社団法人　電子情報通信学会
THE INSTITUTE OF ELECTRONICS,
INFORMATION AND COMMUNICATION ENGINEERS

信学技報
TECHNICAL REPORT OF IEICE.

時刻認証グリッドの構築と基礎実験

西川　武志†　　松岡　　聡†,††

† 東京工業大学　〒 152–8559 東京都目黒区大岡山 2–12–1
†† 国立情報学研究所　〒 101–8430 東京都千代田区一ツ橋 2–1–2
E-mail: †t.nishikawa@gsic.titech.ac.jp, ††matsu@is.titech.ac.jp

あらまし　デジタルデータがある時点で存在したこと，改竄されていないことを証明する有効な手段にデジタル署名による時刻認証法がある．従来の単独時刻認証局には運用コストや性能スケーラビリティの問題が存在し，その解決を試みた従来の分散時刻認証法には多数の時刻認証局をどうやって設置するかの問題が存在する．既存のこれらの問題を，我々は，多数の時刻認証ユニットを用いて相互に時刻認証を行う $K = L + M$ among N for G–generation 分散時刻認証法で解決出来ることを示して来た．またローカルネットワーク上のクラスタ環境で動作パラメータ依存性等の基本動作を検証して来た．今回はインターネット上の複数のサイトで分散時刻認証ユニットを設置し，すなわち分散時刻認証グリッドを構築し，ネットワーク遅延時間が認証時刻にどのような影響があるかを調査した．さらにその結果から認証時刻を，応答があった場合の算術平均とした場合に時刻認証ユニット数を幾つ用いすれば 1 秒以内となるかを検討した．その結果，本報告で検討した動作パラメータでは TSU が 256 よりも多く用意すれば認証時間の算術平均が 1 秒以内となることが明らかとなった．最頻値を用いることで算術平均よりも遅延の小さな認証時間を得られる事が明らかとなった．

キーワード　時刻認証，タイムスタンプ，グリッド

Building Time-Stamp Authority Grid and Basic Experiment

Takeshi NISHIKAWA† and Satoshi MATSUOKA†,††

† Tokyo Institute of Technology　2–12–1 Oookayama, Meguro-ku, Tokyo, 152–8550 JAPAN
†† National Institute of Informatics　2–1–2 Hitotsubashi, Chiyoda-ku, Tokyo, 101–8430 JAPAN
E-mail: †t.nishikawa@gsic.titech.ac.jp, ††matsu@is.titech.ac.jp

Abstract　Digital time stamping is a technique to prove the existence of a digital data prior to a specific point in time. The centralized time-stamping authority has two major difficulties. One is administrative costs and another is scalability of performance. Distributed time-stamping methods were proposed to solve the problems. But they still have the problem to prepare many TSAs. We have shown that $K = L + M$ among N for G–generation method is able to solve these existing problems. That method does the time-stamping to the mutuality by using many time-stamping units. And we also reported basic characteristics of the method and dependence on configuration parameters at the computer cluster environment on the LAN. In this report, we described that we built TSA Grid on the Internet to install several distributed TSU. And we investigated the influence of network time delay to the authorized time . We also considered that how many time-stamping units enable to be arithmetic mean value of responding authorized times within 1-second. As that result, if it prepared more than 256 TSU that arithmetic mean of the authentication time became within 1 second. It became clear that the mode is able to get smaller delay time than that of arithmetic-mean.

Key words　time-stamping, time-stamp, Grid

1. はじめに

税法や商法等の各種法令により，民間企業が作成・保存することを義務付けられている文書や帳票類の電磁気的記録を認める法律，通称 e-文書法が 2005 年 4 月 1 日に施行された．では真正性，すなわち他人のなりすまし防止，文書の改ざん防止，

事実発生の事後否認の防止が求められ，それには作成者の電子署名だけでは不十分であり，電子署名に正確な時刻データを付与するタイムスタンプの利用が必須であり，普及が始まっている．

他方で，デジタル時刻認証は科学技術分野でも知的財産優先権の確保や不正が行われていないことの証明に役立つことが期待されるにもかかわらず，その経済性，高コストであることが第一の要因となって普及していない．実験機器の電子化，デジタル化等により出力がデジタルデータとしてのみ保存され，またバイオインフォマティクスのように学問手法自体が，従来の紙媒体の実験ノートに保存しきれない大量のデータが生成される場面では，元のデジタルデータのハッシュを生成し，元のデータそのものを第三者に渡さずに存在や非改竄の証明ができるデジタル時刻認証は原理的に有用な手段と分かっていながら普及していない．様々なイベント毎に個別にタイムスタンプを発行出来れば使い勝手が向上するにもかかわらず，費用の点からまとめざるを得ない．

