社団法人 情報処理学会 研究報告
IPSJ SIG Technical Report

2004－DPS－117 (3)
2004－CSEC－ 24 (3)
2004／3／4

時刻認証子による配信時刻証明方式

久保寺 範和* 島 成佳* 石崎 健太郎† 小松 文子*

E-mail: n-kubotera@bp.jp.nec.com, shima@ap.jp.nec.com, ishizaki-kxb@necst.nec.co.jp, a-komatsu@ay.jp.nec.com

* 日本電気株式会社
システム基盤ソフトウェア開発本部
〒108-8557 東京都港区芝浦 2-11-5

† NEC システムテクノロジー(株)
〒731-3168 広島市安佐南区伴南 1-40-1

概要： タイムスタンプはタイムスタンプ局(TSA：Time-Stamping Authority)から発行されており、データの存在時刻を証明する。存在時刻は TSA の内部時計を使用する。TSA は内部時計の正確性を保持するため、標準時配信局(TA：Time Authority)と時刻同期をとっているが、時刻同期によって保証された時刻を証明することは困難である。時刻の配信元を特定する方法は、前回「時刻認証子による時刻証明方式」[3]で課題となっており、本稿では、時刻の配信記録を残し、後に時刻の配信元を確認可能となる方式を提案する。

A Method of Time Distribution using Time Evidence

Norikazu KUBOTERA*, Shigeyoshi SHIMA*, Kentaro Ishizaki †, Ayako KOMATSU*

Email: n-kubotera@bp.jp.nec.com, shima@ap.jp.nec.com, ishizaki-kxb@necst.nec.co.jp, a-komatsu@ay.jp.nec.com

* NEC Corporation,
System Platform Software Development Division
2-11-5 Shibaura, Minato-ku, Tokyo, 108-8557, Japan

† NEC System Technologies, Ltd.
1-40-1 Tomominami, Asamimami-ku, Hiroshima-si,
Hiroshima, 731-3168, Japan

Abstract： A Time-Stamp is issued by Time-Stamping Authority (TSA) and is evidence of Time-of-existence of digital documents. Time-of-existence is used by a local clock of TSA. In order to keep accuracy of TSA's local clock, TSA synchronizes time of Time Authority (TA). But it is difficult to verify time guaranteed by synchronized time. A method of identifying TA is a task in "A Method of Time Certification using Time Evidence"[3]. In this paper, we propose method of identifying TA.

1. はじめに

近年、文書の電子化を推進するための電子署名やタイムスタンプを使用したシステムの市場が拡大している。またタイムスタンプサービス行う企業も出てきている。電子文書は公開鍵基盤(PKI)を利用した電子署名が施されている場合、発行者と内容を保証することができる。電子署名は、認証局が発行した証明書によって認証される。しかし電子署名は、証明書の有効期限が切れた後、署名の正当性を証明できない。よって署名者は電子署名を否認で

きる。さらに電子署名を施した日時の改ざんを署名者が行うこと可能であるため、過去のある時点での電子文書の存在保証はできない。署名者による不正行為を防止するためには、ある時点における電子文書の存在証明を保証し、信頼できる第三者機関としてタイムスタンプ局(TSA : Time-Stamping Authority)を使用する。TSA は、電子文書のハッシュ値に対しタイムスタンプ[1]を付与することにより、タイムスタンプの時刻に電子文書の存在を保証する。TSA は、発行するタイムスタンプの時刻に TSA の内部時計を使用している。TSA の内部時計は、信頼できる時刻を保つために、標準時配信局(TA : Time Authority)から NTP(Network Time Protocol)[2]などを利用して時刻監査を受けている。TA は正しい時刻を管理しており、上位の国家時刻標準機関(NTA : National Time Authority)と同期を取っている。特定の TA、TSA と時刻同期を行ったことを検証するためには、時刻同期をとった時刻の証明が不可欠であり、その時刻を付加するのは困難である。

2. 時刻配信方法の問題点

NTP による時刻同期では、時刻配信元を特定し、それを第三者が確認することは難しい。そのため TSA が不正な TA と時刻同期をとり、不正な時刻のタイムスタンプを発行している場合、確かめることは困難である。

