経済産業省委託調査

電子署名文書長期保存に関する実用化動向調査報告書

電子商取引推進協議会
財団法人日本情報処理開発協会 電子商取引推進センター

←（背表紙）

（表紙）

経済産業省委託調査

平成１５年度ＥＣ技術基盤の相互運用性に関する調査研究事業
（電子署名生成・検証システムのセキュリティ環境の標準化等調査）

電子署名文書長期保存に関する実用化動向調査報告書

平成１６年３月

電子商取引推進協議会
財団法人日本情報処理開発協会
電子商取引推進センター

（表紙裏）

この報告書は、平成１５年度受託事業として（財）日本情報処理開発協会電子商取引推進センターが経済産業省から委託を受けて、電子商取引推進協議会（ECOM）の協力を得て実施した「平成１５年度ＥＣ技術基盤の相互運用性に関する調査研究事業（電子署名生成・検証システムのセキュリティ環境の標準化等調査）」の成果を取りまとめたものです。

はじめに

ECOM で電子署名文書長期保存に関する検討が始まってはや約 4 年が経過した。平成 12 年度から 13 年度にかけて、電子署名文書の長期保存を可能にする為の技術基盤を洗い出し、その要件を明確にして、電子署名文書長期保存システムのモデル案を作成・提言した。その中で、電子文書の日時を特定するタイムスタンプを活用した電子署名文書の長期保存システムの確立が必要不可欠とのことから、平成 14 年度ではタイムスタンプのサービスに関する調査報告書、タイムスタンプサービスの利用ガイドライン、及び運用ガイドラインを作成・提言した。今年度は、タイムスタンプに関する調査を引き続き行い、PKI を使ったタイムスタンプ方式（シンプルプロトコル方式）のタイムスタンプ局証明書を発行する認証局の運用ガイドラインを作成・提言する。ここでは、タイムスタンプ局専用の認証局は個人用証明書などを発行する一般的な認証局と異なり、タイムスタンプの役割、タイムスタンプ局特有の運用及び鍵管理といったことを前提にして運用ガイドラインを作成した。また、長期署名検証の課題（文書偽造の脅威）を提示し、その対策の提案と認証局の運用ガイドラインも併せて作成した。さらに、ECOM で提言した電子署名文書長期保存モデルシステムをベースにしたシステム及び要素技術が、今年度になり実用化され商品化が始まっているので、代表的な実用化例を併せて紹介したい。

今年度になり政府は、引き続き世界最先端の IT 国家を目指して e-Japan 戦略Ⅱ、e-Japan 重点計画-2003、e-Japan 戦略Ⅱ加速度パッケージを発表した。その中に、電子文書の長期保存の技術開発支援や１０年保管を義務付けている財務関係書類、税務関係書類等の電子的な保存を法律の制定（e-文書法）により認めることなどが書かれている。またタイムスタンプ・プラットフォーム技術の研究開発も進められており、IT 社会基盤の整備が着実に進められている。今後ますます ECOM で取り組んできた電子署名文書長期保存に関する検討成果が注目されるようになるであろう。

本報告書が、皆様のタイムスタンプのより一層の理解に役立つとともに、これまでの ECOM での成果と併せて電子署名文書長期保存に関する技術的課題と課題を解決するための技術要素の理解、及び技術動向、実用化動向の把握に役立てれば幸いである。

平成 16 年３月

財団法人日本情報処理開発協会
電子商取引推進センター
電子商取引推進協議会

目次

図表一覧目次

1．電子署名の有効性を長期にわたり維持するための要件

これまで、電子署名の長期にわたる有効性検証を行う環境について検討を行ってきた。その中において、署名検証に関わる材料に対し時刻保証を伴う仕組みにおいて保存し、必要に応じその環境に基づいて署名検証の有効性を証明することを見出した。本章においては、その問題点、その問題点に基づいた要件、そしてその要件から見出したシステムモデルについて示す。

1.1　電子署名の問題点

契約書等紙文書で受け渡しされていた重要書類は電子化されつつあり、インターネット上において電子文書の受け渡しが広まっている。しかし、そのような電子文書には、一般的に以下のような問題点があると認知されている。

- 本人性の検証が困難
- 完全性の保証が困難

上記のような問題を解決し、電子文書の「本人性」と「完全性」を保証するものとして、「電子署名」が期待されている。

この検証手順の中では、文書作成者の公開鍵は、認証局より公開鍵証明書として発行され、この公開鍵証明書に含まれる情報に基づいて電子署名の検証を行う。これは、電子文書作成者の「本人性」と「完全性」を示すために、作成者が公開鍵暗号方式を利用することにより文書の「完全性」を保証し、また、その公開鍵を認証局に認証してもらうことにより「本人性」を保証する方法である。

このとき、公開鍵証明書の中には、以下の運用が含まれている。

- 公開鍵証明書には、利用している情報自身の変更等に備えて、有効期限が設定されている
- 公開鍵証明書の有効期間中に鍵の問題が生じたときに、公開鍵証明書の有効性をなくす失効手続きが備えられている
- 暗号アルゴリズム自体の危殆化に備えて、失効手続きが備えられている

しかし、上記のような運用があるため、検証したい電子文書の電子署名が生成された当初は有効であっても、公開鍵証明書の有効期限が過ぎたとき、または公開鍵証明書が失効したときには、その検証のもととなる公開鍵証明書が保証されない。そのため、電子署名の有効性も確認できないといったことが問題視されている。

1.2　電子署名の有効性を長期にわたり維持するための要件

前節の問題点に対する解決方法として、過去において電子署名を検証した材料・結果を保存し、過去に有効性を検証した電子署名の再検証を行うことにより、電子署名の有効性を確認可能とするモデルを検討してきた。これは、前節のような問題が発生した後でも、問題が発生する以前に

電子署名の検証を行い、検証に利用したあらゆる材料・結果を時刻を保証しかつ改ざん検知できる手段で残しておき、署名再検証時にそれらの情報を利用可能にしておく方法である。

このときの要件としては、これまでの検討の中で以下が導き出された。

表 1-1 電子署名の有効性を長期にわたり維持するための要件

要件１	署名検証時に、署名再検証に必要な情報を明確にしておくこと
要件２	署名検証時の時刻を明確にしておくこと
要件３	署名再検証に必要な情報を改ざん検出可能な状態にすること
要件４	署名再検証に必要な情報を保存すること

これを電子署名文書長期保存システムの要件として記述しなおすと、次の 4 項目として示すことができる。

表 1-2 電子署名の有効性を長期にわたり維持するためのシステム要件

システム要件１	署名再検証に必要な情報を収集すること ※電子署名長期保存システムとしての機能するためには、署名検証時に、署名再検証に必要な情報収集しておくことが必要である。
システム要件２	署名再検証に必要な情報を収集した時刻を確認可能にすること ※システム要件２に伴い、それらの情報を収集した日時を再検証時に確認可能とする。
システム要件３	署名再検証に必要な情報を改ざん検出可能な状態にすること
システム要件４	署名再検証に必要な情報を保存すること

本ガイドラインでは、この 4 つのシステム要件を満たすことのできるシステムを解説していく。

1.3 実装モデル

電子署名の有効性を長期にわたり維持するために、前節で導き出された４つのシステム要件を受けて、実装すべきモデルを次から示す構成要素で実現できるモデルであることを定めた。その中においては、上記のシステム要件以外にも、ポリシーに基づいた処理が必要であることも判明している。

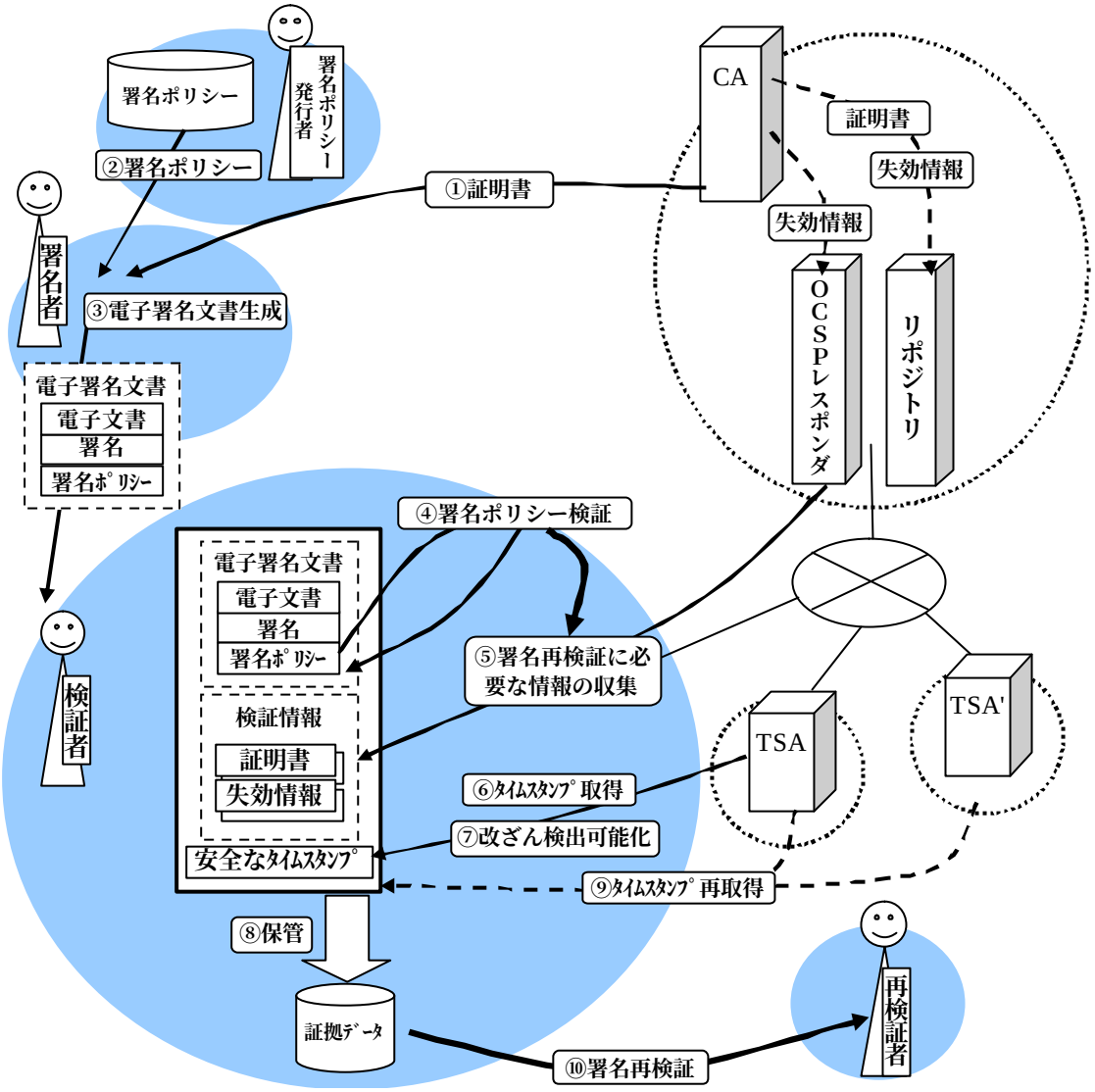

図 1-1 モデル構成

(1) 署名者：電子署名文書を生成する。
(2) 検証者：電子署名文書の有効性を検証する。
(3) 再検証者：電子署名文書の有効性を再検証する。
(4) 署名ポリシー発行者：署名ポリシーを発行する。
(5) CA/OCSP レスポンダ／リポジトリ：証明書の生成／配布、証明書失効情報の生成／配布を行う。
(6) TSA/TSA’：安全なタイムスタンプを発行する。

また、図 1-1 で示したモデルにおいて、証明書発行から署名検証に至るまでの実際の運用フローにおいて、各処理事例を以下に示す。

表 1-3 基本モデルシステムの処理内容

項目	内容	要件
①証明書の発行	CA/OCSP レスポンダ／リポジトリが署名者に対して証明書を発行する。必要に応じて検証者に対する証明書及び証明書失効情報の配布が可能となるように、証明書及び証明書失効情報の管理を行う。	（準備）
②署名ポリシーの発行	署名ポリシー発行者が署名者に対して署名ポリシーを発行する。ただし、署名ポリシーはドメイン内で別途（オフラインで）決められている場合あり。	（準備）
③電子署名文書作成	署名者が、CA から発行を受けた証明書及び署名ポリシー発行者から発行を受けた署名ポリシーに基づき、電子文書に対する電子署名を生成する。更に電子署名を電子文書及び署名ポリシーと結合し、電子署名文書として検証者に転送する。ただし、署名ポリシーが電子署名内やシステム内に明示的に含まれない場合もある。	（準備）
④署名ポリシー検証	検証者が、署名ポリシーの内容を検証する。署名ポリシーの内容により、署名の有効性を検証する基準を知ることができる。	（準備）
⑤署名再検証に必要な情報の収集	検証者が、署名ポリシーに基づき、署名の有効性を検証するために必要な情報を収集する。情報収集先は、CA/OCSP レスポンダ／リポジトリであり、収集対象は、信頼する CA（信頼点）の証明書とそこに至るパス上の証明書、及びそれらに関する失効情報（信頼点までのパス上の証明書が失効していないことを保証する情報：CRL、OCSP レスポンスなど）である。これらの情報が署名再検証時に至るまで有効であることが保証できれば、署名再検証にも利用可能である。	システム要件 1
⑥タイムスタンプ取得	電子署名文書及び前ステップで収集した情報に対して検証者が TSA よりタイムスタンプを取得する。	システム要件 2
⑦改竄検出可能化	電子署名文書、前々ステップで収集した情報、及びタイムスタンプを検証者は改竄検出可能な状態とする。タイムスタンプとして安全なタイムスタンプを利用する場合、（その有効期限までの間は、）安全なタイムスタンプ自体が電子署名文書、前々ステップで収集した情報、及びタイムスタンプの改竄を検出するための手がかりとなる。	システム要件 3
⑧保管	検証者は，前ステップで改竄検出可能とした全データを保管する。このデータは、電子署名文書が有効であった事実を証明するための証拠データとなる。	システム要件 4
⑨タイムスタンプ再取得	安全なタイムスタンプに有効期間がある場合、有効期限到来以前に、検証者は、全データ（電子署名文書、前々項に示した収集情報、及び安全なタイムスタンプ）に対して更に安全なタイムスタンプを取得する。この場合、新たに取得したタイムスタンプまでを含めたデータ全体が証拠データとなる。	システム要件 3

⑩署名再検証	再検証者は、証拠データを受取り、決められた手順にしたがって、署名の有効性を検証する。検証の手順は実装するモデルにより異なるが、少なくとも、証拠情報の非改竄性の情報、タイムスタンプの検証、署名再検証情報の検証、署名の検証を実施する。検証処理そのものをオーソリティに委託する必要がある場合もある。	（検証）

電子署名の種類、署名再検証情報の種類、署名再検証情報の収集主体、タイムスタンプの方式、署名再検証情報の非改ざん保証の方法、署名再検証情報の非改ざん保証の主体、電子署名文書の保存主体、署名再検証情報の保存主体などの相違により、電子署名文書長期保存システムのモデルにはいくつかのバラエティが考えられる。次節より、この相違点の中の技術要素について解説する。

2．要素技術

2.1　技術要件

前章でまとめたデジタル署名の有効性を長期的に維持するための要件をさらに技術要件としてまとめる。

1．署名検証時に、署名再検証に必要な情報を明確にしておくこと

後日における署名再検証によって、過去の特定日時におけるデジタル署名文書の有効性を確認可能とするためには、その確認に必要な情報をあらかじめ署名者と検証者の間の合意によって明確化しておき、署名検証時及び署名再検証時において、その情報をもとにデジタル署名文書の有効性を確認できることが必要となる。デジタル署名文書の有効性確認に必要な情報とは、まさに、署名検証時点において「有効な署名として成立」した事実を示す情報で、「デジタル署名文書に付与された署名の本人性および、署名者と検証者が合意した署名規則（署名ポリシー）のもとに成された署名であることが確認されること」が必要である。署名の本人性を確認するためには、署名者の公開鍵証明書が信頼された発行者(CA)により発行されたものであることを示すCAの公開鍵証明書、さらに、これらの公開鍵証明書が検証時点で無効化されていないことを示す情報等が挙げられる。署名者と検証者による署名規則（署名ポリシー）の合意を確認するためには、署名検証にまつわる技術的、運用的な合意事項を記述した情報や、その情報に署名者および検証者が合意したことの証となる情報等があげられる。これらの情報を検証者が収集し、証拠情報としてデジタル署名文書とともに保持するための技術が必要となる。

2．署名検証時の時刻を明確にしておくこと

デジタル署名の有効性は、その状態が時間の経過とともに変化しうるものであるため、デジタル署名文書に付与された署名が有効な署名として成立した時刻（署名検証時刻）が、信頼された時刻源から提供された時刻情報を用いて確定された事実を示す情報を、証拠情報として残すことが必要となる。有効な署名として成立した事実は、1 の要件を満たすために収集した証拠情報によって示すことができるため、その事実の成立時刻を示すためには、この証拠情報と信頼された時刻情報とを結びつけるための技術が必要となる。

3．署名再検証に必要な情報を改ざん検出可能な状態にすること

「署名再検証に必要な情報」とは、上記１、２の要件を満たすために収集した証拠情報となるため、署名検証後、長時間経過した後でも、その証拠情報が署名検証時と変わらず改ざんされていないことを、署名再検証時に確認できる手段あるいは情報が提供されていることが必要となる。１、２で収集した証拠情報に含まれる情報には、例えばデジタル署名文書や公開鍵証明書など、それぞれにその完全性を確認できるデジタル署名が付与されているが、これらのデジタル署名の有効性が確認できる期間は、その署名者の公開鍵証明書の有効期間内に限られる（公開鍵証明書の無効化が発生した場合にはさらに短い期間に限

定される）ため、収集した証拠情報をそのまま保管した場合には、それらの情報に対して署名者の意図しない改ざんの発生の有無を確認できる期間が、その公開鍵証明書の有効期間内に限定されてしまうことになる。したがって、１、２で収集した証拠情報内に含まれる署名の有効期間に依存することなく、長期にわたって収集した証拠情報すべての非改ざん性を確認可能とする技術が必要となる。