高コストが解消されたとしても科学技術分野での個別イベント毎に大量のタイムスタンプ要求がなされると現状の集中型時刻認証局（Time-Stamping Authority:TSA）では性能スケーラビリティに問題がある．

時刻認証局の分散 DoS 攻撃耐性や性能スケーラビリティの問題を解決する手段として複数の TSA を利用する分散時刻認証法がこれまで提唱されている. [5]〜[7] しかしこれらの手法では多数の TSA を用意しなければならず，設置・運用・監査のコストの問題が存在する．定期的な監査の間隔中での時刻改ざん等の不正に対する対策も考慮されていない．

これまで我々は $K = L + M$ among N for G–generation と名付けた本分散時刻認証手法を提案し，時刻認証を利用したい者が相互に多数の時刻認証ユニット（Time-Stamping Unit:TSU）を設置し，互いに時刻認証し合う事で，いつでもどこでも時刻認証が安全に利用出来る時刻認証のためのグリッド構築の仕組みを提唱して来た．分散時刻認証局グリッド，TSA Grid は，時刻認証を利用したい TSU が集まって形成しており，これまでに提唱された分散時刻認証法に対して設置・運用・監査のコストの問題を解決する方法となっている．

同様に我々は Java servelt として実装した，TSA Grid の実装系 tsagrid ミドルウェアにより，その動作パラメータ K, L, M, N, G に対する動作依存性をローカルネットワーク上のクラスタ環境で調査して来た．その結果，毎秒百万スタンプの発行が TSA Grid 全体で可能である事を示した（図 1）．

本報告では，$K = L + M$ among N for G–generation 分散時刻認証手法を実装した TSA Grid をインターネット上に構築し，ローカルネットワーク上のクラスタ環境での同一動作パラメータで動作させた場合に，ネットワーク遅延時間の影響がどのように現れるかの検証を行った．

2. 既存時刻認証法の問題

既存の時刻認証局をネットワーク上に分散させて実現するという提案の多くは，時刻認証局の信頼性の確立の視点に重きを置いたものが多い．

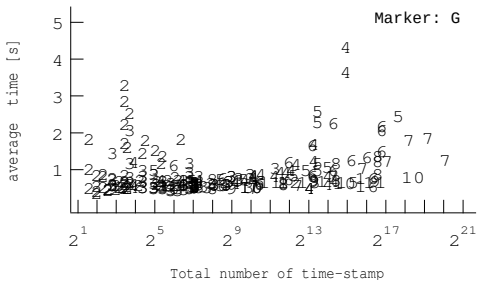

図 1　動作パラメータを変化させたときの認証時刻の平均値と総発行可能タイムスタンプ数への依存性

Fig. 1　Dependency of average response times with various configuration parameters v.s. total number of time-stamp.

一定数以上の時刻認証局の署名が集めて，多数決，平均値等でタイムスタンプを生成すると定義するものでは，多数が結託あるいは障害が発生しない限り，タイムスタンプを偽造・改竄することが困難であることに基づいている．

これらの分散時刻認証局の仕組みでは各々の時刻認証局が信頼できなくても，多数の時刻認証局が結託する可能性は比較的小さいと考えられるため安全な時刻認証が可能であるというものである．この代表的なものとして秘密鍵分散技術を利用した NTT の分散時刻署名システムがあげられる．[4] これは如何に安全に秘密鍵の部分鍵を安全に複数の TSA に配るかという問題が存在し，かつ複数 TSA の個数は秘密鍵長を部分鍵長で分割する個数となり，非常に少数の TSA から構成され性能スケーラビリティの問題が存在する．これに対し A. Bonnecaze 等が提案する k among n スキーム [5], [6] や D. Tulone が提唱する RSTS [7] では n 個の時刻認証局から k 個を乱数で選択し多数決により時刻認証を行う仕組みである．この分散時刻認証局に分散 DoS 攻撃を行う者は n サーバすべてに攻撃をかけねばならず，分散 DoS 攻撃に耐性があると唱えている．しかしながら A. Bonnecaze 等が提案する k among n スキームでは認証される時刻の期待値の計算が困難である．D. Tulone が提唱する RSTS では client から TSA までの round trip time(RTT) の平均値で期待値が計算出来るとしている．表 1 に東工大からの主たる公開 TSA サイトおよび TSA Grid サイトへの RTT を示す．

