時刻信頼性を検証するタイムスタンプ検証サーバ

谷川嘉伸[†] 手塚悟[†]
小黒博昭[††] 橋川善之[†††]

タイムスタンプの統一的な検証およびタイムスタンプの時刻信頼性の検証を実現するタイムスタンプ検証サーバを開発した．実用性と拡張性を重視し，RFC 3029 を拡張したタイムスタンプ検証プロトコルを設計・開発し，任意の方式で作成されたタイムスタンプを統一的に検証できるようにした．また，検証対象とするタイムスタンプ方式において商用サービスの独自方式を含めることを目標とし，非公開の検証モジュールをタイムスタンプ検証サーバにアドオンするために使用できる共通インタフェースを設計・開発した．さらに，時刻信頼性検証方式として時刻監査証明書確認方式と時刻監査レポート確認方式を設計・開発した．実装したタイムスタンプ検証サーバを用いて RFC 3161 ベースのタイムスタンプと ISO/IEC 18014-2 アーカイブ方式ベースのタイムスタンプの検証動作を確認し，対話環境において実用性があることを確認した．

Development of Time-stamp Validation Server Supporting Time Reliability Verification

Yoshinobu Tanigawa,[†] Satoru Tezuka,[†] Hiroaki Oguro[††] and Yoshiyuki Hashikawa[†††]

We developed a time-stamp validation server which provides universal time-stamp verification and time reliability verification. We designed and developed the time-stamp validation protocol based on RFC 3029 for its practicality and extensibility. Moreover, we designed a common interface for verification modules of original non-standard time-stamping schemes. As time reliability verification schemes, we also designed time attribute certificate verification scheme and time audit report verification scheme. We constructed a prototype client-server system, and confirmed that the server successfully performed time-stamp validation of two types of time-stamps, which are based on RFC 3161 and the ISO/IEC 18014-2 archiving scheme.

1. はじめに

電子データの完全性を保証する技術の 1 つとして，タイムスタンプ技術が注目されている．タイムスタンプ技術とは，電子データと時刻情報を暗号学的に結び付ける技術であり，電子データの存在証明機能と電子データの完全性証明機能を有している[1)]．この機能により，タイムスタンプ技術の利用者および検証者は，タイムスタンプが付与された電子データがいつ存在したのか，また，それ以降に改ざんされていないのかどうかを確認することができる．

† 株式会社日立製作所システム開発研究所
Systems Development Laboratory, Hitachi Ltd.
†† 株式会社 NTT データ技術開発本部
Research and Development Headquarters, NTT DATA CORPORATION
††† NTT データ・セキュリティ株式会社
NTT DATA SECURITY CORPORATION

これまでにタイムスタンプ技術に関して様々な研究開発が行われてきている．タイムスタンプ作成メカニズムの観点からシンプルプロトコル，リンキングプロトコル，分散プロトコルと呼ばれる様々な方式が提案されている[1)]．これらのうち，シンプルプロトコルとリンキングプロトコルのアイデアに基づく技術は，RFC 3161[2)] や ISO/IEC 18014[3)~5)] などの標準規格となっている．また，e-文書法の施行などもあり，タイムスタンプ技術は，実社会の IT 基盤技術として利用されつつある[6)]．

近年，様々なタイムスタンプ方式を包括的な視点からとらえた研究がいくつか行われている．特に，宇根らは，文献 7) において，タイムスタンプ検証に注目し，様々なタイムスタンプ方式における可用性および安全性の観点から議論している．ここでは，検証者が，検証に用いるデータを他のエンティティから入手できない場合にどのような検証が可能になるのかの検討を

行い，タイムスタンプ方式の詳細仕様を知ることなく，タイムスタンプ方式ごとの可用性と安全性を分類できることを明らかにしている．

しかしながら，従来の研究では，様々なタイムスタンプ方式のタイムスタンプが流通した後のタイムスタンプ検証にともなう課題に関しては十分に議論されているとはいい難い．

一般的に，タイムスタンプ検証者は，タイムスタンプ方式に応じた検証ソフトウェアをそれぞれ作成あるいは導入するという手間が発生する．今後，様々なコミュニティがオープンにつながれば，標準技術に準拠したタイムスタンプや独自方式のタイムスタンプなど様々なタイムスタンプが流通することが想定される．よって，検証者の利便性を高めると同時にタイムスタンプの相互運用性を確保することが重要である．

また，タイムスタンプに含まれる時刻情報の信頼性をどのように検証するのかという課題も重要である．ここでいう時刻の信頼性とは，タイムスタンプ局が，タイムスタンプ作成時に協定世界時（UTC: Coordinated Universal Time）に同期していた時計に基づき時刻情報を作成していたことを表す．従来は，この観点からの研究はほとんど行われていない．だが，実社会において，実際にタイムスタンプが使用されるようになってきていることをふまえると，タイムスタンプに含まれる時刻情報の信頼性を確認することは重要なことだと思われる．

このような背景の下，総務省は，e-Japan II 戦略の一環として，2003 年度から 3 カ年計画で，協定世界時（UTC）と同期する日本標準時（JST: Japan Standard Time）☆を利用し有効かつセキュリティの高いタイムスタンプを高速に付与することができる「タイムスタンプ・プラットフォーム技術」を確立するための研究開発を，産学官の連携のプロジェクト（以降，本プロジェクトと呼ぶ）により推進してきた[8)~10)]．研究開発のテーマの 1 つに，ユーザ利便性の向上とタイムスタンプの相互運用性の確保を目的としたタイムスタンプ検証技術の確立があった．

これに対応し，我々は，様々な方式によって作成されたタイムスタンプを時刻の信頼性も含めて統一的に検証する複数方式タイムスタンプ検証サーバを開発した．タイムスタンプ検証サーバは，本プロジェクトで開発された RFC 3161 方式のタイムスタンプサーバと ISO/IEC 18014-2 アーカイブ方式のタイムスタンプサーバのタイムスタンプを検証するサービスを提供する．タイムスタンプ検証サーバは，検証者の利便性を高めるために，タイムスタンプ検証を統一的に解釈し，検証処理を 4 つのサブ検証処理から構成されるとして抽象化する．時刻の信頼性の要件として，タイムスタンプ局が使用した時計が日本標準時（JST）に追跡可能であること，さらに，タイムスタンプに含まれた時刻情報と日本標準時の差分を後日確認できること，とした．

開発したタイムスタンプ検証サーバは，実証実験を通じて動作確認および性能確認が行われ，対話的に使用する環境では，実用性に問題がないことを確認した．

本論文の構成は以下のとおりである．2 章では，本プロジェクトで開発された時刻認証基盤技術実験装置におけるタイムスタンプ方式，タイムスタンプ検証に係るデータ検証プロトコル技術，およびタイムスタンプ検証にともなう課題について述べる．3 章では，タイムスタンプ検証サーバの目的と要件について述べる．4 章では，タイムスタンプ検証プロトコルの検討を述べ，5 章では，タイムスタンプ検証サーバの設計について記述する．6 章では，実装と実験について述べる．7 章では，考察を行い，8 章では，結論と今後の課題についてまとめる．

2. 時刻認証基盤技術実験装置におけるタイムスタンプ方式と検証の課題

時刻認証基盤技術実験装置では，国家時刻標準機関である NTA（National Time Authority），時刻配信局である TA（Time Authority），タイムスタンプ局である TSA（Time Stamping Authority）から構成されたシステムにて実現された 2 つの方式のタイムスタンプ方式が開発された．本章では，この 2 つのタイムスタンプ方式を説明する．また，タイムスタンプ検証に係るデータ検証プロトコル技術，およびタイムスタンプ検証にともなう課題を述べる．

2.1 用語の定義

本論文で使用する用語を以下に定義する．

タイムスタンプトークン：電子データの特徴量であるハッシュ値と時刻情報が暗号技術により結び付けられたもの．TSA は，タイムスタンプサービスの一環として，このデータを作成し，タイムスタンプ要求者へ送付する．本論文では，TST（Time-stamp Token）と略記する場合がある．