前回「時刻認証子による時刻証明方式」[3]において、TA および TSA で時刻認証子を生成することにより、TSA の時刻改ざん検出や、第三者による監査結果の再検証を可能とする方式を提案したが、時刻のトレーサビリティを証明することはできなかった。TSA が特定の TA と時刻同期を行っていたことを証明できず、同様に、TA が特定の NTA と時刻同期を行っているのかは確認することができなかった。

また、実際に時刻認証子を使用したシステム構築では、時刻認証子のパラメータの再検討が必要であった。

3. 配信時刻証明方式

3.1. 方式の概要と特徴

本方式では、時刻の配信元を特定するためのトレーサビリティの情報を、TA、TSA で共に生成される時刻認証子に含める。さらに実際に時刻認証子を使用した時刻配信するシステムの構築の際、必要と思われるパラメータ(時刻の表記方法、ハッシュ関数の種類など)を追加した。

時刻認証子の生成モデルは[3]と同様であるが、図 1 に示す。図 1 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と流れている。TA、TSA は、時刻の流れを示す矢印上に横線が入っている部分で時刻認証子を生成する。

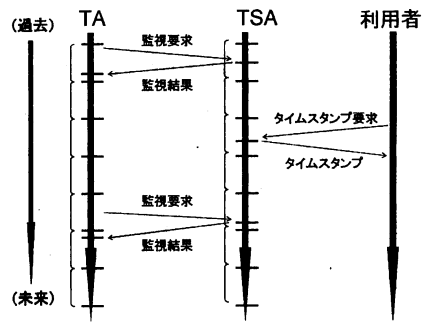

図 1 : 時刻認証子生成の図

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 それぞれで直前に生成された時刻認証子を使用して生成されているため、時刻認証子の改ざんと不正な時刻認証子生成を防止している。

本方式では、時刻配信元を確認可能となる領域としてトレーサビリティ領域を新たに付加した。トレーサビリティ領域には、時刻認証子を受信するもの(機関または利用者)に対して、受信した時刻認証子の配信ルートである NTA、TA、TSA を確認可能となる情報を記録している。
また本稿では、利用者が、タイムスタンプ要求時に生成された時刻認証子の検証を行う方法を示している。

3.2. 時刻認証子の生成方式

時刻認証子は次のような 3 つのデータ領域から構成されている。

TE ：HF ‖ CF ‖ HVF

TE：時刻認証子
HF：ヘッダー領域
CF：ハッシュ対象領域
HVF：ハッシュ値領域

(1) HF：ヘッダー領域
ヘッダー領域には、時刻認証子の基本的な情報を記録している。

data_type：時刻認証子の種類を表す識別子
trace_type：時刻認証子のトレーサビリティ方式を表す識別子
hash_type：ハッシュ領域(HVF)で使用するハッシュ関数の種類を表す識別子
id_type：時刻認証子を生成した機器を示す番号の種類を表す識別子
id_len：時刻認証子を生成した機器を示す番号の長さ
id：時刻認証子を生成した機器の番号

trace_type には、トレーサビリティ領域の有無や、確認可能なトレーサビリティの段数を記録している。また機器を特定する記述には IP アドレス、FQDN などあり、これら複数の記述方式に対応するため、記述方法を識別する id_type を付加している。

(2) CF：ハッシュ対象領域
ハッシュ対象領域は、データ領域(DF)とトレーサビリティ領域(TR)から構成される。CF の値は、ハッシュ値領域 HVF を作成するのに用いられる。

(2.1) DF：データ領域
データ領域は、時刻認証子の生成タイミングによって次の(ア)～(エ)の 4 種類がある。

(ア) 一定時間ごとの場合(TA, TSA 共に使用)
ヘッダー領域の data_type には 0000(すべてのビットが 0)の値がはいり、一定時間ごとに生成された時刻認証子であることを示す。

time_id：時刻表記の種類を示す識別子
time_len：生成した時刻の長さ
time：生成した時刻
random_len：ランダム値の長さ
random：ランダム値

時刻の表記方法は、ISO 標準表記や UNIX タイムなどといったものがあり、時刻の表記方法を示しているのが time_id である。ランダム値は時刻認証子を第三者に予測されないための値である。

(イ) TA が監視結果を受けた場合
ヘッダー領域の data_type には 0001(1 ビット目のみ 1)の値がはいり、TA が監視結果受信時に生成された時刻認証子であることを示す。