しかし，応答しない TSA サイトの RTT は無限大となり，そういったサイトが一つでもあれば期待値の平均値は無限大となってしまう．

またこれらの手法では分散時刻認証法で必要となる複数個の TSA の，設置・運用・監査のコストの問題に対する解決策を示していない．また定期的な監査の間隔中での時刻改ざん等の不正に対する対策も考慮されていない．

3. 複数世代分散時刻認証手法：$K = L + M$ among N for G–generation 法

本報告で提案する TSA Grid では，ネットワーク上に多数の時刻認証要素（Time Stamping Unit:TSU）が相互に時刻認証

表 1　東工大からの主たる公開 TSA サイトおよび TSA Grid サイトへの RTT
Table 1　RTT from Tokyo-Tech to some major public TSA sites and some TSA Grid sites

target	min	avg	max	stddev
www.opentsa.org	317.9	321.5	328.9	2.4
tsp.iaik.at	294.9	295.2	295.4	0.1
www.edelweb.fr	263.9	264.7	268.4	1.3
dse200.ncipher.com	255.3	257.0	259.4	1.2
gcxp011.exp-net.osaka-u.ac.jp	11.2	11.4	11.7	0.1
roboc03.roboc.com	8.4	9.5	12.6	1.0
fs.qcgrid.jp	6.8	9.1	18.3	3.9
local network segment	0.1	0.2	0.5	0.1

し合うことで同時に多数の時刻認証局が結託して時刻を詐称することが困難であること，同じく多数の時刻認証局がネットワーク上に分散していることで分散 DoS 攻撃に耐性を持つであることの仮定に基づいている．

TSA Grid は，時刻源としてインターネット上の NTP [8] サーバを用いた TSU が，互いに一つの時刻認証要求に複数 TSU が複数世代に渡って，NTP により同期した絶対時刻をデジタル署名するタイムスタンプを発行する．全体としてお互いに発行し合ったタイムスタンプに含まれる時刻に基づいて自律的に遅延や時刻の誤りが無いかを評価し合い，信用出来る TSU のリストを形成して行くことで分散時刻認証局として監査プロセスをその動作に内在させている．一つの時刻認証要求に複数のタイムスタンプ発行され，そのタイムスタンプに含まれる時刻に関して多数決や平均値や最頻値を採用することでデジタルデータの時刻認証を行う TSA を構成する．

TSA Grid で発行されたタイムスタンプの検証は，分散時刻認証局から源となるタイムスタンプから連環して複数世代に渡って発行された多数のタイムスタンプに記録された時刻から多数決や平均値や最頻値などの統計処理値に基づいている．従って TSA Grid のタイムスタンプは，タイムスタンプ間の順序を保証するものでは無い．TSU 数が増えれば増えるほど統計値の分散が低減する．本方式には TSU の総数 N，世代数 G，信頼できる TSU のリストの要素数 L，$(N-L)$ 個の TSU からランダムに選ぶ TSU の数 M という設定パラメータが存在する．各世代毎のタイムスタンプ発行要求数 K は L と M の和になっている．世代数 G と信頼出来る TSU のリストの要素数 L の導入が既存の単一 TSA や分散 TSA に対する優位性をもたらしている．

一つの時刻認証要求に複数 TSU が G 世代に渡ってタイムスタンプを発行することで，既出の分散時刻認証法より少ない TSU の総数 N での運用が可能になりコスト削減が図れる．時間の一方向性により複数世代に渡って発行されたタイムスタンプが認証する時刻は後の世代のものは前の世代のものより後の時間となることが要請される．ここに世代が異なるのに全く同じ時刻や後世代が前世代より先の時刻であればその二つの TSU 間でどちらかの時刻に問題があることが検出可能となる．問題が有った TSU 以外で多数決を取ることで問題のある TSU を特定し修正を行うことが可能になる．