以上より、デジタル署名の有効性を長期的に維持するための証拠情報を生成するためには、その生成過程において以下の要件を満たす技術を適用することが必要であると言える。

①署名規則の合意形成及び確認可能な技術
②署名規則に基づいて確実に署名検証を行うための技術
③署名検証に用いた情報を証拠情報としてデジタル署名文書とともに保持するための技術
④証拠情報と信頼された時刻情報とを結びつけるための技術
⑤署名検証時に収集した証拠情報の長期にわたる非改ざん性を確認可能とする技術

これらの要件を満たす基本技術として、

1)タイムスタンプ技術（④、⑤）
2)署名検証技術（②）
3)署名ポリシー合意形成技術（①）
4)署名フォーマット形成技術（③）

が存在しており、次節以降ではこれらの基本技術についてそれぞれ説明する。

2.2 タイムスタンプ技術

2.2.1 タイムスタンプの要件

容易に変更可能なデジタルデータの使用には、それらの情報がいつ作成されたか、または最後にいつ変更されたかをいかに証明するかという課題が存在する。デジタルタイムスタンプは、このような時間の証拠保全に役立てなければならない。したがって、デジタルタイムスタンプは、以下の要件を満たさなければならない。

- 時刻変数は、ある情報が特定の時刻より以前に存在したことを保証する証拠を提供するために、暴露されない方法で、その情報と結合していなければならない。（存在証明）
- 情報が暴露されない方法で提供されている可能性（非改ざん証明）

情報の正当性と機密性の制御のために、情報のハッシュ値に対してタイムスタンプすることによって、これらの要件を解決している。情報自体が、タイムスタンプから取り出されることはない。情報のハッシュ値はある TSA によるカレント時間と暗号手段で結合している。この結合が、時間の正当性と認証を証明する。タイムスタンプトークンは、これらの要素を提供し、タイムスタンプの要求者へ返答する。タイムスタンプトークンは、以前に作成したトークンに関連した情報を含むこともある。ここで、情報とタイムスタンプ要求以前のタイムスタンプされた情報に含

まれる追加情報は、タイムスタンプ処理の入力情報となる。

2.2.2 タイムスタンプサービスのモデル

タイムスタンプサービスは、図 2-1 に示すモデルにおいて TSA（Time Stamping Authority）により提供される。タイムスタンプサービスモデルは、タイムスタンププロトコルによりモデルが異なる。例えば、IETF で 2001 年に標準化が完了した RFC3161 に準拠した独立トークン方式は、時刻証明書を TSA の公開鍵証明書を用いて検証する方式で、公開鍵証明書を発行する認証局は必須である（2.2.5 独立トークン方式を参照）。一方、リンクトークン方式は、時刻証明書の偽造を行うことができないように過去に発行した時刻証明書すべてとのリンクを作成し、このリンク情報の偽造を困難にするため定期的に新聞等でリンク情報を衆目にさらす運用を行っている。リンク情報公開先は必須である（2.2.6 リンクトークン方式を参照）。

・国家時刻標準機関－NTA（National Time Authority）

国家標準時を生成・維持・配信する国家機関を示す。

・標準時刻配信事業者－TA（Time Authority）

NTA から時刻配信を受け、TSA へ標準時を配信する。定期的に TSA の時刻監査も行う。信頼できる第三者機関である。

・時刻認証事業者－TSA（Time Stamping Authority）

TA から時刻配信を受け、利用者からの要求に応じて時刻証拠となる時刻証明書を作成し、発行する。

・認証局－CA（Certification Authority）

NTA、TA、TSA に対して認証、あるいは署名用の公開鍵証明書を発行する。信頼できる第三者機関である。デジタル署名を用いたタイムスタンプ方式には必須である。

・リンク情報公開先

主に新聞会社である。定期的に時刻証明書のリンク情報を新聞等で衆目にさらすことにより、リンク情報の偽造をさらに困難にする。リンクトークン方式を用いたタイムスタンプには必須である。

・時刻利用者

電子データの存在と非改ざんを証明してもらうため、TSA に対して時刻証明書を要求する。また、取得した時刻証明書を検証するための検証主体に対して検証を要求する。

・時刻証明書

電子データの存在と非改ざんを証明するための電子データ

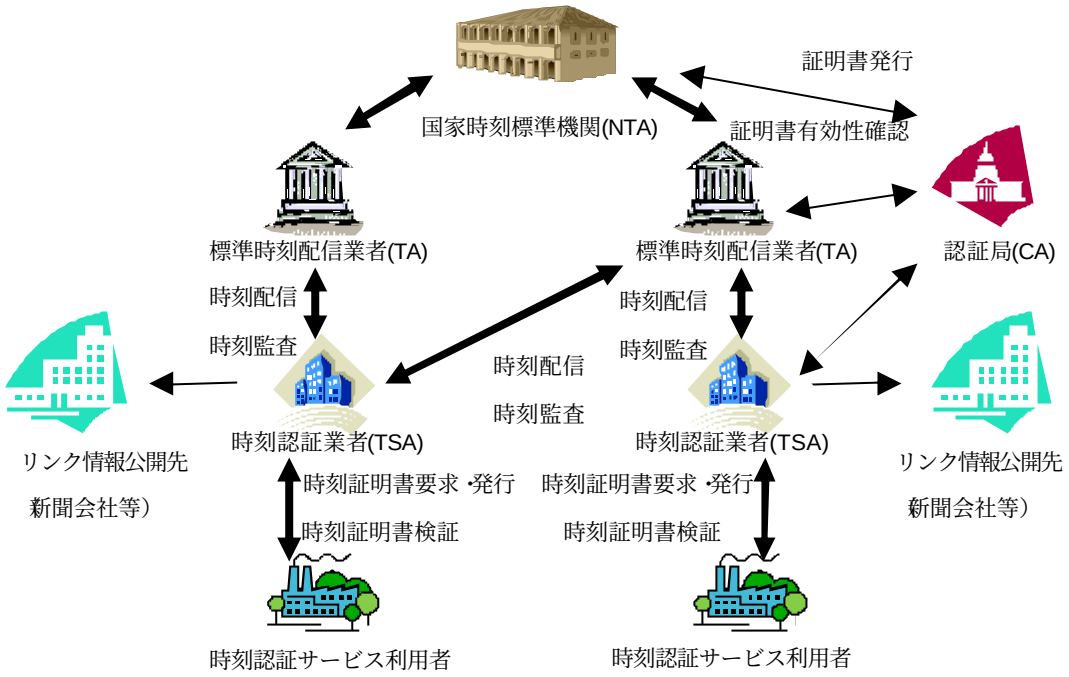

図 2-1 タイムスタンプサービスのモデル

2.2.3 タイムスタンプ方式の特徴

タイムスタンプには、「独立トークン方式」「リンクトークン方式」などの種類がある。ここでは製品化やサービス化が行われている「独立トークン方式」と「リンクトークン方式」についてサービスの特徴を比較する。

表 2-1 は、タイムスタンプの利用者がタイムスタンプサービス選択時に必要とする情報を仮定した比較項目である。

表 2-1 タイムスタンプ方式の比較

	独立トークン方式を利用したサービス（PKI ベース）	リンクトークン方式を利用したサービス
有効期間	・ タイムスタンプ証明書の有効期間と同じ期間 ※例：RSA1024 の場合、５年程度	・ ハッシュ関数の強度に依存 ※タイムスタンプの有効期間を長くするために異なるハッシュ関数を用いる場合がある
タイムスタンプの検証	・ タイムスタンプのデジタル署名の検証に必要な証明書と CRL を取得できれば、TSA 以外の第三者による検証が可能である ・ タイムスタンプのデジタル署名の検証に必要な証明書と CRL がそろっていれば、オフラインによるローカル検証が可能である	TSA もしくは検証機関に検証を要求する必要がある

タイムスタンプに含まれる情報	・ 文書のハッシュ値 ・ 時刻 ・ TSA ポリシー ・ 順序性（ordering） ・ 時刻の精度（accuracy） など	・ 文書のハッシュ値、 ・ 時刻 ・ タイムスタンプの正当性を証明するための必要情報（リンク情報など）
信頼の拠所	・ ハッシュ関数・公開鍵暗号に安全性を依存 ・ 認証機関を信頼の拠所とする	・ ハッシュ関数に安全性を依存 ・ リンク情報の公開・検証により信頼性を確保
信頼性低下、および、信頼喪失の可能性	・ TSA の証明書失効や秘密鍵の危殆化 ・ タイムスタンプの署名有効期間を超えて保証する場合、タイムスタンプ更新の失敗 ・ CA の秘密鍵の危殆化 ・ TSA の時刻の改ざん	・ システム内のハッシュ値の改ざん ・ リンク情報の消失 ・ TSA の時刻の改ざん
不正行為の可能性	・ TSA と署名者が結託することで、タイムスタンプの改ざんが可能である（耐タンパモジュールが必須） ※時刻証明を行う TA の属性証明書を利用することにより、TSA の時刻の改ざんを防止可能	・ TSA と利用者が結託することで、TSA に登録したという事実を抹消できる（ただし TSA のリンク情報を検証することで不正の事実を特定できる）
利用形態	・ ローカル検証が可能であるため、大量にタイムスタンプの発行される（大量の検証処理が発生する）環境に適している	・ 比較的長期間保管が必要な電子文書の利用に適している
標準化	・ RFC3161（IETF PKIX WG） ・ ISO/IEC 18014-2：2002	・ ISO/IEC 18014-3：2002
システム構成	・ クライアントと TSA 以外に、認証機関と連携したシステムの構成	クライアントと TSA（TSA には過去の履歴を保管するデータベースが必要）
コスト	・ タイムスタンプの利用 ・ タイムスタンプの更新とその回数	・ タイムスタンプの利用 ・ 検証処理を TSA や検証機関が行うため、タイムスタンプ検証時に費用が発生する可能性がある

本比較表は、各方式の一般的な特徴を比較したものであり、アルゴリズムの改良や機能追加したものを対象としていない。

2.2.4 標準化動向

(1) 国際規格

ISO/IEC 18014-1 Information technology– Security techniques– Time stamping services

・Part 1: Framework

タイムスタンプサービスの要件、スコープ、提供サービス・機能などの枠組みについて規定している。タイムスタンプの証拠となるトークンについて、独立トークンおよびリンクトークンの 2 つの実現方式の概要を記述している。

・Part 2: Mechanisms producing independent tokens

独立トークン方式のメカニズムについて定義する。電子署名を使用したトークン（RFC3161 互換)、MAC を使用したトークン、アーカイブを使用したトークンという３種類のトークンを規定している。

・Part 3: Mechanisms producing linked tokens

リンクトークン方式のメカニズムを定義する。電子署名を使用したトークンと電子署名を使用しないトークンの２種類のトークンを規定する。

(2) インターネット標準（IETF：Internet Engineering Task Force）

・RFC3161 Internet X.509 Public Key Infrastructure Time-Stamp Protocol（TSP）

インターネットで PKI を利用する上での標準群のひとつ。タイムスタンプモデルにおける手順とフォーマットであるプロトコル、タイムスタンプトークンについて規定する。

・IETF Policy Requirements for Time-Stamping Authorities（draft – ietf – pkix – pr – tsa-00.txt）(2002-03)

TSA の運用ポリシーに関する要件について規定する。

(3) 欧州標準

・ETSI TS 101 861v1.2.2 Time Stamping Profile

RFC3161 に基づく仕様書。タイムスタンプクライアントとタイムスタンプサーバが従う要件を定義する。

(4) OASIS Digital Signature Services Technical Committee（DSS TC）

OASIS DSS TC は、デジタル署名を Web サービスで実現するためのインタフェースやプロトコルの開発を行う目的で 2002 年 10 月に設立された団体。

・Tokens and Protocol for the Temporal Integrity Markup Language

TIML（Temporal Integrity Markup Language）と言い、RFC3161 をベースにほぼ 1 対 1 対応で XML に置き換えた XML タイムスタンププロトコル。RFC3161 のオプションなどで冗長性のある部分を省いている。

2.2.5 独立トークン方式

(1) 方式の概要

独立トークンは国内ではシンプルプロトコルとも呼ばれている。この方式を代表するもの

に、PKI を用いたタイムスタンプがありこれは IETF で RFC3161 として標準化されている。時刻情報に TSA の電子署名を付与し、TSA による第三者保証をしたものである。ユーザは、タイムスタンプの対象文書のハッシュ値（メッセージダイジェストと呼ぶ）を含む決められたフォーマットのタイムスタンプ要求を TSA に送付する。TSA は、受信したメッセージダイジェストと受付時刻を含む規定のフォーマット（タイムスタンプトークンの形式）の文書に電子署名を付け、タイムスタンプトークン（TST：Time Stamp Token）を作成して、ユーザに返送する。ユーザは、受け取ったタイムスタンプトークンを保管しておき、将来、元の文書のタイムスタンプトークンが必要になった時点で、タイムスタンプトークンを TSA の公開鍵証明書を用いて検証することにより、時刻証明書が発行された時点において、元の文書が存在していたことを証明することができる。この方式の特徴は、発行されたタイムスタンプトークンおよびその証明書作成に用いられた公開鍵暗号の公開鍵証明書を用いるだけでタイムスタンプトークンの検証が可能であるということである。この方式が有効であるためには、TSA は信頼のおける第三者機関（Trusted Third Party: TTP）でなければならない。

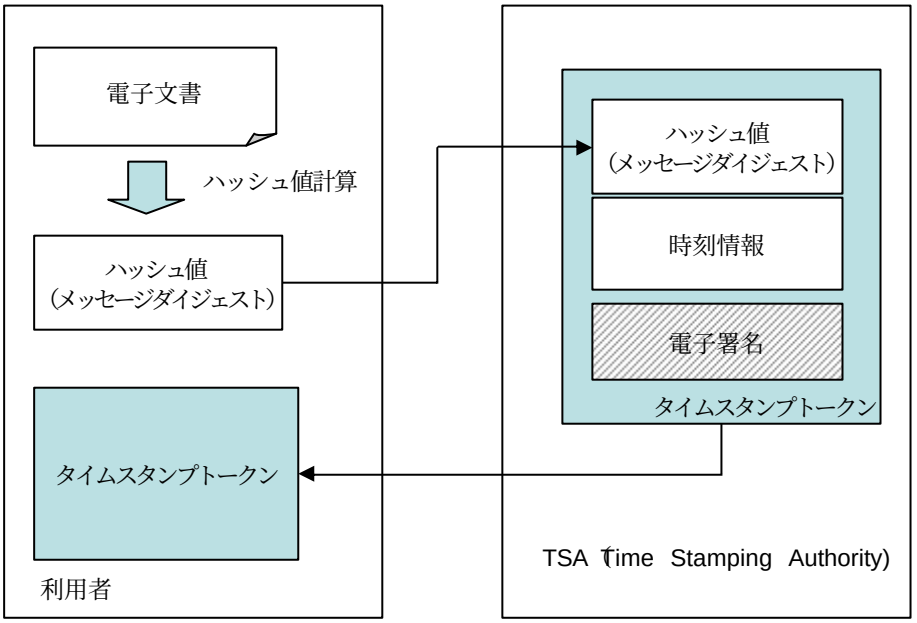

図 2-2 独立トークン方式のタイムスタンプ（電子署名）

タイムスタンプの付いたデータは、TSA の公開鍵証明書の有効期限が切れた場合や署名アルゴリズムが弱体化した場合、タイムスタンプの信頼性が喪失するので、これらの事象が起こる前に、有効期限に余裕のある TSA の公開鍵証明書で再度タイムスタンプを付け直したり、より強度のある署名アルゴリズムでタイムスタンプを付け直す必要がある。

独立トークン方式によるタイムスタンプには、他にメッセージ認証コード（MAC：Message Authentication Code）を用いる方式とアーカイブ方式がある。

MAC 方式は、ISO/IEC 18014 において標準化されているが、図 2-2 で説明した方式とは、電子署名のかわりにメッセージ認証コードを用いるところが異なる。この方式では、TSA が所

有する秘密鍵を用いて、MAC を作成する。すなわち、検証には TSA が秘密にして保持する秘密鍵が必須である。このため、時刻証明書を受け取ったユーザが時刻証明書だけを用いて、証明書の正しさを検証することができない。この方式では、TSA は第三者信頼機関でなければならない。この性質を第三者機関による TSA の監査によって保証する枠組みが必要となる。すなわち、時刻証明書を受け取ったユーザが時刻証明書の検証に一切かかわることがないため、TSA が正しい時刻証明書を発行し続けていることの監査が必要となる。

アーカイブ方式は、TSA にアーカイブされているメッセージダイジェストと時刻情報の対応関係への参照情報を時刻証明書に含める時刻証明方式である。TSA は偽操作を検出できる外部証拠を持たないので、完全に信頼されなければいけない。TSA は適切な運用が行われていることを証明するため、送受信ログ等運用履歴を保管し、定期的に外部の監査を受けることが望ましい。

(2) 時刻証明書の有効期限

ここでは PKI を用いたタイムスタンプの時刻証明書有効期限について述べる。PKI を用いたタイムスタンプの時刻証明書の有効期限は、TSA の公開鍵証明書を発行する認証局の証明書ポリシーに依存する。従って、時刻証明書の有効期限は、認証局の証明書ポリシーで規定した公開鍵証明書の有効期限よりも長くすることはできない。このことを踏まえ、現時点での秘密鍵と有効期間に関する推奨例を提案する。

表 2-2 鍵長に対する有効期間の推奨例

秘密鍵	証明書有効期間	証明書検証可能期間
1024bit　RSA 相当	６年以内	５年以内
2048bit　RSA 相当	１１年以内	１０年以内

詳しくは付録の「TSA 証明書を発行する認証局の運用ガイドライン」を参照して頂きたい。

(3) 長期署名検証に対する課題（文書偽造の脅威）

例えば２０年以上電子署名文書を長期保管する場合、PKI を用いたタイムスタンプを利用すると、時刻証明書の有効期限等から、複数回タイムスタンプを付け直す必要がある。通常、長期署名フォーマットとして広く普及している ETSI ES 201 733 Electronic Signature Formats（RFC3126）に従って、再検証に必要な失効情報、証明書チェーン等の情報にタイムスタンプを付与し、デジタル署名の有効期限が切れた場合でも、署名時点の署名の有効性を再検証できる措置を講じる。ところが、個人又はサーバ認証を目的とした一般認証局の証明書を使ったタイムスタンプでは、証明書の有効期限が過ぎると、タイムスタンプ付与時有効であったことを証明する失効情報等の証拠情報が抹消されるため過去の検証ができなくなってしまう。すなわち、信頼根拠であったタイムスタンプ付与時のタイムスタンプそのものの有効性を検証するための証拠情報がないため、タイムスタンプが有効であったことを確認することができなくなるのである。従って、悪意を持った者は、文書を偽造し、PC の時刻を２０年前に戻して、危殆化した秘密鍵で署名生成し、偽 CA、偽 TSA を使って、時刻を２０年前

