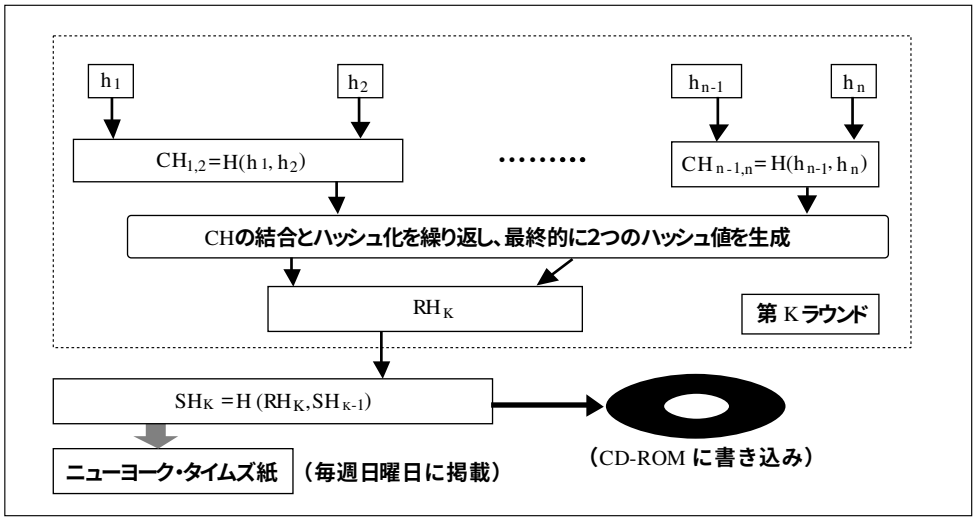
下のとおり（図16参照）。

（A）同一ラウンド（図16では第Kラウンド）において受信したハッシュ値（$h_1,...,h_n$）（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（$CH_{1,2},..., CH_{n-1,n}$）を生成。
（B）複数のCHを再び結合・ハッシュ化し、最終的に288 bitのRH_Kを生成したうえで、CD-ROMに記録。ただし、RH生成用のハッシュ関数は、CH生成用のハッシュ関数とは異なる。
（C）RH_KとSH_{K-1}を結合し、再びハッシュ化してSH_Kを作成。
（D）SHは、Surety社のCD-ROMに記録されるほか、その一部は毎週日曜日のニューヨーク・タイムズ紙に掲載。

本システムでは、CHとRHのハッシュ関数が異なっている点が特徴である。証明書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証明書のハッシュ値が同一であることを確認。
（B）検証者は、証明書の情報から当該ラウンドのRHを生成し、CD-ROMのRHやSHを用いてSHの系列を生成。生成したSHと、ニューヨーク・タイムズ紙に掲載されているSHを比較。

図16　第Kラウンドにおける SHの作成方法



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[36]。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[37]。

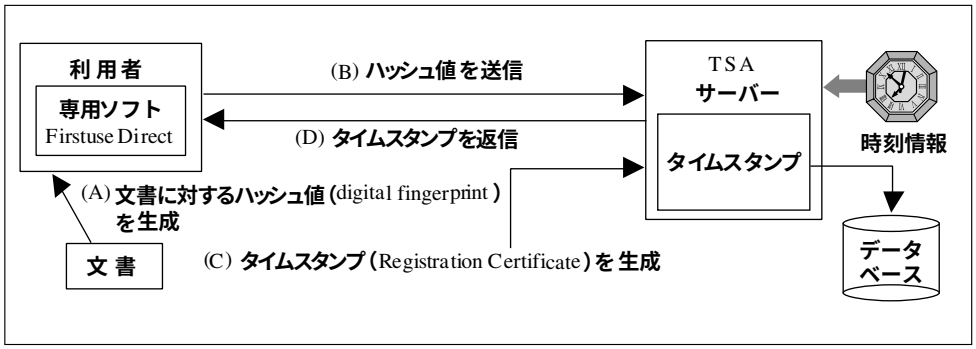
ロ. Firstuse.com（Firstuse.com社）とe-TimeStamp（DigiStamp社）

