時刻認証基盤技術実験装置
―統合化プラットフォームシステム―

評価報告書

平成 18 年 2 月 28 日

【評価報告書の構成】

サブシステム

・(1)-(ii)　配信時刻高精度高信頼化サブシステム-2（配信時刻の高信頼化）

・(1)-(iii)　配信時刻高精度高信頼化サブシステム-3（時刻情報のトレーサビリティの保証）

・(4)-(i)　高速・高セキュリティタイムスタンプ付与・検証サブシステム-1

・(4)-(ii)　高速・高セキュリティタイムスタンプ付与・検証サブシステム-2

評価報告書

配信時刻高精度高信頼化サブシステム-2(配信時刻の高信頼化)
サブシステム(1)-(ii)

平成 18 年 2 月 28 日

目次

第1章 評価の目的

1. 目的

本サブシステムは、NTA/TA/TSA 間で行う時刻配信において、配信元のなりすましや配信される時刻情報の改ざんが行われない、高信頼な日本標準時配信の実現を目的に、平成 15 年度に開発した装置を基にした以下の機能の追加、性能向上についての開発・評価を行う。

(1) 時刻情報配信機能
統合化プラットフォームシステム上において、ネットワーク環境や機器性能等を示した上で、日本標準時との同期精度として、NTA から TA 間でミリ秒以内、NTA から TSA 間で数ミリ秒以内を達成する。

(2) 時刻情報トレース機能（認証連鎖方式）
統合化プラットフォームシステム上における、時刻配信を受けた TSA が付与するタイムスタンプの検証において、タイムスタンプに係る時刻情報の配信経路と誤差を確認できる。

(3) セキュリティ評価
統合化プラットフォームシステムに係るセキュリティ評価に伴い、認証連鎖方式＊1による時刻配信機能について必要となるセキュリティ評価を行う。

＊１：ここでいう認証連鎖方式とは、PKI(Public Key Infrastructure)認証技術を利用して TA が時刻配信先の時計を特定すると同時に時刻の計測と配信を行い、その計測結果を時刻監査記録として更に時刻配信先の時計に連鎖していく方式。

なお、セキュリティ評価については、セキュリティ評価報告書に記載する。

第2章 評価対象環境

1. ネットワーク構成図、機能ブロック図、測定環境

評価対象環境について、ネットワーク構成図、機能ブロック図、測定環境を以下に記載する。
※機能ブロック図は、本年度の評価に関係のある項目のみ記載する。

なお、統合化プラットフォームシステムにおける NTA、TA、TSA の名称を、以降次のように記載する。

表 2-1　統合化プラットフォームシステムにおける略称

名称	略称
配信時刻高精度高信頼化サブシステム－２と接続するための仮想国家時刻標準機関	NTA1
配信時刻高精度高信頼化サブシステム－２	TA1
高速・高セキュリティタイムスタンプ付与・検証サブシステム－１	TSA1
高速・高セキュリティタイムスタンプ付与・検証サブシステム－２	TSA2

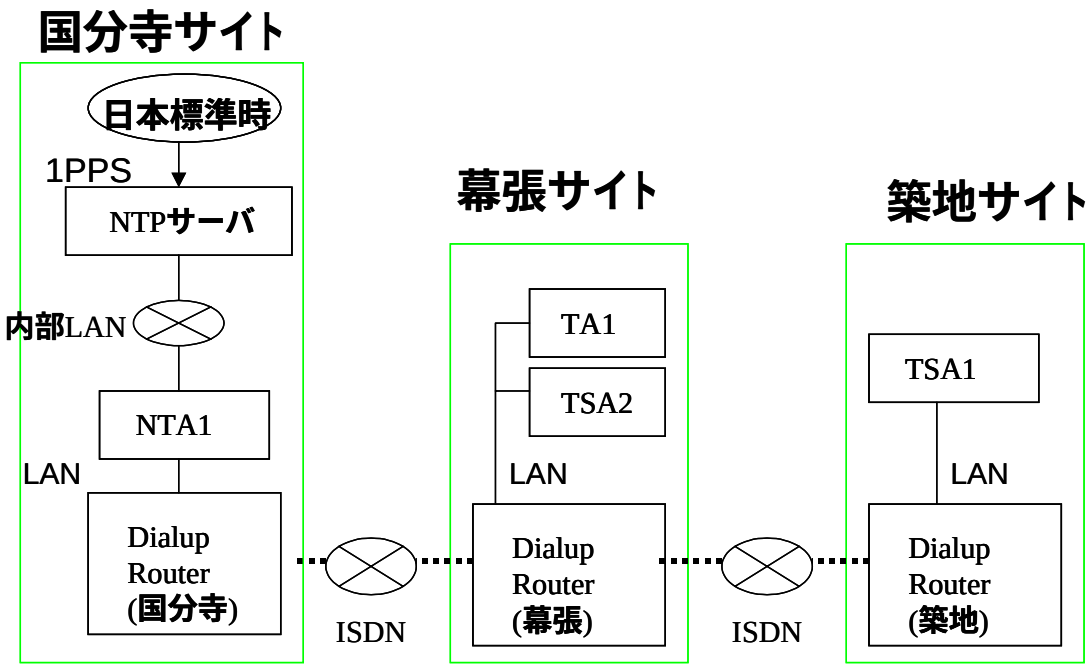


図 2-1　ネットワーク構成図

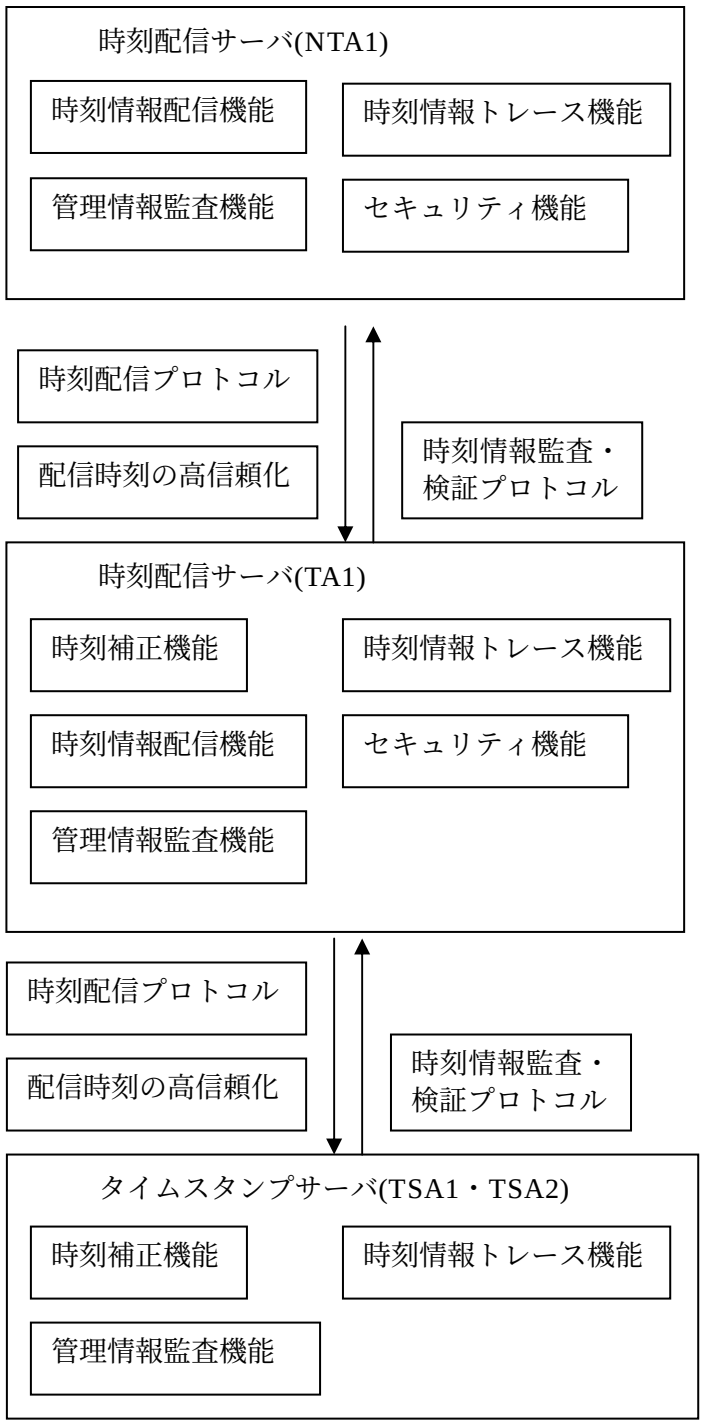

時刻配信サーバ(NTA1)
時刻情報配信機能
時刻情報トレース機能
管理情報監査機能
セキュリティ機能
時刻配信プロトコル
配信時刻の高信頼化
時刻情報監査・
検証プロトコル
時刻配信サーバ(TA1)
時刻補正機能
時刻情報トレース機能
時刻情報配信機能
セキュリティ機能
管理情報監査機能
時刻配信プロトコル
配信時刻の高信頼化
時刻情報監査・
検証プロトコル
タイムスタンプサーバ(TSA1・TSA2)
時刻補正機能
時刻情報トレース機能
管理情報監査機能

図 2-2　　機能ブロック図

表 2-2　測定環境

	NTA1	TA1	TSA1	TSA2
機器名	DELL PowerEdge 600SC	DELL PowerEdge 650	IBM xSeries 225	DELL PowerEdge 650
CPU	Celeron 2GHz	Pentium4 2.4GHz	Xeon 2.80GHz×2	Pentium4 2.4GHz
メモリ	512MB	1GB	2GB	1GB
OS	Red Hat Professional Workstation	Red Hat Professional Workstation	Red Hat Linux Advanced Server 2.1	Red Hat Professional Workstation
時刻配信・受信に使用する時刻	システム時刻	システム時刻	システム時刻	HSM 時刻
時刻配信・受信に使用するソフトウェア	時刻情報配信ソフトウェア(TACServer) (配信)	時刻情報配信ソフトウェア(TACServer) (配信・受信)	評価用クライアントソフトウェア(TSAProxy) (受信)	タイムスタンプソフトウェア(SecureTSS) (受信)
ネットワークインタフェース	1000BASE-T	1000BASE-T	100BASE-T	1000BASE-T
回線に接続するためのネットワーク機器	NTA1-TA1：ダイヤルアップルータ	TA1-TSA1：ダイヤルアップルータ TA1-TSA2：ダイヤルアップルータ	TA1-TSA1：ダイヤルアップルータ	TA1-TSA2：ダイヤルアップルータ
回線	NTA1-TA1：ISDN (64Kbps)	TA1-TSA1：ISDN (64Kbps) TA1-TSA2：LAN	TA1-TSA1：ISDN (64Kbps)	TA1-TSA2：LAN
配信間隔	NTA1-TA1：20 分	TA1-TSA1：120 分 TA1-TSA2：1 分	ー	ー

※HSM 時刻：HSM の時計の時刻と、TA から受信した時刻情報をもとに生成した時刻

第3章 評価内容・評価結果

1. 評価内容・評価結果

本サブシステムでは、下記の内容に関する評価を行い、すべて期待する結果が得られることを確認した。

・社内試験：7 項目
本試験は、仕様書に記載された要求項目に関する試験である。
（詳細は社内試験成績書を参照。）

・総合試験：46 項目
本試験は、開発されたシステムおよび各機能に対して、社内環境にて実施した試験である。
（詳細は総合試験成績書（社内試験)を参照。）

・試験結果まとめ

全試験項目数	：53 項目
合格した試験項目数	：53 項目
不合格の試験項目数	： 0 項目

第4章 本サブシステムの成果

本サブシステムでは、NTA/TA/TSA 間で行う時刻配信において、配信元のなりすましや配信される時刻情報の改ざんが行われない、高信頼な日本標準時配信の実現を目的とし、平成 16 年度に開発した装置に対する機能の追加、性能向上についてのシステムの開発・評価を行った。その結果、以下の成果を得ることができた。

1. 時刻情報の配信

本サブシステムでは、時刻情報の配信間隔の設定を最適化し、統合化プラットフォームシステム上において、日本標準時との同期精度として、NTA から TA 間でミリ秒以内、NTA から TSA 間で数ミリ秒以内を達成することができた。以下に、各測定環境における測定結果を示す。（各同期精度の測定に関する、ネットワーク環境や機器性能等の測定条件は、第 2 章を参照。）

1-1 NTA1-TA1 間の時刻同期精度

1-1-1 測定結果

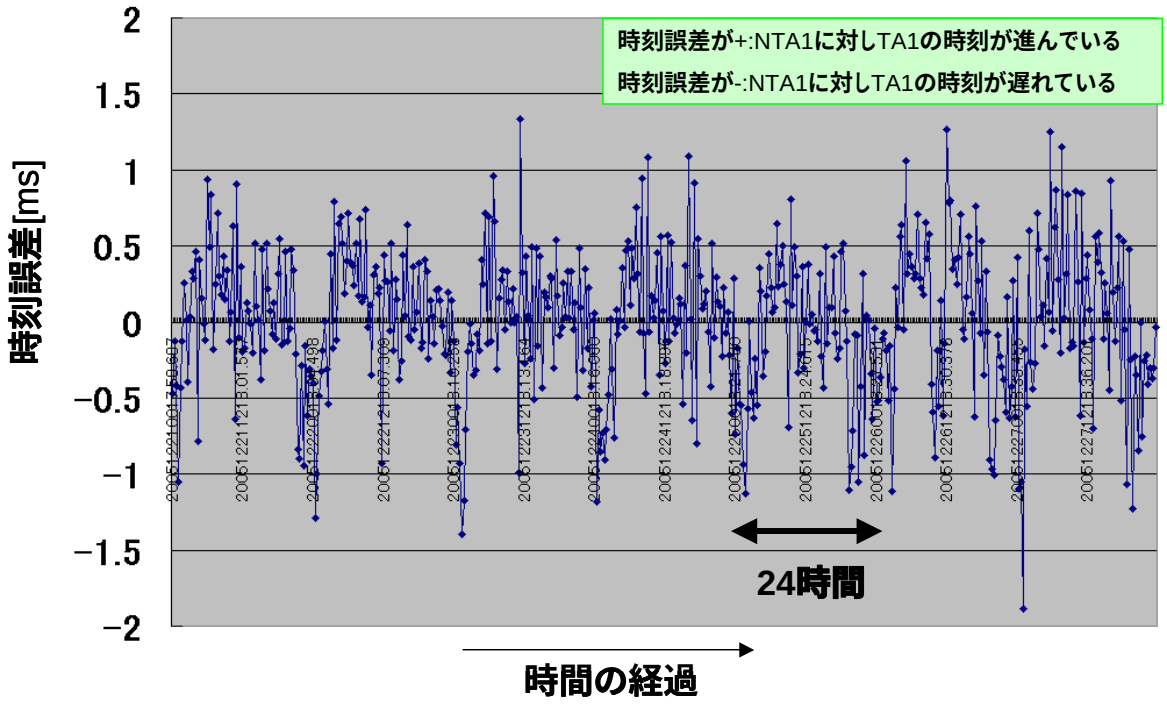


図 4-1　NTA1-TA1 間の時刻同期精度(2005/12/21-2005/12/27(UTC))

図 4-1より、計測期間中の時刻誤差がほぼ±1[ms]以内に収まっている。
より詳しいデータは、以下のとおり。

・計測期間　　：2005 年 12 月 21 日 00 時～2005 年 12 月 27 日 23 時(UTC)
・期間中の時刻配信回数　　：504 回
・期間中に NTA1-TA1 間の時刻誤差が±1[ms]を超えた回数　　：22 回
・期間中に NTA1-TA1 間の時刻誤差が±1[ms]を超えた割合　　：約 4.37%
・期間中の時刻誤差の最小値　　：-1.886[ms]
・期間中の時刻誤差の最大値　　：1.337[ms]

上記の結果より、NTA1 から TA1 間でミリ秒以内を達成することが確認できた。
　なお、ミリ秒以内の時刻同期精度の実現には、温度環境、割り込み処理などの要因も影響すると考えられる。より高い水準で時刻同期精度を維持するには、これらを考慮し、対策する必要がある。