タイムスタンプ：時刻情報，あるいは，タイムスタンプトークンを示す．本論文では，特に，記載がなければ，タイムスタンプトークンと同じ意味を示す．

タイムスタンプトークン検証：タイムスタンプトークンの完全性（非改ざん性）を検証する．

タイムスタンプ検証：タイムスタンプトークン検証だ

☆ NICT は，UTC と高精度に同期する UTC（NICT）を生成する．UTC（NICT）にプラス 9 時間したものが，JST である．

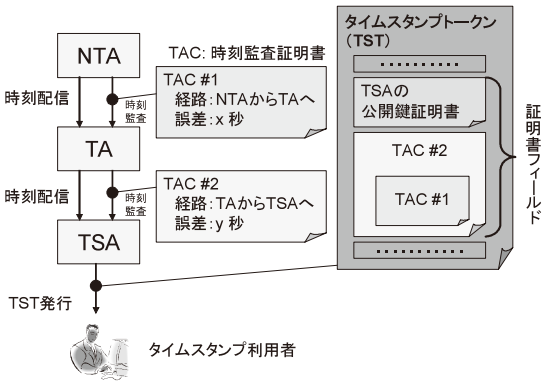

図 1 RFC 3161 ベースタイムスタンプ方式
Fig. 1 The RFC 3161 based time stamping scheme.

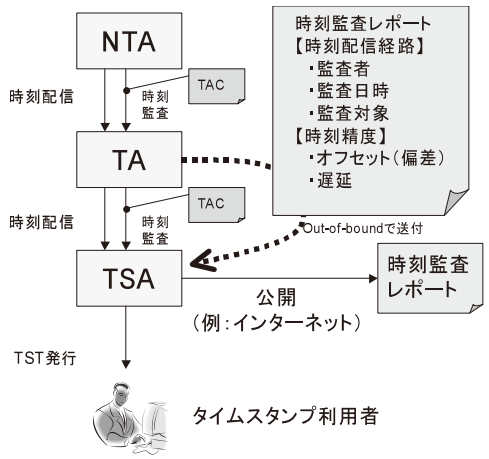

図 2 ISO/IEC 18014-2 アーカイブ方式ベースタイムスタンプ方式
Fig. 2 The ISO/IEC 18014-2 archive method based time stamping scheme.

けでなく，タイムスタンプトークンが付与された電子データの存在時刻と完全性（非改ざん性）を検証することも含む．

2.2 タイムスタンプ方式

2.2.1 RFC 3161 ベースのタイムスタンプ方式

本プロジェクトにおける RFC 3161 ベースのタイムスタンプを発行するタイムスタンプ局は，TA から正確な時刻配信サービスおよび時刻監査サービスを受けている．また，TA は，協定世界時（UTC）と高精度に同期する JST を生成する NTA から時刻配信サービスと時刻監査サービスを利用している．

時刻配信サービスとは，正確な時刻情報を提供するサービスであり，サービス享受者は，この時刻情報に基づいて，自身の時計を校正する．時刻監査サービスとは，サービス享受者の時計の正確性を監査するサービスである．サービス提供者は，監査結果として，時刻監査証明書（TAC: Time Attribute Certificate）と呼ばれる RFC 3281 準拠の属性証明書をサービス享受者に対して交付する．この証明書の中には，時刻配信経路や時刻精度に関する情報が含まれている．また，サービス提供者のデジタル署名が付与されている．

この RFC 3161 ベースのタイムスタンプの中には，TA から TSA に対して発行された時刻監査証明書が含まれている（**図 1** 参照）．また，この時刻監査証明書の拡張領域には，NTA から TA に対して発行された時刻監査証明書が含まれている．このように，NTA から TSA までの時刻配信経路と時刻の精度に関して確認できる情報が，タイムスタンプに含まれている．

なお，商用の時刻配信サービスの中には，時刻監査証明書を用いているものがある．たとえば，クロノトラスト^{TM}時刻配信サービス☆は，属性証明書を用いた時刻監査証明書を配信先へ送付する[11]．本論文の時刻監査証明書は，より上位の機関（NTA）の時刻監査証明書が入れ子構造的に含まれているため，上記の商用サービスの時刻監査証明書の拡張形式といえる．

2.2.2 ISO/IEC 18014-2 アーカイブ方式ベースのタイムスタンプ方式

本プロジェクトにおける ISO/IEC 18014-2 アーカイブ方式ベースのタイムスタンプを発行するタイムスタンプ局は，発行したタイムスタンプの完全性を保証するためにリンク技術を利用している．TSA は，タイムスタンプ発行時に発行したタイムスタンプと以前に発行したタイムスタンプのハッシュリンクを作成し，TSA 内で安全に保管する．

また，このタイムスタンプ局は，RFC 3161 ベースのタイムスタンプ局と同様に，TA から正確な時刻配信サービスおよび時刻監査サービスを受けている．さらに，TSA は，TA から定期的に，時刻監査レポートを Out-of-bound で受け取っている（**図 2** 参照）．

本プロジェクトにおける時刻監査レポートは，TA が作成するデジタル署名付きの PDF データである．この中には，一定期間における NTA から TSA への時刻配信経路および時刻精度に関する情報が含まれている．TSA は，受け取った監査レポートを保存し，インターネットから閲覧可能な状態とする．

本方式を基にした実サービスとして，SecureSeal® standard ☆☆がある[12]．このサービスでは，時刻配信

☆ クロノトラストは，セイコーインスツル株式会社の商標です．

☆☆ SecureSeal は，株式会社 NTT データの商標または登録商標です．

業務認定事業者[13]から時刻配信および時刻監査を受け，監査結果をインターネット上に毎月公開している．

2.3 検証プロトコル

2.3.1 ISO/IEC 18014-1 検証プロトコル

ISO/IEC 18014-1で規定されたタイムスタンプ検証プロトコルは，タイムスタンプトークン検証のためのプロトコルである．主に，リンクトークン方式のタイムスタンプのように，タイムスタンプを発行したタイムスタンプ局へタイムスタンプの検証を委任するときや，第三者検証サービスを提供する検証局へタイムスタンプの検証を委任するときに使用される．

2.3.2 DVCS プロトコル

DVCSプロトコルは，データ認証とデータ検証に関する4つのサービスを定義している．cpd（Certification of Possession of Data）サービスとccpd（Certification of Claim of Possession of Data）サービスは，任意の電子データの所有を認証するサービスである[14]．vsd（Validation of Digitally Signed Document）サービスは，デジタル署名文書を検証するサービスであり，vpkc（Validation of Public Key Certificates）サービスは，公開鍵証明書の検証サービスである．

DVCSプロトコルの大きな特徴は，複数の検証サービスを1種類の要求メッセージと応答メッセージで扱っていることである．そのため，複数の検証サービスを統一的に扱うための工夫が施されている．なお，DVCSプロトコルは，タイムスタンプ検証を明示的に扱っていない．

2.4 タイムスタンプ検証の課題

タイムスタンプ方式は，時刻認証基盤技術実験装置にて実装された2種類のタイムスタンプ方式だけでなく，様々な方式が存在する．たとえば，ISO/IEC 18014-3は，リンキング方式のタイムスタンプを定義している．また，商用のタイムスタンプサービスは，独自な方式のタイムスタンプを利用しているものもある[6]．

今後，様々な方式のタイムスタンプが流通するようになると，検証者は，様々な方式のタイムスタンプを検証する必要性が出てくる．

このとき，検証処理の実装が問題となる．なぜならば，検証者が操作するクライアントPC上で，様々な方式のタイムスタンプを検証する処理を実装する場合，タイムスタンプ方式ごとに検証ソフトウェアの開発・導入・運用の手間が発生する．また，検証処理を安全に実行するためには，検証者が信用する検証ソフトウェアが安全に動作する環境を用意しなくてはならない．企業内での検証利用シーンを考えた場合，これらのことは，検証者となる従業員の利便性の低下，従業員が操作するPCのTCO（Total Cost of Ownership）の増大，PCのセキュリティ確保問題などと見なすことができる．