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時間利用可能なサービスであり、1998年にサービスが開始されている[38]。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単純なタイムスタンプ・プロトコルが採用されている。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とおり（図17参照）。

（A）利用者は、あらかじめパソコンに専用ソフトFirstuse Directをインストールし、タイムスタンプの対象データをハッシュ化してdigital fingerprintと呼ばれるハッシュ値を生成。
（B）利用者は、インターネット経由でFirstuse.comのTSAサーバーにハッシュ値を送付。
（C）TSAサーバーは、ハッシュ値、時刻情報、利用者名、住所等を含むRegistration Certificateと呼ばれるタイムスタンプを生成[39]。
（D）TSAサーバーは、タイムスタンプ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。

36 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。
37 ZANTAZ社に関する情報については、http://www.zantaz.com/を参照。
38 Firstuse.com社に関する情報については、http://www.firstuse.com/を参照。また、DigiStamp社に関する情報は、http://www.e-timestamp.com/を参照。なお、具体的なサービス利用状況については、これらのウェブサイトには掲載されていない。
39 Firstuse.com社のホームページによると、タイムスタンプ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、その具体的な方法については記述されていない。

図17　Firstuse.comのタイムスタンプ生成手順



タイムスタンプの検証はFirstuse Directにおいて実行される。その手順は、(A) タイムスタンプの真正性を確認、(B) 対象データのハッシュ値とタイムスタンプに含まれているハッシュ値を照合、(C) タイムスタンプのハッシュ値とFirstuse.comが管理するデータベースのハッシュ値を照合、の３段階で構成される。

ハ. Stamper（I.T. Consultancy社）

Stamperは、PGP[40]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のタイムスタンプ生成方法は単純なタイムスタンプ・プロトコルであり、タイムスタンプをインターネット上に公開することでタイムスタンプの改ざんを困難にする仕組みを採用している[41]。

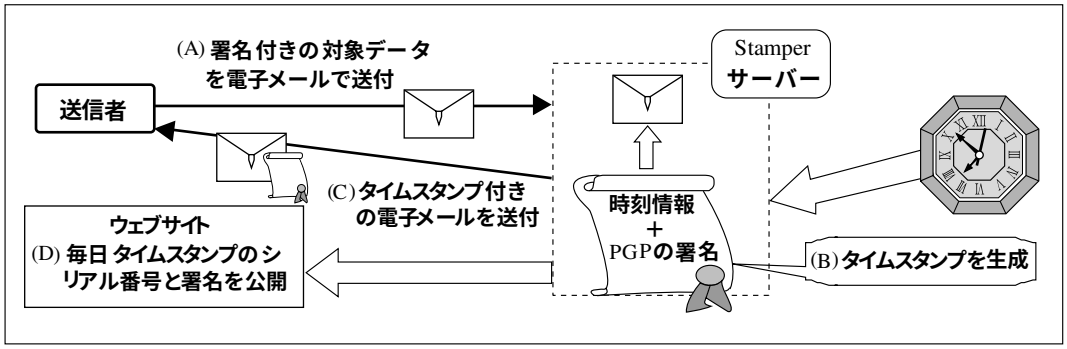
Stamperにはタイムスタンプに関連する６種類のサービスが準備されており、通常のタイムスタンプサービスに対応するPGPモードのほか、電子メールの配達記録証明サービスとしてPOSTモードが用意されている。利用者は、電子メールの送付先を変更することで、サービスを選択できる。例として、PGPモードにおけるタイムスタンプの生成・検証方法は以下のとおり（図18参照）。

（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。
（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と呼ばれるタイムスタンプを生成。
（C）Stamperサーバーは、利用者から送付されたデータとタイムスタンプを利用者に送付。同時に、I.T. Consultancy社のウェブサイト上にシリアル番号と