1-2 TA1-TSA1 間の時刻同期精度

1-2-1 測定結果

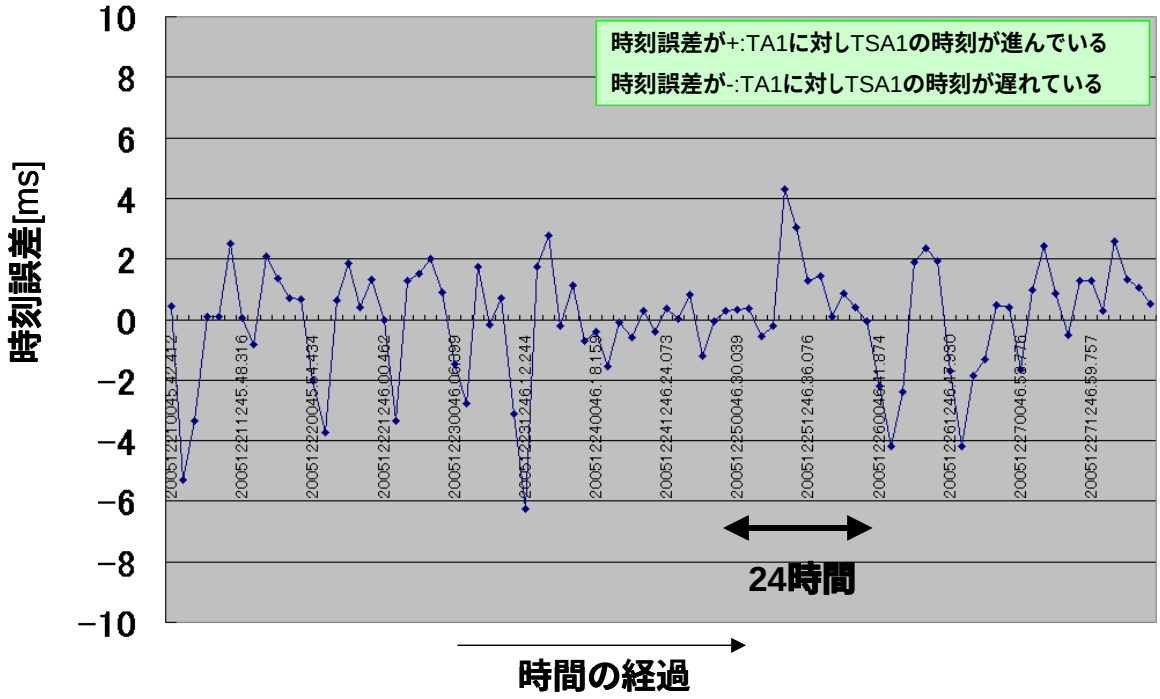


図 4-2　TA1-TSA1 間の時刻同期精度(2005/12/21-2005/12/27(UTC))

図 4-2より、計測期間中の時刻誤差が±10[ms]以内に収まっている。
NTA1-TA1 の時刻誤差を考慮した、NTA1-TA1-TSA1 間の時刻同期精度も含め、より詳しいデータは、以下のとおり。

・計測期間　：2005 年 12 月 21 日 00 時～2005 年 12 月 27 日 23 時(UTC)
・期間中の時刻配信回数　：84 回
・期間中に TA1-TSA1 間の時刻誤差が±9[ms]を超えた回数　：0 回
・期間中に TA1-TSA1 間の時刻誤差が±9[ms]を超えた割合　：0%
・期間中の時刻誤差の最小値　：-6257[ms]
・期間中の時刻誤差の最大値　：4320[ms]
・期間中に NTA1-TA1-TSA1 間の時刻誤差が±10[ms]を超えた回数　：0 回
・期間中に NTA1-TA1-TSA1 間の時刻誤差が±10[ms]を超えた割合　：0%

上記の結果より、NTA1 から TSA1 間で数ミリ秒以内を達成することが確認できた。
なお、今回の測定環境では、時刻の配信間隔が NTA1-TA1 間は 20 分間隔なのに対し、TA1-TSA1 間では 120 分間隔である。時刻の配信間隔以外の環境は NTA1-TA1 間、TA1-TSA1

間とも同等（むしろ、ハードウェア性能的には TA1-TSA1 間の方が高い）であることから、時刻の配信間隔を NTA1-TA1 間と同じ 20 分間隔にすることで、NTA1-TA1 間と同等の時刻同期精度が得られると考えられる。また、NTA1-TA1 間と同様、より高い時刻同期精度を実現する際には、温度環境、割り込み処理などの要因を考慮する必要がある。

1-3 TA1-TSA2 間の時刻同期精度

1-3-1 測定結果

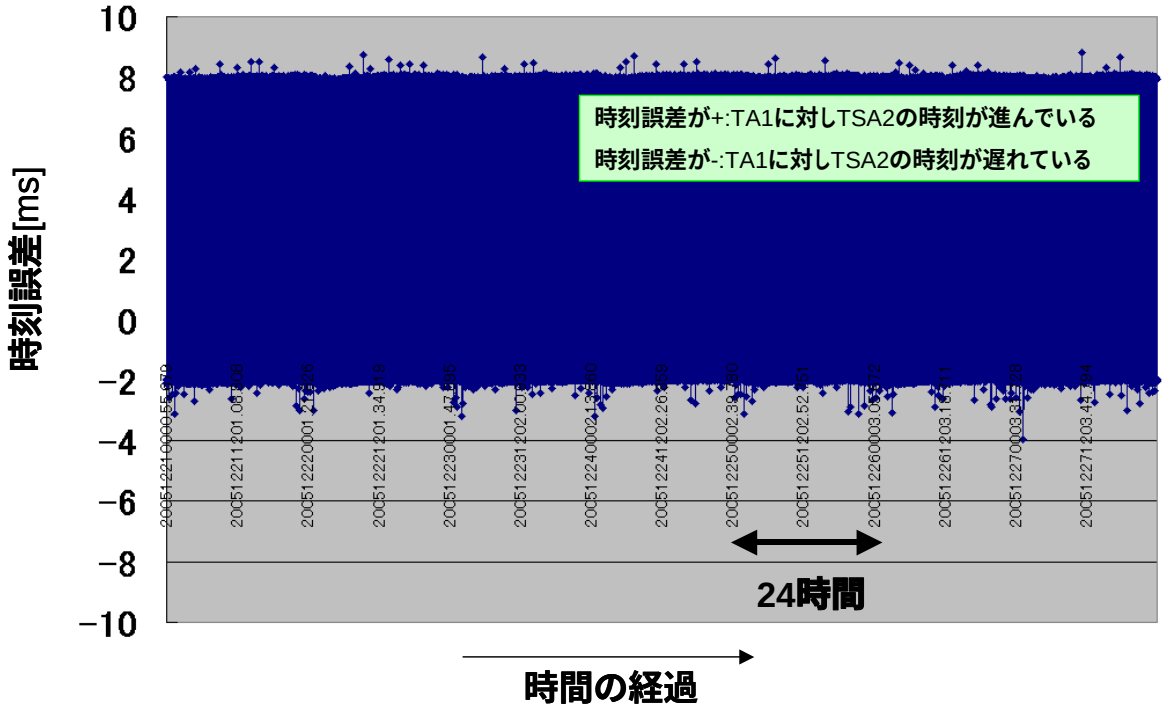


図 4-3　TA1-TSA2 間の時刻同期精度(2005/12/21-2005/12/27(UTC))

図 4-3より、計測期間中の時刻誤差が±10[ms]以内に収まっている。
NTA1-TA1 の時刻誤差を考慮した、NTA1-TA1-TSA1 間の時刻同期精度も含め、より詳しいデータは、以下のとおり。

・計測期間　　：2005 年 12 月 21 日 00 時～2005 年 12 月 27 日 23 時(UTC)
・期間中の時刻配信回数　　：10077 回
・期間中に TA1-TSA2 間の時刻誤差が±9[ms]を超えた回数　　：0 回
・期間中に TA1-TSA2 間の時刻誤差が±9[ms]を超えた割合　　：0%
・期間中の時刻誤差の最小値　　：-3976[ms]
・期間中の時刻誤差の最大値　　：8798[ms]

・期間中に NTA1-TA1-TSA2 間の時刻誤差が±10[ms]を超えた回数　：0 回
・期間中に NTA1-TA1-TSA2 間の時刻誤差が±10[ms]を超えた割合　：0%

上記の結果より、NTA1 から TSA2 間で数ミリ秒以内を達成することが確認できた。

なお TA1 から TSA2 の配信間隔が 1 分と短いのは、HSM の時計がずれやすい(1 分間で-2[ms]前後の時刻誤差が出る)ためである。また、時刻誤差が 8[ms]前後となる場合が多いのは、HSM の時計が、数分間隔で約 10[ms]時刻を進める特性を持つためである。以下に、TA1 からの時刻配信・監査が無い状態で、HSM の時計の時刻とシステム時刻の差を 1 秒間隔で出力した際のグラフを示す。

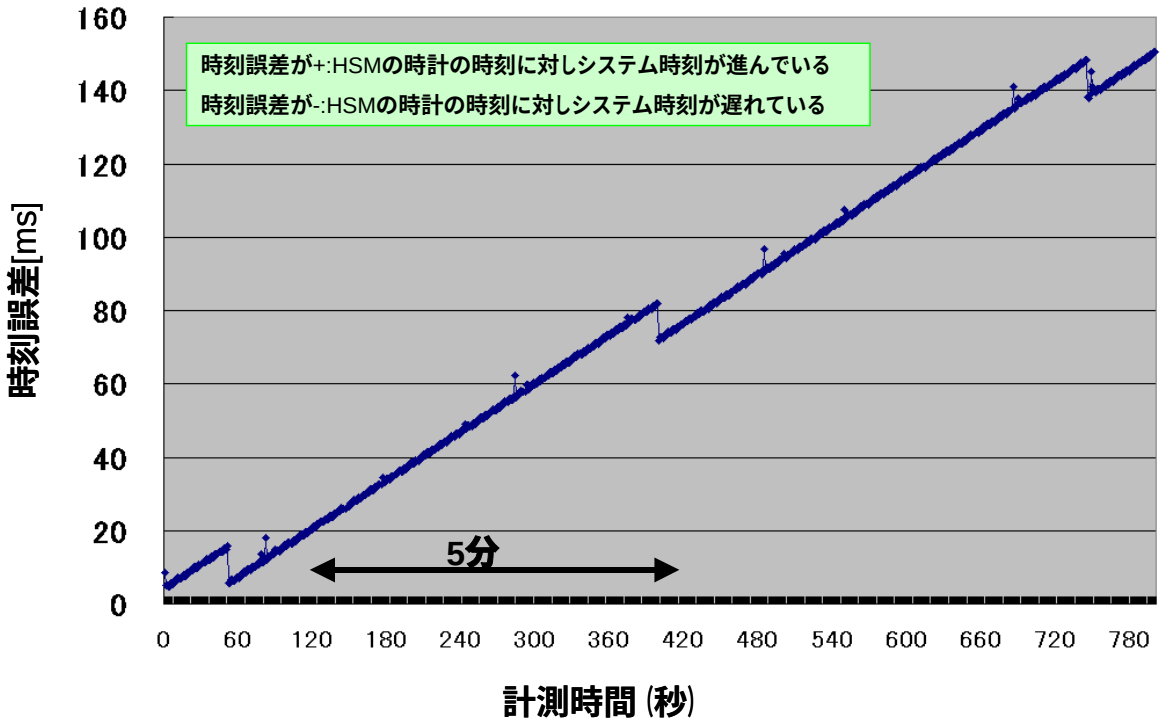


図 4-4　HSM の時計の時刻-システム時刻間の時刻誤差の推移

図 4-4より、HSM の時計の時刻は、システム時刻に対して少しづつ遅れ(=システム時刻との差が広がる)、数分間隔ごとに約 10[ms]、HSM の時計の時刻がシステム時刻に対して相対的に急に進んでいる(=システム時刻との差が縮まっている)ことが分かる。

※なお、図 4-4では、ミリ秒レベルの時刻誤差の推移をクローズアップするため、スケールを調整している。(HSM の時計の時刻とシステム時刻は、実際には数分以上ずれている。)

今回、TSA2 ではセキュリティを考慮して HSM の時計の時刻を利用しているが、上記の

ような内容から、より高い時刻同期精度の実現には、HSM の時計自体に課題があることが分かった。この課題は、HSM 全般の課題ではなく、製品固有の特性によるものではあるが、タイムスタンプにはセキュリティと時刻同期精度の両立が求められるため、タイムスタンプサーバ開発の際はこの両方について考慮する必要があることを、改めて明らかにすることができた。

2. 時刻情報のトレース（認証連鎖方式）

本サブシステムでは、平成 16 年度に複数方式タイムスタンプ検証サブシステム、高速・高セキュリティタイムスタンプ付与・検証サブシステム-1、高速・高セキュリティタイムスタンプ付与・検証サブシステム-2 と連携し、タイムスタンプトークンに含まれる時刻情報のトレーサビリティについて、認証連鎖方式による検証方法の検討を行った（本内容については、平成 17 年度の取扱説明書の添付資料 6「認証連鎖方式における時刻情報トレース機能」を参照)。平成 17 年度は、本検討内容を統合化プラットフォームシステムの NTA1-TA1-TSA1、および NTA1-TA1-TSA2 に適用した。

本サブシステムの評価により、統合化プラットフォームシステム上における、時刻配信を受けた TSA が付与するタイムスタンプの検証において、タイムスタンプに係る時刻情報の配信経路と誤差を確認できることが確認された。

評価報告書

配信時刻高精度高信頼化サブシステム-3(トレーサビリティの保証)
サブシステム(1)-(iii)

平成 18 年 2 月 28 日

目次

第1章 本サブシステムの評価と課題

1. 本方式の評価と課題

本サブシステムでは、NTA-TA-NTP サーバ間で行う時刻配信において、高信頼度な時刻情報の配信および高い時刻同期精度の確保、時刻情報のトレース機能の実現、本サブシステムのセキュリティ評価を行った。ただし、NTP サーバは、TSA 相当機器として取り扱っている。

1-1 本システムの機能内容とその評価

1-1-1 時刻情報配信機能

統合化プラットフォームシステム上において、ネットワーク環境や機器性能等を示した上で、日本標準時との同期精度として、NTA から TA 間でミリ秒以内を達成する上での課題と見通しを明確化するとともに、NTA から時刻配信先のサーバ等の機器間で数ミリ秒以内を達成すること。

(1) 実現方法

TA で NTA-TA 間の同期精度を確認・調査し、時刻誤差がミリ秒を達成する課題と見通しを立てる。

高信頼度な時刻情報の生成方法や時刻同期のタイミングを調査し、時刻同期の課題を明らかにし、同期精度を向上させる。その結果、NTA から時刻配信先のサーバ機器間での時刻同期精度を数ミリ秒以内を達成する。

(2) 評価

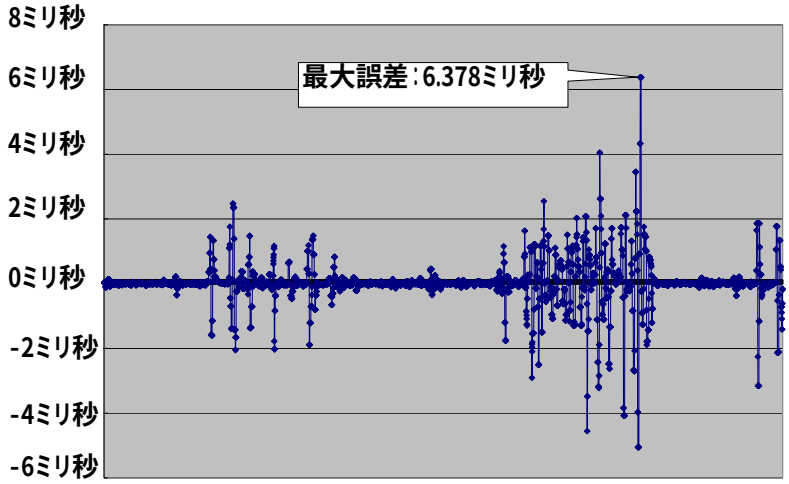
評価を行った環境を以下に記す。

ネットワーク環境	・NTA-TA 間はローカルネットワークで接続。ただし、ローカルネットワーク上には、他機器が接続されている。 ・TA-NTP サーバ間はインターネットで接続されている。（10Mb/s、上り下り対称）
他ソフトウェア	・TA 機器にアンチソフトウェアプログラムをインストールしている。

以下に、本サブシステムを使用した時刻配信の時刻誤差のグラフを記す。各グラフの縦軸は時刻誤差を表し、横軸は調査した時刻を表している。なお、調査した期間は 1 日であり、1 日の時刻誤差の推移を表している。