さらに，タイムスタンプに含まれる時刻情報の信頼性に関する問題も十分に議論されていない．従来は，TSAは，TTP（Trusted Third Party）であるという理由から，タイムスタンプに含まれる時刻情報は，時刻精度が高く信頼性があると見なされてきた．そのため，検証者が自身でタイムスタンプに含まれる時刻情報の信頼性を検証することに関する検討は不十分であった．

今後，タイムスタンプが流通することを想定すれば，タイムスタンプ検証者は，タイムスタンプを取得した利用者とは異なる場合がある．このとき，タイムスタンプ検証者が，タイムスタンプ利用者と同様の信頼をTSAに対しておいているとは限らない．よって，タイムスタンプに含まれる時刻情報の信頼性を検証する方法を検討することは重要だと考える．

そこで，我々は，上記の問題を解決するタイムスタンプ検証サーバを設計・開発した．

3. タイムスタンプ検証サーバの目的と要件

本章では，タイムスタンプ検証サーバの目的と要件について述べる．

3.1 目　　的

タイムスタンプ検証サーバは，ユーザ利便性の向上とタイムスタンプ相互運用性を確保することを目的としている．様々な方式で作成されたタイムスタンプが流通した環境を想定し，以下の目的を設定した．

(g1) 統一的なタイムスタンプ検証

タイムスタンプ検証者がタイムスタンプ方式ごとに検証クライアントを用意・運用する手間を削減させること，また，様々な方式のタイムスタンプを統一的に検証することを目標とする．

(g2) 時刻信頼性検証

タイムスタンプ検証サーバは，タイムスタンプに含まれる時刻情報の信頼性を検証する機能を提供することも目標とする．

なお，タイムスタンプ検証サーバのモデルは，検証者の検証処理を代行するモデルとなる．タイムスタンプ検証サーバの想定運用者としては，TTPとしての検証局，統一的なポリシの下で検証を行う企業内利用，が考えられる．

3.2 要　　件

3.1節で述べた目的を達成するために，タイムスタンプ検証サーバの要件を以下のように定めた．

表 1 ISO/IEC 18014-1 検証プロトコルと DVCS プロトコルの比較
Table 1 Comparison between ISO/IEC 18014-1 verification protocol and DVCS protocol.

要件	項目		ISO/IEC 18014-1	DVCS
r1	統一的検証	タイムスタンプ検証	△（タイムスタンプトークン検証のみ）	△（新規検証サービス追加により対応可）
		検証結果の信頼性	×（検証応答に署名無し）	○（検証応答に署名あり）
r2	時刻信頼性検証		×	△（新規検証サービス追加により対応可）
r3	拡張性	タイムスタンプ方式追加	×	△（新規検証サービス追加により対応可）
		機能拡張	×	○（拡張領域の利用）

(r1) 統一的なタイムスタンプ検証

RFC 3161，ISO/IEC 18014-2 アーカイブ方式，商用の独自方式など様々な方式で作成されたタイムスタンプの妥当性を統一的に検証できるようにする．また，検証結果に信頼性を与えるとともに，検証者にタイムスタンプ方式を意識させないようにする．これにより，目的 (g1) を達成することができる．

(r2) 時刻の信頼性検証

タイムスタンプに含まれる時刻情報の信頼性を検証する．ここで，時刻情報の信頼性とは，タイムスタンプ作成時に TSA が参照する時刻情報が，協定世界時（UTC）と同期していたことを後日に確認できることとする．これにより，目的 (g2) を達成することができる．

(r3) 拡張性

検証機能のエンハンスに耐えうる拡張性を持つようにする．検証機能のエンハンスは，2 つの観点から考えられる．1 つは，検証対象としてサポートするタイムスタンプ方式を増やすこと，もう 1 つは，検証機能の高度化である．この拡張性は，要件 (r1) をサポートするものである．

4. タイムスタンプ検証プロトコルの検討

3 章の要件を備えたタイムスタンプ検証サーバを設計するにあたり，タイムスタンプ検証やデータ検証に係る標準的な仕様および公開された仕様を参考にすることとした．ベースとなるタイムスタンプ検証プロトコルの候補として，ISO/IEC 18014-1 と RFC 3029（Data Validation and Certification Server Protocols: DVCS）プロトコルに注目し，検討を行った．

タイムスタンプ検証サーバが満たすべき要件の観点から ISO/IEC 18014-1 検証プロトコルと DVCS プロトコルを比較した結果を**表 1** に示す．DVCS プロトコルと比較して，ISO/IEC 18014-1 検証プロトコルは，問題点が多い．

ISO/IEC 18014-1 検証プロトコルがかかえる課題を列挙すると，以下のとおりである．

(1) タイムスタンプ方式が限定される

検証対象として指定するタイムスタンプトークンには，作成メカニズムを示すオブジェクト識別子が含まれる必要がある．そのため，オブジェクト識別子を持たないタイムスタンプトークンをサポートできない．商用のタイムスタンプサービスの中にはオブジェクト識別子を持たないものもあるため，様々なタイムスタンプ方式を扱うことには問題がある．

(2) タイムスタンプトークン検証のみ

検証対象としてタイムスタンプ付与対象の電子データを含めることができない．そのため，タイムスタンプ検証者の一番の関心事であるタイムスタンプ付与対象の電子データの存在時刻と完全性を検証するサービスを提供できない．

(3) 検証結果の信頼性を示せない

検証結果の信頼性を示す情報を格納する領域が確保されていないため，検証クライアントに対して検証結果の信頼性を示すことができない．

(4) 独自の情報を扱うことができない

拡張データ領域が用意されていないため，独自の情報を扱うことができない．そのため，検証機能の拡張に耐えうるとはいい難い．また，時刻信頼性検証をサポートしたとしてもその結果を格納することは難しい．

一方，DVCS プロトコルでは，データ検証とデータ認証に係る 4 つのサービスを 1 種類の検証要求メッセージと DVC（Data Validation Certificates）と呼ばれるデジタル署名が付与された検証応答メッセージで表現している．つまり，サービスの種類ごとに異なる検証・認証に係るデータ項目を統一的に扱う枠組みを持っている．また，検証サービスの追加を許す構造を持っているとともに検証要求メッセージと検証応答

メッセージには，ユーザ定義の拡張情報を格納する領域が確保されている．

このように，DVCS プロトコルが持つ拡張性を利用すれば，タイムスタンプ検証サーバの要件を満たせる見通しが得られた．そこで，我々は，DVCS プロトコルをベースにタイムスタンプ検証プロトコルを設計することとした．なお，検証プロトコルとして，DVCS プロトコルを採用する方法としては，石本らの文献 15) がある．本論文でも同様のアイデアを採用する．

5. タイムスタンプ検証サーバの設計

本章では，タイムスタンプ検証サーバの設計について述べる．タイムスタンプ検証の抽象化，時刻信頼性検証仕様，タイムスタンプ検証プロトコル，共通インタフェース，そして検証クライアントについて記述する．

5.1 タイムスタンプ検証の抽象化

タイムスタンプ検証サーバ上で，統一的なタイムスタンプ検証を実現するために，タイムスタンプ検証内容を抽象化した．RFC 3161 や ISO/IEC 18014 などの標準化技術仕様に記載されたタイムスタンプ検証内容を分析し，タイムスタンプ方式に依存しない共通的な検証項目として，以下の (v1)，(v2)，(v3) の検証内容に整理した．我々は，これらの検証を総称してタイムスタンプの正当性検証と呼ぶ．さらに，4 つ目の検証処理 (v4) として，タイムスタンプに含まれる時刻情報の信頼性検証を定義した．

(v1) タイムスタンプトークンの形式検証

タイムスタンプ方式が規定するフォーマットに合致しているのかどうかを検証する．また，複数方式タイムスタンプ検証サーバが扱うことのできるタイムスタンプトークンなのかどうかを検証する．RFC 3161 準拠のタイムスタンプの場合，ASN.1 バイナリ符号化の妥当性，タイムスタンプトークンに含まれるデータ間の整合性，暗号アルゴリズムのサポート有無などを確認する．