信頼できる L 個の TSU のリスト形成は独立した多数の時刻認証要求の履歴により各 TSU 毎に独立したリストが，第三者によって事前に予測することができないタイミングで形成される．従って既存の TSA に存在する監査と次の監査の間で時刻改ざん等の不正の可能性を本手法では排除することが出来る．タイムスタンプ発行と検証を相互に行うこと自体が TSU 相互の監査になる．すなわち on-the-fly で監査をし合うことで，既存 TSA の定期的監査を不要とし，監査間の時刻詐称可能性を排除し，かつ定期監査のコストを削減出来る．既存の分散 TSA では認証される時刻の期待値の推計が困難であるが，本手法では L 個の TSU から得られた統計量から認証される時刻の推定が可能となる．

3.1　TSA Grid の要求要件

TSA Grid は，以下の要求，仕様を満たす．

- 各 TSU は RFC3161，”Internet X.509 Public Key Infrastructure Time-Stamp Protocol” [1] に互換の Simple Time-Stamp Unit ととして動作する．したがって各 TSU が発行するタイムスタンプは一意のシリアル番号を持つ，
- 各 TSU は TSA Grid 運用規則に則って独立に管理される，
- 各 TSU は独立にインターネット上の NTP サーバと時刻を同期すること，
- 各 TSU は互いに監査し合う，
- 各 TSU は特定のセキュリティハードウェアを使用することなく，汎用の計算機上で動作可能であること，
- 各 TSU は絶対時刻をある一定の精度でタイムスタンプとして応答可能であること，
- 簡便，効率的，頑強な統計的処理に基づく検証プロトコルを持つこと，
- Denial of Service (DoS) 攻撃に耐性を持つこと，
- 同時多数のリクエストに応えられるようにすること，
- ネットワーク切断・重大遅延等のネットワーク障害に耐性を持つこと，
- タイムスタンプの検証プロセスには十分に長い時間を要しても統計的に十分なサンプルを集めること，
- 発行プロセスは検証プロセスよりプライオリティを高くすること，
- ネットワークトラフィックを過度に消費しないこと，
- 時刻が過失により正確でないノードや悪意を持った第三者による時刻を改ざんしたノードが存在することへの耐性を持つこと
- TSA Grid 全体として単一障害点を持たないこと．

TSA Grid はタイムスタンプを利用したい主体がタイムスタンプを要求，発行し合うとともに相互に監査を行う TSU を用意することで形成される相互結合ネットワークから構成される．時刻認証を利用したい者が TSU を用意し集まることで TSA Grid を形成する．従って既存の分散時刻認証法が抱えていた多数の TSU を如何にして用意するかという問題への解決法を提示している．また各 TSU が RFC3161 のインターネット標準

に準じることで既存の RFC3161 の時刻認証局ならびにその利用者を TSA Grid へ参加出来ることも目指している．さらに世代数 G を導入することで既存の分散時刻認証法よりも少ない TSU の個数で多数のタイムスタンプ発行が可能となる．

時刻源に関して「インターネットマルチフィード (MFEED) 時刻情報提供サービス for Public」 [11] 等の様にインターネットを通じて高精度な時刻情報を無償にて提供するサービスが存在し，世界協定時 UTC から 1 ミリ秒以内の誤差で時刻を同期することが可能である．各々が相互にタイムスタンプを発行，要求し合うプロセスとその発行されたタイムスタンプに納められた時刻に基づいて各 TSU が他の TSU の信頼性を監査すること，相互監査をシステムに内在することで時刻源および TSA に関する監査費用の低減をはかっている．

3.2 タイムスタンプ発行プロセス

概略として以下の手順を繰り返す．クライアントからのタイムスタンプの発行要求からログの記録までの詳細は次の通りである．

(1) $G = 0$ すなわちクライアントにおいてデジタルデータからハッシュを作成する（mk hash）.

(2) 利用者は RFC3161 に従って時刻認証を行いたいファイルのハッシュからタイムスタンプクエリを作成する（dispatch tsq）.