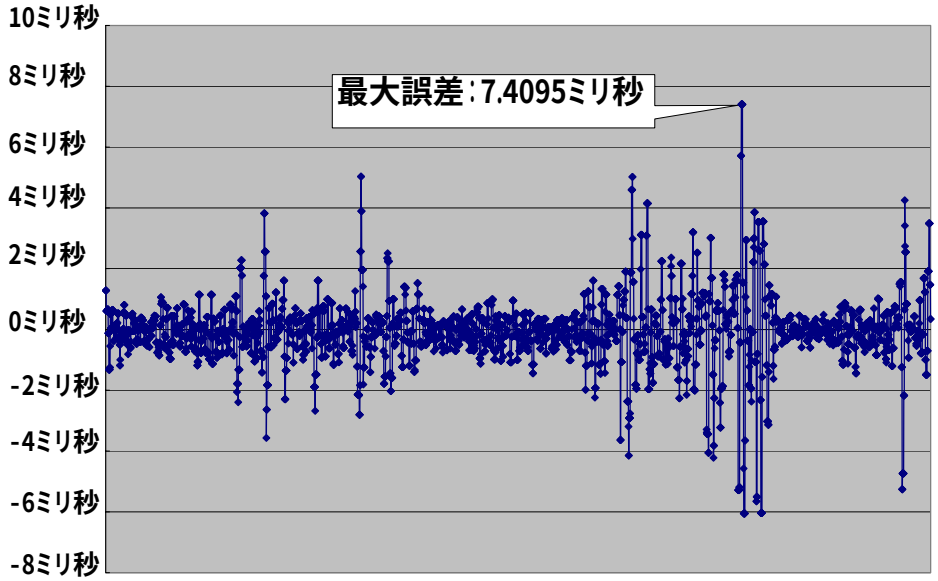
ア NTA-TA 間の時刻誤差

NTA-TA 間の時刻誤差の以下のグラフで表す。なお、NTA-TA 間はローカルネットワークで接続されている。グラフから NTA-TA 間の時刻誤差の最大値は、約 6.4 ミリ秒であることがわかる。



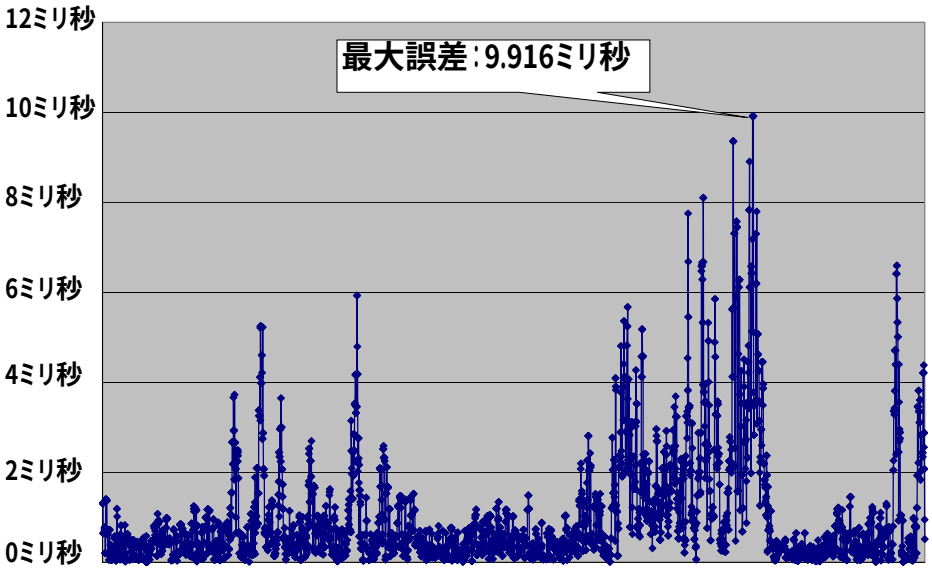
イ TA- NTP サーバ間の時刻誤差

TA- NTP サーバ間の時刻誤差の以下のグラフで表す。なお、TA- NTP サーバ間はインターネット経由で接続されている。グラフから TA- NTP サーバ間の時刻誤差の最大値は、約 7.4 ミリ秒であることがわかる。



ウ NTA-TA- NTP サーバ間の時刻誤差

NTA-TA- NTP サーバ間の時刻誤差の以下のグラフで表す。なお、グラフに表されている時刻誤差は、NTA-TA 間の時刻誤差の絶対値と TA- NTP サーバ間の時刻誤差の絶対値の和である。グラフより、NTA-TA- NTP サーバ間の時刻誤差の最大値は、約 9.9 ミリ秒であることがわかる。

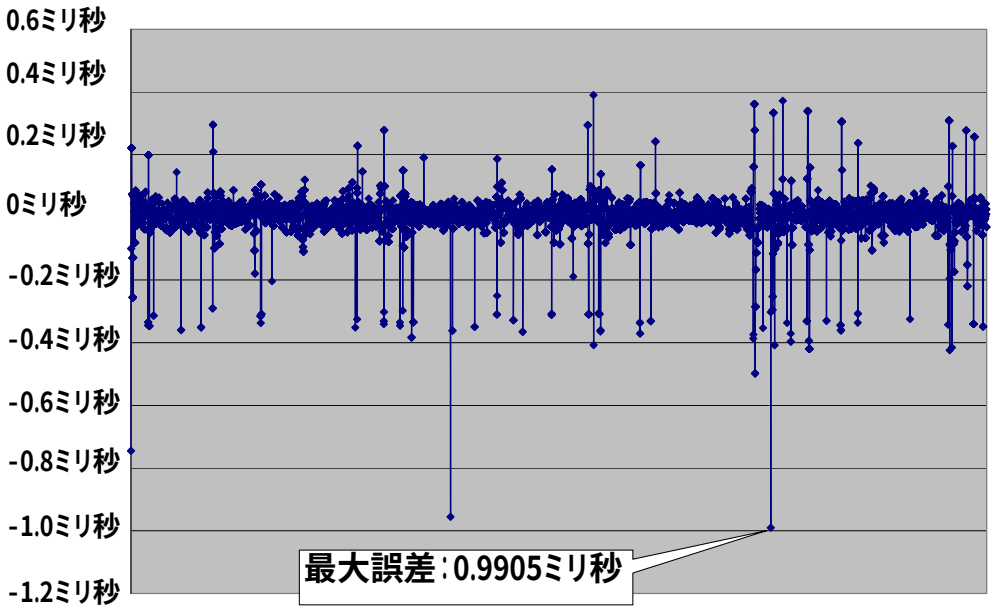


エ NTA-TA 間の時刻誤差ミリ秒達成の見通し

NTA-TA 間はローカルネットワークで接続されているが、ローカルネットワーク上には NTA および TA 以外の機器が存在する。そのため、他の機器の通信状況によっては NTA-TA 間の時刻誤差が大きくなる可能性がある。以下に調査した環境を記す。

ネットワーク環境	・NTA-TA 間はローカルネットワークで接続。ただし、ローカルネットワーク上には、他機器は接続されていない。
他ソフトウェア	・TA 機器には、時刻配信に関するソフトウェア以外はインストールされていない。

以下の時刻誤差を表したグラフも、NTA 機器と TA 機器をローカルネットワークで接続したときの時刻誤差を示しているが、このローカルネットワーク上には、NTA 機器と TA 機器以外の機器は接続されていない。以下のグラフから NTA-TA 間の最大の時刻誤差が約 0.99 ミリ秒であることが判明し、ローカルネットワーク上に NTA 機器、TA 機器以外の機器を接続しない場合、NTA-TA 間の時刻誤差がミリ秒以下になることが判明した。



1-1-2 時刻情報トレース機能

統合化プラットフォームシステム上において、サーバ等の機器に時刻配信を行い、当該機器内に保存される記録について、時刻情報の配信経路と誤差を確認可能とする。

(1) 実現方法

統合化プラットフォーム上の NTP サーバからサーバ機器へ高信頼度な時刻情報を送受信することにより、高信頼度な時刻情報の利用とその配信経路および日本標準時との時刻誤差を確認する。

統合化プラットフォーム上の TA2 と NTP サーバはインターネットを介して接続されている。また、配信経路の検証を行うには、NTA2 および NTP サーバから検証に必要となる時刻認証子をオンラインで TA2 に収集する必要がある。その際、必要な時刻認証子の検索、ネットワーク遅延等により配信経路検証を正常に行えないケースが発生していた。そこで、時刻認証子の保存方法や検索方法を改良し、インターネット経由でも正常に配信経路検証を行い、日本標準時との時刻誤差を確認可能とした。

(2) 評価

インターネットを介した時刻認証子の配信および配信経路の検証を可能とした。また、統合化プラットフォーム上の NTP サーバからログサーバへ時刻認証子を配信し、ログの生成時刻証明に時刻認証子を利用することが可能となった。これにより、時刻リンク方式時刻配信の実用化が可能となった。

1-1-3 セキュリティ評価

セキュリティ評価は、NTA2 および TA2 のセキュリティ評価報告書を参照のこと。

1-2 今後の課題と改善方法

1-2-1 前年度の課題

前年度の評価報告書にて取り上げた課題について、今年度の対応を以下に記す。

(1) 時刻情報トレース機能

ア 配信ルート検証

(ア) 昨年度の課題

改造した NTP を用いて時刻配信を行っているが、時刻情報の生成のため時刻誤差が大きくなる(常時、数 10 ミリ秒の誤差が発生していた)。

(イ) 今年度の対応

1-1-1 時刻情報配信機能に記載している。

イ 検証・管理機能

(ア) 昨年度の課題

上位の時刻配信機関が下位の機関への監査処理に時間がかかる。

(イ) 今年度の対応

時刻認証子の保存ログの形式を変更し、検索時間を短縮、監査完了までの時間削減を達成した。

以下のように保存ログの形式を変更した。

A 保存ログの形式

(A)　昨年度版までの形式

1 日分の時刻認証子を 1 ファイルに保存する。

ファイル名の例　　time_20060117.dat

(B)　今年度版の形式

1 日分の時刻認証子を複数のファイルに保存する。

ファイル名の例　　time_20060117_12345.dat
time_20060117_12346.dat
time_20060117_12347.dat
time_20060117_12348.dat

B 監査に要する時間

1 日で生成される時刻認証子（総数：86509 個、サイズ：22470592 バイト）を監査した場合、監査に要する時間が以下のように短縮された。

(A)　昨年度版

監査所要時間：21 分 32 秒

(B)　今年度版

（1 日分の時刻認証子を 12 ファイルに分割して保存）

監査所要時間： 3 分 42 秒

ウ 保証範囲

(ア) 昨年度の課題

時刻認証子の生成間隔を数百ミリ秒としたが、さらに範囲を狭めることが可能か検討する。

(イ) 今年度の対応

NTPD の動作周期を、現状の 1 秒単位に固定されている状態から、下記に示すミリ秒単位の周期に変更できるように修正した。

1000[ms]
500[ms]
250[ms]
200[ms]
125[ms]
100[ms]

その結果、NTPD の動作周期を変更することにより、保証範囲を狭めるように調整可能になった。

(2) 拡張性

ア 昨年度の課題

特になし。

イ 今年度の対応

特になし。

(3) その他機能追加

ア NTA 監査

昨年度版においては、NTA に対して定期的に監査を実施する機能が無く、NTA が生成した時刻認証子を監査できなかった。また、TA 及び TSA では、定期監査時に時刻認証子の保存ログファイルが分割されているのに対し、NTA では定期監査が実施されないため、時刻認証子の保存ファイルの分割は行われていなかった。

今年度版において、NTA が NTA 自身に対して監査を実施する機能を追加した。その結果、NTA が生成した時刻認証子を監査できるようになった。また、NTA においても、時刻認証子の保存ログファイルの分割が行えるようになった。

イ 監査時のエラーについて変更

昨年度版においては、監査中に監査対象の時刻認証子から 1 つでもエラーが見つかった

場合、その場で監査を完了していた。そのため、監査対象の時刻認証子に複数のエラーがある場合でも、全てのエラーを確認できなかった。

今年度版においては、監査中に監査対象の時刻認証子からエラーが見つかっても、最後まで監査を実施するよう変更した。その結果、監査対象の時刻認証子にある全てのエラーを確認できるようになった。

ウ 監査時の採取データの追加

今年度版において、監査対象の時刻認証子内の、時刻情報の offset 値について、平均値/最大値/最小値を計算し、監査ログに記録できるように変更した。

エ WEB 監査の画面変更

昨年度版においては、WEB 監査の依頼画面と結果確認画面は同じ画面となっていた。また、WEB 上で確認できる監査結果は、最後に実施された定期監査と全監査についてのみであり、過去に実施された定期監査と全監査、および、全ての WEB 監査については WEB 画面上で結果を確認できなかった。

今年度版においては、WEB 監査の依頼画面と結果確認画面は別画面とした。また、結果確認画面において期間を指定し、その期間中に実施された監査の結果を WEB 画面上で確認できるように変更した。

1-2-2 今後の課題と改善方法

今年度の開発において発生した課題とその改善方法を以下に記す。

(1) 検証・監査の処理時間

ア 課題

検証・監査の処理時間に対し、時刻認証子の検索処理にかかる時間が大きく影響を与えている。また、検索の処理時間は保存ログファイルのサイズに影響されている。定期監査(保存ログの分割)の頻度が少ない場合、保存ログファイルのサイズが大きくなり、その結果、検索処理にかかる時間が長くなる。最終的には検証・監査の処理時間が長くなってしまう。

イ 改善方法

保存ログ分割のタイミングを見直して、保存ログのサイズを小さくして、検索時間を短縮する。

今年度版においては、定期監査及び NTA 監査が実施されたタイミングで、時刻認証子の保存ログファイルの分割が実施されている。保存ログ分割のタイミングを監査時ではなく、時刻認証子をログファイルに保存する時に変更し、保存ログファイルのサイズを小さく維持する。その結果、定期監査(保存ログの分割)の頻度が少ない場合でも、検証・監査の処理時間に影響を与えないように出来る。

(2) 拡張性

ア 課題

今年度版のシステムでは、時刻配信ホスト情報として、IP アドレスを使用している。

そのため、一度、ネットワーク構成を決めてシステム稼動させ時刻認証子を生成した後で、ホストの IP アドレスを変更した場合、時刻認証子中の時刻配信ホスト情報と異なってしまう。その結果、IP アドレス変更以前に作成した時刻認証子については、検証や WEB 監査が出来なくなってしまう。

イ 改善方法

ネットワーク構成と、時刻認証子中の時刻配信ホスト情報との間に、違いが生じた場合の対応を検討する。

(3) 冗長性

ア 課題

今年度版のシステムでは、上位の NTP サーバは 1 台のみであったため、上位の NTP サーバに障害が発生した場合の冗長性がない。システム上、上位に複数の NTP サーバを持つことは可能であるが、複数の上位サーバを使った評価をしていないため、冗長性の確認がされていない。

また、上位に通常の NTP サーバを置いて時刻同期する場合、上位サーバが NTP の認証機能である Autokey を使用しなければ、時刻同期できるが、Autokey を使用する場合、改造 NTP サーバでは、上位に時刻認証子要求のパケットを送信してしまう為、通常の NTP サーバとは時刻同期できなくなる。

イ 改善方法

上位サーバとして、Autokey を使用する通常の NTP サーバを使用できるように、改造 NTP サーバを修正する。

上位に複数の NTP サーバを置き、上位の NTP サーバに障害が発生した場合の冗長性を確認する。

2. システム構築と管理

2-1 システム構築と管理の評価

2-1-1 ユーザインタフェース

NTA、TA、TSA の運用では、管理者がコンソールからコマンドラインにより操作している。ただし、下位の機関が上位の配信機関へ監査の依頼を行うのは、Web を用いる。大部分の操作をコマンドラインから実施可能とすることにより、機能の拡張性と他サブシステムとの連携を検討できる。また配信された時刻情報の検証には Web が用いられており、時刻情報を受け取ったユーザが簡単に時刻情報の検証ができるようになっている。

2-1-2 ネットワーク構成

統合化プラットフォーム環境では、TA-NTP サーバ間でインターネットを使用しており、一

般に使用されているネットワーク構成に近い環境にしている。

2-1-3 クライアントへの時刻配信

今年度は NTP サーバから複数の機器への時刻配信を行った。NTP サーバからログサーバへ時刻配信を行い、NTA-TA-NTP サーバ-ログサーバまでの時刻配信経路の特定を可能とした。