に戻したタイムスタンプで付与し直し、信頼根拠を偽造して、最新のタイムスタンプだけ信頼できる CA、TSA のものを使えば、表面的には長期署名検証を正しく行うことができるのである。なお、リンクトークン方式の場合、PKI により時刻証明書の発行者を保証する方式ではないので、上記のような課題はない。

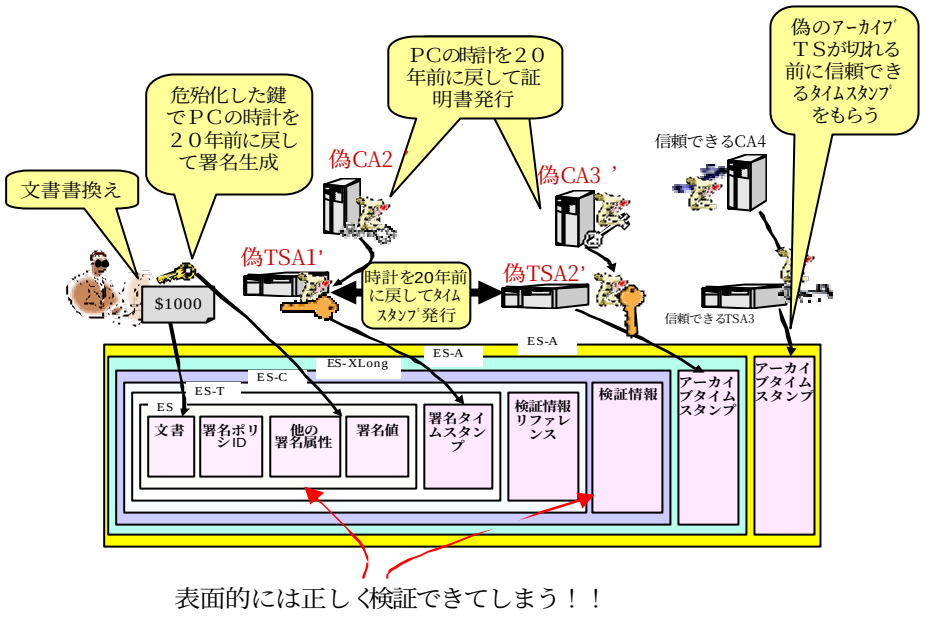

図 2-3 長期署名検証時の文書偽造例

(4) 長期署名検証時における文書偽造の脅威に対する対策

このような偽造問題を解決する方法として、アーカイブタイムスタンプの取得者、或いは発行者が内側のデータ（オリジナル署名、署名タイムスタンプ、検証情報、前世代のアーカイブタイムスタンプ等)の正当性を検証した上で処理することが考えられる。しかしながら、証拠性を示すためにはアーカイブタイムスタンプの取得者及び発行者は、データや検証結果の長期保存及びそれらの信憑性を証明する必要があり、各々に多大な負担が発生する。むしろ現実的な対策として長期署名検証のための TSA 専用の認証局を構築し、その認証局でタイムスタンプ付与時有効であったことを証明する信頼点となる認証局自身の証明書や失効情報等の証拠情報を十分長い期間保管し、長期署名の検証において必要に応じて照合できる状態を永続的に維持する方法が考えられる。これより、アーカイブタイムスタンプの取得者及び発行者は、運用の負担が減り、必要に応じて信頼点となる認証局へ照合を要求すれば、保管している証拠情報によりタイムスタンプの有効性を証明することができる。従って、長期保存における電子署名文書の信憑性を証明することができる。また信頼点となる認証局で保存された失効情報等との照合により、中間 TSA の偽造を検出することができる。従って、(3) 長期署名検証に対する課題（文書偽造の脅威）のような偽 TSA を使った文書偽造はできないのである。

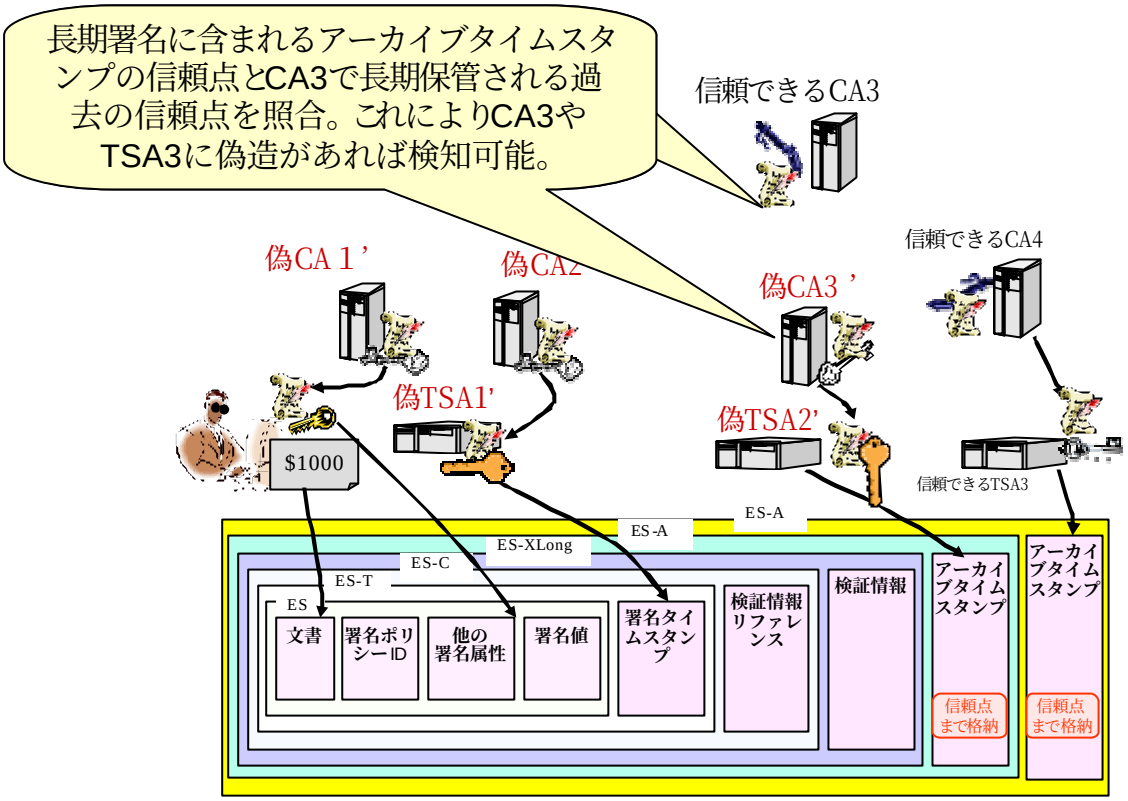

図 2-4 長期署名検証時の文書偽造対策例

(5) TSA 専用の認証局

TSA 専用の認証局とは、TSA の証明書を専門に発行する認証局のことである。認証局のポリシーは、タイムスタンプの役割、TSA 特有の運用、鍵管理は HSM（Hardwere Security Module：FIPS140-2 レベル 3 以上の認定が望ましい）に保管といったことが前提になるので、個人用証明書などを発行する一般的な認証局と異なる。また、長期署名検証に対応する認証局の場合、証明書の有効期限が過ぎても、タイムスタンプ付与時有効であったことを証明するための失効情報等の証拠情報を長期間保管する必要がある。さらに保管した情報の信憑性を証明できる措置を講じておく必要がある。詳しくは付録の「TSA 証明書を発行する認証局の運用ガイドライン」を参照して頂きたい。

2.2.6 リンクトークン方式

(1) 方式の概要

リンクトークン方式はハッシュアルゴリズムの安全性に依存する方式である。利用者から電子文書のハッシュ値を受け取り、証拠となるリンクトークンを返す。また定期的に全体のハッシュ値を歴史的な証拠となるように新聞等に公開している。

図 2-5 は、TSA が直前あるいは時間的に近傍で受け付け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とハッシュ関数により関連付けた時刻証明書を発行する方式である。この方式では、結果的に過去に発行した時刻証明書すべてとのリンクが作成されることになる。このため、過去に発行したすべての証明書との整合性を取らない限り、時刻証明書の偽

造を行うことができない。また、定期的に時刻証明書のリンク情報を新聞等で衆目にさらすことにより、リンク情報の偽造をさらに困難にするとともに、リンク情報の検証を定期的な公開期間内だけで済ませることが可能となる。この方式では発行された時刻証明書の検証に TSA が常に必要となる。リンク情報とは、ある特定時刻（ｔ）に受け付けたハッシュ値を集約して複数の要求を代表する１つのハッシュ値を作成し、その直前のリンク情報（ｔ-1）から新しいリンク情報（ｔ）を作成したものである。また、時間との関連づけを持たせたハッシュ値であり、検証のためのデータとして後に利用されるものである。

リンクトークン方式には、直前あるいは時間的に近傍で受け付け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とハッシュ関数により関連付けた時刻証明書にデジタル署名をして発行する電子書名付きのリンクトークン方式がある。独立トークン方式同様、公開鍵証明書の有効期限内であれば、公開鍵証明書を用いるだけで時刻証明書の検証が可能である。公開鍵証明書の有効期限が切れた場合でも、安全なハッシュ関数を採用した TSA であれば、アーカイブへアクセスするだけで検証が可能である。

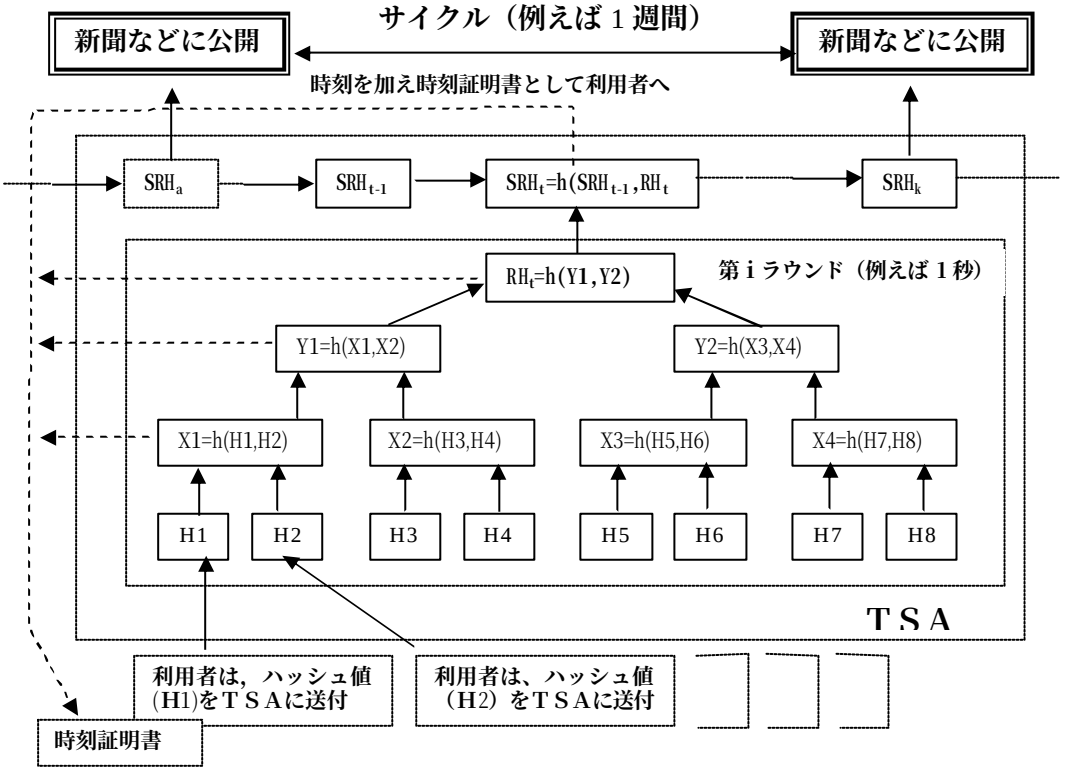

図 2-5 リンクトークン方式

(2) リンクトークン方式の特徴

リンクトークン方式の特徴は、時刻証明期間が長期で高速に処理することができることである。また、リンク情報そのものの信憑性も証明することができることである。

① 時刻証明書の有効期限

時刻証明書の有効期限はハッシュ関数の強度のみに依存する。ハッシュ関数の特徴には、

ハッシュ値から元のデータを計算することが計算量的に困難である「一方向性」とハッシュ値が同じとなる２つの異なる入力を見つけることが計算量的に困難である「衝突困難性」がある。これらの特徴から、計算した元データのユニーク性が担保され、時刻と関係付けることにより、元データが存在した時刻と改ざんされていないことを証明することができる。暗号技術評価プロジェクト（CRYPTREC：Cryptography Research and Evaluation Committees）の暗号技術評価報告書（2002 年度）では、少なくとも 2002 年度から 10 年間安全であるハッシュ値は 160 ビット以上必要であることが報告されている。CRYPTREC の電子政府推奨暗号リストに掲載されたハッシュ関数は RIPEMD-160（160 ビット）、SHA-1（160 ビット）、SHA-256（256 ビット）、SHA-384（384 ビット）、SHA-512（512 ビット）である。なお、（ ）はハッシュ値長である。これまで広く利用されてきた MD5（128 ビット）は現時点で電子政府推奨暗号リスト案に含まれていないので、これを単独でタイムスタンプ発行サービスに用いることは望ましくない。但し、前述のハッシュ関数と MD5 の併用によるサービスの提供は、必要な技術要件を十分に満たしているものと考えられる。時刻証明書による証明期間の長さは、ハッシュ値長に依存する。それぞれの証明期間に必要なハッシュ値長の目安は、現在のところ以下の通りである。[1]

2013 年まで有効（現在から 10 年間）・・・160 ビット
2018 年まで有効（　〃　15 年間）・・・168 ビット
2023 年まで有効（　〃　20 年間）・・・176 ビット
2028 年まで有効（　〃　25 年間）・・・184 ビット
2033 年まで有効（　〃　30 年間）・・・192 ビット

強度の強いハッシュアルゴリズム（複数のハッシュ関数を組み合わせたアルゴリズムも含む）を採用したタイムスタンプであれば、時刻証明書そのものの有効期限を 20 年以上にすることができる。PKI により時刻証明書の発行者を保証する方式ではないので、2.2.6(3)で示した長期署名検証に対する課題のような文書偽造の脅威はない。

② ハッシュアルゴリズム危殆化に対する対策

TSA が使用しているハッシュアルゴリズムは、時間の経過と共に危殆化する可能性が出てくると考えられる。ハッシュアルゴリズムが危殆化しそうになった場合には、TSA は速やかにハッシュアルゴリズムの更新を行ない、利用者は過去に保管されたすべてのデータとタイムスタンプを結合したものに対するハッシュを生成し、これを TSA に送信し、新たなタイムスタンプの取得を行なう。この方式により時刻証明書の寿命を延長することができる。

③ リンク情報の信憑性

TSA に保管しているリンク情報を消失、改ざん、順序変更することは技術的に可能であ

[1] A.K. Lenstra, E.R. Verheul, Selecting Cryptographic Key Sizes, 1999 のデータを参考に算出

る。従って、リンク情報の信憑性が疑われた場合、TSA はリンク情報を消失、改ざん、順序変更していない事を証明する必要がある。リンキングプロトコルの場合、新聞等に公開するサイクル（例えば１週間）単位でリンクしている個々のハッシュ値から計算した結果と新聞等に公開したハッシュ値を比較して一致すればリンク情報の信憑性を証明することができる。いったん作成したリンク情報は、発行した時刻証明書を回収し整合性を取らない限り、痕跡を残さないで改ざんすることはできない。つまりいったんリンク情報を改ざんすると、矛盾のないリンク情報に修復することは事実上不可能なのである。そのため TSA は、利用者と結託し悪意をもってリンク情報の消失、改ざん、順序変更できないのである。

④ 運用

TSA は、リンク情報等のデータを外部の脅威から守るため、運用規定を作成し、セキュアな管理環境、物理的にセキュアな環境及び TSA 管理者および TSA 責任者の二重管理のもとで運用が行われている。リンク情報の生成及び保持はセキュリティ対策が施された安全なシステムで行われている。リンク情報へのアクセスは TSA 責任者、TSA 管理者の両者合意のもとでしかできないようになっている。従って、運用担当者はもとより、TSA 責任者、TSA 管理者単独でリンク情報をアクセスすることはできない。TSA は、運用の正当性を証明するために定期的にリンク情報を公開するが、公知の事実をできるだけ多くすることが望ましいので、最低でも１週間単位で新聞等へ公開している。出版会社等との原稿はサービス提供中保持し続ける。リンク情報を地震等の災害から守るため TSA のバックアップセンタを物理的に離れた箇所に設置し、メインセンタとバックアップセンタ間リンク情報の同期をとり、お互いのリンク情報を保有しあうことにより、サービスを継続することを可能にしている。各々のセンタでは、システム故障等によるデータ消失に備え、リンク情報等のデータをバックアップテープに１日１回保存している。また、ホットスタンバイ方式なので他方がシステム故障してもサービスを継続できるので、年間を通して１日 24 時間サービスを提供し続けることを可能にしている。

詳しくは ECOM 平成１４年度「タイムスタンプサービス運用ガイドライン」の「リンキングプロトコルを用いたタイムスタンプサービス運用規定例」を参考にして頂きたい。

2.3 署名検証技術

2.3.1 署名検証技術とは

デジタル署名が付加された文書に対して、デジタル署名の有効性を検証するための技術について説明する。

デジタル署名技術において「署名生成」と「署名検証」は対となるものであり、「署名生成」は署名者の秘密鍵を用いて行われ、「署名検証」は、検証者が署名者の公開鍵を用いて行う。そのため、「署名検証」にあたって、デジタル署名の有効性を検証するためには、署名者の秘密鍵に対応する公開鍵の証明書の有効性を示すことが求められる。