(v2) タイムスタンプトークンの正当性検証

タイムスタンプトークン検証を表す．RFC 3161 準拠のタイムスタンプの場合，タイムスタンプに付与された TSA の署名や TSA の公開鍵証明書の検証を行う．また，ISO/IEC 18014-2 アーカイブ方式のタイムスタンプの場合，そのタイムスタンプを発行したタイムスタンプ局に問い合わせることによりタイムスタンプの改ざん有無を確認する．

(v3) タイムスタンプトークンと電子データの対応検証

タイムスタンプが使用するハッシュ関数を用いて電子データのハッシュ値を求め，その値が，タイムスタンプトークンに含まれる該当ハッシュ値と同一なのかどうかを確認する．

(v4) 時刻情報の信頼性検証

タイムスタンプに含まれる時刻情報の信頼性を確認する．

以上の検証のうち，(v1)，(v2)，(v3) の一連の検証により，タイムスタンプ付与対象の電子データが，ある時点に存在し，それ以降改ざんされていないことを確認することができる．さらに，(v4) の検証により，タイムスタンプに含まれる時刻情報の信頼性を確認することができる．

5.2 時刻信頼性検証の要件

タイムスタンプに含まれる時刻の信頼性とは，TSA がタイムスタンプ作成時に参照する時刻情報が，協定世界時（UTC）と同期していたことを後日確認できることである．そこで，時刻信頼性検証の要件として以下のように定義した．

(1) 時刻配信経路を確認できること

時刻配信経路とは，TSA を出発点として，TSA，TA，NTA までの時刻配信サービスに係るエンティティの経路を示す．時刻配信経路を確認することにより，タイムスタンプに含まれる時刻情報の時刻ソースとして，協定世界時（UTC）と高精度に同期する NTA に追跡可能なのかどうかを確認する．

(2) 時刻精度を確認できること

時刻精度を確認できるとは，タイムスタンプに含まれる時刻情報が，NTA により生成・配布・管理される JST とどの程度差があったのかを確認できることとする．

5.3 時刻信頼性検証方式

時刻認証基盤技術実験装置における TSA を対象モデルと見なし，タイムスタンプに含まれる時刻情報の信頼性を検証する方式として，2 つのモデルを策定した．1 つは，時刻監査証明書確認方式，もう 1 つは，時刻監査レポート確認方式である．

5.3.1 時刻監査証明書確認方式

時刻認証基盤技術実験装置における RFC 3161 ベースのタイムスタンプに対する時刻信頼性検証方式は，時刻監査証明書確認方式である．

この方式では，タイムスタンプ検証サーバは，タイムスタンプトークンに含まれる時刻監査証明書（TAC）の真正性の検査および TAC に記載された時刻配信経路と時刻精度の情報を検証する．TAC が入れ子構造の形で複数含まれている場合は，すべての TAC をふまえて検証する．検証手順は，以下のとおりである．

(1) TAC のデジタル署名を検証する

TAC に含まれる公開鍵識別子をキーとして，ディレクトリサーバから該当する公開鍵証明書を取得する．次に，この公開鍵を用いて，TAC のデジタル署名を検証する．

(2) 時刻配信経路を検証する

TAC の中には，時刻配信・時刻監査サービスの主体者と享受者のエンティティ名が含まれている．タイムスタンプ検証サーバは，時刻配信・時刻監査サービスに係るエンティティの連鎖を作成する．さらに，タイムスタンプトークンに含まれるタイムスタンプ発行者（TSA）と上記のエンティティの連鎖につながりがあるのかどうかを確認する．

(3) 時刻精度を検証する

TAC の中には，時刻監査対象の時刻精度が記載されている．時刻精度とは，監査実施エンティティの時刻と監査対象エンティティの時刻の偏差（d_i）を示す．タイムスタンプ検証サーバは，時刻配信経路における最上位のエンティティから TSA までの時刻精度の総和を計算する．次に，タイムスタンプトークンに含まれる時刻精度（$TST_{accuracy}$）が，上記で求めた総和よりも大きいことを確認し，タイムスタンプトークンに記載された時刻精度に虚偽がないことを検証する．

$$TraceVerify(tst) = \begin{cases} valid & \left(\text{if} \sum_{i \in \{TSA,\ TA\}} d_i \leq TST_{accuracy}\right) \\ invalid & \left(\text{if} \sum_{i \in \{TSA,\ TA\}} d_i > TST_{accuracy}\right) \end{cases}$$

5.3.2 時刻監査レポート確認方式

時刻認証基盤技術実験装置における ISO/IEC 18014-2 アーカイブ方式ベースのタイムスタンプに対する時刻信頼性検証方式は，時刻監査レポート確認方式である．

この方式のタイムスタンプの中には，時刻監査証明書（TAC）は含まれていない．そのため，TAC に基づいて時刻の信頼性を確認することはできない．そこで，タイムスタンプ検証サーバは，タイムスタンプトークンを発行した TSA を特定し，次に，その TSA がインターネット上で公開する時刻監査レポートの位置（URL）を検証者へ通知する．検証者は，通知された URL に基づき，Web ブラウザを用いて，時刻監査レポートにアクセスする．時刻監査レポートは，TA のデジタル署名が含まれた PDF 形式である．このように，本方式の信頼性は，TA のデジタル署名に基づいている．

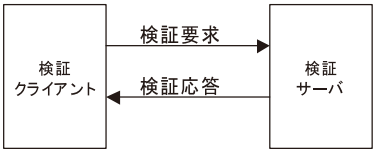

図 3 タイムスタンプ検証プロトコル
Fig. 3 Time-stamp verification protocol.

フィールド		データ型	説明
DVCSRequest		SEQUENCE	検証要求
	requestInformation	DVCSRequestInformation	検証要求情報
	data	Data	検証対象データ
	transactionIdentifier	GeneralName OPTIONAL	トランザクション識別子

フィールド		データ型	説明
DVCSRequestInformation		SEQUENCE	
	version	INTEGER	バージョン
	service	ServiceType	検証サービスタイプ
	nonce	Nonce OPTIONAL	ナンス
	requestTime	DVCSTime OPTIONAL	検証評価時刻
	requester	GeneralNames OPTIONAL	検証要求者
	requestPolicy	PolicyInformation OPTIONAL	検証ポリシー
	dvcs	GeneralNames OPTIONAL	検証サーバ
	dataLocations	GeneralNames OPTIONAL	検証対象データ位置
	extensions	Extensions OPTIONAL	拡張領域

図 4 タイムスタンプ検証要求フォーマット
Fig. 4 Time-stamp verification request format.

検証者は，Adobe® Reader®☆などのツールを用いて時刻監査レポートのデジタル署名を検証し，時刻監査レポートに記載された時刻配信経路や時刻精度を目視により確認する．

5.4 タイムスタンプ検証プロトコル

タイムスタンプ検証プロトコルは，タイムスタンプ検証を要求するクライアントとタイムスタンプ検証サーバ間で規定されるアプリケーションレベルの通信プロトコルである（**図 3**）．以降において，タイムスタンプ検証要求・応答メッセージを示す．

5.4.1 タイムスタンプ検証要求メッセージ

タイムスタンプ検証プロトコルにおけるタイムスタンプ検証要求メッセージは，RFC 2630 で定義された CMS ContentInfo である．ContentInfo 内の content は，RFC 3029 で定義された DVCSRequest を表す（**図 4**）．

我々は，RFC 3029 の DVCSRequest で定義される version，service，data の 3 つのフィールドを拡張した．バージョンを示す version は，RFC 3029 で指定された 1 ではなく，2 とした．サービスの種類を示

☆ Adobe Reader は，Adobe Systems Incorporated. の商標または登録商標です．

```
TimeStampInfo ::= SEQUENCE {
        tstType   ContentType   OPTIONAL,
        tst           OCTET STRING ,
        content   OCTET STRING OPTIONAL   }
```

図 5 TimeStampInfo の構造
Fig. 5 Structure of TimeStampInfo.