(3) RFC3161 のタイムスタンプ要求 (TSQ) を $G = 1$ である root TSU に発行する（tsq()）．発行を要求した時刻を t_0 とする．

(4) root TSU は，時刻 t_1 を署名したタイムスタンプ応答 (TSR) を生成し，a) 非同期に TSR を要求元に返す．このタイムスタンプを root TS と呼ぶ．

(5) 次世代の K 個の TSU は L 個の信頼できるものと $(N-1-L)$ 個の TSU からランダムに選んだ M 個のものから選択し TSQ の発行準備をするとともに，拡張プロトコルで保持している現在何世代目かを記録するフラグを 1 減ずる．

(6) 次世代 TSU へ TSQ を要求（g_tsq()）として発行する．

(7) 次世代 TSU による TSR 生成，応答が行われ，a) 非同期に TSR を要求元に返す．受け取った前世代 TSU は g_tsq による TSR を手元に保存し，c) どの次世代 TSU からの TSR であったかを記録する．従ってタイムスタンプ時刻は TSQ が TSU に届いて処理される過程で認証されるため，TSQ 要求から TSR 処理，ログの記録までの全体の処理が終わった時刻よりも前の時刻が記録される．

(8) タイムアウトになっていないか，世代数が上限に達していないかを確認し，いずれでも無い場合，再び次世代の TSU，K 個をこれまで述べたアルゴリズムに従って選択し TSQ の発行準備をするとともに，現在何世代目かを記録するフラグを 1 減ずる．

(9) 次世代 TSU へ TSQ を $(N, K = L + M, G)$ アルゴリズムによる要求 (g_tsq) として発行する．

(10) 以降，(7)〜(9) を G 上限まで繰り返していく．

G が 3 以上の時，各々の G 世代の TSU は $(G-1)$ 世代から受け取ったタイムスタンプ要求を自己と $(G-1)$ 世代の直接上流の TSU を除いた $(N-2)$ の TSU の内から $K = (L + M)$ 個を選んでこれまでと同様に世代数を減じて転送要求するその結果，各 G 世代で $1, K, K^2, ..., K^{G-1}$ 個のタイムスタンプ要求がなされ，全体では $(K^G - 1)/(K - 1)$ 個のタイムスタンプ要求がなされる．その結果，最大で $(K^G - 1)/(K - 1)$ 個のタイムスタンプが生成されるが，現実の実装では途中のネットワーク遮断等なんらかの理由により応答が無い TSU があることが考えられる．3 世代以上の間では次世代への転送 TS 要求がループすることがあり得る．本 TSA Grid で拡張したプロトコルでは各 TSU はどの TSU からのタイムスタンプ要求を受け応答したか，そのタイムスタンプ要求をどの子世代の TSU に転送したか，それが何世代目だったかを記録する．このようにして各 TSU は親 TSU に次世代への転送 TS を返答し，子 TSU から次世代への転送 TS を受け取る．受け取った子 TSU からの最大 K 個のタイムスタンプに署名された時刻を使って信頼出来る L 個の TSU のリストを更新する．なお時刻は RFC3161 で規定されたタイムスタンプに署名された時刻の解像度が 1 秒のため，root TS に署名された時刻を t_1 に対する次世代への転送 TS の時刻との差 Δt は 1 秒単位毎となる．TSA Grid としての時刻認証は root TS に署名された時刻 t_1 に対し Δt の各種統計処理値（平均値，分散，最頻値，etc.）によって表現された時刻によってなされる．

従って，本 TSA Grid を使って 1 日の解像度でデジタルデータの時刻認証を行う場合にはタイムスタンプ時刻自体の解像度 1 秒は十分な精度である．

3.3 タイムスタンプ検証プロセス

TSA Grid に於いてのタイムスタンプ検証プロセスは，非改竄や存在を証明したいデジタルデータの RFC3161 に基づいたタイムスタンプ要求を新たに生成するか，既存のタイムスタンプ要求を利用して行う．検証したいタイムスタンプをキーとしてタイムスタンプ要求を行ったパスに検証要求を同じく行う．このとき上位の TSU は下位の TSU に検証要求を出し応答を待つ上限時間，タイムアウトをタイムスタンプ発行要求より長くする．タイムスタンプ発行要求は，直ちに上位から下位へ転送されるように実装する．他方で検証プロセスは発行要求よりもプライオリティを低く実装する．これは検証プロセスの負荷が発行プロセスを阻害しないようにし，発行要求を受け付けた TSU が要求時間から遅延が少なくなるよう時刻認証をするためである．

タイムアウト時間内に集まった次世代への転送 TS の統計値により，root TS の確からしさを検証する．例えば，Δt の平均値を取った場合は root TS の時刻は $t_1 + EX(\Delta t) \pm VAR(\Delta t)$ と検証される．統計処理は最頻値の算出，四分上位数，最尤度推定等様々なものが目的に応じて利用可能である．