3. 時刻認証子による時刻証明方式での課題解決

時刻認証子による時刻証明方式は、TSA で生成される時刻認証子の保証を TA 内の時刻認証子を用いて行う方式である。TSA 内の時刻認証子の保証のために、TA 内の時刻認証子を TSA に送信する必要がある。以下、TA 内の時刻認証子を TSA に送信することを「監視」と記す。（本サブシステムでは、監視を NTP を用いて実施している）

最新の監視と次の監視との間隔を短くすると TSA 内の時刻を改ざんできる範囲を小さくできるが、TA と TSA 間のネットワーク遅延や時刻誤差のため、正しく監視できない場合がある。監視と監視の間隔の決定は、以前の論文でも問題となっていた。本論文では、TA と TSA をインターネットを介して接続し、ネットワーク遅延や時刻誤差から監視の間隔の最適な値を求める。さらに、求めた最適値が、時刻改ざんの防止に有効な値であることを示す。

なお、以下に示した時刻配信システムは、平成 15 年度開発のものを使用しており、今年度のシステムとは異なる。しかし、TA-TSA 間の監視頻度、TA および TSA での時刻認証子生成の間隔の決定は今年度のシステムでも共通の問題であり、今後の時刻認証子時刻配信システムの改良の参考とする。

3-1 配信時刻証明方式の概要と特徴

3-1-1 時刻認証子概要

TA および TSA で生成される時刻認証子は以下のような特徴を持つ。

(1) 生成された時刻認証子の改ざん検出を可能とする。
(2) 時刻認証子を予測して事前に生成するのは困難である。
(3) 時刻情報の配信経路の特定が可能である。

(1)の改ざん検出を可能とするには、信頼できる第三者機関である TA から監視を受ける必要がある。

3-1-2 時刻認証子の生成方法とフォーマット

TA および TSA の時刻認証子の生成モデルを図 1 に示す。図 1 では、TA、TSA の時刻認証子の生成タイミングを示しており、時間が矢印の方向に過去から未来へ流れている。TA および TSA では、時刻の流れを示す矢印上に横線にて、時刻認証子が生成されたことを示す。

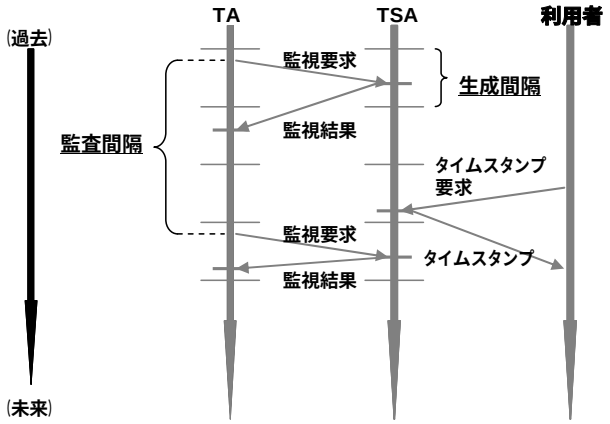


図 1：時刻認証子生成図

時刻認証子の生成タイミングは、①一定時間ごと、②TA からの監視要求を受けたとき、③TSA から監視結果を受けたとき、④利用者からタイムスタンプ要求を受けたときの 4 種類である。各時刻認証子は、TA、TSA にて、それぞれで直前に生成された時刻認証子のハッシュ値を基に生成されているため、改ざんが困難である。また監視の際、TA の時刻認証子を TSA に送付し、TA の時刻認証子のハッシュ値を基に TSA の時刻認証子を生成する。

本章で算出する「監視間隔」とは、TA から監視を実施する間隔のことである。また、監視間隔は一定時間ごとの時刻認証子の生成される間隔（以下「生成間隔」とする）に依存する。時刻認証子に格納されるデータは、主に以下がある。

- 時刻認証子を生成した機関（TA または TSA）を示す情報（例：IP アドレス）
- 時刻認証子を配信した機関（TA）を示す情報（例：IP アドレス）
- ランダム値（事前予測を防止するため）
- ハッシュ値（時刻認証子の非改ざん性を検証するための情報）

3-2 生成・監視間隔の算出方法

3-2-1 生成・監査間隔の計算式

時刻認証子の適正な監視間隔および生成間隔を算出するために TA と TSA をインターネットを介して接続した。なお、インターネットへ接続する通信速度は 10Mbps としたが、TA、TSA 以外の通信装置もあり、実使用に近い形態とした。

TA は、TSA が不正な時刻認証子を生成することを防止するために監視を行う。TSA 内で生成される時刻認証子は、ハッシュ値によりリンクされているため改ざんは困難であるが、監視間隔が長くなった場合、ハッシュ値が衝突する可能性が高くなるため監視間隔は短いほうが望ましい。

監視間隔を生成間隔の 2 倍にした場合（図 2）、TA からの監視(te_s_1)と監視の間(te_s_4)に生成される時刻認証子は 2 個(te_s_2 と te_s_3)になる。

te_s_1 と te_s_4 は、TA の時刻認証子を元に生成され、TA にて保存されているため、TSA で改ざんを行った場合、検出が可能である。te_s_2, te_s_3 はそれぞれ、改ざんの検出が可能

な te_s_1, te_s_4 とハッシュリンクしており、ハッシュの衝突可能性を低減している。

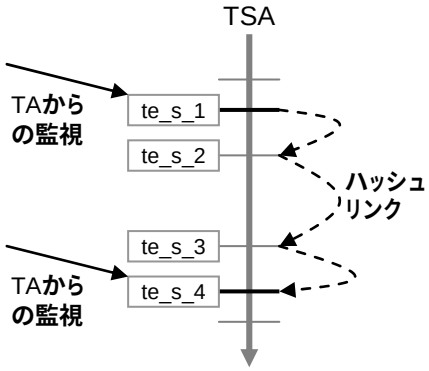


図 2：監視間隔算出図

監視間隔を生成間隔の 2 倍以下にした場合、TSA で生成される一定時間ごとの全ての時刻認証子は監視要求を受けたときの時刻認証子とリンクするため、改ざんはより困難となる。監視間隔を生成間隔の 2 倍以上にした場合、監視要求を受けたときに生成される時刻認証子と直接リンクしない時刻認証子が生成されるため、ハッシュの衝突発生の危険性が高まる。よって、生成間隔は監視間隔の 2 倍以下が望ましい。

TSA が監視結果を意図的に遅らせることで、TSA は結果を遅らせた分の時刻を改ざんすることが可能となる。したがって、監査要求を送信してから監視結果を受信するまでの時間は、生成間隔より短くてはならない（図 3）。

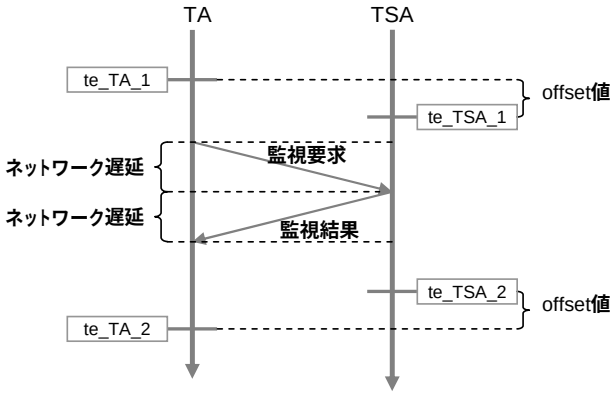


図 3：生成間隔算出図

te_TA_1, te_TA_2 は TA で、te_TSA_1, te_TSA_2 は TSA で、それぞれ一定時間ごとに生成される時刻認証子を示している。監視にかかる時間はインターネットを介したネットワーク遅延の 2 倍以上かかり、時刻誤差（以下 offset 値と記述する。[2]参照）を考慮した場合、以下の計算式を得る。

$T_g \geqq 2(T_o + T_d)$

$0 < T_a \leqq 2T_g$

T_g：時刻認証の生成間隔
T_a：TA～TSA 間の監視間隔
T_o：TA～TSA 間の offset 値の絶対値
T_d：TA～TSA 間のネットワーク遅延

3-2-2 測定値と生成・監査間隔の最適値

実際に計測した、offset 値、ネットワーク遅延を以下に示す。計測データは、NTPv4 を使用して約 1 週間かけて収集したものである。

●offset 値
最大　19.014 ミリ秒
最小　-11.708 ミリ秒
平均　0.113265 ミリ秒

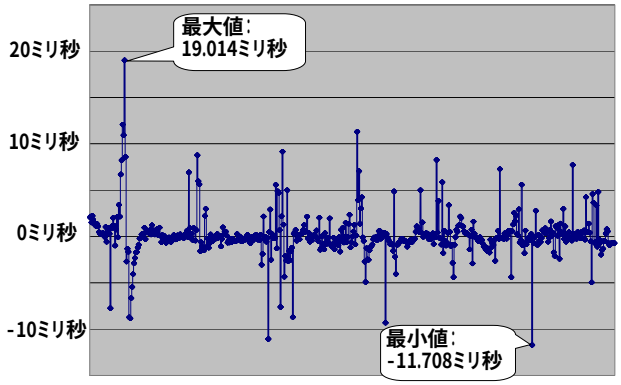


図 4：offset 値

●インターネットを介したネットワーク遅延
最大値：79.071 ミリ秒
平均値：22.02011 ミリ秒

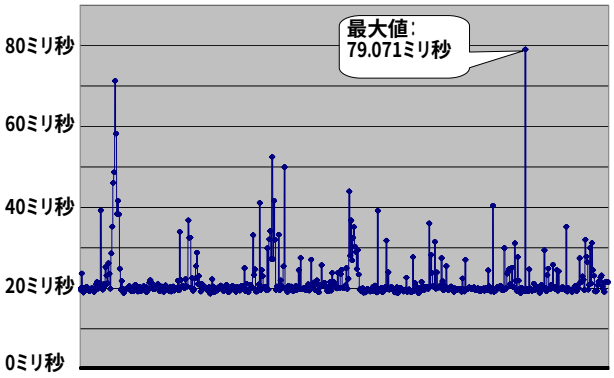


図 5：インターネットを介したネットワーク遅延

また、2 (T_o + T_d) の値は以下のようになった。
最大：181.558 ミリ秒
平均：46.5722 ミリ秒

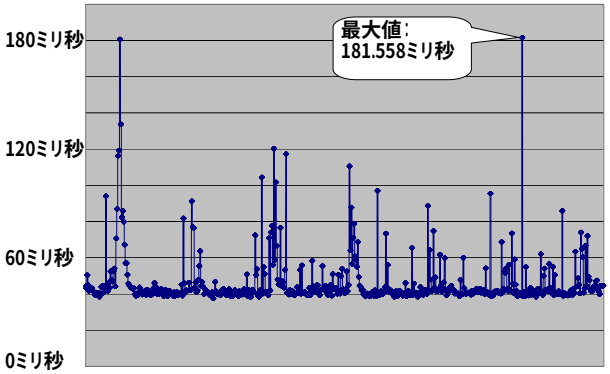


図 6：2 (T_o + T_d) の値

1 週間で取得した offset 値、ネットワーク遅延のサンプル数は 9477 個であった。そのサンプルに対し、以下の生成間隔に設定した場合の監視の成功率を示す。

①生成間隔　182 ミリ秒
全体サンプル数：9477
監視成功数：9477
監視成功率：100%

生成間隔を 182 ミリ秒以上に設定した場合、監視は全て成功する。時刻認証子による時刻配信を実施する際には、事前にネットワーク遅延、offset 値を調査し、監視成功率 100%となる値を用いることを推奨する。また、時刻認証子配信に専用線を使用できない場合、想定外のネットワーク遅延、offset 値が発生する場合が考えられるので、生成間隔を大きめに設定したほうがよい。

②生成間隔　181 ミリ秒
全体サンプル数：9477
監視成功数：9460
監視成功率：99.82061%

生成間隔を 181 ミリ秒以下に設定した場合、監視が失敗するケースが発生する。監視が成功するまで、時刻認証子の非改ざん性を保証できない。

④生成間隔　37 ミリ秒以下（参考）
全体サンプル数：9477
監視成功数：0
監視成功率：0%

生成間隔を 37 ミリ秒以下に設定した場合、監視はまったく成功しない。

3-3 本方式の考察

実際のインターネットを使用した場合の、時刻認証子の監視間隔、生成間隔の理想値を求めることができた。よって、TA による TSA の時刻の正当性をきめ細かく検証することが可能となり、TSA 管理者による時刻改ざんの不正行為の検出・防止が可能となる。今回、実際のインターネット上で計測した値から、時刻認証子の生成間隔は 182 ミリ秒以上が理想的な値となった。よって、TSA 管理者が時刻認証子を改ざんした場合でも、180 ミリ秒以内の改ざんしかできない。現在、1 秒未満の精度を必要とするアプリケーションは見当たらないため、この生成間隔で時刻認証子を生成するのは有効である。また、今後ミリ秒単位の精度を必要とするアプリケーションが普及した場合でも、有効と思われる。

また、生成間隔を 182 ミリ秒に設定した場合でも、offset 値、ネットワーク遅延によっては、監視が成功しないケースが発生する可能性がある。そのような場合、次回以降監視が成功すれば、TSA の時刻認証子の非改ざん性を保証できる。しかし頻繁に監視が失敗するのであれば、生成間隔を再検討しなければならない。

今回の実証実験は、時刻認証子の生成と監査間隔を共に 1 秒に設定して実施した。1 週間の調査期間中で TSA にて生成された時刻認証子のデータ量は、135.5M バイトであった。また、1 週間分の時刻認証子の正当性検証には、約 6.6 秒の時間がかかった（実行環境：ペンティアム 4 3.2GHz、メモリ　1.0Gbyte）。生成間隔、監視間隔を共に 200 ミリ秒にした場合、1 年間で生成される時刻認証子のデータ量は、約 33G バイトと予想される。また、1 年間に生成される時刻認証子の検証に要する時間は、約 28 分程度と予想される。したがって、現在一般に市販されている PC を使用して時刻認証子の検証を実施することは、十分実現可能であると考えられる。

3-4 参考文献

[1] C. Adams, C. Cain, D. Pinkas, R. Zuccherato, “Internet X.509 Public Key Infrastructure Time-Stamp Protocol (TSP)”, RFC3161, August 2001

[2] David L. Mills, “Network Time Protocol (Version 3) Specification, Implementation and Analysis”, RFC 1305, March 1992
[3]島成佳，小松文子，時刻認証子による時刻証明方式，コンピュータセキュリティ研究会 2003
[4]久保寺範和，島成佳，石崎健太郎，小松文子，時刻認証子による配信時刻証明方式，コンピュータセキュリティ研究会 2004

以上

評価報告書

高速・高セキュリティタイムスタンプ付与・検証サブシステム-1
サブシステム(4)-(i)

平成 18 年 2 月 28 日

目次

第1章 評価の目的

1. 目的

リンクトークン方式※1のタイムスタンプについて、統合化プラットフォームシステム上における処理性能を明らかにするとともに、本研究開発の最終的な到達目標である毎秒 10,000 スタンプ以上を可能とする場合の環境を明らかにすることを目的とする。

※１：ここでいうリンクトークン方式タイムスタンプとは、TSAがタイムスタンプ対象データのハッシュ値に対して他のハッシュ値と関連付けるリンク情報を生成し、その時点までに生成したタイムスタンプと関連性を明らかにして有効性を証明する方式。

第2章 評価環境

1. ハードウェア構成

本性能評価で用いるハードウェアの構成を表 2-1に示す。

表 2-1　ハードウェア構成

種別	数量	構成	備考
タイムスタンプサーバ	1	IBM xSeries225 CPU：Xeon 2.80GHz×2 MEM：2GByte HDD：220GB	
クライアント	1	IBM ThinkCentreS50 CPU：Pentium4 2.80GHz×1 MEM：512MByte HDD：40GB	
ルータ 1	1	NTT-ME MN8300 ポート：10Base-T/100BASE-TX	
ルータ 2	1	YAMAHA RTX1100 ポート：10Base-T/100BASE-TX	

2. ソフトウェア構成

本性能評価で用いるソフトウェアの構成を表 2-2に示す。

表 2-2　ソフトウェア構成

種別	ソフトウェア	備考
タイムスタンプサーバ	Red Hat Enterprise Linux AS 2.1	
	PostgreSQL	
	Apache	
	タイムスタンプソフトウェア	
	時刻情報受信ソフトウェア(TSAProxy)	
クライアント	Windows 2000 Professionl	ServicePack4
	性能測定用クライアントソフトウェア	