す service には，タイムスタンプ検証サービスを示す vtst(5) を追加した．data は，検証対象のデータを格納するフィールドである．ここでは，RFC 3029 で規定されている任意のバイナリデータを示す OCTET STRING データ型を持つ message フィールドを利用する．ただし，タイムスタンプ検証サーバは，検証要求メッセージに含まれる message フィールドのバイナリデータは，新規に設計した TimeStampInfo データ型を持つタイムスタンプ情報であると解釈する．

TimeStampInfo は，タイムスタンプ方式を示す tstType フィールド，タイムスタンプトークンのバイナリデータ列を示す tst フィールド，タイムスタンプ対象電子データのデータ列を表す content から構成される（**図 5**）．タイムスタンプ情報の必須フィールドは，tst フィールドだけである．そのため，検証クライアントは，検証対象データとして，タイムスタンプトークンだけを指定するだけでもよい．また，tst フィールドは，任意のバイナリデータ列を格納することができるので，任意の方式のタイムスタンプトークンを検証対象にすることができる．つまり，本検証プロトコルは，サポートするタイムスタンプ方式を容易に増やせるという拡張性を持つ．

5.4.2 タイムスタンプ検証応答メッセージ

タイムスタンプ検証プロトコルにおけるタイムスタンプ検証応答メッセージは，RFC 2630 で定義された CMS SignedData である．SignedData 内の encapContentInfo は，RFC 3029 で定義された DVCSResponse を表す（**図 6**）．

DVCSResponse は，dvCertInfo，あるいは，dvErrorNote から選択する．前者は，検証サービスの実行が成功したときの応答であり，後者は，検証サービスの実行に失敗した場合に発行される．なお，dvCertInfo が選択された SignedData は，DVC（Data Validation Certificate）と呼ばれる．

以下，検証結果を格納する dvCertInfo の詳細を説明する．

(1) dvCertInfo

タイムスタンプ検証サービス向けに拡張したフィールドは，version，dvStatus，certs，extensions である．version は，RFC 3029 で指定された 1 ではなく，2 とする．dvStatus と certs は，RFC 3029 の考え方と同様，検証結果を格納するフィールドとして使用する．したがって，dvStatus は，グローバルな検証結果を示し，certs は，詳細な検証結果を表すのに使用する．extensions は，タイムスタンプ方式に依存せずに共通的に存在する時刻情報やドキュメントハッシュ値などのタイムスタンプトークンの内容を示す情報を格納する．

フィールド		データ型	説明
DVCSResponse		CHOICE	検証応答
	dvCertInfo	DVCSCertInfo	検証結果
	dvErrorNote	DVCSErrorNotice	検証エラー通知

フィールド		データ型	説明
DVCSCertInfo		SEQUENCE	
	version	INTEGER	バージョン
	dvReqInfo	DVCSRequestInformation	検証要求情報
	messeageImprint	DigestInfo	検証対象データのダイジェスト値
	serialNumber	INTEGER	シリアル番号
	responseTime	DVCSTime	検証時刻
	dvStatus	PKIStatusInfo OPTIONAL	検証結果
	policy	PolicyInformation OPTIONAL	検証ポリシー
	reqSignature	SignerInfos OPTIONAL	検証要求者情報
	certs	SEQUENCE SIZE (1..MAX) OF TargetEtcChain	検証結果(詳細)
	extensions	Extensions OPTIONAL	拡張

図 6 タイムスタンプ検証応答フォーマット
Fig. 6 Time-stamp verification response format.

以下，拡張したフィールドを説明する．

(a) dvStatus

dvStatus は，グローバルな検証結果を示す．RFC 2510 で定義される PKIStatusInfo データ型である．グローバルな検証結果とは，タイムスタンプ検証サーバが実行する複数の検証処理結果から判断される総合的な検証結果である．

(b) certs

certs は，詳細な検証結果を格納し，複数の TargetEtcChain から構成される．1 番目の TargetEtcChain は，正当性検証の個々のサブ検証結果を格納する．2 番目は，時刻信頼性検証の詳細な検証結果を格納する．

実装した正当性検証結果の詳細情報を格納する TargetEtcChain の仕様は，**表 2** のとおりである．

target は，クライアントから送信される検証対象データの種類に応じて解釈される．検証対象として，タイムスタンプトークンだけが指定された場合は，タイムスタンプトークン自体の正当性検証結果を表す．一方，タイムスタンプトークンとタイムスタンプ付与対象となる電子データが指定された場合は，タイムス

表 2 正当性検証における TargetEtcChain の仕様
Table 2 Specification of TargetEtcChain for time stamp token validity verification.

フィールド名	説明
target	総合的な正当性検証結果を示す．PKIStatusInfo で表現する．
chain	正当性検証結果の詳細．以下の順番で並んでいる (1) タイムスタンプトークン形式検証 (2) タイムスタンプトークン正当性検証 (3) タイムスタンプトークンと電子データの対応検証
pathProcInput	-

表 3 RFC 3161 ベースのタイムスタンプの時刻信頼性検証における TargetEtcChain の仕様
Table 3 Specification of TargetEtcChain for time reliability verification for RFC 3161 based time-stamp.

フィールド名	説明
target	総合的な時刻信頼性検証結果を示す．PKIStatusInfo で表現する．
chain	正当性検証結果の詳細．以下の順番で並ぶ． (1) 時刻監査証明書の形式検証 (2) 時刻配信経路検証 (3) 時刻精度検証
pathProcInput	-

タンプトークン自体の正当性だけでなく，電子データの存在時刻と完全性の検証結果として解釈できる．

chain には，正当性検証の 3 つの検証内容を格納する．それぞれ，タイムスタンプトークン形式検証，タイムスタンプトークン正当性検証，タイムスタンプトークンと電子データの対応検証である．

RFC 3161 ベースのタイムスタンプトークンに係る時刻信頼性検証結果の詳細情報を格納する TargetEtcChain の仕様は，**表 3** のとおりである．

(c) extensions

extensions は，RFC 2459 で定義された Extension のシークエンスである．タイムスタンプ検証サーバは，2 つの Extension を使用する．1 つ目は，タイムスタンプ方式に依存しないタイムスタンプトークンの内容を示すデータである．このデータとして，RFC 3161 や ISO/IEC 18014 で規定される TSTInfo というデータを採用する．これにより，タイムスタンプフォーマットを解析できない検証クライアントに対して，タイムスタンプトークンが主張する時刻情報，時刻精度情報，ハッシュ情報（ハッシュ関数の識別子とハッシュ値）を提示することが可能になる．検証クライアントは，これらの情報に含まれるハッシュ情報を用いることで，タイムスタンプトークンと電子データの対応検証をローカルに実行することもできる．2 つ目は，時刻の信頼性に関するデータを格納する．このデータは，時刻配信経路および時刻精度に関する情報，あるいは，時刻監査レポートの位置（URL）を格納する．

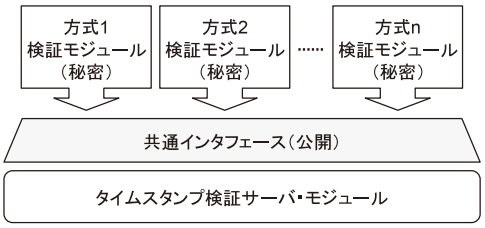

図 7 共通インタフェース
Fig. 7 Common interface.

5.5 共通インタフェース

商業的な観点からタイムスタンプサービスを考察すると，TSA 事業者は検証ソフトウェアを，クライアント側の様々なアプリケーションに対応させるため，ライブラリの形態で提供することが考えられる．さらに，TSA 事業者は，自身が提供するタイムスタンプサービスを普及させるため，その検証サービスを，タイムスタンプ検証サーバが提供するサービスの 1 つに追加してほしいという要望を持つと考えられる．一方，TSA 事業者は，自身の検証ライブラリのソースコードおよび内部仕様を企業秘密とし，検証サーバ事業者に公開したくないという要望もあると考えられる．

このような商業的な事情を想定し，**図 7** に示すように，TSA 事業者が提供する検証ライブラリとタイムスタンプ検証サーバを接続するためのラッパの役割を果たす共通インタフェースを設計した．共通インタフェースは，5.1 節で述べた抽象化された検証項目 (v1)，(v2)，(v3)，および (v4) をサポートしている．

これにより，TSA 事業者は共通インタフェースに則したラッパ・モジュールを開発することで，自身が保有する検証ライブラリの内部仕様を改造することなく隠蔽したまま，検証サーバがサポートするタイムスタンプに自身のタイムスタンプの追加を要求することが可能になる．よって，レガシー資産を有効活用できる．さらに，検証サーバ事業者側においては，タイムスタンプ方式特有の性質が共通インタフェースで吸収されるため，検証サービスの追加が容易となる．

共通インタフェースの仕様を**表 4** に示す．共通イ

表 4 共通インタフェースの仕様
Table 4 Specification of common interface.