40 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。
41 Stamperの内容については、http://www.itconsult.co.uk/stamper/stampinf.htmを参照。

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、誰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。

図18　PGPモードのタイムスタンプ生成手順



タイムスタンプの真正性は、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。また、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もできる。

4．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、現在さまざま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。主な標準化の動向を整理すると以下の表９のとおり。

表９　デジタルタイムスタンプ技術に関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[42]としてPKIX-TSP（Internet X.509 Public Key Infrastructure Time Stamp Protocols, 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、単純なタイムスタンプ・プロトコルがベースとなっているとみられるものの、リンキング・プロトコル等さまざま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。

イ. 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の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。

ロ. PKIX-TSPの概略

PKIX-TSPのベースとなっているのは単純なタイムスタンプ・プロトコルである（図19参照）。

図19　PKIX-TSPのプロトコルの概略

42 Internet Draft：IETFのワーキング・グループ等から標準の原案として提案される文書。Internet Draftは、ワーキング・グループ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み込まれることとなる。なお、標準化提案となった文書は、さまざま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。

本プロトコルでは、以下の２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[43]、タイムスタンプの生成方法に関するポリシー情報[44]等から構成される。

②TimeStampResp

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）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さまざま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[45]。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。

ハ. 標準化と特許問題

現在、今後のPKIX-TSPの標準化について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のプロトコルが既存の特許に抵触する可能性があることが指摘されている。PKIX-TSP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。ただし、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ない。このため、PKIX-TSPの標準化が完了しても、特許に抵触する可能性が残されている場合、標準として利用できない可能性もある。

43 1999年7月11日現在のPKIX-TSPの最新バージョンは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。

44 ポリシー情報は、TSA等におけるログの取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。

45 本プロトコルでは２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、単純なタイムスタンプ・プロトコル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うえで、通信オプションを利用する。例えばリンキング・プロトコルの場合には、TSA内部でリンク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にリンク情報等を入力して、リンキング・プロトコルに必要なデータ交信を行うといった実装が可能となる。

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。

イ. 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。そのうえで、「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、一般的な否認防止トークン（Generic Non-repudiation Token）とタイムスタンプ・トークン（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。

ロ. 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のサービス関連の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。

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（オンラインやオフライン）について解説したうえで、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うえ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うえで、これらのサービスを実施するために必要となるCRL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。

５．デジタルタイムスタンプ技術の主な関連特許

第４章で触れたように、デジタルタイムスタンプサービスの標準化を行う際には、関連する特許の存在が問題となることが多い。例えば、PKIX-TSP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[46]等を参考に、わが国で出願されている関連特許を調査すると、表10の９件の特許が出願されていることがわかる（1999年8月末現在）。今後、デジタルタイムスタンプ技術を利用する際には、こうした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ろう。

表10　わが国で出願されている関連特許（1999年8月末現在）

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. RE034954 平成7年5月30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

46 PKIX-TSPには、表10の①～④の４件の米国特許を含め、合計６件の米国特許が記載されている。

６．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ひとつとして、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２章（２）イ. 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、さまざま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の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、これらのどこか一か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うえで、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、安全性の高いデジタルタイムスタンプを利用可能にするためには、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・高度化を進めるうえ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。

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、日本銀行金融研究所、1999年4月

岩本信正、『公証人法』、明文社、1981年

宇根正志・岡本龍明、「最近のデジタル署名における理論研究動向について」、『金融研究』第19巻別冊第１号、日本銀行金融研究所、2000年4月

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・電子認証について－ 技術面、標準化に関する最近の動向を中心に」、『金融研究』第19巻別冊第１号、日本銀行金融研究所、2000年4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

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) ,” October 1999. (ftp://ds.internic.net/ internet-drafts/draft-ietf-pkix-time-stamp-04.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/infose -curity/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, 1999.