3.4 時刻検証プロセス

個々のタイムスタンプ要求の応答結果から Δt が負の値が得られたとする．この場合，要求した自身の時計が遅れているか，要求した相手の時刻が進んでいるかが考えられる．負の Δt を返答した TSU が含まれる，Δt が負の値が含まれるパス以外のパスを調べ，該当 TSU が他の上位の（root TS 発行 TSU より

の）次世代への転送 TS よりも小さな Δt を返しているパスが見つかれば該当 TSU の時刻が進んでいると検出され，そのようなものが見つからなければ自身の時計が遅れている可能性が検出される．前者では該当 TSU に時刻を NTP に同期するよう要求を送り，後者の場合は自己の時計を NTP に同期する．

3.5 信頼性検証プロセス

上記に述べたタイムスタンプの検証プロセスおよび時刻検証プロセスを経て，各 TSU は過去に自分が要求したタイムスタンプ要求に応答した他の TSU の信頼性を評価することができる．定期的に二つの検証プロセスを行うことでタイムスタンプ応答の可能性が高い TSU や時刻の誤差が小さい TSU を信頼度が高い TSU として信頼出来る TSU リストに加えるだけでなく，リストに保持する TSU の個数 L よりも多くの TSU の信頼性を検証することが出来る．本スキームでは TSU は独立に管理されており信頼出来る TSU のリストは各 TSU が独立に作成し保持するため，悪意ある第三者がある TSU についてこのリスト入手したとしても他の TSU のリストとは異なるため，man-in-middle 攻撃等を回避出来る．

3.6 本手法の頑強性

本手法で $G = 1$ とすれば既存の RFC3161 のシンプルプロトコルと同一となるが，多数の TSU が TSA Grid を形成した場合には N のなかから一つ選ぶため，障害が f 個の TSU で発生しているとすれば f/N となる．G=2 の場合，Δt を評価する際に $N/2$ 未満の個数の TSU が DoS 攻撃にあっても時刻認証が可能であるという頑強性を定義すれば，A. Bonnecaze 等の論文で提案する分散時刻認証局と同じ頑強性となる．すなわち N 個の時刻認証局から K 個のタイムスタンプを得るとするときに，K 個のタイムスタンプから多数決でグローバルタイムスタンプ (GTS) を構成する場合，過半数 $(K/2)$ を超える時刻認証局からのタイムスタンプが分散 DoS 攻撃でサービス不能もしくは大幅に遅れたタイムスタンプが発行された場合，システムとして時刻証明が出来なくなる．その確率は，F をサービス障害の出た時刻認証局の総数，f を集めたタイムスタンプに障害が出た時刻認証局から受け取った個数とすると

$$P(f \geqq \frac{K}{2}) = \frac{1}{\binom{N}{K}} \sum_{\frac{K}{2} \leqq i \leqq F} \binom{F}{i} \binom{N-F}{K-i} \qquad (1)$$

で表される．

G が 3 以上の場合は A. Bonnecaze 等の k among n スキームにおいて，$((N-2)^G - 1)/((N-2)-1) - 1$ の n，最大 $(K^G - 1)/(K-1) - 1$ の k 個のタイムスタンプを得るもの見なすことが出来る．

4. 評価実験

ローカルネットワーク上のクラスタ環境と同一動作パラメータ（$G = 5, K = 2, L = 1, M = 1, N = 4$，図 1 に置いて平均認証時刻が 1 秒であった 1 例）で動作させた場合に，ネットワーク遅延時間の影響がどのように現れるかの検証を行った．$K = L + M$ among N for G–generation 分散時刻認証手法の実装は，Java および Bouncy Castle API [12] を用い Servlet として実装した．実装系である tsagrid プログラムを Tomcat を Servlet/JSP アドオンとして Apache 上で実行した．

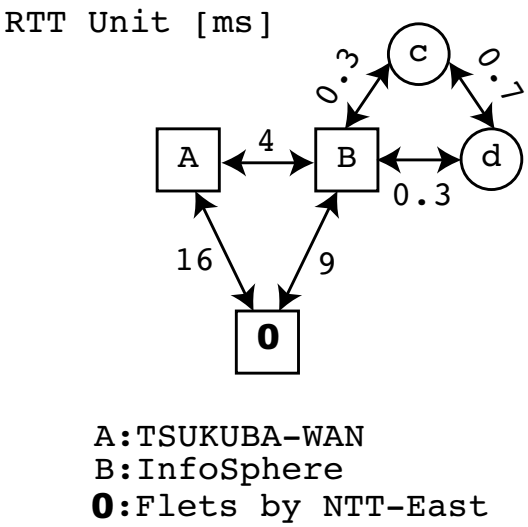

図 2 インターネットを経路に含む TSA Grid の配置
Fig. 2 Configuration of TSA Grid that includes the Internet in the netowrk routing.