(1) タイムスタンプトークンの形式検証	isSStoken(struct SSParam param, struct Result *result, int token_length, unsigned char *token);
(2) タイムスタンプトークンの正当性検証	SSVerify(struct SSParam param, struct Result *result, int token_length, unsigned char *token);
(3) タイムスタンプトークンと電子データの対応検証	SSVerifyWD(struct SSParam param, struct Result *result, int token_length, unsigned char *token, int doc_length, unsigned char *doc);

表 5 検証結果の構造
Table 5 Structure of verification result.

	型	説明
struct Result		
	struct TargetEtcChain	検証結果情報項目
	struct TSTInfo	タイムスタンプ情報項目
	struct TSTInfo_DER	DER形式のTSTInfoのバイト長
	struct TimeTraceabilityInfo	時刻トレーサビリティ情報
	char	メッセージ情報項目

ンタフェースの基本的な使用法は，共通パラメータ，対象データ（タイムスタンプトークンや電子データなど），および対象データ長を入力し，検証結果を**表 5** に示すような構造体 result へ格納する．

5.6 検証クライアント

検証クライアントは，タイムスタンプ検証プロトコルをサポートした専用クライアントとして設計されている．検証クライアントは，検証要求としてタイムスタンプトークンと電子データを含んだ要求情報をタイムスタンプ検証サーバへ送信する．また，検証要求としてタイムスタンプトークンのみを送信することも可能である．検証クライアントは，タイムスタンプ検証サーバから送信される検証結果に付与されたデジタル署名を検証し，検証結果の真正性を確認する．

検証クライアントは，タイムスタンプ検証プロトコルを実装したライブラリとユーザインタフェース部分に分離可能である．そのため，このライブラリを利用して検証者の利便性をさらに向上させることが可能である．たとえば，汎用 PC に標準的に備わっている Web ブラウザに対する社内向け Proxy サーバを構築することが可能である．このとき，Proxy サーバ上では，タイムスタンプ検証クライアントライブラリを利用したアプリケーションサーバを動作させて，Web ブラウザに対して HTTP ベースの検証サービスを代理的に提供する（検証結果を HTML として送信）ことが想定される．

なお，検証クライアントとして，汎用的な Web ブラウザやモバイルコード（Java Applet）などを用いる方法は，下記の点から問題があると思われる．汎用的な Web ブラウザの場合，デジタル署名を検証することができない．また，モバイルコードの場合，社内のセキュリティポリシのため，確実に利用できるとは限らないという問題がある．

6. 実装と実証実験

本章では，タイムスタンプ検証サーバの実装と実証実験について記述する．

6.1 実　　装

タイムスタンプ検証サーバを Sun Blade^{TM} 150，CPU：550 MHz × 1，メモリ：640 MB，OS：Solaris^{TM} 8 上で C 言語を用いて実装した☆．また，nShield^{TM} を用いて私有鍵などの暗号情報を保護するとともにデジタル署名作成などの暗号処理を実装した☆☆．

タイムスタンプ検証サーバは，RFC 3161 ベースのタイムスタンプ検証ソフトウェアをベースに開発した．TSA 公開鍵証明書の検証に関しては，トラストアンカとなる CA のルート証明書をあらかじめ検証サーバに導入するモデルとした．また，証明書検証結果として，GPKI の政府認証基盤相互運用性仕様書で規定されている証明書検証結果コードを使用した[16]．共通インタフェースの動作確認のため，RFC 3161 準拠以外のタイムスタンプ検証にこの共通インタフェースを利用した．

実装したタイムスタンプ検証サーバがサポートするタイムスタンプ形式は 2 つである．1 つは，デジタル署名技術を使用した RFC 3161 ベースのタイムスタンプである．もう 1 つは，リンク技術を使用した ISO/IEC 18014-2 アーカイブ方式ベースのタイムスタンプである．ISO/IEC 18014-2 アーカイブ方式ベースのタイムスタンプを検証するモジュールは，共通インタフェースの規約に基づいて構築されている．

☆ Sun，Blade，および Solaris は，米国および他の各国における Sun Microsystems, Inc. の商標または登録商標です．
☆☆ nShield は，英国 nCipher Corporation Ltd. の商標または登録商標です．

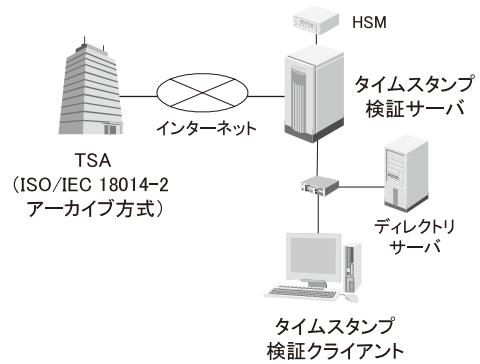

図 8 タイムスタンプ検証システム構成
Fig. 8 Time-stamp verification system configuration.

6.2 実証実験

図 8 のシステム構成で，タイムスタンプ検証サーバの実験を行った．タイムスタンプ検証サーバは，同一 LAN 内で Microsoft® Windows® XP ベースの検証クライアントとディレクトリサーバと接続している☆．タイムスタンプ検証クライアントとタイムスタンプ検証サーバ間は，HTTP を用いた．ディレクトリサーバは，RFC 3161 ベースのタイムスタンプに係る公開鍵証明書の失効リストを公開している．タイムスタンプ検証サーバは，このディレクトリサーバへアクセスし，公開鍵証明書の失効リストを取得する．また，タイムスタンプ検証サーバは，インターネットを介して ISO/IEC 18014-2 アーカイブ方式ベースの TSA と通信を行い，ISO/IEC 18014-2 アーカイブ方式ベースのタイムスタンプ検証の一部であるタイムスタンプトークンの正当性検証処理を委任する☆☆．

タイムスタンプ検証サーバの動作確認を行ったところ，2 方式のタイムスタンプ検証が可能であることが確認された．**図 9** は，検証クライアントのディスプレイ上に表示される RFC 3161 ベースのタイムスタンプに対する正当性検証に関する結果画面の例である．

検証結果画面の上側のエリアは，タイムスタンプの検証結果を示す．上から順に，総合的な検証結果，形式検証，正当性検証，タイムスタンプトークンと電子データの対応検証の結果が表示される．一方，検証結果画面の下側のエリアは，タイムスタンプトークンの内容を示す TSTInfo の情報である．このように，複数方式タイムスタンプ検証サーバは，複数の方式のタイムスタンプを統一的に検証した結果を発行していることが示される．

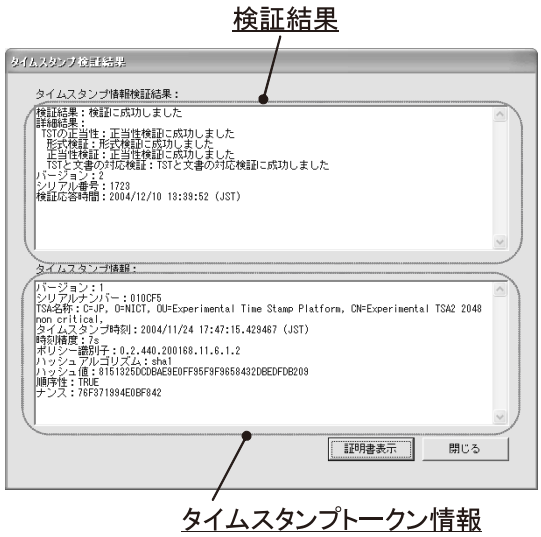

図 9 正当性検証結果画面
Fig. 9 A screenshot of time-stamp token validity verification result.