検証者は、デジタル署名を行った証明書について以下の検証を行う。全てが確認されることが求められる。

① 認証パスの検証

階層構造の場合、認証パスを構築している証明書のチェーンが構成され、最上位認証局の証明書は自身の自己署名であることを確認する。相互認証の場合、認証パスを構築している証明書を順次検証し、自身の証明書の認証パスに存在する認証局との相互証明書の検証までを確認する。

② 有効期間の検証

有効性を検証すべき時刻が、証明書の有効期間に含まれていることを確認する。

③ 証明書の状態の検証

証明書が、検証すべき時刻に失効していないことを確認する。

④ ポリシーの検証

証明書に関するポリシーと整合性があることを確認する。

⑤ 証明書の署名の検証

署名アルゴリズムの計算により、署名が改竄されていないことを確認する。

上記の有効性の検証を行うためには､少なくとも以下の５つの情報が必要であることがわかる。

①署名者の公開鍵証明書

②証明書の公開鍵証明書の認証パスに存在する認証局の公開鍵証明書

③証明書の失効情報（＝検証時に公開鍵証明書が有効であったことを示す情報）

④検証時刻

⑤署名ポリシーの合意情報

2.3.2 署名検証技術の概要

ここでは、2.3.1 に示したデジタル署名の有効性、そのために必要となる公開鍵証明書の有効性を検証するために必要となる以下の技術の概要を説明する。

(1) 証明書の有効性検証技術

(2) 署名の有効性検証技術

(3) データ検証技術

(1) 証明書の有効性検証技術

証明書を利用する場面において､利用する証明書が有効であることを確認する必要がある。証明書の有効性を確認するための技術として以下の２種類を説明する。

① CRL/ARL を利用するもの

発行した証明書が無効となる要因が発生した時点で、認証局が証明書の失効情報リスト（CRL/ARL）を作成し証明書を利用する検証者に配布する（または検証者がダウンロードする)｡検証者は証明書を利用するタイミングで失効情報リストを参照し利用する証明書が無効でないことを検証する。

② OCSP レスポンダにオンラインで問い合わせるもの

CRL/ARL を利用する方法は、検証者自身が CRL/ARL を参照して証明書の有効性を検証する方式であることに対し、検証を行うサーバに対してオンラインで検証要求を送信し、検証結果を受け取る方式も実施されている。

代表的なものが OCSP で RFC2560 として規定されている。

(2) 署名の有効性検証技術

証明書の有効性をオンラインで検証する方式と同様に、デジタル署名の有効性の検証をオンラインで行う技術も存在する。代表的な技術として DVCS（RFC3029） がある。DVCS の特徴としてデータ検証証明書（DVC）を DVCS サーバが発行する。DVCS の利用者は DVC を保存しておくことにより、次の機能（サービス）が実現される。

①過去の時点における証明書の有効性を検証可能となる
②過去の時点に電子文書が存在していたことを証明可能となる。
③その文書を所有していたことを証明可能となる。
④デジタル署名の有効性を検証可能となる（電子文書が改竄されていないことを証明可能となる）
⑤検証したデータを DVCS が保管するアーカイブ機能を持ち､データを保管する TTP としてサービスを提供することも可能である。

(3) データ検証技術

公開鍵証明書やデジタル署名の検証だけでなく、ある時刻におけるデジタル文書の所有を証明するための技術も提案されている。代表的な技術として DVCS がある。

上記に説明した各技術とそれらを実現するための代表的な技術を表 2-2 にまとめる。

表 2-3 署名検証関連技術

項番	技術分類（標準化状況）	証明書の有効性検証	署名の有効性検証	データ検証	認証パス構築	認証パス検証
1	CRL/ARL	○				
2	OCSP（RFC2560）	○				
3	DVCS（RFC3029）	○	○	○	○	○

2.3.3 署名再検証に必要な情報の保存について

(1) 署名再検証に関連する時点

電子文書の検証技術について述べてきたが、次に、電子文書を長期間保存した後に再検証するために保存しておかなければならない情報について検討する。３つの時点について定義を確認しておく。図 2-6 を参照。

①署名時点：署名者がデジタル署名を作成した時刻

②検証時点：検証者がデジタル署名を検証した時刻

③再検証時点：長期間経過後に、再度署名検証が必要となり検証する時刻

一般に、署名時点と検証時点は近いことを想定している。(例えば、契約書作成時には、署名およびその検証により契約が完了する。)

一方、再検証は５年、１０年といった長い期間が経過した後を想定している。(例えば、紛争解決のために過去の契約書の内容を確認する場合。)

再検証を行うために、署名時点・検証時点で保存しておくべき情報をここでは検討する。

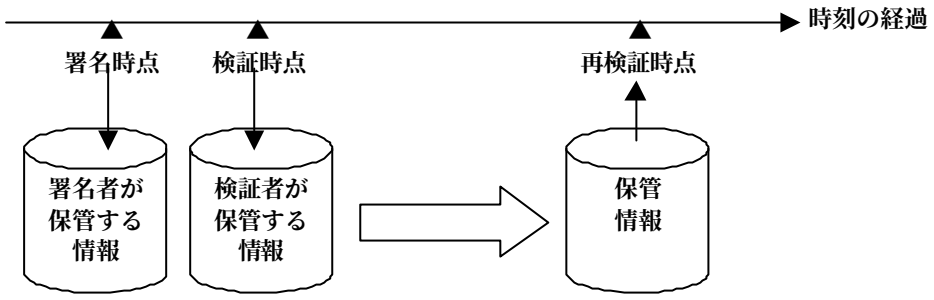

図 2-6 署名検証に関係する時点

(2) 署名再検証のために保存すべき情報

デジタル署名の再検証に必要な証拠情報として、2.3.1 では、次の５つを示した。

①署名者の公開鍵証明書

②証明書の公開鍵証明書の認証パスに存在する認証局の公開鍵証明書

③証明書の失効情報（＝検証時に公開鍵証明書が有効であったことを示す情報）

④検証時刻

⑤署名ポリシーの合意情報

これらの情報のうち、３点目の失効情報については署名検証時の方式により保管する情報が異なる。表 2-4 に技術ごとに保存すべき情報とその時の注意事項を纏める。

表 2-4 再検証のために保存するべき情報

利用する技術	保存する情報	注意事項・補足
CRL	検証時点の最新 CRL	CRL が検証時点で最新であったことを示すのは一般に困難である。 少なくとも、検証時点が CRL 作成時点よりも後、かつ、CRL の定期更新・次回更新時点よりも前であることを確認しておく
OCSP	OCSP レスポンダからの応答メッセージ	OCSP レスポンダからの応答メッセージ自身の再検証に必要な情報も保管する必要がある。例えば、 ・OCSP レスポンダの公開鍵証明書およびその認証パス上の認証局の公開鍵証明書 ・タイムスタンプサーバを使用している場合、タイム

		スタンプサーバの公開鍵証明書およびその認証パス上の認証局の公開鍵証明書　等が必要となる。
DVCS	DVCS サーバからの応答メッセージ ※1	OCSP と同様にDVCS サーバからの応答メッセージ自身の再検証に必要な情報も保管する必要がある。 例えば、 ・DVCS サーバの公開鍵証明書およびその認証パス上の認証局の公開鍵証明書 ・タイムスタンプサーバを使用している場合、タイムスタンプサーバの公開鍵証明書およびその認証パス上の認証局の公開鍵証明書　等が必要となる。

※　1：DVCS の利用形態として、データをアーカイブするための TTP としての機能を持たせることも可能である。この場合には、長期間経過後も、アーカイブされていたデジタル署名された電子文書の正当性は TTP としての DVCS により保証されることになる。ここでは、アーカイブ機能を想定しない DVCS を利用する場合に保存すべき情報を示す。

(3) 再検証のための情報の保管者

図 2-6 に示したように、署名再検証のためには署名時点、検証時点の情報を保存することが求められる。これらの情報を誰が保管すべきかは、取り扱われる署名の目的・ビジネスモデルに依存するため一概には決めることはできない。一般論としては、将来自身の利益を保護するために再検証が必要となる者が保存することを求められると考えられる。

2.4　署名フォーマット形成技術

長期署名フォーマットは、時間の経過と共に失われる可能性があるデジタル署名の有効性を維持するために、RFC2630 で規定されている署名フォーマットを拡張したフォーマットであり、「ESTI TS 101 733 Electronic Signature Formats」において規定されている。長期署名フォーマットは、RFC3126 にもなっている。

長期署名フォーマットは、タイムスタンプ発行局やリポジトリ等の信頼サービスプロバイダを利用してデジタル署名の有効性を維持するための情報を生成する方式と、その情報をフォーマット内に格納する方式ついて規定したものである。

デジタル署名の有効性を維持するためにデジタル署名に情報を追加する様子を図 2-7 に示す。

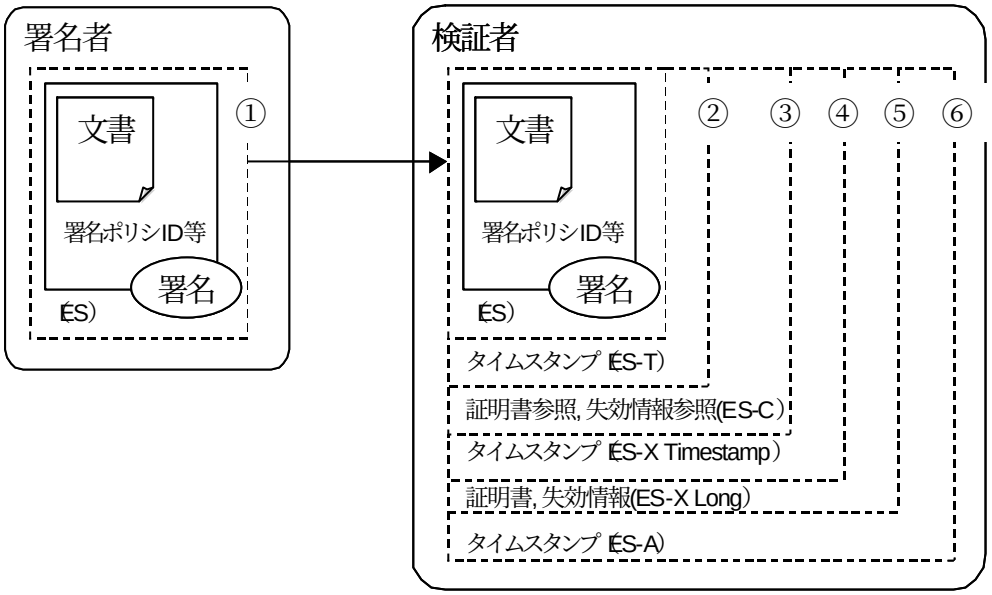

図 2-7 長期署名フォーマット

① ES

いわゆる署名付き文書である。署名者は、デジタル署名を生成する時、文書に加え、署名ポリシーID や署名時刻などの情報を含めたデジタル署名を生成できる。

② ES-T

ES に対して ES の署名値に対するタイムスタンプを付加したデータである。

③ ES-C

ES-T に対して、証明書パス上の公開鍵証明書へのリファレンスと公開鍵証明書の失効情報へのリファレンスを付加したデータである。

④ ES-X Timestamp

CA の鍵が危殆化する場合に備えて、ES-C 全体もしくは、ES-C で追加した情報に対するタイムスタンプを取得し、これをデジタル署名に添付する。

⑤ ES-X Long

署名再検証に必要な情報を保存するため、証明書パス上の公開鍵証明書、公開鍵証明書の失効情報をデジタル署名に追加する。

⑥ ES-A

署名再検証に必要な情報を改ざん検出可能な状態にするため、デジタル署名とデジタル署名に追加した情報に対するタイムスタンプを取得し、これをデジタル署名に添付する。タイムスタンプは署名フォーマット内の非署名属性に含まれる。

2.5 署名ポリシー合意形成技術

署名ポリシーは、署名者と検証者がデジタル署名を有効とみなすための規則を集めたものであり、「ESTI TS 101 733 Electronic Signature Formats」において規定されている。署名ポリシーは、RFC3125 にもなっている。この署名ポリシーの記述形式には以下の２つの形式がある。

1. 可読形式
   文書に添付されたデジタル署名が、法的あるいは契約の要件に適合するか否かを人間が判断できるように署名ポリシーを可読にした形式。
2. 計算機で処理可能な形式
   文書に添付されたデジタル署名が、署名ポリシーに準拠しているか否かを計算機で判別できるようにした形式。

署名ポリシーを用いたデジタル署名の生成と検証は、図 2-8 に示すようになる。

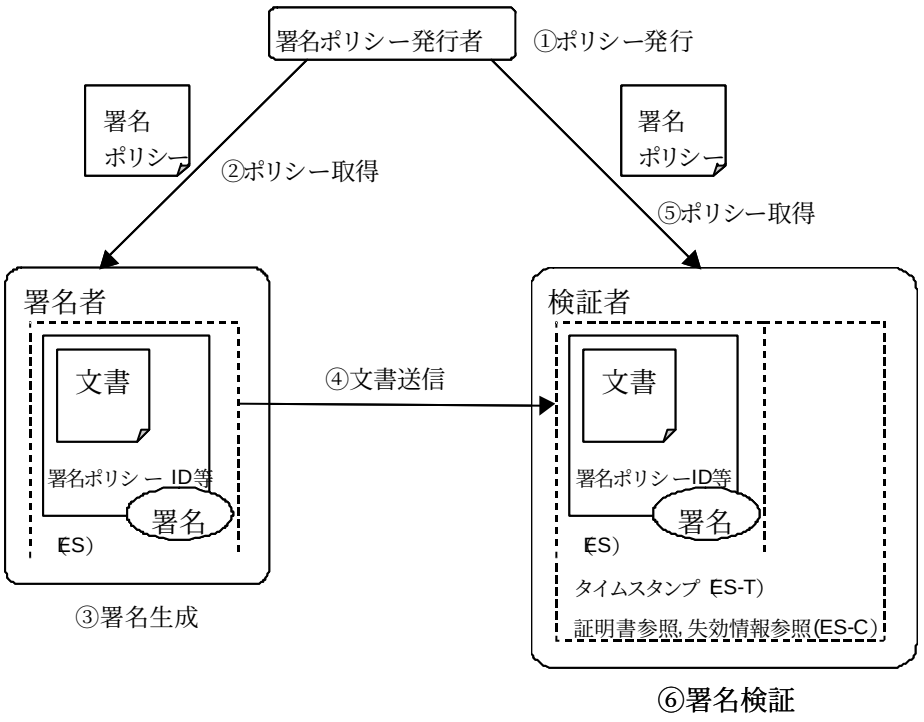

図 2-8 署名ポリシーを用いたデジタル署名の生成と検証

① 署名ポリシー発行者は、署名ポリシーを公開する。
② 署名者は、要件に合致する署名ポリシーを署名ポリシー発行者から入手する。
③ 署名者は､署名ポリシーに記述された規則に従ってデジタル署名を生成する｡この時、署名ポリシーを特定する署名ポリシーID をデジタル署名に付加する。
④ 署名者は、検証者に文書とデジタル署名を送信する。
⑤ 検証者は、デジタル署名に付加された署名ポリシーID を参照し、この署名ポリシーを署名ポリシー発行者から入手する。
⑥ 検証者は、署名ポリシーに記述された規則に従って、デジタル署名を検証する。この時、デジタル署名の再検証に必要となる情報をデジタル署名に追加する。

2.6 長期署名フォーマットと署名ポリシーのプロファイル

2.6.1 長期署名フォーマットのプロファイル

ETSI TS 101 733（RFC3126）で規定した電子署名フォーマットは極めて汎用的に定められており各種のタイプの署名フォーマットや多くのオプションを持っている。ここでは長期署名検証の観点から推奨する署名フォーマットのタイプを次の通り限定することにする。

① ES：基本署名フォーマットで署名ポリシー、署名属性、署名値からなる
② ES-T：ES にタイムスタンプを付加したもの
③ ES-C：ES-T に認証パス上にある全ての証明書と CRL または OCSP 応答の参照値を付加したもの

拡張署名フォーマット（ES-X）には目的により以下のタイプがある

④ ES-X Long：ES-C に認証パス上にある全ての証明書と CRL または OCSP 応答を添付したもの
⑤ ES-A：ES-X にタイムスタンプを付加したもの

2.6.2 署名ポリシーのプロファイル

ETSI の規定した上記２の ASN.1 構文の署名ポリシーは、汎用的に定義されており、多くのオプションを可能にしているので、ここでは長期署名保存の観点から署名ポリシーとして最低限必要な署名ポリシーのプロファイルを規定する。

この署名ポリシーのプロファイルでは関連する属性証明書に関するものは含めないことにした。また署名アルゴリズムについては CMS 署名フォーマット、証明書に指定されるので割愛した。

署名ポリシー（SignaturePolicy）プロファイル

```
SignaturePolicy ::= SEQUENCE {
        signPolicyHashAlg       AlgorithmIdentifier,  --ハッシュアルゴリズム
        signPolicyInfo          SignPolicyInfo,
        signPolicyHash          SignPolicyHash      OPTIONAL }
```

署名ポリシーを保護するために、ハッシュ値を付けることが出来る（オプション）。

```
SignPolicyHash ::= OCTET STRING
```

```
SignPolicyInfo ::= SEQUENCE {
        signPolicyIdentifier            SignPolicyId,  --この署名ポリシーのOID
        dateOfIssue                     GeneralizedTime,  --ポリシー発行日
```

```
        policyIssuerName                PolicyIssuerName,  --ポリシー発行者名
        fieldOfApplication              FieldOfApplication,  --適用分野
        signatureValidationPolicy       SignatureValidationPolicy,
                                                    --署名有効性ポリシー
        signPolExtensions               SignPolExtensions  OPTIONAL }

SignPolicyId ::= OBJECT IDENTIFIER
PolicyIssuerName ::= GeneralNames
FieldOfApplication ::= DirectoryString
```

signPolExtensions は署名ポリシーを拡張するもので、どのようなものでも定義して加えることが出来る。

署名有効性検証ポリシー（SignatureValidationPolicy）

署名者が指定するデータ要素と、検証者がこの署名ポリシーの元に付けなければならないデータ要素を定義する。

```
SignatureValidationPolicy ::= SEQUENCE {
        signingPeriod           SigningPeriod,    --署名ポリシーの適用日
        commonRules             CommonRules,      --署名ポリシーの共通規則
        commitmentRules         CommitmentRules,  --署名者が約束する規則
        signPolExtensions       SignPolExtensions   OPTIONAL}

SigningPeriod ::= SEQUENCE {
        notBefore       GeneralizedTime,  ポリシー適用開始日
        notAfter        GeneralizedTime   OPTIONAL  –ポリシー適用終了日 使用しない}
```

notAfterは、長期署名の観点から本プロファイルでは含めないことにする。

共通規則（CommonRules）

すべての CommitmentRules タイプに共通する規則。

属性証明書（attributeTrustCondition）と署名アルゴリズム（algorithmConstraintSet）については本プロファイルでは指定しないことにする。署名アルゴリズムは CMS 署名構文や証明書に指定したものに従う。

```
CommonRules  ::= SEQUENCE {
        signerAndVeriferRules       [0]  SignerAndVerifierRules  OPTIONAL,
        signingCertTrustCondition   [1]  SigningCertTrustCondition  OPTIONAL,
        timeStampTrustCondition     [2]  TimestampTrustCondition  OPTIONAL,
```

```
        attributeTrustCondition     [3]  AttributeTrustCondition  OPTIONAL,
                                                                  --指定しない
        algorithmConstraintSet      [4]  AlgorithmConstraintSet   OPTIONAL,
                                                                  --指定しない
        signPolExtensions           [5]  SignPolExtensions   OPTIONAL}
```

CommonRules に以下の領域が含まれていた場合は、それぞれの CommitmentRules にはこの領域は含めてはいけない。

* signerAndVeriferRules;
* signingCertTrustCondition;
* timeStampTrustCondition.