3. システム構成

3-1 ネットワーク構成

タイムスタンプサーバとクライアントをインターネット経由で接続する。
ネットワーク構成を図 2-1に示す。

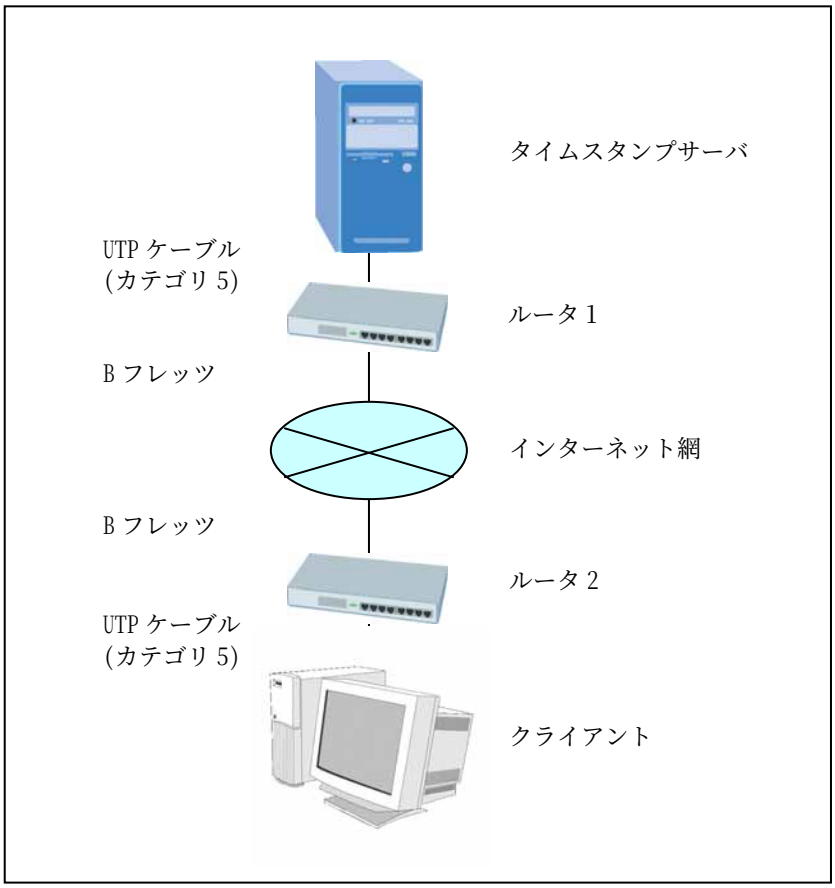


図 2-1　ネットワーク構成

3-2 通信速度及び通信モード

タイムスタンプサーバとクライアント間の通信速度及び通信モードを表 2-3に示す。

表 2-3　通信速度及び通信モード

#	ネットワーク機器	接続先	通信速度	通信モード
1	ルータ 1	タイムスタンプサーバ	100Mbps	全二重
2		WAN	100Mbps	全二重
3	ルータ 2	WAN	100Mbps	全二重
4		クライアント	100Mbps	全二重

第3章 処理概要

1. タイムスタンプ発行機能

1-1 機能概要

タイムスタンプ発行機能の概要について、図 3-1に示す。

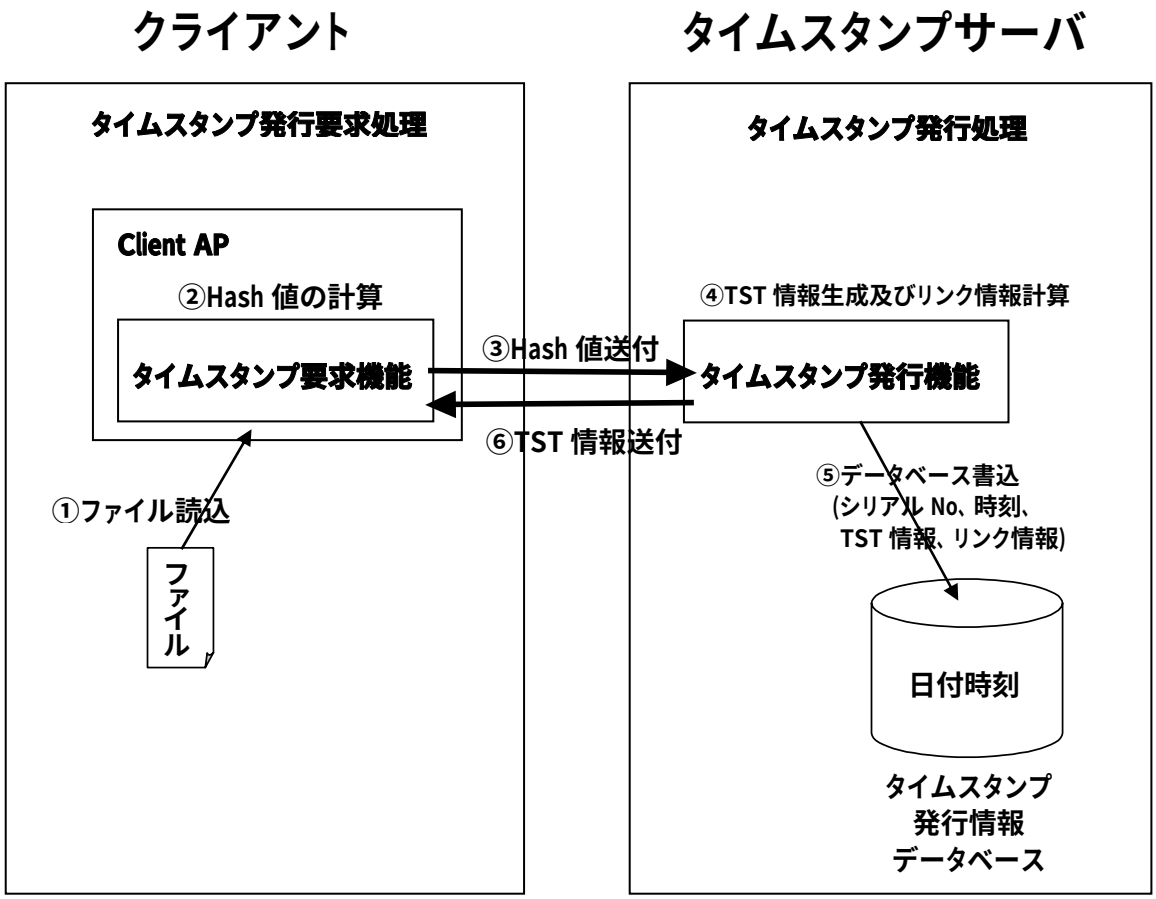


図 3-1　タイムスタンプ発行機能

本性能評価は、タイムスタンプサーバのタイムスタンプ発行処理に関する処理性能を明らかにすることを目的としているため、タイムスタンプサーバの処理性能に係らないクライアントでの TST 情報のファイルへの書き込みは行っていない。

1-2 処理フロー

タイムスタンプ発行機能の処理フローについて、以下に示す。

① (クライアント)：原本となるファイルを読み込む。
② (クライアント)：読み込んだファイルの Hash 値を計算する。
③ (クライアント)：タイムスタンプ発行要求を行う。(Hash 値の送信)
※原本ファイルは送信しない。
④ (タイムスタンプサーバ)：受信した Hash 値に対する TST 情報を生成して、リンク情報の計算を行う。
⑤ (タイムスタンプサーバ)：シリアル No、時刻、TST 情報、リンク情報をデータベースへ書き込む。
⑥ (タイムスタンプサーバ)：タイムスタンプ発行応答を行う。

性能測定用クライアントソフトウェアで②、③の処理を同時並行で複数行うことにより、タイムスタンプサーバに大量のタイムスタンプ発行要求を送信する。

第4章 評価方法

1. 評価概要

タイムスタンプサーバに対して、クライアントから大量のタイムスタンプ発行要求を送信して、タイムスタンプサーバのタイムスタンプ発行処理に関する処理性能を明らかにする。

2. 評価条件

2-1 ネットワーク条件

本性能評価では、インターネットを介してタイムスタンプサーバに接続して、測定を行った。

2-2 サーバ条件

タイムスタンプ発行処理において、タイムスタンプ発行情報データベースに対して行う処理は、insert 処理となる。

データベースへの insert 処理においては、登録されている全体件数による処理性能への影響は無いと考えられる。

従って、測定の間は、各測定点でデータベースのクリアは行わないこととした。

2-3 クライアント条件

大量のタイムスタンプ発行要求を送信するために、タイムスタンプサーバに係らないクライアントでの TST 情報のファイルへの書き込みは行わない。

3. 評価項目

本性能評価では、以下の評価項目について測定値を取得して、タイムスタンプサーバのタイムスタンプ処理に関する処理性能を明らかする。

(1) 平均処理件数

リンクトークン方式のタイムスタンプサーバでは、タイムスタンプを発行する前に、タイムスタンプ発行情報データベースにタイムスタンプ情報の書き込みを行う。従って、タイムスタンプ発行情報データベースを参照することにより、1 秒あたりの平均処理件数を確認する。

(2) 処理時間

タイムスタンプ発行情報データベースを参照することにより、処理時間を確認する。

(3) 平均 CPU 使用率

sar コマンドで処理中の CPU 使用率を確認する。

(4) 平均 DISK スワップ領域使用率

sar コマンドで処理中の DISK スワップ領域使用率を確認する。

(5) TST 発行件数

タイムスタンプ発行情報データベースを参照することにより、測定毎の TST 発行件数を確認する。

4. 測定手順

測定手順を以下に示す。

(1) タイムスタンプサーバの稼動状況を取得

タイムスタンプサーバの稼動状況を秒単位で取得するために、以下のコマンドを投入する。

投入するコマンド：「sar -o *file.log* 1 0 > /dev/null」

(2) タイムスタンプ発行要求の送信

性能測定用クライアントソフトウェアを用いてタイムスタンプサーバへ大量のタイムスタンプ発行要求を送信する。

(3) 取得ファイルの形式変換

処理完了後、sar コマンドを停止させる。sar コマンドで取得したファイルはバイナリ形式なので、以下のコマンドでテキスト形式に変換する。

投入するコマンド：「sar -A -f *file.log* > *file.log.txt*」

(4) 処理結果確認

タイムスタンプ発行情報データベースのテーブルを参照して、単位時間あたりの処理件数を確認する。データベースにログイン後、以下の SQL 文を投入する。

投入するコマンド：「select time, count(*) from notarized_info where time
between '*YYYY-MM-DD hh:mm:ss*'　← 開始日時
and
'*YYYY-MM-DD hh:mm:ss* '　← 終了日時
group by time order by time;」

また、sar コマンドで取得したタイムスタンプサーバの CPU 使用率及び DISK スワップ領域使用率を確認する。

5. タイムスタンプ発行要求の送信方法

クライアントから送信するタイムスタンプ発行要求は、性能測定用クライアントソフトウェアに表 4-1に示す「スレッド数」、「1 スレッドの発行要求数」を設定して行う。

表 4-1　タイムスタンプ発行要求の送信方法

測定方法	スレッド数	1 スレッドの発行要求数	送信件数
測定 1	10	200	2000
測定 2	20	100	2000
測定 3	40	50	2000
測定 4	50	40	2000

第5章 測定結果

1. 平均処理件数の算出方法

処理件数の平均値の算出においては、以下の方法でデータを集計することで、平均値の精度を高めた。

- 処理開始直後の 3 秒間及び処理終了直前の 3 秒間は、タイムスタンプ発行要求数が安定しないため、集計の対象外とする。

日付	時間	処理件数	
2005/12/13	10:50:44	4	対象外
2005/12/13	10:50:45	105	
2005/12/13	10:50:46	96	
2005/12/13	10:50:47	84	
2005/12/13	10:50:48	90	
2005/12/13	10:50:49	81	
2005/12/13	10:50:50	66	
2005/12/13	10:50:51	61	

・
・
・

2005/12/13	10:51:04	82	
2005/12/13	10:51:05	89	
2005/12/13	10:51:06	89	
2005/12/13	10:51:07	74	
2005/12/13	10:51:08	80	対象外
2005/12/13	10:51:09	52	
2005/12/13	10:51:10	29	

図 5-1　平均処理件数の算出方法

2. 測定データ

2-1 測定値

タイムスタンプ発行処理の測定結果を以下に示す。

■ 測定1： スレッド数： 10
1スレッドの発行要求数: 200
送信件数： 2000

表 5-1　測定 1 の測定結果

実施回数	1	2	3	4	5	平均
平均処理件数(件数/秒)	75.24	80.65	80.95	81.26	81.05	79.83
処理時間 (秒)	27	26	26	25	26	26
平均CPU使用率 (%)	21.4	22.34	23.08	23.57	23.3	22.74
平均DISKスワップ領域使用率 (%)	0.00	0.00	0.00	0.00	0.00	0.00
TST発行件数 (件数)	2000	2000	2000	2000	2000	2000

■ 測定2： スレッド数： 20
1スレッドの発行要求数: 100
送信件数： 2000

表 5-2　測定 2 の測定結果

実施回数	1	2	3	4	5	平均
平均処理件数(件数/秒)	77.95	78.21	80.95	81.74	79.2	79.61
処理時間 (秒)	27	25	26	25	26	26
平均CPU使用率 (%)	23.11	22.76	23.88	24.61	23.48	23.57
平均DISKスワップ領域使用率 (%)	0.00	0.00	0.00	0.00	0.00	0.00
TST発行件数 (件数)	2000	2000	2000	2000	2000	2000

■ 測定3： スレッド数： 40
1スレッドの発行要求数: 50
送信件数： 2000

表 5-3　測定3の測定結果

実施回数	1	2	3	4	5	平均
平均処理件数(件数/秒)	80.9	81.55	81.16	79.67	81.5	80.96
処理時間 (秒)	26	26	25	27	26	26
平均CPU使用率 (%)	23.48	23.84	23.18	23.42	23.6	23.5
平均DISKスワップ領域使用率 (%)	0.00	0.00	0.00	0.00	0.00	0.00
TST発行件数 (件数)	2000	1996	1987	2000	2000	1997

■ 測定4： スレッド数： 50
1スレッドの発行要求数: 40
送信件数： 2000

表 5-4　測定4の測定結果

実施回数	1	2	3	4	5	平均
平均処理件数(件数/秒)	81.25	80	80.1	81.32	79.19	80.37
処理時間 (秒)	27	25	25	26	27	26
平均CPU使用率 (%)	25.45	23.98	23.77	23.29	23.06	23.91
平均DISKスワップ領域使用率 (%)	0.00	0.00	0.00	0.00	0.00	0.00
TST発行件数 (件数)	1997	1990	1997	1999	1990	1995

2-2 測定結果グラフ及び考察

2-2-1 平均処理件数

(1) 測定結果グラフ

実施回数毎の平均処理件数の推移を図 5-2に示す。

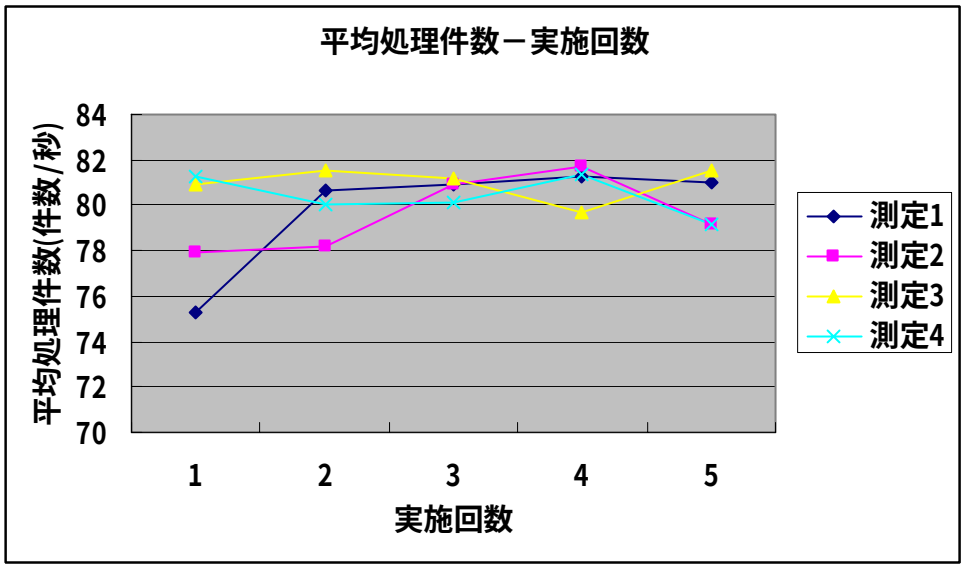


図 5-2　各測定における平均処理件数

(2) 平均処理件数の考察

性能測定用クライアントソフトウェアのスレッド数を変更しても、平均処理件数に大きな変化は見られなかった。そして各測定における平均処理件数は毎秒 80 件程度であった。従って、タイムスタンプサーバの平均処理件数は毎秒 80 件程度であると考えられる。