TE_REQ_LEN:TSA にて監視要求時に生成された時刻認証子の長さ

TA_REQ : TSA にて監視要求時に生成された時刻認証子

(ウ) TSA が監視要求を受けた場合

ヘッダー領域の data_type には 0010(2 ビット目のみ 1)の値がはいり、監視要求時に生成された時刻認証子であることを示す。

NTE_{TA} : 時刻監視回数を示すカウンタ。時刻監視を行ったときの数値を示している。TA にて監視ログとして、管理されている。

id_type : 機器の識別子

id_len : 時刻監視を行う機器(TA)を示す番号の長さ

id : 時刻監視を行う機器(TA)の番号

NTE_{TA} に何回目かの時刻監視であるかを示す値を記録する。このカウンター値は、時刻監視を行う TA にて管理されている。id に監視要求を出した機器(TA)の番号が記録する。

(エ) TSA がタイムスタンプ要求を受けた場合

ヘッダー領域の data_type には 0011(下位 2 ビットが 1)の値がはいり、TSA がタイムスタンプ要求を受けた時に生成された時刻認証子であることを示す。

hash_type : タイムスタンプ対象データのハッシュ値の種類

hash_value : タイムスタンプ対象データのハッシュ値

タイムスタンプを要求された場合、タイムスタンプ対象データのハッシュ値とそのハッシュ関数を示す値を記録する。

(2.2) TR : トレーサビリティ領域

TRN : トレーサビリティの段数

id_type : 時刻認証子を生成した機器を示す番号の種類を表す識別子

id_len : 時刻認証子を生成した機器を示す番号の長さ

id : 時刻認証子を生成した機器を示す番号

トレーサビリティ領域では、時刻のトレーサビリティの情報を記録する。

NTA を 0 段目の機器とする(TRN の値が 0)。NTA から時刻の配信をされた機器を 1 段目とし、1 段目の機器から時刻の配信をされた機器を 2 段目とする。以下、時刻の配信を受けた機器ごとに 3 段目、4 段目、・・・、n 段目とする。

時刻の配信を受けた機器は、トレーサビリティの段数と上位の配信機器を示す情報を記録し、時刻認証子に付加していく。

(3) HVF : ハッシュ値領域

この領域には、直前に生成された時刻認証子とハッシュ対象領域の値(もしくはハッシュ対象領域に関連するデータ)のハッシュ値が記録される。直前に生成された時刻認証子も含めてハッシュ値を計算することにより、時刻認証子の改ざん・不正生成を防止する。

- 監視要求時に生成される時刻認証子の場合

  HVF: Hash(lastTE | TE_{TA} | TR…TR)

  Hash( ) : ハッシュ関数

  lastTE : 前回生成された時刻認証子

  TE_{TA} : TA で生成された時刻認証子

- 上記以外の時刻認証子の場合
  HVF: Hash(lastTE | CF)

3.3. TA の監視ログ生成方法

TA は監査後に TSA の監査結果を元に監査ログを作成する．監査ログは第三者が TSA の時刻監査の結果を再検証するために利用する．

監査ログは次の３つのデータ領域から構成されている．

$$NTE_{TA} \| TSA_{num} \| TE_{TA}$$

NTE_{TA} ：時刻監査の回数。TSA 時刻認証子の検証時に時刻監査時の TA 時刻認証子を検索するための検索キーとして利用。

TSA_{num} : TSA を識別するためのデータ。TA の監査時に生成する認証子のデータ識別領域の TSA を示す値．この領域には以下の 3 つのデータが記録される。

TSA_{num}: id_type || id_len || id

id_type : TSA 機器を示す番号の種類を表す識別子

id_len : TSA 機器を示す番号

id : TSA 機器を示す番号

TE_{TA} ：時刻監査で使用した TA の時刻認証子

4. 時刻認証子の検証方法

前回[3]では、TSA の生成したすべての時刻認証子を検証することにより、TSA に対する時刻監視の再検証の方法を示した。本稿では、利用者が受け取った時刻認証子の検証方法を示す。

タイムスタンプ時に生成された時刻認証子の検証方法は、TSA の時刻認証子と TA の監査ログを使用して行う。例として、タイムスタンプ要求した時刻 i+j に生成された時刻認証子 TE_{TSAi+j} の検証方法を示す。図 3 には、時刻 i-1 から i+j+k までの間に TSA で生成された時刻認証子を示し、図 4 には、時刻 i と i+j+k で時刻監視を行った監査ログを示している。なお図 3 では、時刻認証子のヘッダー領域に、data_type を記載している。