コミットメント規則（CommitmentRules）

CommitmentRules は有効性検証のための幾つかの CommitmentRule から構成される。

```
CommitmentRules ::= SEQUENCE OF CommitmentRule

CommitmentRule  ::= SEQUENCE {
        selCommitmentTypes                  SelectedCommitmentTypes,
        signerAndVeriferRules          [0]  SignerAndVerifierRules  OPTIONAL,
        signingCertTrustCondition      [1]  SigningCertTrustCondition  OPTIONAL,
        timeStampTrustCondition        [2]  TimestampTrustCondition   OPTIONAL,
        attributeTrustCondition        [3]  AttributeTrustCondition  OPTIONAL,
                                                                  --指定しない
        algorithmConstraintSet         [4]  AlgorithmConstraintSet  OPTIONAL,
                                                                  --指定しない
        signPolExtensions              [5]  SignPolExtensions  OPTIONAL}

SelectedCommitmentTypes ::= SEQUENCE OF CHOICE {
        Empty                          NULL,
        recognizedCommitmentType       CommitmentType }
```

SelectedCommitmentTypes で empty を指定した場合は、CommitmentType が提示されていないものとする。すなわち実質的な CommitmentType はこのポリシーにはなく、メッセージ本体に含まれているとする。そうでなければ、かならず CommitmentType で指定するものでなければならない。

```
CommitmentType ::= SEQUENCE {
```

```
        Identifier                  CommitmentTypeIdentifier,
        fieldOfApplication      [0] FieldOfApplication OPTIONAL,
        semantics               [1] DirectoryString OPTIONAL }
```

署名者と検証者の規則（SignerAndVeriferRules）

この規則は ETSI の長期署名フォーマットに適用される。これは署名者規則と検証者規則からなる。

```
SignerAndVerifierRules ::= SEQUENCE {
        signerRules      SignerRules,
        verifierRules    VerifierRules }
```

署名者規則（SignerRules）

```
SignerRules ::= SEQUENCE {
        externalSignedData          BOOLEAN       OPTIONAL,
                        -- True 署名データがCMS構造の外にある場合
                        -- False 署名データがCMS構造に含まれる場合
                        -- どちらでも良ければこの領域は現れない
        mandatedSignedAttr          CMSAttrs,  --必須とするCMS署名属性
        mandatedUnsignedAttr        CMSAttrs,  --必須とするCMS非署名属性
        mandatedCertificateRef      [0] CertRefReq DEFAULT signerOnly,
                                -- 必須とする証明書参照
        mandatedCertificateInfo     [1] CertInfoReq DEFAULT none,
                                -- 必須とする証明書情報
        signPolExtensions           [2] SignPolExtensions        OPTIONAL  }

CMSattrs ::= SEQUENCE OF OBJECT IDENTIFIER

CertRefReq ::= ENUMERATED {
                signerOnly (1),  --署名者証明書のみを必須とする
                fullpath   (2)   --信頼点までの完全な認証パスへの参照を要する  }
```

mandatedCertificateInfo 領域は署名者が、署名者の証明書のみか、認証パス中の全ての証明書を CMS の SinedData 付けなければならないかを指定する。

```
CertInfoReq ::= ENUMERATED {
                None        (0),    --必須要件なし
                signerOnly (1),     --署名者の証明書のみを必須とする
                fullpath   (2)      --信頼点までの完全認証パスの証明書    }
```

検証者規則（VerifierRules）

検証者規則は、このポリシーで示された非署名属性が存在しなければならないことを示す。もし、この非署名属性が署名者によって提示されていない場合は、署名フォーマットに検証者が付加えなければならない。（これはタイムスタンプなどに適用される）

```
VerifierRules ::= SEQUENCE {
                mandatedUnsignedAttr    MandatedUnsignedAttr,
                signPolExtensions       SignPolExtensions  OPTIONAL   }

MandatedUnsignedAttr ::=  CMSAttrs  --必須とするCMS非署名属性
```

署名証明書信頼点条件（SigningCertTrustCondition）

署名証明書信頼点条件は信頼点証明書の集合（CertificateTrustTrees）と証明書失効の要件（CertRevReq）からなる。

```
SigningCertTrustCondition ::=   SEQUENCE {
     signerTrustTrees              CertificateTrustTrees,
     signerRevReq                  CertRevReq  }
```

信頼点証明書の集合（CertificateTrustTrees）

```
CertificateTrustTrees ::=   SEQUENCE OF CertificateTrustPoint

CertificateTrustPoint ::= SEQUENCE {
        Trustpoint              Certificate, --自己署名証明書
        pathLenConstraint       [0] PathLenConstraint   OPTIONAL,  --指定しない
        acceptablePolicySet     [1] AcceptablePolicySet OPTIONAL,  --指定しない
        nameConstraints         [2] NameConstraints     OPTIONAL,  --指定しない
        policyConstraints       [3] PolicyConstraints   OPTIONAL  --指定しない }
```

このプロファイルでは､証明書の信頼点にはトラストアンカーとなる CA の証明書のみを指定し、その他のオプションは指定しない。これらのオプションの値は通常信頼点証明書に指定されているからである。

失効情報要件（CertRevReq）

```
CertRevReq ::= SEQUENCE {
        endCertRevReq   RevReq,
        caCerts     [0] RevReq  }
```

＊endCertRevReq 領域はエンド･エンティティの証明書の失効要件で、署名者証明書、タイムスタンプ（TSA）証明書を含む。

＊caCerts 領域は CA 証明書に関する失効要件である。

```
RevReq ::= SEQUENCE  {
        enuRevReq  EnuRevReq,
        exRevReq   SignPolExtensions OPTIONAL}

EnuRevReq  ::= ENUMERATED {
        clrCheck   (0),  --現在のCRLs（ARLs）をチェックしなければならない
        ocspCheck  (1), -- OCSPで証明書状態をチェック
        bothCheck  (2),  --CRLsとOCSP共にチェック
        eitherCheck  (3),  --どちらかをチェックしなければならない
        noCheck    (4),  --チェックなし
        other      (5)  --署名ポリシー拡張で定義した方法でチェック、使用しない   }
```

タイムスタンプ信頼点

タイムスタンプの信頼点に関する条件は、ttsCertificateTrustTrees 領域として TSA の証明書の信頼点(複数可)(オプション)と TSA の失効情報について指定する。CautionPeriod と signatureTimestampDelay についてはオプションである。

TtsCertificateTrustTrees がなかった場合は CA と同じ信頼点を用いる。

TtsRevReq は TSA の失効状態について最低 CRLs または OCSP の状態情報を用いる。

```
TimestampTrustCondition ::= SEQUENCE {
    ttsCertificateTrustTrees    [0]    CertificateTrustTrees  OPTIONAL,
    ttsRevReq                   [1]    CertRevReq   OPTIONAL,
    ttsNameConstraints          [2]    NameConstraints   OPTIONAL,  使用しない
    cautionPeriod               [3]    DeltaTime  OPTIONAL,
    signatureTimestampDelay     [4]    DeltaTime   OPTIONAL }

DeltaTime ::= SEQUENCE {
        deltaSeconds    INTEGER,
        deltaMinutes    INTEGER,
        deltaHours      INTEGER,
        deltaDays       INTEGER }
```

cautionPeriod は、完全な検証情報が得られなかった時に、つぎに検証するまでの待ち時間である。

SignatureTimestampDelay は、署名時間からタイムスタンプを付けるまでの許容時間で

ある。

署名ポリシー拡張

署名ポリシー拡張は本署名ポリシーの各所で独自に定義できるようになっており、OID を指定しても散ることが出来る。

```
SignPolExtensions ::= SEQUENCE OF SignPolExtn

SignPolExtn ::= SEQUENCE {
                extnID      OBJECT IDENTIFIER,
                extnValue   OCTET STRING
```

2.7 ヒステリシス署名

ヒステリシス署名とは、電子データに施された電子署名が当該利用者が生成したものなのか、あるいは不正者によって偽造されたものなのかということを判別するために、事後になっても署名の正当性を確認可能とする技術である。

図 2-9 はヒステリシス署名の生成方法を示している。通常の電子署名の生成と異なり、前回生成した署名を用いることを特徴としている。これにより、各署名間に連鎖関係が構築される。また、定期的に最新の署名を新聞に公開したり、信頼できる第三者機関に預託したりすることで、当該署名を信頼ポイントとすることができる。これにより、署名者本人も信頼ポイント以前の署名履歴の偽造が不可能となり、署名履歴の整合性を証明できる。

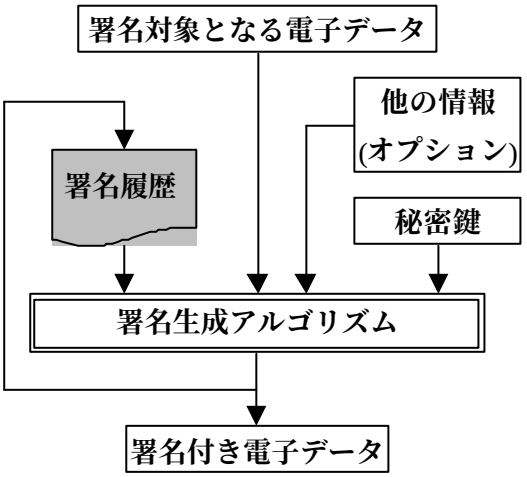

図 2-9 ヒステリシス署名概要

また、ヒステリシス署名は pkcs#7 や XML 署名フォーマットなどの標準の署名フォーマットに対応可能であり、公開鍵証明書の有効期間内であれば一般に流通している署名検証ソフトウェアで検証できる。公開鍵証明書の有効期限後は、最新の署名もしくは信頼ポイントから署名履歴の整合性検証を行うことで電子署名の偽造の有無を確認できる。

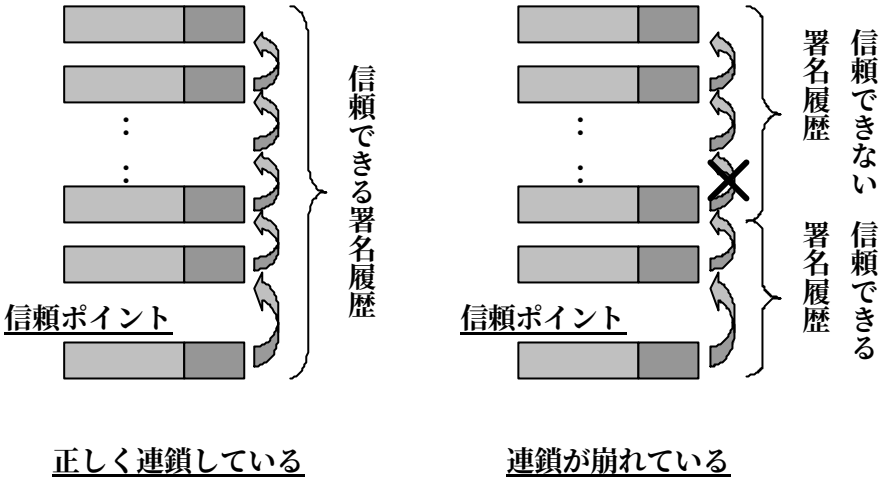

図 2-10 署名履歴の検証

不正者Bが「利用者Aの過去の電子署名α」を偽造するためには、電子署名αを生成して以降の全ての電子署名を整合的に偽造することが必要となる。すなわち、利用者は自己の署名履歴を調停者に提示することによって、当該利用者が生成した個々の電子署名の現在に至るまでの順序関係を示すことで、それ以外の電子署名を生成していないことを証明できる。

●履歴交差プロトコル

ヒステリシス署名においては、各利用者の署名履歴を交差させることで改竄を著しく困難にすることができる。

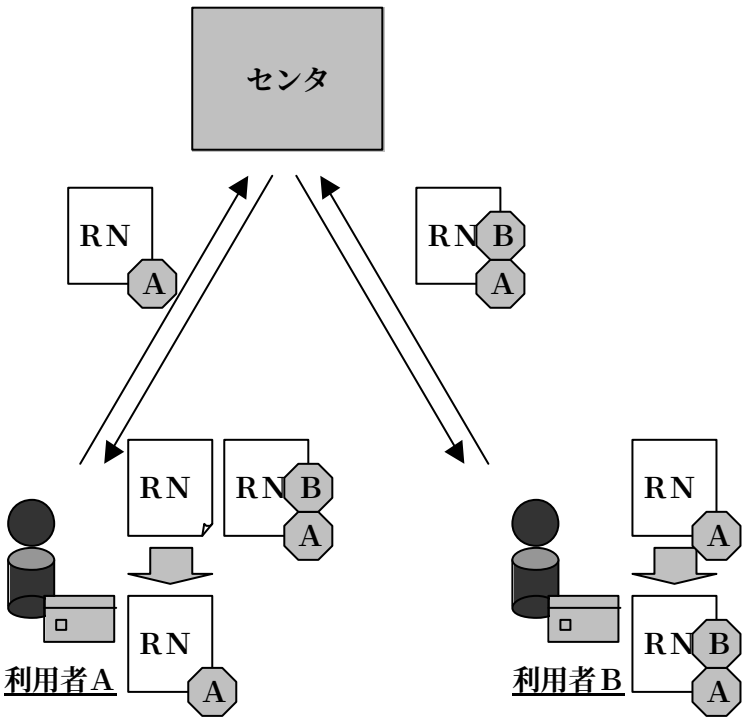

図 2-11 履歴交差プロトコル概要

1．　利用者Ａは乱数を生成して電子署名を施し、当該乱数に対する署名結果を署名履歴に含ませる。
2．　利用者Ａは署名付き乱数をセンタに送付する。
3．　センタはランダムに選択した利用者Ｂに利用者Ａから送られてきた署名付き乱数を送付する。
4．　利用者Ｂはセンタから送られてきた署名付き乱数に電子署名を施し、当該データに対する署名結果を署名履歴に含ませる。
5．　利用者Ｂは二者署名付き乱数をセンタに返送する。
6．　センタは利用者Ｂから返送されてきた二者署名付き乱数を利用者Ａに送付する。

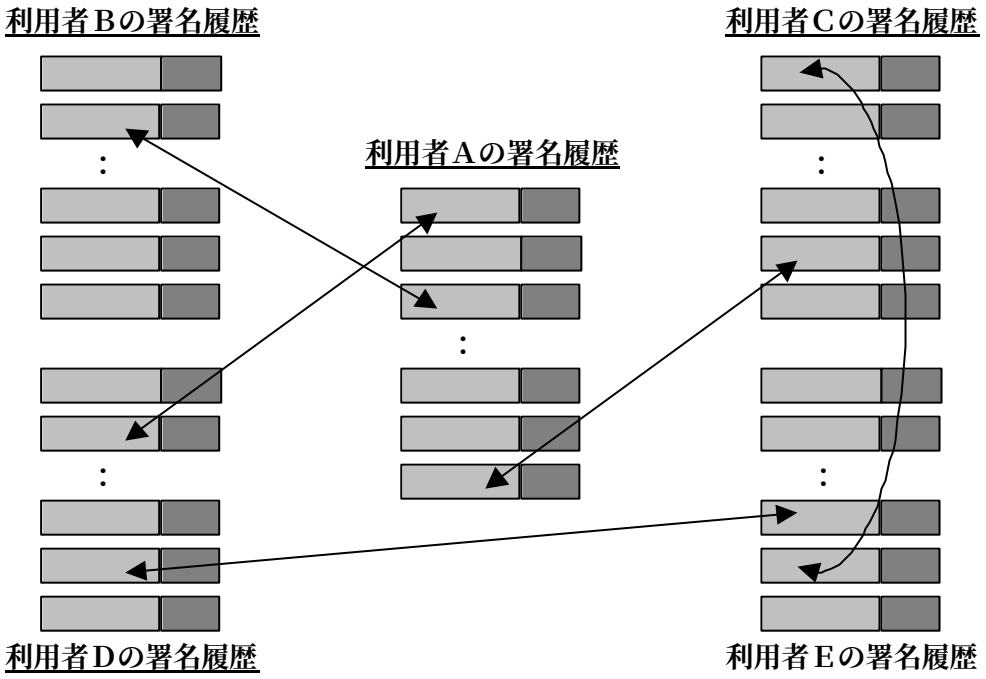

図 2-12 履歴交差している署名履歴

署名履歴の中に他の利用者と履歴交差を行った部分があれば、当該交差相手の署名履歴を確認することによって、欠落した履歴データ以前に署名されたものの真偽も検証可能となる。