図 10 時刻監査証明書確認方式画面
Fig. 10 A screenshot of Time Attribute Certificate verification scheme validation result.

時刻信頼性検証に関しては，タイムスタンプの正当性検証が成功したものだけを対象として実施した．時刻監査証明書確認方式および時刻監査レポート確認方式ともに動作確認を行った（**図 10**，**図 11**，**図 12** 参照）．

RFC 3161 ベースのタイムスタンプに関しては，正当性検証および時刻信頼性検証を含む検証処理時間を

☆ Microsoft および Windows は，米国 Microsoft Corporation の米国およびその他の国における登録商標または商標です．

☆☆ ISO/IEC 18014-2 アーカイブ方式のタイムスタンプの場合，タイムスタンプ検証サーバは，タイムスタンプトークンの正当性検証 (v1)，タイムスタンプトークンと電子データの対応検証 (v2)，時刻情報の信頼性検証 (v4) を実施する．

図 11 時刻監査レポート確認方式画面
Fig. 11 A screenshot of Time Audit Report verification scheme validation result.

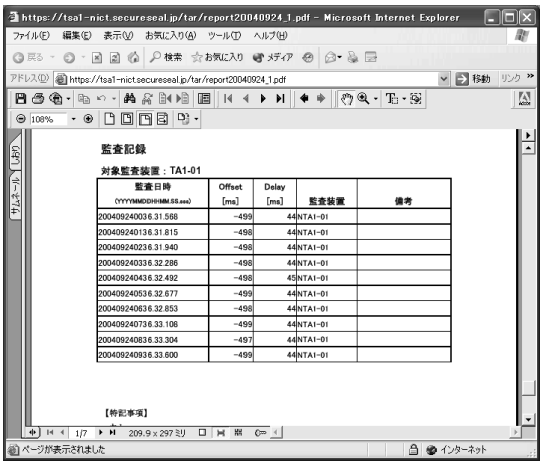

図 12 時刻監査レポート
Fig. 12 A screenshot of Time Audit Report.

計測した．タイムスタンプ検証サーバが，検証処理を受け付けてから検証応答を送信するまでの時間をサーバ内のログにより確認した．40 回の試行により，平均して 1.8 秒かかることが分かった．

7. 考　　察

開発したタイムスタンプ検証サーバについて，任意のタイムスタンプ方式の検証可能性，タイムスタンプ正当性検証と時刻信頼性検証の関係，性能，および安全性について考察する．

7.1 任意のタイムスタンプ方式の検証可能性

開発したタイムスタンプ検証サーバを用いて，RFC 3161 ベースのタイムスタンプと ISO/IEC 18014-2 アーカイブ方式ベースのタイムスタンプを統一的に検証することができた．

タイムスタンプ検証プロトコルの検証要求では，タイムスタンプ方式を特定する情報を必須としておらず，任意の方式のタイムスタンプを検証要求として含めることができる．そのため，タイムスタンプ検証プロトコルは，今回サポートしたタイムスタンプ方式だけでなく，商用の独自方式のタイムスタンプを含めた様々な方式のタイムスタンプも対応することができるといえる．

商用の独自方式のタイムスタンプ検証をサポートするために，共通インタフェースを開発した．今回は，共通インタフェースの実用性を確認するために，ISO/IEC 18014-2 アーカイブ方式ベースのタイムスタンプを検証対象とした検証モジュールを構築した．任意の方式のタイムスタンプをサポートできるのかどうかについては，タイムスタンプトークン情報のマッピングが可能かどうかに依存する．共通インタフェースは，タイムスタンプ検証モジュールに対して，TSTInfo 構造のデータ作成を求めている．TSTInfo には，バージョン，シリアル番号，ハッシュ値，時刻情報などが含まれるが，商用など独自方式のタイムスタンプの中には，これらの情報の一部を含まない，あるいは，表現が異なっている可能性がある．これらのマッピングに関しては十分に検討する必要がある．

7.2 正当性検証と時刻信頼性検証の関係

時刻信頼性検証は，タイムスタンプの正当性検証の補完となる．タイムスタンプ検証サーバは，タイムスタンプの正当性を確認した後，時刻信頼性検証を実行する．つまり，タイムスタンプ検証サーバは，タイムスタンプの正当性が確認できない場合，時刻信頼性検証を実行する必要性はないと判定する．

タイムスタンプ正当性検証と時刻信頼性検証は，検証可能期間が異なる可能性がある．たとえば，本プロジェクトにおける RFC 3161 方式ベースのタイムスタンプの場合，タイムスタンプ正当性検証の検証可能期間は，タイムスタンプに付与されたデジタル署名の検証可能期間となる．これは，通常，デジタル署名を検証するための公開鍵証明書（TSA の公開鍵）の有効期間が該当する．一方，時刻信頼性検証の検証可能期間は，TAC に付与されたデジタル署名の検証可能期間となる．この期間は，TAC を発行した TA や NTA の公開鍵証明書の有効期間となる．また，本プロジェクトにおける ISO/IEC 18014-2 アーカイブ方式のタイムスタンプの場合，タイムスタンプ正当性検証の検証可能期間は，TSA のサービス継続期間となる．さらに，

時刻信頼性検証の検証可能期間は，時刻監査レポートのデジタル署名検証可能期間（TA の公開鍵証明書の有効期間）となる．このような検証可能期間の違いにより，タイムスタンプの正当性検証が可能であったとしても，時刻信頼性検証が実行できないという状況が想定される．

7.3 性　　能

RFC 3161 ベースのタイムスタンプに対して，開発したタイムスタンプ検証サーバは，検証クライアントとの間の通信時間を除き，タイムスタンプ正当性検証および時刻信頼性検証に関しては，1 検証要求あたり，1.8 秒以内で検証を実行した．ユーザが GUI などを介して対話的に検証する方法であれば，実用性があると思われる．なお，アプリケーションの要件によっては，より高性能を求められる可能性がある．たとえば，2005 年 4 月から施行された e-文書法においては，国税関係書類にタイムスタンプを付与することが求められている．ここでは，後日，複数のタイムスタンプを一括して検証することが要求されるため，検証速度が重要になる．検証速度向上に関しては，今後の課題である．

なお，開発したタイムスタンプ検証サーバの設計・実装では，スケーラビリティに係る考慮は十分には行われていない．しかしながら，以下で示すように，スケーラビリティ確保のための見通しおよびタイムスタンプ検証サーバならではの効果があると考える．同時アクセス数への対応としては，既存の技術であるサーバの負荷分散技術（例：負荷分散装置の利用）の手法を用いることが考えられる．また，タイムスタンプ検証特有の処理として，コストの高い公開鍵証明書処理がある．公開鍵証明書検証では，証明書パス構築，証明書パス検証など複雑な処理から構成されており，クライアントによる検証処理よりも，キャッシュ機能を利用したサーバ側で中央集中的に処理する方が高速になると報告されている[17]．本タイムスタンプ検証サーバも公開鍵証明書検証に係る同様なキャッシュ機能を利用しており，クライアント側で検証するよりも，高速に検証できることが期待できる．

7.4 安　全　性

タイムスタンプ検証サーバは，タイムスタンプ検証要求者の検証を代行するサービスを提供する．さらに，検証結果には，タイムスタンプ検証サーバの署名が付与されるため，その検証結果は，一種の証明書となる．これらのことから，タイムスタンプ検証サーバの安全性の確保は重要である．

タイムスタンプ検証サーバの安全性確保としては，一般的なサーバに対する安全性およびタイムスタンプ検証サーバに特化した安全性に関する視点がある．一般的なサーバの安全性確保の観点からは，(1) 入退出管理が行われた物理的に安全な場所にサーバを設置することで第 3 者からの物理的攻撃を防止すること，(2) 適切な情報セキュリティ教育により管理者と運用者の故意や不注意による脅威を抑止すること，(3) OS やネットワークなどの脆弱性に対して適切に対処することでネットワーク経由の脅威に対処すること，などがあげられる．