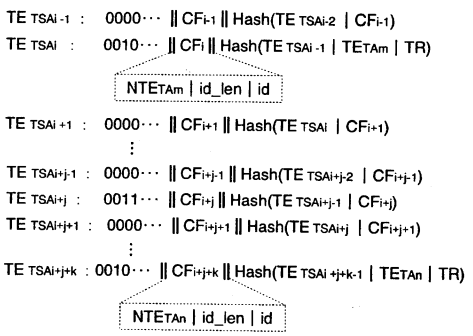

図 3：TSA で生成された時刻認証子

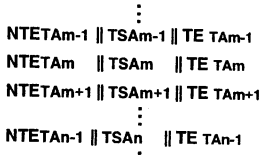

図 4：監視ログ

data_type : 0010 を検索することにより、時刻認証子 TE_{TSAi+j} の前後に生成された「監視要求時生成の時刻認証子」TE_{TSAi} と $TE_{TSAi+j+k}$ を見つけ出す。「監視要求時生成の時刻認証子」は TA からの時刻認証子を使用して生成されているため、第三者から見て、信頼できる時刻認証子となっている。よって、TE_{TSAi+j} の順序性は、時刻 i-1 から i+j+k までに生成された時刻認証子のハッシュ値を確認することにより、検証される。

ハッシュ値と時刻認証子の整合性の確認方法は、

data_type が 0010 でない時刻認証子（例えば $TE_{TSAi+j-1}$)の場合、$TE_{TSAi+j-1}$ と CF_{i+j-1} のハッシュ値が TE_{TSAi+j} のハッシュ値領域の値と同じであることを確かめることである。data_type が 0010 の場合（例えば TE_{TSAi})、CF_i から監視ログの情報をキーとして、図 4 の監視ログから TE_{TAm} を得る。そして、TE_{TSAi-1} 、TE_{TAm} と TR のハッシュ値を計算し、TE_{TSAi} のハッシュ値領域と同じであるか検証する。

時刻 i-1 から i+j+k まで、ハッシュ値を計算・検証し、不整合がなければ時刻認証子 TE_{TSAi+j} の正当性の証明となる。もし不整合があれば、TE_{TSAi+j} が不正なものである可能性があり、タイムスタンプの時刻の正当性は保証できない。

5. 本方式の考察

本方式では、TA と TSA において生成される時刻認証子に時刻のトレーサビリティの情報を記録し、後に時刻の配信元を特定することが可能となった。さらに、利用者が受け取った時刻認証子の正当性の検証を可能とした。また、時刻認証子にシステム構築の時に必要と思われるパラメータを追加し、実際にシステム構築を行った。

本稿で検討した時刻認証子には、以前[3]で提案した方式より多くの情報が含まれている。しかし、構築したシステムで使用している時刻認証子は 100 バイト程度であり、このサイズは[3]で想定していた大きさであった。よって[3]と同様に、30 年間で生成される時刻認証子のデータ量は 150G バイト程度と想定される。また、時刻認証子の可読性を高めるためには、500 バイト程度の大きさが必要である。このとき 30 年間で生成される時刻認証子のデータ量は 750G バイトであり、大容量のストレージを用意することで可能となる。

1 秒ごとに時刻認証子を生成した場合、24 時間で生成された時刻認証子すべての検証に要する時間は、15 秒程度であった(実行環境：ペンティアム 4　2.8GHz, メモリ　768Mbyte)。単純に計算すると、1 年間で生成された時刻認証子の検証に要する時間は 90 分程度となり、本システムは実用可能であると考えられる。

本方式では、TA に時刻の配信を行っている配信元の情報は時刻認証子に記録されているが、その検証方法は確立していない。また時刻認証子の検証は、TA の監視ログの信頼性に依存するが、監視ログの正当性検証、保管場所や取得方法については課題としたい。

6. おわりに

本稿では、TA、TSA 双方がある時刻の証拠となる時刻認証子に、時刻の配信元を特定するトレーサビリティ情報を追加し、取得した時刻認証子の配信元の特定が可能となった。今後、本方式の実用性を高める検討や残る課題の改善をする。

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