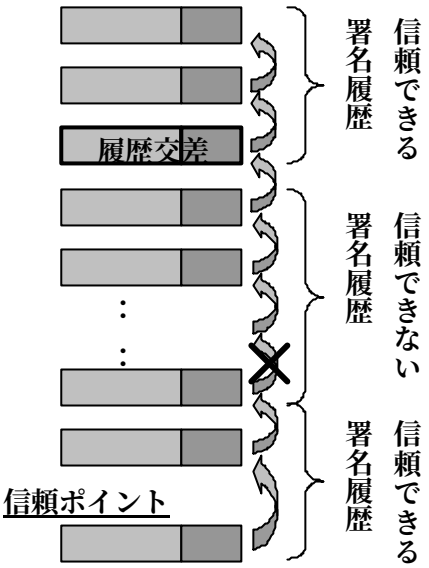

図 2-13 履歴交差の検証

3．製品紹介

3.1 製品動向概要

これまで電子署名文書の長期保存に関する機能要件、及び当該技術を述べてきたが、ここからは現在の電子署名文書長期保存に関する製品動向について述べる。

2 章で示した当該技術を組み合わせて電子署名文書長期保存サービスを実現している製品については、表 3-1 で示されるような事例がある。

表 3-1 電子署名文書長期保存に関する製品特徴

製品名称		PKIサーバ/Carassuit	署名ﾌﾟﾗｯﾄﾌｫｰﾑ(仮)	セキュアな長期原本保管システム	原本性保証システムDP1/Proofbox2	三菱署名有効性延長システムMISTYGUARD <EVERSIGN>	電位文書流通ﾌﾟﾗｯﾄﾌｫｰﾑSecurePod
開発元		日本電気（株）	NTTコムウェア（株）	NTTコミュニケーションズ（株）	（株）日立製作所	三菱電機（株）	（株）NTTデータ
利用技術	時刻証明	RFC3161ﾀｲﾑｽﾀﾝﾌﾟ または ｵﾘｼﾞﾅﾙﾀｲﾑｽﾀﾝﾌﾟ	XMLﾀｲﾑｽﾀﾝﾌﾟ（OASIS TIML Working Draft 01,6 September 2002）	ｵﾘｼﾞﾅﾙﾀｲﾑｽﾀﾝﾌﾟ	なし	RFC3161ﾀｲﾑｽﾀﾝﾌﾟ	リンキング方式
	非改竄証明	RFC3161ﾀｲﾑｽﾀﾝﾌﾟ または ｵﾘｼﾞﾅﾙﾀｲﾑｽﾀﾝﾌﾟ	XMLﾀｲﾑｽﾀﾝﾌﾟ（OASIS TIML Working Draft 01,6 September 2002）	秘密分散法	ヒステリシス署名	RFC3161ﾀｲﾑｽﾀﾝﾌﾟ	リンキング方式、デジタル署名
	署名延長/過去検証	ｵﾘｼﾞﾅﾙﾌｫｰﾏｯﾄ（RFC3126の必要な情報を付加）	XAdESフォーマット	ｵﾘｼﾞﾅﾙﾌｫｰﾏｯﾄ(RFC3126に必要な情報付加)	ヒステリシス署名により長期保証	RFC3126長期署名フォーマット	オリジナル指定時刻検証
	原本保管・管理	独自仕様（原本、システムログ等を3DESにて暗号化する）	文書管理システム等との連携が可能	秘密分散法により分散して保管	データベースで原本保管。外部文書管理システムとの連携可能	ファイリングシステム、文書管理システムとの連携が可能	毎日バックアップ、2拠点保管,文書管理ｼｽﾃﾑ等との連携可能
	見読性確保	特になし	XML署名	なし	外部文書管理システムにおいて表示	特になし（PDF署名、XML署名（予定）に対応）	PDFの利用を推奨
	機密性確保	独自仕様（利用者毎にロール(役割)を与え、権限を制限する）	特に無し（文書管理システム等に依存）	秘密分散法による符号化等	アクセス制御、アクセス履歴の管理、操作履歴の管理、ユーザー認証、暗号化	特になし（原本保管・管理システムに依存）	アクセス制御、ユーザ認証等
	送達確認	特になし（送達確認サーバとの連携は可）	特に無し（別途ログ監視サーバと組み合わせる事により可能）	なし	なし	特になし	電子メール(オプション：ﾊｯｼｭ比較、ﾚｼﾞｭｰﾑ機能)
	鍵管理	独自仕様	HSM	なし（鍵は使用しない）	ファイルとして保管	長期署名の有効性に影響を与える重要な鍵を持たない	鍵は使用しない
対応データ		任意（XML用に独自に変更する）	XML署名データ	任意	制限なし	CMS署名データ、PDF署名データ、XML署名データ（予定）	任意、署名検証はPDF署名ﾃﾞｰﾀのみ、XML署名ﾃﾞｰﾀ（予定）
データ保障期間		超長期間	超長期間	期限なし	期限なし	超長期間	超長期間
利用者AP		独自AP	任意（JAVAライブラリを利用して開発可能）	SOAPインタフェース	クライアントAPIを利用して作成	任意（クライアントライブラリを利用して開発可能）	ブラウザ、接続API提供
利用者鍵管理		特になし	HSM	不要（利用者の署名は不要）	不要（利用者の署名は不要）	制限なし	eToken等のデバイス推奨

この表から分かるように、各企業において 2 章で示している各技術を組み合わせて、電子文書長期保存サービスを実現している。次節からは、上記製品の実現内容、及びそのサービス状況について詳細を示す。

3.2 PKI サーバ/Carassuit 原本保管サーバ（日本電気（株））

3.2.1 はじめに

「PKI サーバ／Carassuit 原本保管サーバ」は、電子文書の改ざんや不正使用などの防止を可能とすることで、電子文書の原本性を保証し、安全かつ確実な電子文書の管理を実現するソフトウェアである。このソフトウェアは、電子自治体市場を中心に重要文書の電子化を進める官公庁・自治体・企業向けのセキュリティ製品である。

「PKI サーバ／Carassuit 原本保管サーバ」は、電子署名技術や暗号技術を活用することで、電子文書の原本性を確保するための要件である、(1) 電子文書に対する改変履歴を記録することで改ざんを防止する「完全性」、(2) 保管する電子文書ごとにアクセス制御を行い、電子文書の盗難や漏洩などを防止する「機密性」への対応を実現した製品である。

3.2.2 機能と特徴

「PKI サーバ／Carassuit 原本保管サーバ」は、保管したデータが流出した場合でも内容の流出を防ぐために文書を 3DES にて暗号化を行っている。さらにデータが改ざんされた場合には検出することは可能となっている。また原本保管時から長期間改ざんされずに保管していることを保証するために、タイムスタンプを繰り返し付与することが可能となっている。

「PKI サーバ／Carassuit 原本保管サーバ」の主な特徴は「システムログ機能」「ロールベース利用者管理機能」「バックアップ／リカバリ機能」であり、これらの機能によりシステムのセキュアな運用が可能となる。

- システムログ管理機能
  原本へのアクセスログ、システムへのアクセスログを 3DES により暗号化して保管し、ログの改ざん検知を行う。
- ロールベース利用者管理機能
  利用者を登録し、ロール（役割）を与えることができる。ロールには、サーバ管理者としてのシステム管理、文書管理者としての原本保管・更新・削除などのアクセス権を与えることができる。
- バックアップ／リカバリ機能
  原本データ、システムデータのバックアップ・リカバリ機能によりシステム障害に備える。

その他の機能として以下のような機能がある。

- バージョン管理機能
  原本に対する保管・更新・削除操作を管理し、旧バージョンの文書の取得が可能である。
- レシート発行機能
  原本保管・更新・削除時に操作を行ったことを証明するレシートを発行する。レシートの提示により原本の取得・更新・削除が可能となる。また、レシートには原本のバージョン情報が記載されており、バージョン毎の文書の管理が容易となる。
- Java 言語連携機能
  Java 言語での操作 API を提供し、他システムとの連携が可能である。

3.2.3 システム構成

以下の図のように NEC の PKI 製品ラインナップにより、原本保管の実現に必要な PKI 機能を、標準技術をベースにトータルに提供している。(図中の（ ）は主な標準技術を表している)

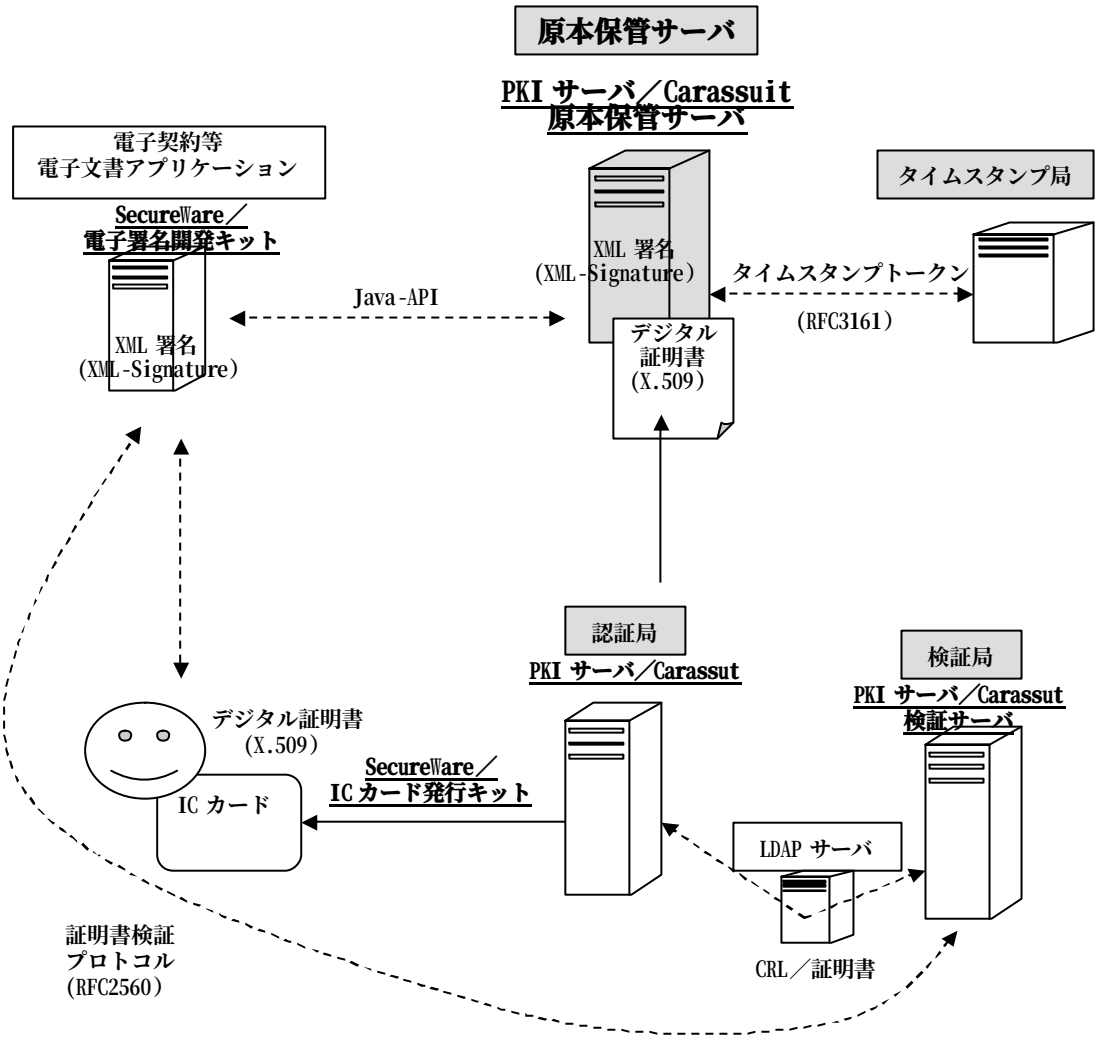

図 3-1 「PKI サーバ／Carassuit 原本保管サーバ」システム構成

3.2.4 運用方法

原本保管サーバの運用イメージとして、「利用者の登録」「原本の登録手順」「原本の取得手順」「原本の更新手順」を以下に示す。

■ 利用者登録

管理者が原本保管サーバに利用者の登録を行う。その際、ユーザ毎に以下の権限が登録可能である。

- 原本管理権限

  原本の登録、更新、削除が可能となり、また原本の取得履歴一覧を参照することが可能となる。また、レシート再発行、原本有効性確認、アーカイブタイムスタンプ再付加することが可能となる。

- 利用者管理権限

利用者登録の更新、削除、セキュリティ属性といった利用者の管理が可能となる。

- 履歴管理権限
  アクセスログ参照、システムログ参照することが可能となる。
- 記憶領域管理権限
  DB 領域参照、変更が可能となる。

■ 原本登録

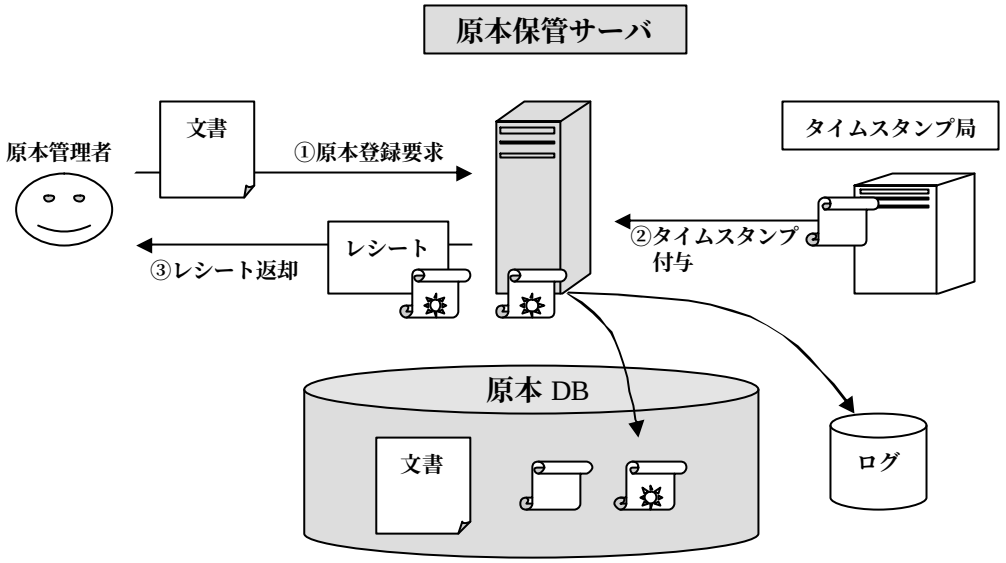

図 3-2 原本登録

① 原本管理権限を持つ人が原本を登録する。
② タイムスタンプを付与し、原本 DB に登録する。(改ざん防止や存在証明の為、原本データのハッシュ値、タイムスタンプ情報を付加し、文書に 3DES アルゴリズムを使用して暗号化を行う。さらにログ情報を出力する。)
③ 文書の保守に必要となるレシートが返却される。その際、レシートにも署名がなされ、改ざんは不可能となる。

■ 原本取得

① 取得希望文書のレシートを提示する。
② レシートの署名を検証し、有効性を確認する。
③ レシートより原本を特定し、写しを作成する。(文書の改ざんをチェックし、復号を行う。さらにログ情報が出力される。)
④ 文書(原本の写し)が返却される。

■ 原本更新

① 更新された文書と更新前文書のレシートを提示する。
② レシートの署名を検証し、有効性を確認する。
③ レシートより原本を特定し、新しい文書によるアーカイブを最新バージョンとして登録する。(改ざん防止のため、原本データのハッシュ値、タイムスタンプ情報が付加される。また、文書は暗号化される。さらにレシートが更新され、ログ情報が出力される。)

3.2.5 動作環境

■ 動作環境

□ 必要なソフトウエア

- Java J2SE v1.4.1
- Oracle8i Enterprise Edition R8.1.7
  ※アーカイブタイムスタンプにタイムスタンプサーバを用いる場合は「PKI サーバ／Carassuit タイムスタンプサーバ Ver1.0」が必要となる。

□ 必要なハードウエア

- Solaris8 （SPARC 版）
- CPU　：UltraSPARC III 440MHz 以上
- 実装メモリ　：256MB 以上

3.3 署名プラットフォーム（仮）（NTT コムウェア（株））

3.3.1 はじめに

企業間における EDI の増加など、ネットワークを介して電子文書をやり取りする機会が増加している。特に政府・自治体が推進する電子申請においては、発信者の身元と電子文書の完全性を確認するために PKI を用いた電子署名が適用されている。しかし従来の電子署名は、通常は数年程度である証明書の有効期限が切れると安全性が確認できなくなるという課題が存在する。弊社ではこの課題を解決するために、長期間にわたって電子文書を安全に保存するための製品を開発した。

この製品は、電子署名が付与された文書の有効性を長期間確認するための技術である XML 長期保存フォーマット（XAdES）を採用している。XAdES は WEB 技術における標準化団体である W3C が推奨する標準仕様であり、弊社が参加する電子商取引推進協議会においても提言されている。この製品によって、安全な長期保存が義務付けられている文書の電子化を可能としている。

3.3.2 製品の特徴

（1） 日本で初めて長期保存対応の XML 電子署名技術（XAdES）を実装し、電子文書の長期に渡る安全な保管を可能としている。

（2） サーバ署名に使う秘密鍵や証明書を HSM*1 で管理することにより、高い安全性を確保している。
＊1HSM：Hardware Security Module

（3） 電子署名・署名検証・XML 長期署名・タイムスタンプ付与といった豊富な機能を API*2 として提供しているため、 容易な他システム連携を可能としている。
＊2API：Application Program Interfase、JAVA ライブラリで提供

（4） タイムスタンプ付与に使用する時刻は、電子商取引推進協議会のガイドラインに従い、タイムスタンプ局から取得し、タイムスタンプ局は署名付与サーバ内部、または外部に構築が可能としている。