タイムスタンプ検証サーバ特有の安全性確保に関しては，以下の問題を考慮する必要があると考える．1 つは，タイムスタンプ検証サーバが利用する暗号技術に係る問題である．タイムスタンプ検証サーバは，検証結果に対してデジタル署名を付与するため，その署名に係る鍵管理の安全性を確保することが重要となる．また，タイムスタンプ検証サーバは，脆弱化した暗号アルゴリズムが使用されたタイムスタンプトークンを検証対象とした場合，悪意者にとって都合の良い検証結果を作成してしまう可能性がある．これに関しては，暗号技術の安全性を厳格に評価・監視する機関（たとえば，CRYPTREC[18]）が公開する情報を監視し，暗号アルゴリズムの安全性をつねに把握する体制が必要だと思われる．なお，デジタル署名の危殆化にともなうデジタル署名検証可能性に関する先行研究[19]があるため，これらの知見をふまえ，タイムスタンプの長期検証可能性についての検討も必要だと思われる．暗号技術の脆弱化をふまえたタイムスタンプ検証技術に関しては，今後の課題である．

もう 1 つは，タイムスタンプ検証要求データに含まれる可能性のあるタイムスタンプ対象データの機密性確保の問題である．タイムスタンプ検証要求者から送信されるタイムスタンプ対象データの中には，営業秘密や個人情報が含まれる可能性がある．タイムスタンプ検証サーバは，検証処理の一環として，これらの情報の内容を解釈せずにビット列として扱い，検証後は，この情報を削除している．このように，機密性が必要になると思われるタイムスタンプ対象データの扱い方に関しては十分考慮する必要がある．

なお，タイムスタンプ検証サーバを利用した第 3 者サービス機関の信頼性確保に関しては，ISMS 認定制度やプライバシマーク制度などを活用し，情報セキュリティ運用管理に係る認証を得ていることを対外的にアピールすることがあげられる．また，タイムスタンプ検証サービスに係る運用規程も公開し，検証要求者がこれらの情報を閲覧できることも重要と考える．

8. ま と め

タイムスタンプの統一的な検証およびタイムスタンプの時刻信頼性の検証を実現するタイムスタンプ検証サーバを開発した．拡張性を重視し，DVCS プロトコルを拡張したタイムスタンプ検証プロトコルを設計・開発し，任意の方式で作成されたタイムスタンプを統一的に検証できるようにした．また，検証対象とするタイムスタンプ方式において商用サービスの独自方式を含めることを目標とし，非公開の検証モジュールをタイムスタンプ検証サーバにアドオンするために使用できる共通インタフェースを設計・開発した．さらに，時刻信頼性検証方式として時刻監査証明書確認方式と時刻監査レポート確認方式を設計・開発した．

実装したタイムスタンプ検証サーバを用いて RFC 3161 ベースのタイムスタンプと ISO/IEC 18014-2 アーカイブ方式ベースのタイムスタンプの検証動作を確認し，対話環境で実用性があることを確認した．

今後は，さらなる実用化を目指し，以下の研究課題に取り組む必要があると考える．1 つは，検証処理の高速化である．また，安全性の観点からは，脆弱化した暗号アルゴリズムが使用されたタイムスタンプ検証可能性に関する研究が必要である．さらに，本研究開発成果を普及させるためには，開発技術を標準化する活動も必要であると思われる．

謝辞 本研究は，総務省が推進しているタイムスタンプ・プラットフォーム技術の研究開発における実証実験プロジェクトの中で行われたものである．

参 考 文 献

1) Schneier, B.: *Applied Cryptography: Protocols, Algorithms and Source Code in C*, 2nd Edition, Wiley (1995).
2) Adams, C., Cain, P., Pinkas, D. and Zuccherato, R.: Internet X.509 Public Key Infrastructure Time-Stamp Protocol (TSP)-RFC 3161, IETF (2001).
3) ISO/IEC: ISO/IEC 18014-1:2002 Information technology — Security techniques — Time-stamping services — Part 1: Framework (2002).
4) ISO/IEC: ISO/IEC 18014-2:2002 Information technology — Security techniques — Time-stamping services — Part 2: Mechanisms producing independent tokens (2002).
5) ISO/IEC: ISO/IEC 18014-3:2004 Information technology — Security techniques — Time-stamping services — Part 3: Mechanisms producing linked tokens (2004).
6) 梅澤淳子，上野祐之，宮田幸夫，佐井川泰治，江並孝之，吉岡克成，松本 勉：インターネット官報データ提供サービスにおけるディジタル証拠性の保証，情報処理学会論文誌，Vol.45, No.8, pp.1954–1965 (2004).
7) 宇根正志，松本 勉：可用性および安全性の観点からみた各タイムスタンプ方式の関係，情報処理学会論文誌，Vol.43, No.8, pp.2644–2658 (2002).
8) 総務省：平成 15 年度版情報通信白書 (2003). http://www.johotsusintokei.soumu.go.jp/whitepaper/ja/h15/index.html
9) NICT：時刻認証基盤技術実験装置（Experimental System of Time Stamping Based on the Japan Standard Time）仕様書 (2003).
10) 総務省：タイムスタンプ・プラットフォーム技術の研究開発 (2006). http://www.soumu.go.jp/menu_02/ictseisaku/ictR-D/051020_2_1_5.html
11) セイコーインスツル株式会社：クロノトラスト時刻配信サービス運用規程 Ver.2.0, 2005 年 3 月 1 日.
12) 株式会社 NTT データ：SecureSeal®standard. http://www.secureseal.jp/
13) インターネットマルチフィード株式会社：TimeFEED サービス. http://www.mfeed.co.jp/tfeed/
14) Adams, C., Sylvester, P., Zolotarev, M. and Zuccherato, R.: Data Validation and Certification Server Protocols-RFC 3029, IETF (2001).
15) 石本英隆，小野 諭，堀田英一：イベント順序証明技術を用いた長期有効性保証タイムスタンプ，信学技報 ISEC2004-86, pp.9–14 (2004).
16) 総務省：政府認証基盤（GPKI）政府認証基盤相互運用性仕様書 (2003).
17) 藤城孝宏，鍛 忠志，橋本洋子，手塚 悟：証明書検証サービスの開発，電子情報通信学会論文誌 D-I, Vol.J87-D-I, No.8 (2004).
18) CRYPTREC. http://www.cryptrec.jp/
19) 田村裕子，宇根正志，岩下直行，松本 勉，松浦幹太，佐々木良一：デジタル署名の長期利用について，金融研究 24 巻別冊 1 号，日本銀行金融研究所，pp.121–176 (2005).

(平成 18 年 11 月 27 日受付)
(平成 19 年 6 月 5 日採録)

谷川 嘉伸（正会員）

1993 年京都大学大学院理学研究科修士課程修了．同年（株）日立製作所に入社．以来，システム開発研究所にて，グループウエア，企業間 EC システム，情報セキュリティシステムに関する研究開発に従事．

手塚　悟（正会員）

1984 年慶應義塾大学工学部数理工学科卒業．同年（株）日立製作所入社．マイクロエレクトロニクス機器開発研究所に勤務し，パーソナルコンピュータのオペレーティング・システム，デバイス・ドライバ，LAN システム等の研究開発に従事．その後，システム開発研究所に勤務．以来，パーソナルコンピュータを中心とした LAN システムの構築・運用管理の研究開発，さらにセキュリティシステムの研究開発に従事し，現在，システム開発研究所第七部部長．工学博士．

小黒　博昭（正会員）

1998 年東京工業大学大学院総合理工学研究科物理情報工学専攻修士課程修了．同年 NTT データ通信株式会社（現，株式会社 NTT データ）入社．以来，同社技術開発本部にて暗号応用，PKI，セキュアシステム開発に関する研究に従事．2000 年から 2002 年 NTT 情報流通プラットフォーム研究所勤務．電子情報通信学会，日本ソフトウェア科学会各会員．

橋川　善之

1997 年東京大学大学院工学系研究科物理工学専攻修士課程修了．同年 NTT データ通信株式会社（現，株式会社 NTT データ）入社．同社ビジネスソリューション事業本部にてタイムスタンプの研究開発に従事．現在，NTT データ・セキュリティ株式会社企画部勤務．