2-2-2 処理時間

(1) 測定結果グラフ

実施回数毎の処理時間の推移を図 5-3に示す。

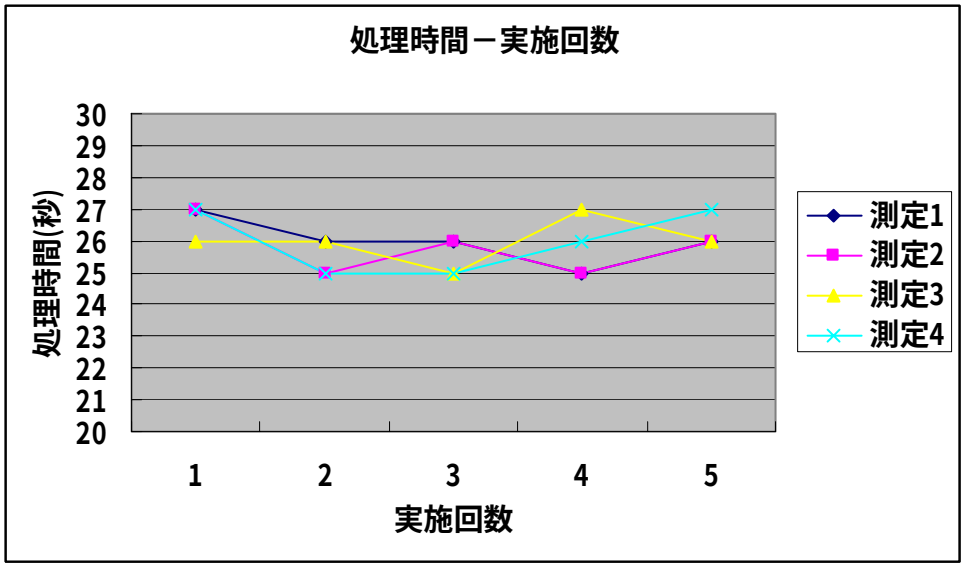


図 5-3　各測定における処理時間

(2) 処理時間の考察

性能測定用クライアントソフトウェアのスレッド数を変更しても、処理時間に大きな変化は見られなかった。

2-2-3 平均 CPU 使用率

(1) 測定結果グラフ

実施回数毎の平均 CPU 使用率の推移を図 5-4に示す。

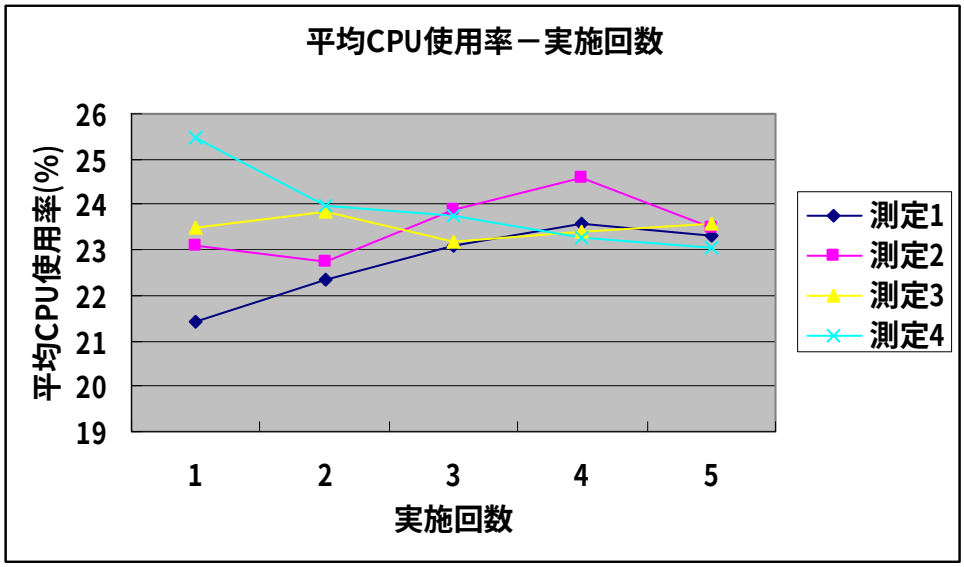


図 5-4　各測定における平均 CPU 使用率

(2) 平均 CPU 使用率の考察

性能測定用クライアントソフトウェアのスレッド数を変更しても、平均 CPU 使用率に大きな変化は見られなかった。また、平均 CPU 使用率は 23%程度で極端に大きな値ではないので、タイムスタンプ発行処理における CPU に対する負荷は低いと考えられる。

2-2-4 平均 DISK スワップ領域使用率

(1) 測定結果グラフ

実施回数毎の平均 DISK スワップ領域使用率の推移を図 5-5に示す。

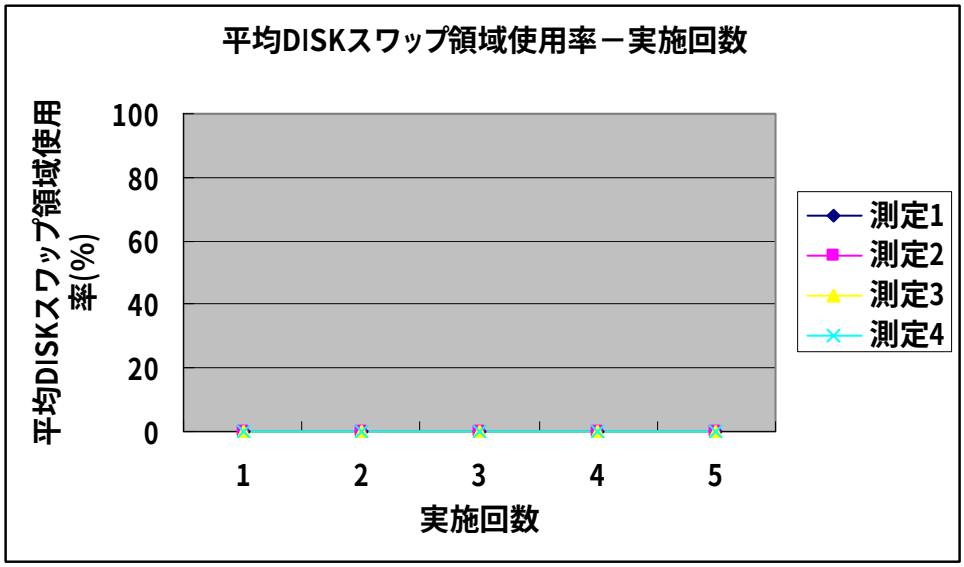


図 5-5　各測定における平均 DISK スワップ領域使用率

(2) 平均 DISK スワップ領域使用率の考察

各測定において DISK スワップは発生しなかった。従って、物理メモリ量は十分に確保されていると考えられる。

2-2-5 TST 発行件数

(1) 測定結果グラフ

実施回数毎の TST 発行件数の推移を図 5-6に示す。

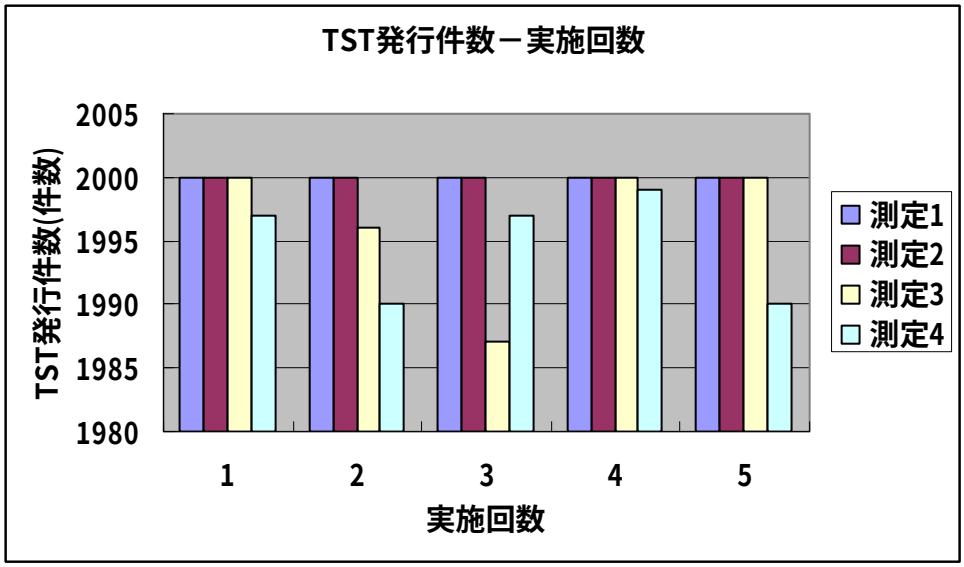


図 5-6　各測定における TST 発行件数

(2) TST 発行件数

測定 3 及び測定 4 において全てのタイムスタンプが正常に発行されなかった。また、その時にデータベースのログにデータベースへの接続に失敗した旨を示すメッセージが出力されていた。これは性能測定用クライアントソフトウェアのスレッド数の増加に伴い、データベースへの接続数が最大接続数に達して書き込みができなかったためであると考えられる。

第6章 到達目標を達成するための環境

1. 到達目標達成に向けた取り組み

最終的な到達目標を以下に示す。

最終的な到達目標： 毎秒 10,000 スタンプ以上処理できること。

最終的な到達目標については、平成１５年度及び平成１６年度に順次性能向上を重視した測定環境を構築し、平成１６年度の性能評価において達成されている。

2. 平成１５年度性能評価結果

到達目標を実現する上で、セキュリティや実証実験にむけた信頼性を重視した今年度の統合化プラットフォームシステムに組み込まれているタイムスタンプサーバ環境においては、まず以下の３点がボトルネックとなっていると考えられた。

- データベースへの書き込みによるオーバヘッド
- SSL 通信におけるサーバ認証、通信データの暗号化及びメッセージ認証
- リンク情報生成におけるハッシュ関数の２重化

平成１５年度の測定においては、性能を重視して、表 6-1に示す通り上記のボトルネックへの対処を行った測定環境における性能評価を実施した。

表 6-1　性能上のボトルネックへの対処（平成１５年度）

#	ボトルネック	対処
1	データベースへの書き込みによるオーバヘッド	●DB への遅延書込み 全てのタイムスタンプ発行要求に対するタイムスタンプ情報をメモリ上に生成してから、データベースへの書き込みをまとめて行うようにプログラムを変更し、データベースへの書き込みによるオーバヘッドを排除した。
2	SSL 通信におけるサーバ認証	●HTTP 通信の採用 タイムスタンプの発行に係る通信において、HTTPS による通信を行わず、HTTP による通信を採用した。
3	リンク情報生成におけるハッシュ関数の２重化	●ハッシュ関数の単一化 リンク情報の生成においては、SHA-512 のみを使用する処理方式に変更した。

平成１５年度の測定環境における性能評価においては、以下の処理性能が得られている。

平均処理件数： 毎秒 3,000 件程度

参考として、平成１５年度の性能評価におけるタイムスタンプ発行処理の測定データを表 6-2に示す。

表 6-2　タイムスタンプ発行処理の測定データ（平成１５年度）

サーバプロセス数			1	2	3	4	6	10	16	25	40
パターン1	クライアント2×3000	最大処理件数(件)	2,036	2,518	2,644	2,624	2,623	2,604	2,593	2,580	2,592
		平均処理件数(件)	2,023	2,503	2,637	2,618	2,616	2,597	2,574	2,575	2,584
		処理時間(秒)	5	4	4	4	3	4	3	4	4
		CPU使用率平均(%)	11.45	14.75	17.38	17.69	19.75	17.31	20.00	20.06	19.38
		スワップ領域使用率平均(%)	5.08	5.08	5.08	5.08	5.08	5.08	5.08	5.08	5.08
パターン2	クライアント5×2000	最大処理件数(件)	1,906	3,058	3,286	3,305	3,412	3,363	3,344	3,366	3,377
		平均処理件数(件)	1,861	3,018	3,236	3,248	3,339	3,313	3,275	3,322	3,330
		処理時間(秒)	10	6	5	6	6	5	6	6	6
		CPU使用率平均(%)	10.28	18.67	22.30	23.46	22.38	28.80	28.63	26.66	26.58
		スワップ領域使用率平均(%)	5.08	5.08	5.08	5.08	5.08	5.08	5.08	5.08	5.08
パターン3	クライアント8×2000	最大処理件数(件)	1,935	3,003	3,332	3,296	3,279	3,396	3,330	3,297	3,446
		平均処理件数(件)	1,889	2,934	3,262	3,249	3,218	3,311	3,201	3,230	3,341
		処理時間(秒)	17	11	10	10	10	9	10	9	10
		CPU使用率平均(%)	10.29	16.05	19.68	21.45	23.53	28.92	27.00	29.25	27.69
		スワップ領域使用率平均(%)	5.08	5.08	5.08	5.08	5.08	5.08	5.08	5.08	5.08

サーバプロセス数			63	100	158	251	398	631	1000	1585
パターン1	クライアント2×3000	最大処理件数(件)	2,602	2,601	2,557	2,576	2,577	2,555	2,556	611
		平均処理件数(件)	2,577	2,576	2,552	2,560	2,570	2,551	2,552	150
		処理時間(秒)	4	4	4	4	4	5	4	155
		CPU使用率平均(%)	17.38	21.00	21.31	19.31	21.19	19.75	18.94	21.64
		スワップ領域使用率平均(%)	5.08	5.08	5.08	5.08	5.08	5.08	5.23	99.95
パターン2	クライアント5×2000	最大処理件数(件)	3,383	3,308	3,323	3,389	3,386	3,396	3,359	
		平均処理件数(件)	3,272	3,288	3,264	3,303	3,348	3,340	3,287	
		処理時間(秒)	6	5	6	6	6	6	6	
		CPU使用率平均(%)	27.67	29.10	28.33	28.92	30.08	30.75	28.67	
		スワップ領域使用率平均(%)	5.08	5.08	5.08	5.08	5.08	5.08	5.23	
パターン3	クライアント8×2000	最大処理件数(件)	3,329	3,388	3,369	3,337	3,335	3,340	3,475	
		平均処理件数(件)	3,245	3,293	3,307	3,254	3,290	3,282	3,387	
		処理時間(秒)	10	10	10	10	10	10	10	
		CPU使用率平均(%)	27.64	27.00	27.33	28.05	27.35	27.92	29.75	
		スワップ領域使用率平均(%)	5.08	5.08	5.08	5.08	5.08	5.08	5.09	