3.3.3 動作環境

(1) 署名付与サーバ

ハードウエア	
OS	Solaris8 2/02 rel.15 (64bit)
CPU	UltraSPARCIII 650MHz 以上
MEMORY	512MB 以上
HDD 容量	36GB 以上
ハードウェア機種	Sun Fire V120
ソフトウエア	
必要ミドルウェア	Luna CA3 ライブラリ (HSM 用)
	JRUN 4.0 SP1a
	Apache 1.3.27
	mod_ssl 2.8.12
	OpenSSL 0.9.6g
	XML パーサ (xalan-j2.3.1、xerces-j1.4.4)
	JP1/Cm2/Operations Assist Agent
	JP1/Extensible SNMP Agent
	JP1/Agent for Process Management
	Log4J 1.2.7

(2) 署名検証サーバ

ハードウエア	
OS	Solaris8 2/02 rel.15 (64bit)
CPU	UltraSPARCIII 650MHz 以上
MEMORY	512MB 以上
HDD 容量	36GB 以上
ハードウェア機種	Sun Fire V120
ソフトウエア	
必要ミドルウェア	JRUN 4.0 SP1a
	Apache 1.3.27
	mod_ssl 2.8.12
	OpenSSL 0.9.6g
	XML パーサ (xalan-j2.3.1、xerces-j1.4.4)
	JP1/Cm2/Operations Assist Agent
	JP1/Extensible SNMP Agent
	JP1/Agent for Process Management
	Log4J 1.2.7

(3) 証明書検証サーバ

ハードウェア	
OS	Solaris8 2/02 rel.15（64bit）
CPU	UltraSPARCIII 650MHz 以上
MEMORY	512MB 以上
HDD 容量	36GB 以上
ハードウェア機種	Sun Fire V120
ソフトウエア	
必要ミドルウェア	JRUN 4.0 SP1a
	Apache 1.3.27
	mod_ssl 2.8.12
	OpenSSL 0.9.6g
	XML パーサ（xalan-j2.3.1、xerces-j1.4.4）
	JP1/Cm2/Operations Assist Agent
	JP1/Extensible SNMP Agent
	JP1/Agent for Process Management
	Log4J 1.2.7

(4) HSM

ハードウェア	
ハードウェア機種	LUNA CA3
	LUNA Xpplus（アクセラレータ）

3.3.4 機能概要

コンポーネント名称	
署名付与サーバ	・秘密鍵、証明書管理機能（HSM アクセス機能など）
	・署名付与機能（XAdES 形式署名付与機能）
	・署名対象物への長期保存用タイムスタンプ付与機能
	・長期保存タイムスタンプ延長機能
署名検証サーバ	・署名検証機能
証明書検証サーバ	・証明書検証機能
	・XAdES 形式のタイムスタンプ用鍵の証明書検証要求

3.3.5 機能構成図

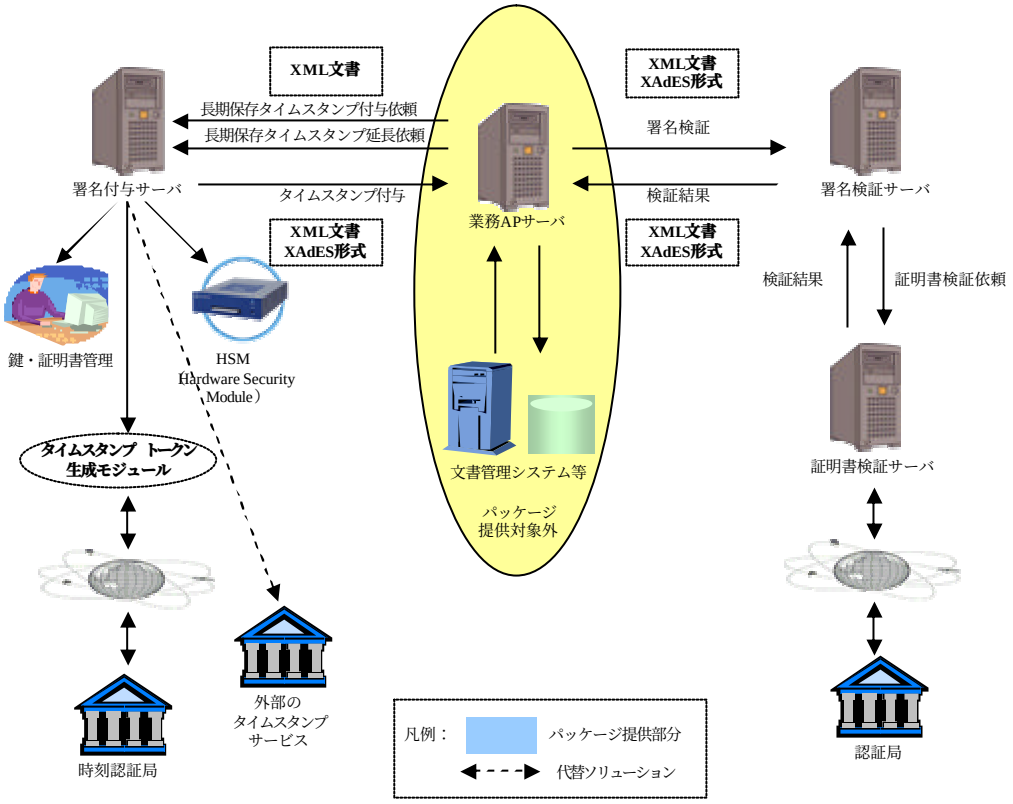

図 3-3 署名プラットフォーム（仮）システム構成

3.4 セキュアな長期原本保管システム（NTT コミュニケーションズ（株））

3.4.1 はじめに

行政、企業等における業務の電子化が進展し、電子データとして保管される情報が増加している。セキュアな長期原本保管システムは、電子文書の長期保管における改竄、漏洩、消失等を防止し、重要データ、機密データであっても長期間、安全に保管することができる。本システムでは、秘密分散技術により電子文書の原本性（完全性、機密性）、保存性を確保した電子文書の長期保管（ロングタームセキュリティ）を実現している。

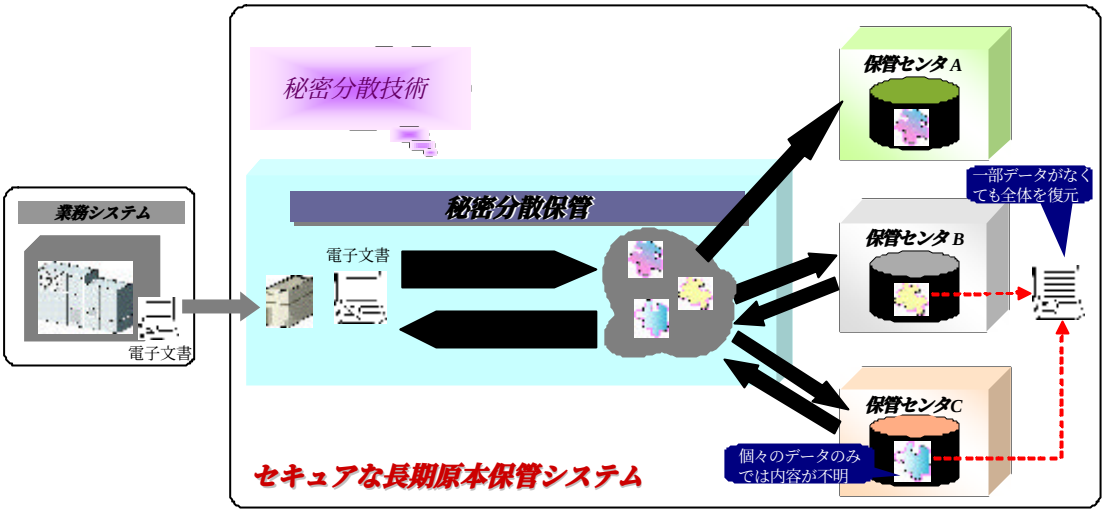

図 3-4 システム概要

3.4.2 特徴

3.4.2.1 保存性の確保

電子文書の長期保管においては、長期に渡って電子文書の消失や破壊がないよう保管する必要があるが、単にバックアップをとるだけでは盗難のリスクを増加させてしまう。

本システムがベースとする秘密分散法とは、秘密情報を安全に保管するために、いくつかの分散情報に符号化し、特定の分散情報の集合に対してのみ秘密情報の復元を可能とする手法である。このうち（k，n）閾値法は、秘密情報を n 個の分散情報に符号化し、そのうちの任意の k 個の分散情報からは元の秘密情報を復元できるが、k－1 個以下の分散情報からは秘密情報に関する情報を一切得ることができないという符号化方法である。

本システムでは、大容量データでも高速処理可能な独自の秘密分散アルゴリズムを実装しているため、保管対象データの制約はない。また、個々の分散情報は異なるサーバに保管されるため、災害等で一部の分散情報が取り出せなくなっても、元のデータが復元可能である。

3.4.2.2 機密性の確保

上記のように、符号化され分散して保管される個々の分散情報からは、元のデータの全体が推定されることはなく、その一部であっても推定することはできない。元のデータを復元するには必ず一定個数の分散情報が必要であり、年々コンピュータの計算能力が向上しても、一定個数に満たない分散情報からは元のデータを復元することができない。暗号化ではないので鍵管理が不要であり、計算量的な解読困難性に依存せず機密性の確保が可能である。

3.4.2.3 完全性の確保

本システムでは図 3-5 のようにハッシュ値を用いた方法により、復元されたデータの完全性を確保している。タイムスタンプの付与を繰り返すことなく、長期に渡り完全性の確保が可能である。

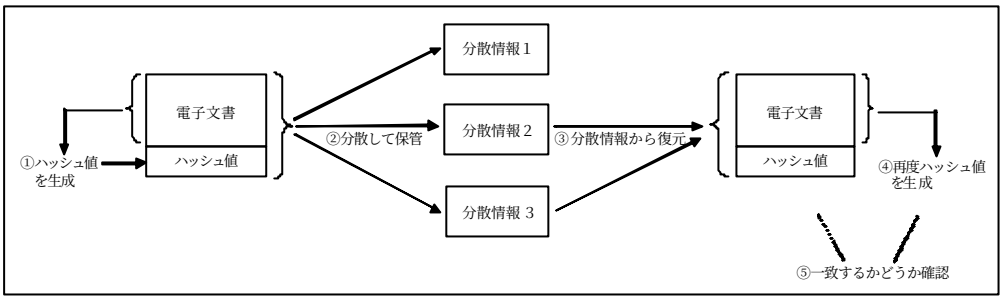

図 3-5 完全性確保方法の概要

3.4.2.4 動作ログ管理機能

システムの動作ログ、保管データへのアクセス履歴は、改竄検知可能な状態で保管し、証拠性、説明性を高める実装をしている。

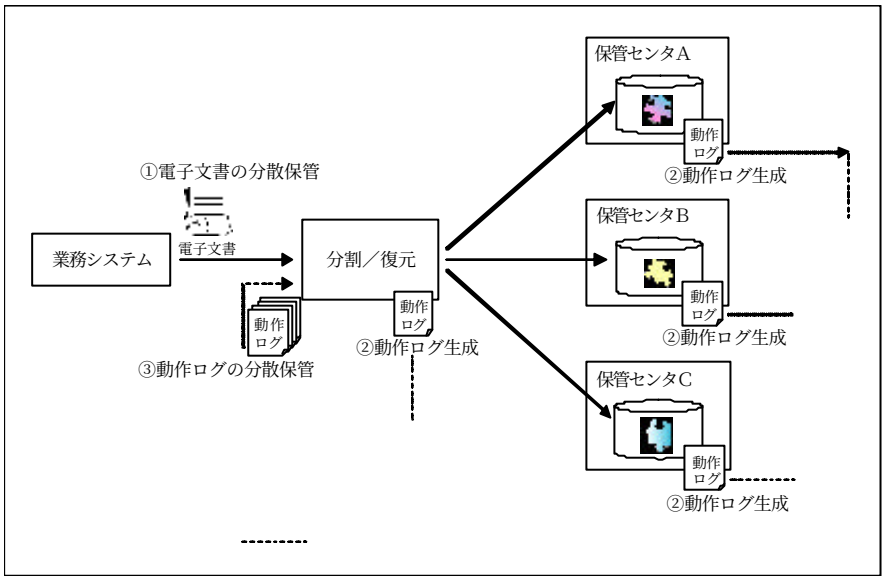

図 3-6 動作ログ管理の概要

3.4.2.5 タイムスタンプ

本システムでは、日本標準時にトレーサブルな時刻情報を用いて、電子文書にタイムスタンプを付与して保管する。

3.4.2.6 電子署名文書の長期保存

上記の通り、本システムでは電子文書にタイムスタンプを付与し、原本性（完全性、機密性）、保存性を確保した長期保管が可能である。電子署名文書の場合は、本システムの利用者が必要な検証情報を収集し、電子署名文書とともに本システムに保管することにより、長期にわたって安全な保管が可能である。

3.4.3　利用事例

秘密分散技術の特徴を生かし、医療情報、CAD データ等の重要な情報を長期間、安全に保管することができる。大容量データのバックアップが不要となり、社内で管理していた機密データまで安心して社外に保管することも可能となる。

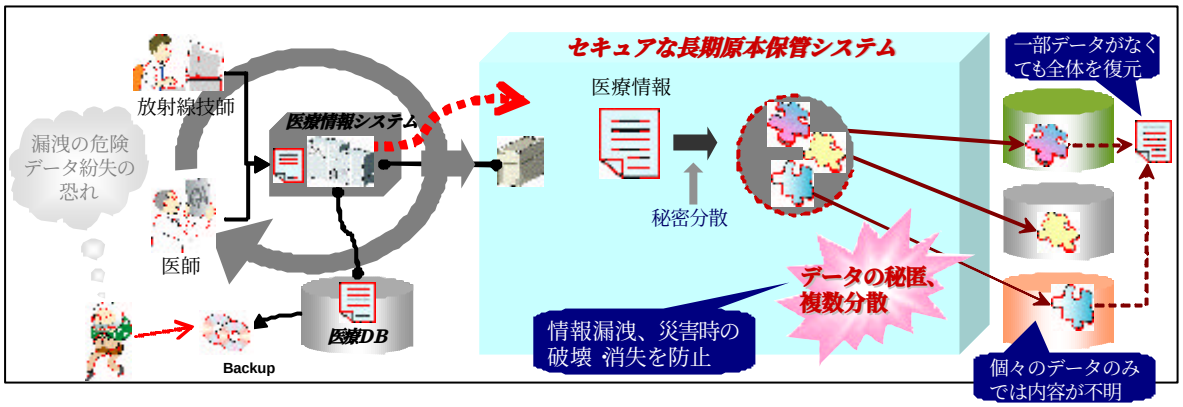

図 3-7 事例：医療情報システム

3.4.4　今後の展開

本システムは、重要データの長期保管が必要となる様々な分野で利用することができる。

電子契約分野・・・契約書・発注書の保管、顧客情報の保管

電子行政分野・・・電子申請、電子調達、電子公文書等の保管

電子医療分野・・・プライバシー情報、検査情報等の保管

3.5　原本性保証システム DP1/Proofbox2（(株) 日立製作所）

3.5.1　はじめに

原本性保証システム DP1/Proofbox2 は 2000 年 3 月に旧総務庁共通課題研究会がまとめた「インターネットによる行政手続き実現のために」に書かれてある原本性確保要件に準拠した製品である（図 3-8 参照）。

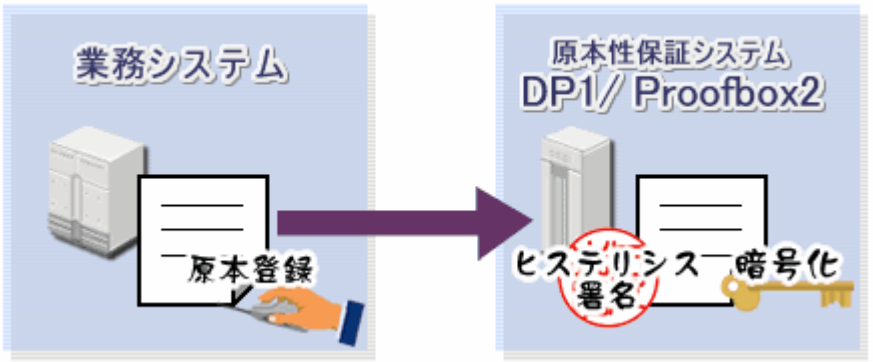

図 3-8 原本性保証システム DP1/Proofbox2

3.5.2　原本性確保要件への対応（完全性・機密性・見読性の確保）

図 3-9 に DP1/Proofbox2 の機能概要を示す。この製品では以下の機能・技術にて原本性確保要件（完全性・機密性・見読性の確保）を満たしている。

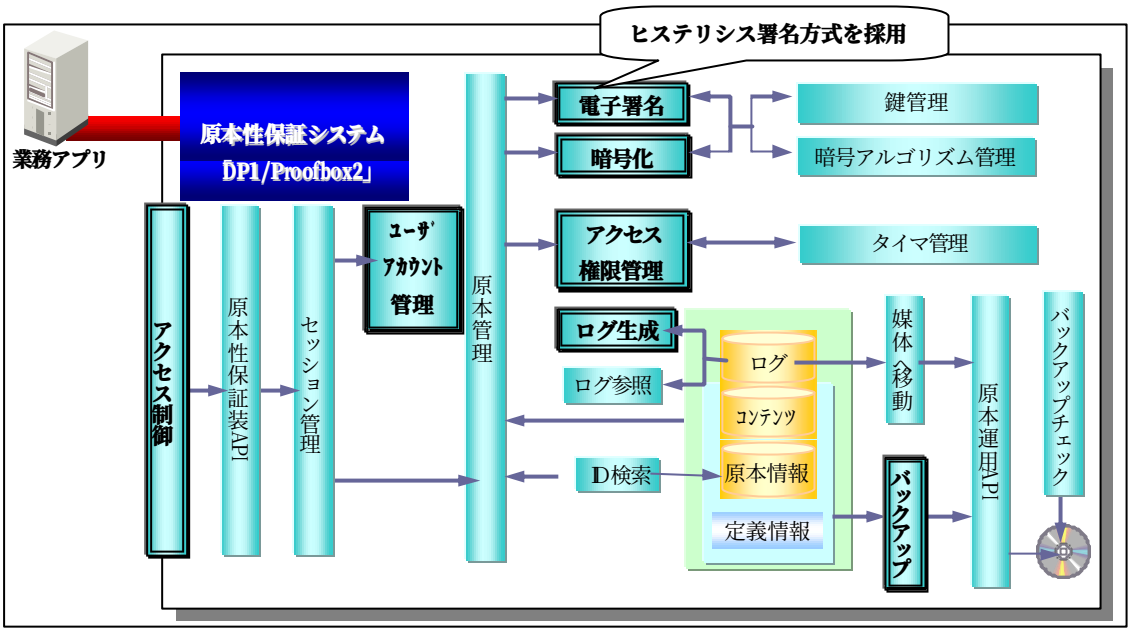

図 3-9 原本性保証システム DP1/Proofbox2 機能概要

3.5.2.1　完全性の確保

(1) 電子署名

原本に対して公開鍵暗号技術に基づく電子署名を付与することにより、電子文書が確定的なものとして作成された以降の改ざん事実の有無を検証可能にする。しかし、電子署名作成に利用する暗号アルゴリズムは計算機パワーの増大により脆弱化する恐れがあり、公開鍵証明書には有効期限が設けてある。また、秘密鍵の漏洩などにより公開鍵証明書は失効する場合があるため、長期保存という観点では電子署名だけでは完全性を保証できない。