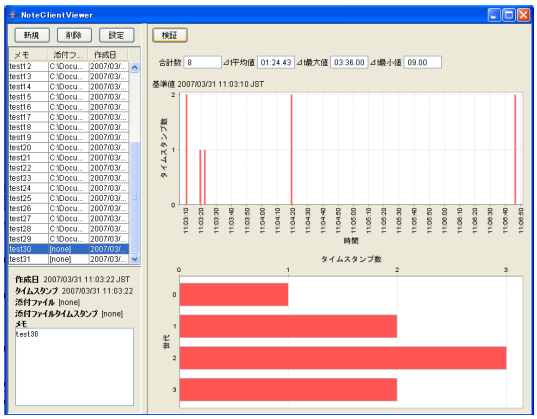

図 3 Note Client 画面
Fig. 3 Snapshot of the Note Client.

図 2 にインターネットを経路に含む TSA Grid の配置を示す．

TSUKUBA-WAN 上に設置した TSU(A)，NTTPC コミュニケーションズが提供する ISP サービス InfoSphere 上に設置した TSU(B,c,d) の 4 つの TSU からなる TSA Grid を構築した．ネットワーク遅延時間を評価しやすくするため，TSU(c) は 1 分，TSU(d) は 3 分，時刻を UTC から遅れさせた．

この TSA Grid に対し，図 3 に示すデジタル実験ノートクライアント（Note Client）を用いて，NTT 東日本が提供する Flets 光ネットワーク上（O）から，10 回の連続したタイムスタンプ要求を 5 分以上の間隔を置いて 3 回を 1 セットとして，3 セット行った．

図 4 にインターネット上での認証時間差と頻度を示す．

図 4 から 0 秒，1 分，3 分の 3 つの基準値から 10 秒から 20 秒の遅延がインターネット上で複数世代に渡って TS を発行する際に発生している事が読み取れる．この遅延の平均時間は 7.6 秒であった．この 7.6 秒の平均遅延時間を取り入れ，67.6 秒およ

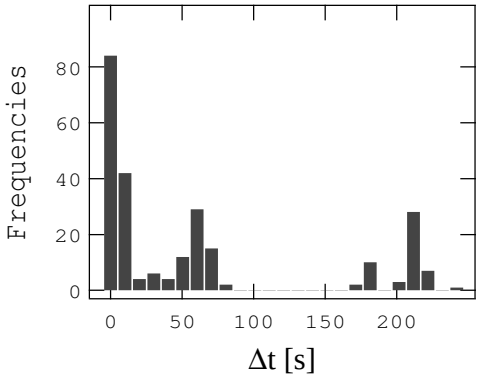

図 4　インターネット上での認証時間差のヒストグラム

Fig. 4　Histgram of response time on the Internet.

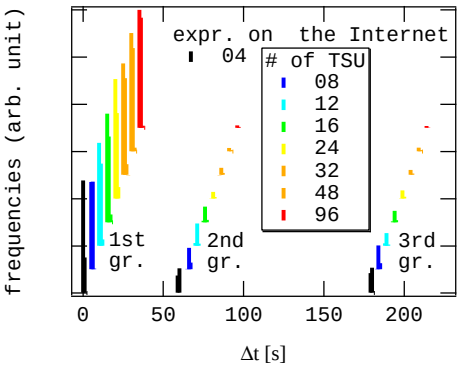

図 5　平均認証時間の頻度分布の TSU 数依存性

Fig. 5　Dependence of Histgram of average response time on nubmer of TSU.