また、平成１５年度の性能評価におけるタイムスタンプ発行処理のサーバの受付プロセス数の変更による、各パターンでの平均処理件数の推移を図 6-1に示す。

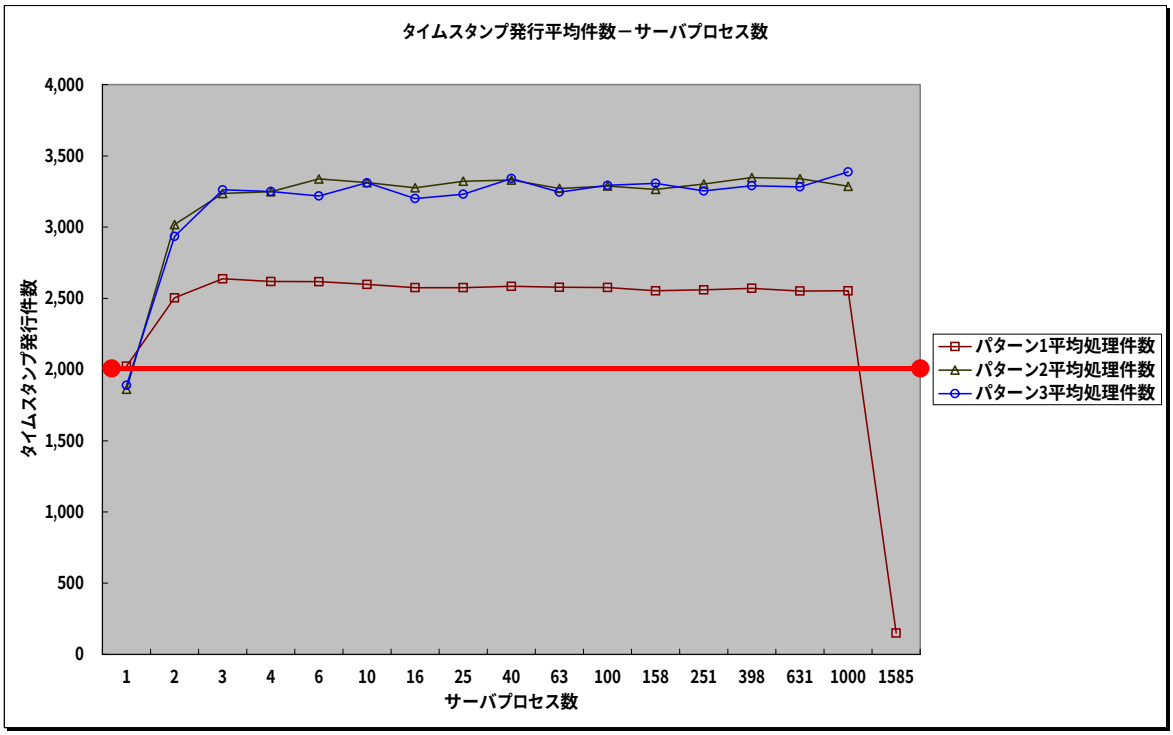


図 6-1　タイムスタンプ発行平均件数（平成１５年度）

3. 平成１６年度性能評価結果

到達目標を実現する上で、平成１５年度の測定環境においては、更に以下の１点がボトルネックとなっていると考えられた。

・　サーバ内プロセス間通信のオーバヘッド

平成１６年度の測定においては、更に性能を重視して、表 6-3に示す通り上記のボトルネックへの対処を行った測定環境における性能評価を実施した。

表 6-3　性能上のボトルネックへの対処（平成１６年度）

#	ボトルネック	対処
1	サーバ内プロセス間通信のオーバヘッド	●発行要求の一括処理 一度だけ受付プロセスからサーバプロセスへの情報の受け渡しを行う。そして、その情報を元に 30 万件のタイムスタンプ情報を生成するようにプログラムを変更し、サーバ内のプロセス間通信のオーバヘッドを排除する。

平成１６年度の測定環境における性能評価においては、以下の処理性能が得られており、最終的な到達目標が達成されている。

平均処理件数：　毎秒 70,000 件程度

参考として、平成１６年度の性能評価におけるタイムスタンプ発行処理の測定データを表 6-4に示す。

表 6-4　タイムスタンプ発行処理の測定データ（平成１６年度）

実施回数	1	2	3	4	5	6	7	8	9	10
最大処理件数(件数/秒)	71,780	71,822	71,732	71,704	71,735	71,731	71,329	71,578	71,805	71,918
平均処理件数(件数/秒)	71,755	71,588	71,604	71,423	71,158	71,164	71,030	71,277	71,010	71,649
処理時間 (秒)	5.0	5.0	5.0	5.0	5.0	5.0	5.0	5.0	6.0	5.0
CPU使用率平均 (%)	24.42	23.19	24.50	24.83	21.88	22.50	24.92	24.75	22.44	23.83
メモリ使用率平均 (%)	15.19	13.59	13.48	13.48	13.46	13.39	14.41	13.45	13.50	13.57

また、平成１６年度の性能評価におけるタイムスタンプ発行処理の測定毎の平均処理件数の推移を図 6-2に示す。

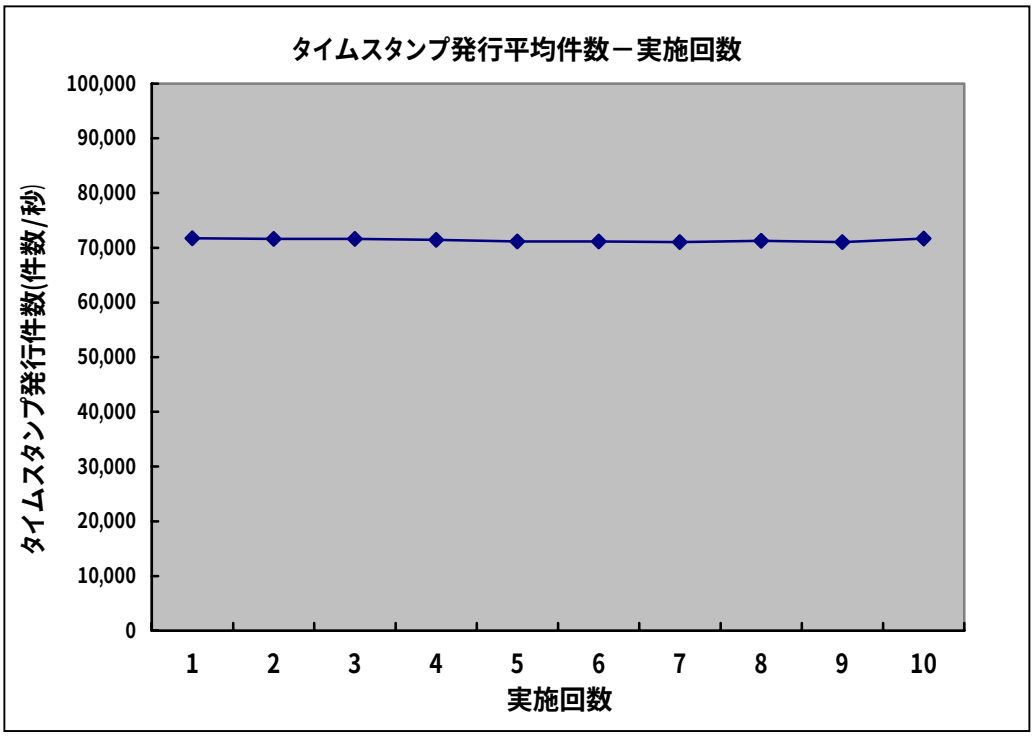


図 6-2　タイムスタンプ発行平均件数（平成１６年度）

4. 今年度の評価環境について

平成１５年度及び平成１６年度においては、性能を重視した評価環境においてタイムスタンプ処理性能を測定し、到達目標を達成したが、今年度は実証実験への適用等を踏まえ、セキュリティや動作の信頼性、多様な利用形態への対応等の実用性を重視した環境を構築し、処理性能の測定を実施している。

平成１５年度及び平成１６年度において実施した性能向上に向けた対処と、それらの対処に対する今年度の性能評価における取扱いを、以下の表 6-5に記載する。

表 6-5　これまでの性能向上に向けた対処の取扱い

#	これまでの対処	今年度の取扱い
1	DB への遅延書込み	データベースに書き込む前に何らかの不具合によってメモリ上のデータが失われるといった事象の発生が想定され、実用性の観点から動作の信頼性をより向上させるため、発行毎にデータベースに書き込む処理の流れとした。
2	HTTP 通信の採用	TSA への成り済まし等の不正が想定され、実用性の観点からセキュリティをより向上させるため、HTTPS を採用した。
3	ハッシュ関数の単一化	単一のハッシュ関数でもタイムスタンプの有効性の確保は可能であるが、ハッシュ関数の脆弱化が発生する状況を想定し、実用性の観点からセキュリティをより向上させるため、ハッシュ関数を２重化した方式を採用した。
4	発行要求の一括処理	今年度の実証実験においては、１件毎に発行要求を送信するアプリケーションを対象としたため、実用性の観点から多様な利用形態への対応を可能とするため、１件毎に発行要求を受付ける方式を採用した。

第7章 まとめ

1. 試験の総評

タイムスタンプサーバの 1 秒あたりの平均処理件数は、以下に示す通りであることが確認できた。

1秒あたりの平均処理件数： 毎秒 80 件程度

また、CPU 使用率及びメモリ容量はボトルネックになっていないことが確認できた。処理のボトルネックと考えられるのは、データベースへの書き込みによるオーバヘッド、SSL 通信におけるサーバ認証、リンク情報生成におけるハッシュ関数の２重化、サーバ内プロセス間通信のオーバヘッドが予測される。

2. 最終的な到達目標

最終的な到達目標を以下に示す。

最終的な到達目標： 毎秒 10,000 スタンプ以上処理できること。

上記の到達目標については、性能を重視した平成１６年度の測定環境における性能評価において以下の処理性能が得られており、達成されている。

平均処理件数： 毎秒 70,000 件程度

評価報告書

高速･高セキュリティタイムスタンプ付与・検証サブシステム-2
サブシステム(4)-(ii)

平成 18 年 2 月 28 日

目次

第1章 評価の目的

1. 目的

本サブシステムは、長期保存にも対応する暗号強度の高い鍵長と、大量のトランザクション要求にも耐えられる処理能力を持つ、高速タイムスタンプサーバの実現を目的に、平成 16 年度に開発した装置を基にした以下の機能の追加、性能向上についての開発・評価を行う。

(1) 安全性の高いハッシュ関数のへの対応機能

SHA-1 の脆弱性についての示唆に伴い、対処が必要な箇所について分析し、必要に応じて安全性の高いハッシュ関数に変更する。

(2) タイムスタンプ処理能力

統合化プラットフォームシステム上において、独立トークン方式＊1のタイムスタンプについて、必要に応じて安全性の高いハッシュ関数へ変更した上で、ネットワーク環境や機器性能等を示し、処理性能を明らかにする。

(3) トレーサビリティ機能との連携

統合化プラットフォームシステム上において、トレーサビリティ機能との連携に必要なインタフェースを備える。

(4) セキュリティ評価

統合化プラットフォームシステムにおいて提供される独立トークン方式のタイムスタンプについて、セキュリティの観点からその妥当性について分析する。

＊1：ここでいう独立トークン方式タイムスタンプとは、TSA がタイムスタンプ対象データのハッシュ値に対してデジタル署名を行い、それぞれのタイムスタンプの有効性を証明する方式。

なお、セキュリティ評価については、セキュリティ評価報告書に記載する。

第2章 評価対象システム

1. ネットワーク構成図、機能ブロック図、測定環境

評価対象環境について、ネットワーク構成図、機能ブロック図、測定環境を以下に記載する。

※機能ブロック図は、本年度の評価に関係のある項目のみ記載する。

なお、統合化プラットフォームシステムにおける NTA、TA、TSA の名称を、以降次のように記載する。

表 2-1　統合化プラットフォームシステムにおける略称

名称	略称
配信時刻高精度高信頼化サブシステム－２と接続するための仮想国家時刻標準機関	NTA1
配信時刻高精度高信頼化サブシステム－２	TA1
高速・高セキュリティタイムスタンプ付与・検証サブシステム－２	TSA2

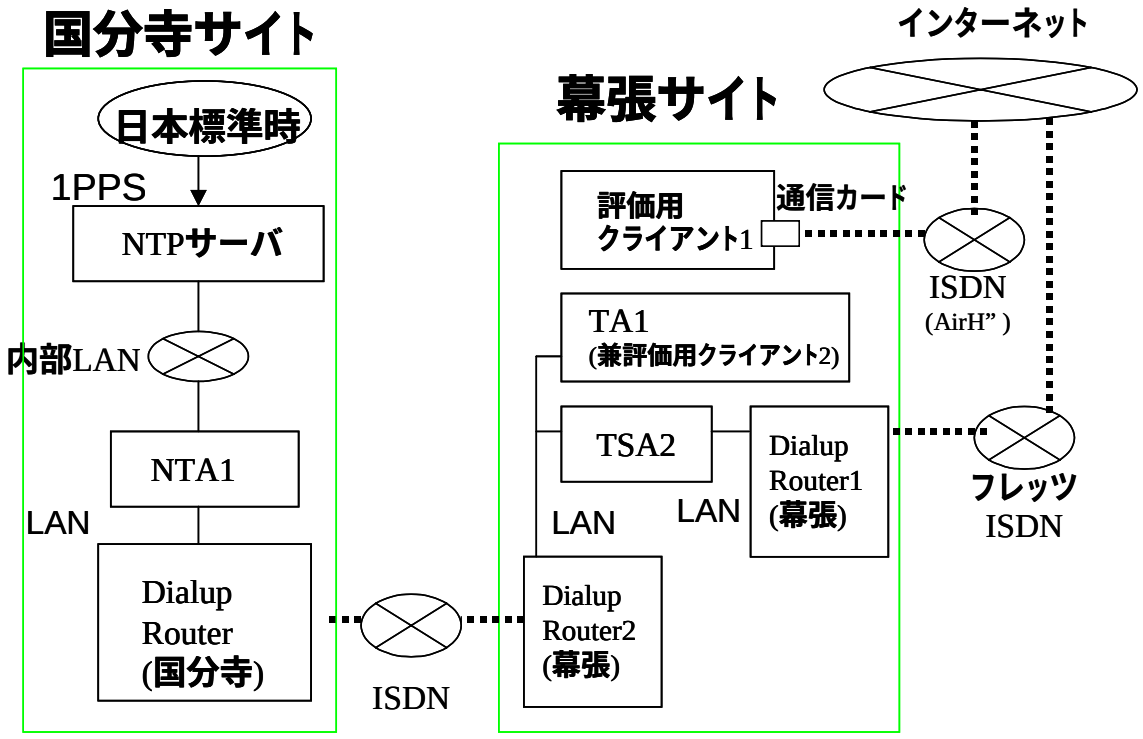


図 2-1　ネットワーク構成図

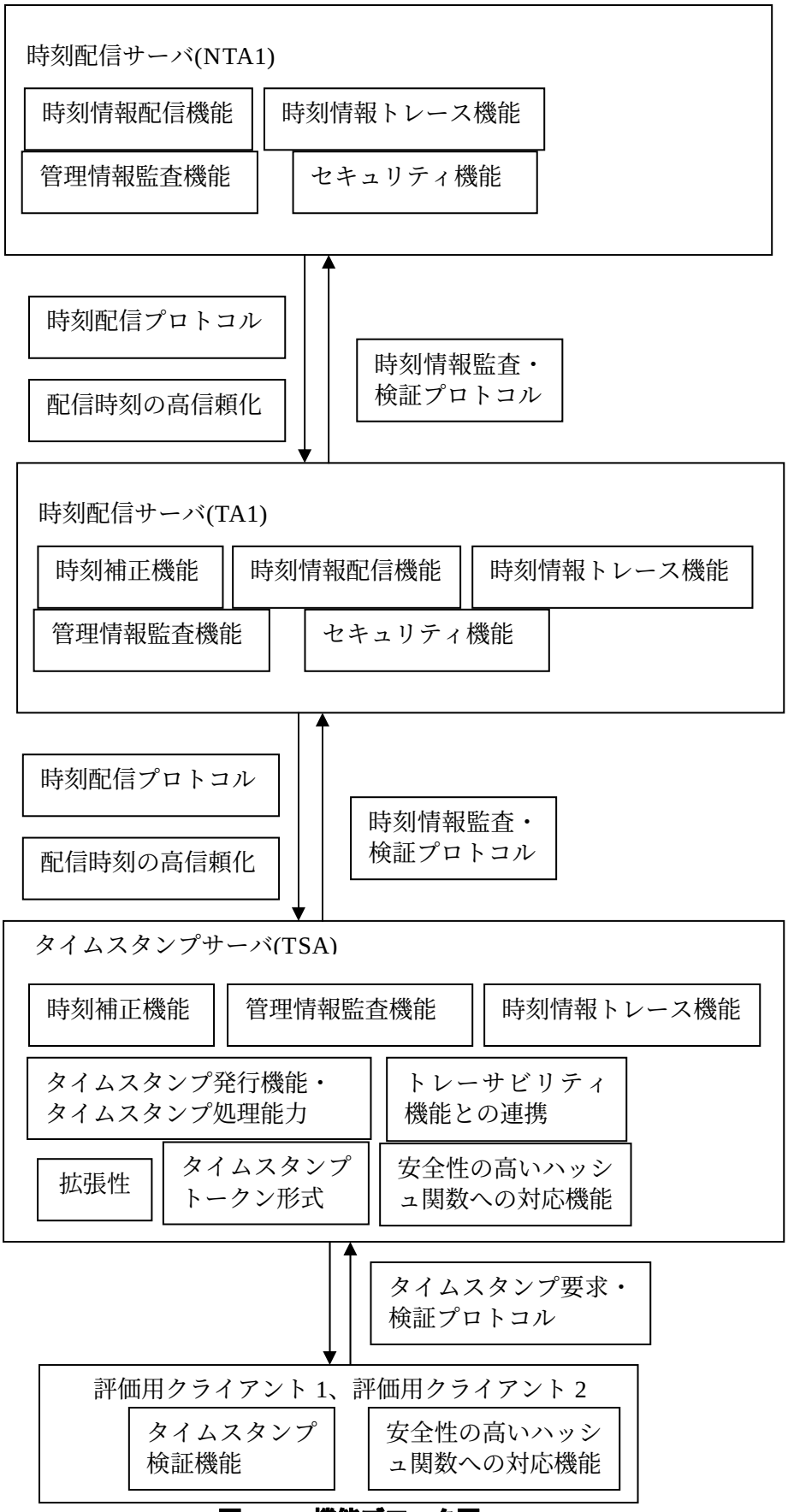

時刻配信サーバ(NTA1)
時刻情報配信機能
時刻情報トレース機能
管理情報監査機能
セキュリティ機能
時刻配信プロトコル
配信時刻の高信頼化
時刻情報監査・
検証プロトコル
時刻配信サーバ(TA1)
時刻補正機能
時刻情報配信機能
時刻情報トレース機能
管理情報監査機能
セキュリティ機能
時刻配信プロトコル
配信時刻の高信頼化
時刻情報監査・
検証プロトコル
タイムスタンプサーバ(TSA)
時刻補正機能
管理情報監査機能
時刻情報トレース機能
タイムスタンプ発行機能・
タイムスタンプ処理能力
トレーサビリティ
機能との連携
拡張性
タイムスタンプ
トークン形式
安全性の高いハッシュ関数への対応機能
タイムスタンプ要求・
検証プロトコル
評価用クライアント1、評価用クライアント2
タイムスタンプ
検証機能
安全性の高いハッシュ関数への対応機能