び 187.6 秒をそれぞれ UTC から送らせた TSU をローカルネットワーク上のクラスタ環境に導入し，全 TSU での認証時間と基準となる時刻の差 Δt の平均値が 1 秒以下になるには幾つ TSU が必要かをシミュレーションにより調べた．クラスタ環境として OrionMultisystems 社製 Orion DS-96(CPU:Transmeta Efficeon 1.2GHz，各ノード 2GB RAM，合計 96 ノード) の均一クラスタを用いた．

図 5 にシミュレーションによる平均認証時間の頻度分布の TSU 数依存性を示す．

TSU 数が増えるに従って，遅延時間が大きな TSU の影響が小さくなっている事が分かる．0 秒を基準とする第 1 群，60 秒を基準とする第 2 群，180 秒を基準とする第 3 群および全体の平均認証時間を TSU 数毎に集計した結果を図 6 に示す．

図 6 から全体の平均として平均認証時間が 1 秒以内となるには TSU 数が 256 以上必要であると読み取れる．また全て TSU 数において最頻値を含む第 1 群の平均認証時間は基準となる時刻から常に 1 秒以内となっていることが読み取れる．

5. おわりに

本報告では，多数の時刻認証ユニットを用いて相互に時刻認証を行う $K = L + M$ among N for G–generation 分散時刻認証法による TSA Grid をインターネット上の複数のサイトで

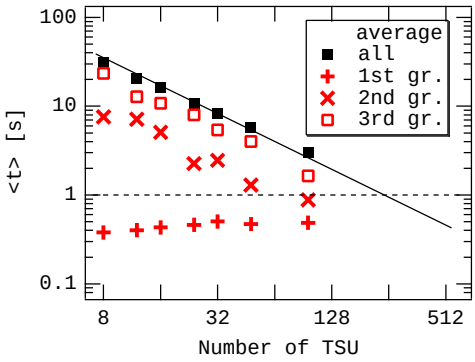

図 6　TSU 数と平均認証時間

Fig. 6　Average response time vs Nubmer of TSU.

TSU を設置し，ネットワーク遅延時間が認証時刻にどのような影響があるかを調査した．その結果，本報告で検討した動作パラメータ（$G = 5, K = 2, L = 1, M = 1, N = 4$）では TSU が 256 よりも多く用意すれば認証時間の算術平均が 1 秒以内となることが明らかとなった．算術平均よりも遅延の小さな認証時間が最頻値を用いることでを得られる事が明らかとなった．

本研究の一部は NEDO 平成 16 年度産業技術研究開発助成事業「量子化学グリッド ASP 実証実験」の支援を受けて行ったものである．

文　献

[1] C. Adams, P. Cain, D. Pinkas and R. Zuccherato, Internet X.509 Public Key Infrastructure Time-Stamp Protocol (TSP), RFC3161, 2001.
[2] タイムビジネス認定センター，時刻認証業務認定事業者一覧，http://www.dekyo.or.jp/tb/
[3] Z. Glózik, Open TSA, http://www.opentsa.org/
[4] A. Takura, S. Ono and S. Naito, Secure and Trusted Time Stamping Authority, Proceedings of IWS' 99, pp. 123–128, (1999).
[5] A. Bonnecaze, P. Liardet, A. Gabillon, K. Blibech, A Distributed Time Stamping Scheme, Proc. of the IEEE conference on Signal and Image Technology and Internet Based Systems. November 2005, Yaoundé Cameroon.
[6] A. Bonnecaze, P. Liardet, A. Gabillon, and K. Blibech,Secure Time Stamping Schemes: A Distributive Point of View, Annals of Telecommunications, Vol. 61, No. 5–6, pp. 662–681 (2006).
[7] Daniela Tulone, A Scalable and Intrusion-tolerant Digital Time-stamping System, Proc. 2006 IEEE Int'l Conf. Communications (ICC'06), Vol. 5, pp. 2357–2363, June 2006.
[8] D. L. Mills, Network Time Protocol (Version 3), Specification, Implementation and Analysis, RFC1305, 1992–3.
[9] 福田晴元, 桂木真一郎, 石本英隆, 小野論：NTP を用いた追跡可能な時刻配送システムの設計，情報処理学会　デジタルドキュメント　研究報告，Vol. 2004, No. 97, pp. 21–27 (2004).
[10] セイコーインスツル, SecureNTP, http://www.sii.co.jp/ni/tss/
[11] 「インターネットマルチフィード (MFEED) 時刻情報提供サービス for Public」, http://www.jst.mfeed.ad.jp/
[12] Bouncy Castle Crypto APIs, http://www.bouncycastle.org/