図 2-2　機能ブロック図

表 2-2　測定環境 1(インターネット経由でのタイムスタンプ)

	TSA2	評価用クライアント 1
機器名	DELL PowerEdge 650	Panasonic CF-R3
CPU	Pentium4 2.4GHz	Pentium M 1.1GHz
メモリ	1GB	256MB
OS	Red Hat Professional Workstation	Windows XP
HSM 性能	1024bit:150 署名/秒 2048bit:非公開	ー
ソフトウェア	タイムスタンプソフトウェア (SecureTSS) (タイムスタンプ発行)	評価用クライアントソフトウェア (タイムスタンプ要求)
ネットワークインタフェース	1000BASE-T	通信カード (AirH" 128kbps)
回線に接続するためのネットワーク機器	ダイヤルアップルータ	通信カード (AirH" 128kbps)
回線	フレッツ ISDN (64Kbps) ※ベストエフォート	ISDN (AirH" 128kbps)
暗号アルゴリズム	・署名アルゴリズム SHA1withRSA	・ハッシュアルゴリズム SHA-512

表 2-3　測定環境 2(LAN 経由でのタイムスタンプ)

	TSA2	評価用クライアント 2
機器名	DELL PowerEdge 650	DELL PowerEdge 650
CPU	Pentium4 2.4GHz	Pentium4 2.4GHz
メモリ	1GB	1GB
OS	Red Hat Professional Workstation	Red Hat Professional Workstation
HSM 性能	1024bit:150 署名/秒 2048bit:非公開	ー
ソフトウェア	タイムスタンプソフトウェア (SecureTSS) (タイムスタンプ発行)	評価用クライアントソフトウェア (タイムスタンプ要求)
ネットワークインタフェース	1000BASE-T	1000BASE-T
回線に接続するためのネットワーク機器	ダイヤルアップルータ	ダイヤルアップルータ
回線	LAN	LAN
暗号アルゴリズム	・署名アルゴリズム SHA1withRSA	・ハッシュアルゴリズム SHA-512

表 2-4　参考：NTA1-TA1-TSA2 の時刻配信に関する環境

	NTA1	TA1	TSA2
機器名	DELL PowerEdge 600SC	DELL PowerEdge 650	DELL PowerEdge 650
CPU	Celeron 2GHz	Pentium4 2.4GHz	Pentium4 2.4GHz
メモリ	512MB	1GB	1GB
OS	Red Hat Professional Workstation	Red Hat Professional Workstation	Red Hat Professional Workstation
時刻配信・受信に使用する時刻	システム時刻	システム時刻	HSM 時刻
時刻配信・受信に使用するソフトウェア	時刻情報配信ソフトウェア (TACServer) （配信）	時刻情報配信ソフトウェア (TACServer) （配信・受信）	タイムスタンプソフトウェア (SecureTSS) （受信）
ネットワークインタフェース	1000BASE-T	1000BASE-T	1000BASE-T
回線に接続するためのネットワーク機器	NTA1-TA1：ダイヤルアップルータ	TA1-TSA2：ダイヤルアップルータ	TA1-TSA2：ダイヤルアップルータ
回線	NTA1-TA1：ISDN (64Kbps)	TA1-TSA2：LAN	TA1-TSA2：LAN
配信間隔	NTA1-TA1：20 分	TA1-TSA2：1 分	ー

※HSM 時刻：HSM の時計の時刻と、TA から受信した時刻情報をもとに生成した時刻

第3章 評価内容・評価結果

1. 評価内容・評価結果

本サブシステムでは、下記の内容に関する評価を行い、すべて期待する結果が得られることを確認した。

・社内試験：5 項目
本試験は、仕様書に記載された要求項目に関する試験である。
（詳細は社内試験成績書を参照。）

・総合試験：40 項目
本試験は、開発されたシステムおよび各機能に対して、社内環境にて実施した試験である。
（詳細は総合試験成績書（社内試験)を参照。）

・試験結果まとめ

全試験項目数	：45 項目
合格した試験項目数	：45 項目
不合格の試験項目数	： 0 項目

第4章 本サブシステムの成果

本サブシステムは、長期保存にも対応する暗号強度の高い鍵長と、大量のトランザクション要求にも耐えられる処理能力を持つ、高速タイムスタンプサーバの実現を目的に、平成 16 年度に開発した装置を基にした以下の機能の追加、性能向上についての開発・評価を行った。その結果、以下の成果を得ることができた。

1. 安全性の高いハッシュ関数への対応機能

本サブシステムでは、SHA-1 の脆弱性についての示唆に伴い、対処が必要な箇所について下記の分析を行い、タイムスタンプ要求に含まれるメッセージダイジェストのハッシュ関数を安全性の高い SHA-512 へ変更した。また、タイムスタンプ要求に含まれるメッセージダイジェストのハッシュ関数が SHA-1 の場合は、そのタイムスタンプ要求を拒否する機能を実装した。

【SHA-1 の脆弱性に関する参考文献】
タイムビジネス推進協議会にて、下記の資料が公開されている。
(http://www.scat.or.jp/time/)
・「SHA-1 脆弱化（衝突困難性の脆弱化）に関する公開情報について　～ハッシュ関数編～」
・「時刻認証サービスに用いる暗号化技術の安全性確保について」
・「タイムスタンプにおける SHA-1 問題に関する対応策検討のための説明補足図」

1-1 ハッシュ関数の脆弱性に関する示唆

今回指摘されているハッシュ関数の脆弱性に関する示唆とは、以下の内容である。
（出典：「SHA-1 脆弱化（衝突困難性の脆弱化）に関する公開情報について　～ハッシュ関数編～」）

・安全なハッシュ関数は、以下の性質を持つ。
『性質１』原像計算困難性：
与えられたハッシュ値から、入力メッセージを得ることが、困難であること。
『性質２』第２原像計算困難性：
与えられた入力メッセージに対して、ハッシュ値が等しくなる、異なる入力メッセージを得ることが、困難であること。
『性質３』衝突困難性：
ハッシュ値が等しくなるような入力メッセージを二以上得ることが、困難であること。

・今回の指摘[*1]の内容は、SHA-1 について、『衝突困難性（性質３）』に係る計算量が 2^80 から 2^69 に低下し、十分な計算量ではなくなったとされるものである。
(＊１：http://theory.csail.mit.edu/~yiqun/shanote.pdf)

・今回指摘されている SHA-1 に係る安全性の低下は、『衝突困難性（性質３）』に関するものであり、『原像計算困難性（性質１）』および『第２原像計算困難性（性質２）』に関しては特に指摘されていない。

・『衝突困難性（性質３）』に係る脆弱化は、ハッシュ値が確定する前について、同じハッシュ値を持つ二つ以上の異なる入力メッセージを見つけ出すことを可能とするが、ハッシュ値が確定した後について、その入力メッセージの一方と同じハッシュ値を持つ別の入力メッセージを見つけ出すことは、『第２原像計算困難性（性質２）』により防止される。

・よって、今回の指摘の影響範囲は、ハッシュ値が確定する前に入力メッセージを自由に設定できるようなハッシュ関数の使い方をしている部分となる。

1-2 ハッシュ関数の脆弱性の影響を受ける箇所および対応

本サブシステムにおいて、ハッシュ関数を利用している箇所およびハッシュ関数への脆弱化への対応は、以下の通りである。
（出典・参考文献：「SHA-1 脆弱化（衝突困難性の脆弱化）に関する公開情報について　～ハッシュ関数編～」）

①利用者が、タイムスタンプの要求において対象ドキュメントのハッシュ値を計算する箇所、および検証者がタイムスタンプの検証において対象ドキュメントのハッシュ値を計算する箇所

・利用者が入力メッセージとなる対象ドキュメントを自由に設定できるため、今回の指摘の影響範囲となる。具体的には、同じハッシュ値を持つ２つの異なる入力メッセージに対して同じタイムスタンプトークンが付与される、という事態の発生する恐れがある。すなわち、一方のメッセージに対して タイムスタンプトークンを取得したとしても、そのトークンが他方のメッセージに対するものだと主張することが可能となる。

・このため、利用者がタイムスタンプの要求において対象ドキュメントのハッシュ値を計算する箇所およびタイムスタンプ要求に含まれるメッセージダイジェストのハッシュ関数として、より安全性の高い、SHA-512 に対応する。
（タイムスタンプソフトウェア、評価用クライアントソフトウェア）

・また、タイムスタンプ要求に含まれるメッセージダイジェストのハッシュ関数が SHA-1 の場合は、そのタイムスタンプ要求を拒否する機能を実装する。
（タイムスタンプソフトウェア）

②TSA がタイムスタンプの付与においてデジタル署名を生成する箇所、および検証者がタイムスタンプの検証においてデジタル署名の検証を行う箇所

・利用者が自由には設定できない内容であるため、今回の指摘による影響はない。

・また、過去に発行されたタイムスタンプトークンの信頼性に関しては、タイムスタンプトークンに含まれるハッシュ値と同じ値になるような別のタイムスタンプ対象データを作り出す事は、『第２原像計算困難性（性質２）』により防止されるため、今回の指摘

による影響はない。

・上記理由から、特に対応は行わない。

③CA が証明書や失効リストの発行においてデジタル署名を生成する箇所、および検証者がタイムスタンプの検証においてデジタル署名の検証を行う箇所

・利用者が自由には設定できない内容であるため、今回の指摘による影響はない。

・また、過去に発行されたタイムスタンプトークンの信頼性に関しては、タイムスタンプトークンに含まれるハッシュ値と同じ値になるような別のタイムスタンプ対象データを作り出す事は、『第２原像計算困難性（性質２）』により防止されるため、今回の指摘による影響はない。

・上記理由から、特に対応は行わない。

④TA と TSA との時刻配信プロトコルにおいて認証・暗号化・改ざん防止(HMAC)を行う箇所

・利用者が自由には設定できない内容であるため、今回の指摘による影響はない。

・また、下記の理由においても、今回の指摘による影響はないと考えられる。
-1 回の通信時間が非常に短く、次の通信では古いセッション鍵は破棄されて新しいセッション鍵が使われることから、不正を行われる機会自体がほとんどない。
-統合化プラットフォームシステムにおいては、NTA1-TA1-TSA1,TSA2 の間を、専用線(ISDN 認証による接続もしくは LAN による接続)にて接続することで、通信の信頼性を確保している。（ハッシュ関数の脆弱性による影響のみに依存しない。）

・上記理由から、特に対応は行わない。

2. タイムスタンプの処理能力

本サブシステムでは、統合化プラットフォームシステム上において、独立トークン方式のタイムスタンプについて、タイムスタンプ要求に含まれるメッセージダイジェストのハッシュ関数を安全性の高い SHA-512 へ変更した上で、ネットワーク環境や機器性能等を示し、処理性能を明らかにした。以下に、各測定環境におけるタイムスタンプの処理能力の測定結果を示す。

2-1 測定結果 1(インターネット経由でのタイムスタンプ)

表 4-1　測定結果 1(インターネット経由でのタイムスタンプ)

鍵長	測定回数	計測結果
1024bit	100 回 (スレッド数:20 ループ数:5)	1 スタンプ/秒
2048bit	100 回 (スレッド数:20 ループ数:5)	1 スタンプ/秒

表 4-1より、インターネット経由でのタイムスタンプは、鍵長 1024bit、2048bit の場合とも 1 スタンプ/秒である。これは、後述の LAN 経由でのタイムスタンプの処理能力を考慮すると、TSA2 の回線速度(フレッツ ISDN)や評価用クライアントの回線速度(AirH”)がボトルネックになっていると考えられる。従って、TSA2 の回線および評価用クライアントの回線を光回線にするなどの対応により、タイムスタンプの処理能力を改善することができる。

【参考】
タイムスタンプ要求のサイズ：約 0.1KB
タイムスタンプ応答のサイズ：約 3.4KB(鍵長 1024bit)
約 3.6KB(鍵長 2048bit)

2-2 測定結果 2(LAN 経由でのタイムスタンプ)

表 4-2　測定結果 2(LAN 経由でのタイムスタンプ)

鍵長	測定回数	計測結果
1024bit	1500 回 (スレッド数:30 ループ数:50)	130 スタンプ/秒
2048bit	1500 回 (スレッド数:30 ループ数:50)	26 スタンプ/秒

表 4-2より、LAN 経由でのタイムスタンプは、鍵長 1024bit の場合 130 スタンプ/秒、2048bit の場合 26 スタンプ/秒である。

このスタンプ数では通信回線速度がボトルネックとなることはないため、HSM の署名

速度がボトルネックになっていると考えられる。すなわち、HSM をより高速なものに交換することで、タイムスタンプ速度を改善することができる。具体的には、表 4-3に示すように、署名鍵 1024 ビットのとき毎秒 578 署名以上、2048 ビットのとき毎秒 116 署名以上の HSM を利用することで、最終目標を達成することが可能である。

表 4-3　タイムスタンプ処理能力の測定結果(平成 15 年度～平成 17 年度)

鍵長	平成 15 年度 目標値	平成 15 年度 結果 (1000 回平均)	平成 16 年度 結果 (1500 回平均)	平成 17 年度 結果 (1500 回平均) ※LAN 経由でのスタンプ数	最終目標
1024 ビット	毎秒 50 スタンプ	毎秒 56 スタンプ HSM 性能: 毎秒 150 署名	毎秒 113 スタンプ HSM 性能: 毎秒 150 署名	毎秒 130 スタンプ HSM 性能: 毎秒 150 署名 HSM 性能/スタンプ数の比: 約 86.6%	毎秒 500 スタンプ 必要な HSM 性能(推定): 毎秒 578 署名 HSM 性能/スタンプ数の比(仮定): 約 86.6%
2048 ビット	毎秒 10 スタンプ	毎秒 19 スタンプ HSM 性能(推定): 毎秒 30 署名	毎秒 26 スタンプ HSM 性能(推定): 毎秒 30 署名	毎秒 26 スタンプ HSM 性能(推定): 毎秒 30 署名 HSM 性能/スタンプ数の比(仮定): 約 86.6%	毎秒 100 スタンプ 必要な HSM 性能(推定): 毎秒 116 署名 HSM 性能/スタンプ数の比(仮定): 約 86.6%

※1024 ビット時の HSM 性能は、カタログ値。
※2048 ビット時の HSM 性能は、平成 17 年度の HSM 性能/スタンプ数の比を、1024 ビット時と同じ約 86.6%と仮定した場合の推定値。
※平成 17 年度の測定結果が平成 16 年度の測定結果より高いのは、タイムスタンプ要求を行うクライアント側の性能が高いため。

3. トレーサビリティ機能との連携

本サブシステムでは、平成 16 年度に開発された、時刻配信プロトコル（認証連鎖方式）によって配信される時刻情報を受信する機能を、統合化プラットフォームシステム上に適用した。本サブシステムの評価により、統合化プラットフォームシステム上において、トレーサビリティ機能との連携に必要なインタフェースを備えることを確認できた。