DP1/Proofbox2 ではヒステリシス署名技術を利用し、署名間に連鎖構造を持たせ、署名の安全性、有効性を長期間維持することができる。ヒステリシス署名技術については 2.7 節にて詳細に説明した。

(2) 書換え、消去不可

原本として登録された文書に対しては一切書換えを許さず、文書内容の更新はすべて新しい版の作成として行う。（更新前の版を残し、更新日時、更新ユーザ等の履歴を記録する。）また、文書の保存期限を設定し、期限前の削除はできないように制御する。

(3) 原本への操作履歴管理

原本として登録した電子文書に対し、どのようなアクセス（登録、更新、参照、削除、署名検

証）が行われたかを自動的に履歴を記録する。

(4) バックアップ・リストア

バックアップ・リストア機能により、記録媒体の経年劣化等による電子文書の消失及び変化を防ぐ。

3.5.2.2 機密性の確保

(1) アクセス制御

DP1/Proofbox2 へのアクセスは専用 API 経由のみとし、それ以外のアクセスは受け付けない。また、不正アクセスを防止するため、システムを利用するクライアントにアカウントを発行し、ID・パスワードによるアクセス制御を行い、ログイン／ログアウトの履歴も記録する。さらに、原本に対するユーザの操作権限や使用しているファイル操作権限を細かく設定できる。

(2) 暗号化

万が一、原本が盗難、盗み見等をされた場合でも情報が漏洩することを未然に防止するため、原本を暗号化し登録する。

3.5.2.3 見読性の確保

DP1/Proofbox2 では登録する原本のデータ形式に制限はない。API 経由により原本は登録された形式のまま、保管され、データ取得時にはそのままの形式で取得が可能である。文書管理ソリューション統合文書管理システム DP1/episimo との連携により、原本の参照、紙による印刷が可能である。

3.5.3 長期保存への対応

DP1/Proofbox2 はヒステリシス署名を用いることにより電子文書の長期保存を実現している。

ヒステリシス署名では、ｎ番目の署名を生成する際、署名対象データのハッシュ値単独ではなく、これにｎ－１番目の署名データを結合したデータを秘密鍵で暗号化することによって署名を行う（図 3-10 参照）。

このようにすることで署名データ間に依存関係ができるため、秘密鍵の漏洩等により攻撃者がある文書を改ざん（署名を偽造）しようとした場合、その署名だけでなく、その署名に依存する署名（その署名以降に生成されたすべての署名）をも偽造しない限り、署名履歴の検証により改ざんの事実が判明してしまう。

攻撃者による署名履歴自体の改竄に対しては、署名履歴を信頼できる第三者機関に寄託する、または署名履歴の一部を公表する等により、トラストアンカーを形成する。トラストアンカー及び署名履歴の検証により、長期に亘って署名の非改ざん性を証明することができる。（図 3-11 参照）。

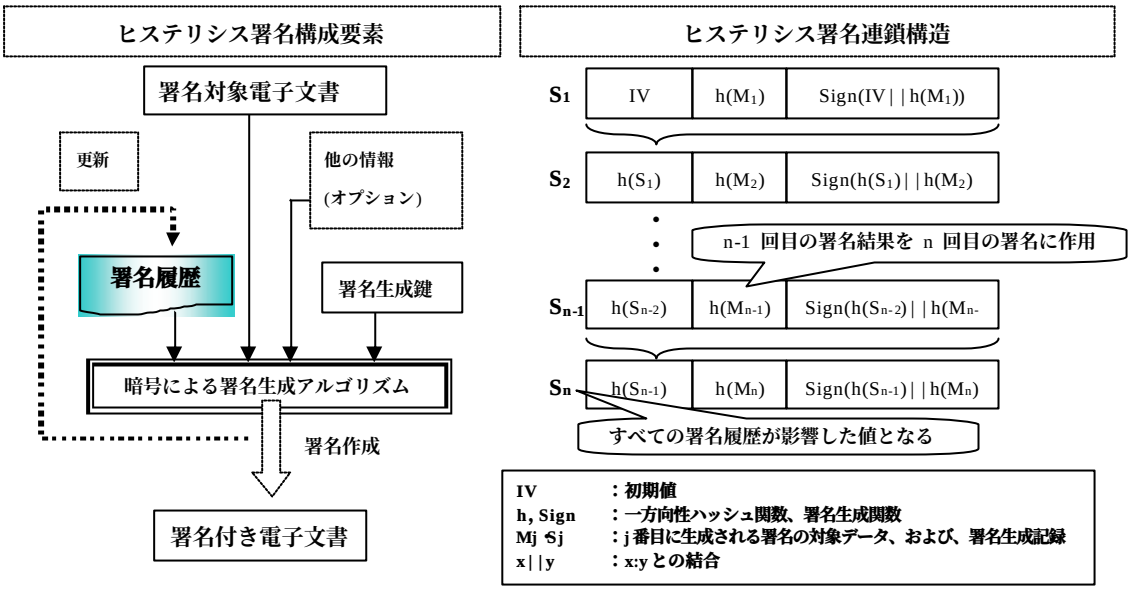

図 3-10 ヒステリシス署名概要

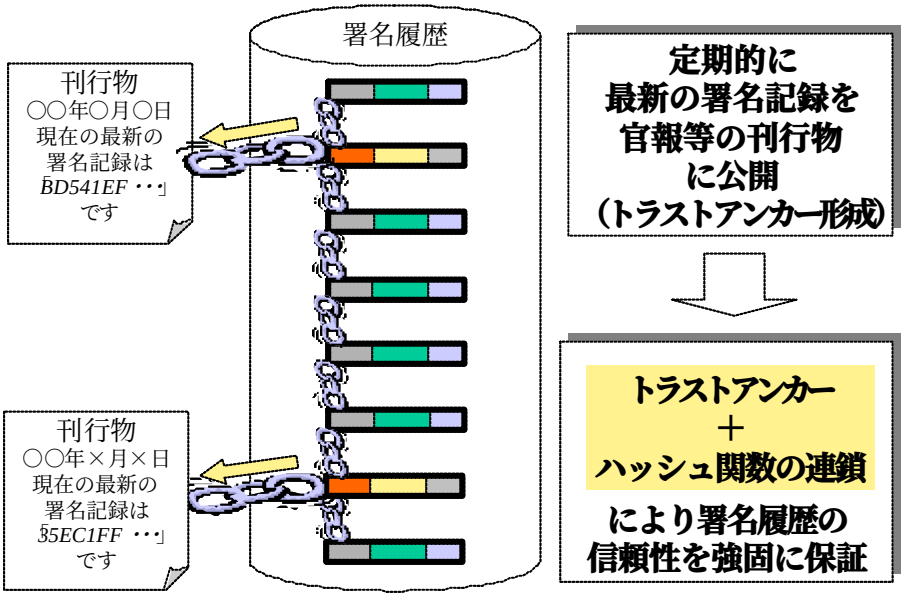

図 3-11 ヒステリシス署名のトラストアンカー形成

3.5.4 DP1/episimo との連携について

DP1/Proofbox2 は行政向け文書管理パッケージ製品である統合文書管理システム DP1/episimo との連携を実現し、行政文書の原本性保証を可能にしている。(図 3-12 参照)。

図 3-12 DP1/episimo での原本性保証システム連携画面

3.5.5 使用事例

DP1/Proofbox2 は統合文書管理システム DP1/episimo とともに自治体等において文書管理システムにおける原本性保証システムとして使用されている。また某自治体では電子申請システムにおける申請文書の長期原本性保証のために DP1/Proofbox2 が使われている。

3.5.6 今後の展開

「e-Japan 戦略」が発表されて以降、官庁を中心に、インターネットを利用した電子申請やワンストップサービスなど、行政サービス向上のためのシステム構築が加速されている。これらのシステムでは、電子文書の原本性確保を必須要件としている。さらに「e-Japan 戦略Ⅱ」により、多くの民間保存文書の電子保存が認可される可能性が高くなっており、それに伴って原本性の確保が要求される文書範囲が拡大し、DP1/Proofbox2 の利用ニーズが高まるものと考える。

3.6 三菱署名有効性延長システム MISTYGUARD<EVERSIGN>（三菱電機（株））

企業などでは、ワープロソフトなどで作成したデジタル文書をデータのまま保管する事やインターネット経由で送受信する事は、業務効率化や費用削減などを目的に普及しており、紙の使用量削減を通じた環境負荷低減の手段としても注目されている。

行政手続のオンライン申請等を実現する「公的個人認証サービス」が開始され、今後、法的に長期保存が義務づけられている文書についても電子化の要求が強まると予想される。

ところが、電子的に署名（捺印）された文書でも、安全性の観点から署名に用いる証明書の有

効期限を 1～数年以内とすることが事実上の基準とされており、文書によっては法的に 10 年以上の保管が義務づけられている事との間にギャップがある。

EVERSIGN はこうした課題に応える製品として、PKI を基盤とし、タイムスタンプ技術と署名有効性延長生成・検証技術を応用した、デジタル署名文書の真正性を長期に保証するソフトウェア製品である。

3.6.1 署名有効性延長システム MistyGuard<EVERSIGN>の特長

- ◆ 署名フォーマットとして標準形式（RFC3126）に準拠した形式を採用
  電子商取引推進協議会（ECOM）が 2003 年 3 月に発表した電子署名文書長期保存に関するガイドラインで推奨する、RFC3126 フォーマットに準拠。
- ◆ 有効性延長済電子文書の真正性の検証が容易
  延長サーバ側での検証に加え、クライアント側で有効性検証を行うための検証ライブラリを提供。
  →申請者又は第三者が、任意に検証を実施するシステムの開発が可能。
- ◆ 外部のファイリングシステムとの連携が可能
  文書保管用の外部ファイリングシステムとの連携用 I/F を提供。
  →連携用モジュールを用意することで､任意のファイリングシステムとの連携が可能。

3.6.2 EVERSIGN システム構成

EVERSIGN システム（サーバとクライアントライブラリ）は、単独で使用するものではない。署名付き電子文書の生成、流通、保管などを行うユーザ側の電子文書管理システムが存在し、EVERSIGN は、そのシステムに長期文書保管、署名有効性延長、検証などの機能を提供するサブシステムとして組み込まれて使用されることを、前提としている。さらに EVERSIGN の利用には、タイムスタンプ発行局（TSA）が必要である。また保管すべき電子文書が多く高速なファイルアクセスが必要となる場合や、既存の文書保管ファイリングシステムを流用するような場合には、ファイリングシステムと連携する構成とする必要がある。

次の図は、EVERSIGN を組み込んだ典型的なシステム構成例を示したものである。この例では、文書管理を行うサーバ（ユーザアプリケーション）を中心として、署名文書を作成したり、有効性の検証を行う文書管理サービス利用者、EVERSIGN サーバ、TSA、およびファイリングシステムによって、システムが構成されている。

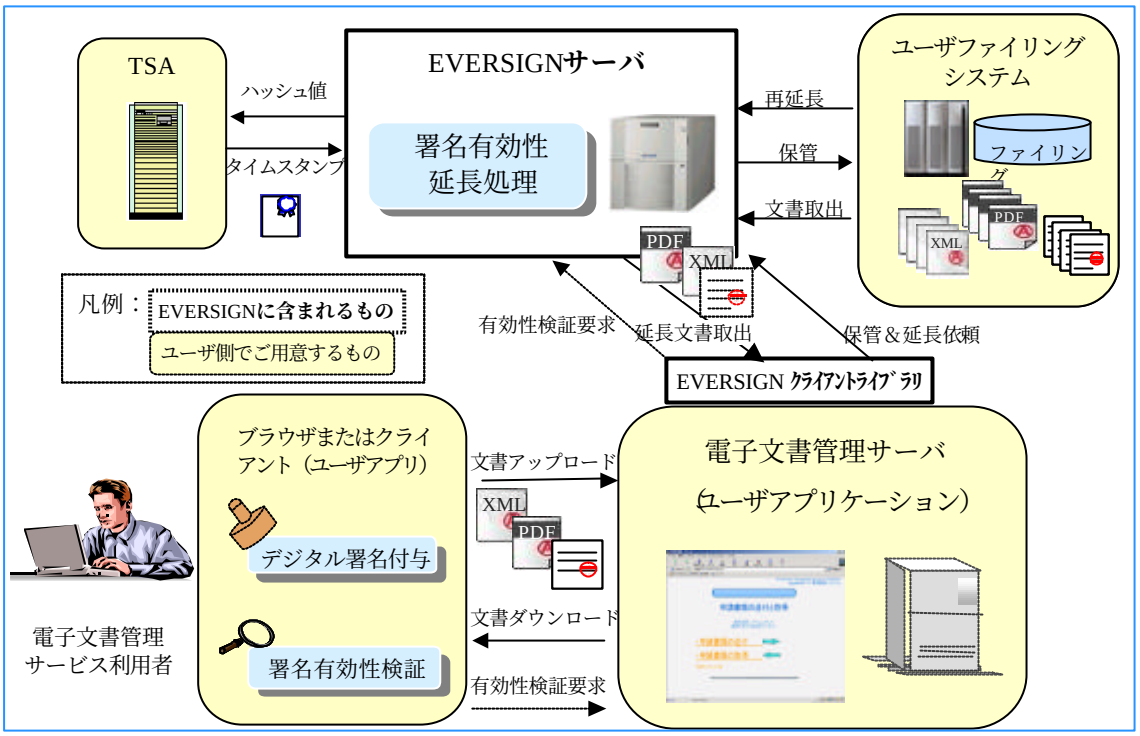

図 3-13 システム構成例

3.6.3 EVERSIGN の機能

署名有効性延長システム MistyGuard<EVERSIGN>は、次の三つのコンポーネントを提供している。

【1】 署名有効性延長、および有効性延長済み文書の保管、取出しなどを行う EVERSIGN サーバ

【2】 EVERSIGN サーバの管理機能を提供する EVERSIGN サーバ管理ツール

【3】 ユーザアプリケーションから EVERSIGN サーバに対して署名文書の登録、有効性延長済み文書の取出しなどを要求する EVERSIGN クライアントライブラリ（開発キットを含む）

上記のうち、サーバ及びクライアントの機能について記す。

(1) EVERSIGN サーバ

EVERSIGN サーバは、署名されたファイルやデータを受け取り、その署名の有効性延長処理、および延長情報と元文書を保管する機能を提供する。延長した文書や延長情報、元文書は、EVERSIGN サーバ自身のデータベースか、外部のファイリングシステムに保管される。有効性延長処理で使用されるタイムスタンプは、外部のタイムスタンプ発行局にアクセスして取得する。

EVERSIGN サーバは、EVERSIGN クライアントライブラリから送られてくる次のような要求を処理し、応答を返す。

- ◆ 電子署名文書ファイルの登録
- ◆ 保管文書の取り出し

◆ 登録したファイルの削除

◆ 延長署名文書の有効性検証

署名の有効性の再延長処理は、設定されたポリシー情報に応じて、EVERSIGN サーバが自動的に実行する。

以上の機能の他、EVERSIGN サーバへクライアントライブラリからの要求受付とその処理の結果、内部動作の一連の動作と結果などについて、ログを出力する機能がある。

(2) EVERSIGN クライアントライブラリ

EVERSIGN クライアントライブラリは、ユーザアプリケーションから、EVERSIGN サーバに対して、署名されたファイルやデータの登録・削除・検証を要求し、結果の応答を受け取り、解析する機能を提供する。EVERSIGN サーバの署名有効性延長機能を利用するには、EVERSIGN クライアントライブラリをリンクし、ユーザアプリケーションからライブラリの API を呼出す。

EVERSIGN サーバとの通信プロトコルは、HTTP を使用する。EVERSIGN サーバへのリクエスト、及びレスポンスのフォーマットは、DVCS をベースにしたオリジナル形式を採用している。

3.7 電子文書流通プラットフォーム SecurePod（(株) NTT データ）

SecurePod（セキュアポッド）は、電子文書の生成から流通、保管、廃棄までのライフサイクルをトータルでサポートするプラットフォームサービスである。このサービスを利用することで、電子契約、申請、調達システムといった、重要文書を安心してインターネット上で交換するシステムを、安全に実現することができる。

3.7.1 はじめに

重要文書の電子化を支えるための技術要素として、電子認証、原本性検証、アクセス管理、セキュリティ対策などが挙げられるが、これらの技術はそれぞれ個別の製品・サービスとして提供されており、実際に重要な電子文書の流通・交換を実現するためにはその度にこれら製品を選定し統合するといった作業が必要となる。このような現状を受けて、重要電子文書の流通・保管に必要となる機能をワンストップで提供する電子文書流通プラットフォーム SecurePod を開発し、現在 ASP サービスとして提供している。

3.7.2 特徴

SecurePod では、デジタル署名、セキュア配送、電子文書保管、原本性検証の４機能を電子文書流通のためのプラットフォームとして提供している。４つの機能は、利用者のニーズに合わせて自由に選択し、業界や分野にあわせ個別の Web アプリケーションを構築することが可能になっている。また、Web アプリケーションのテンプレートを用意し、Web アプリケーションを簡単に作成することもできる。その他に、お客様の既存システムと連携して動作させるために、機能ごとの API を準備しシームレスに連携することが可能となっている。

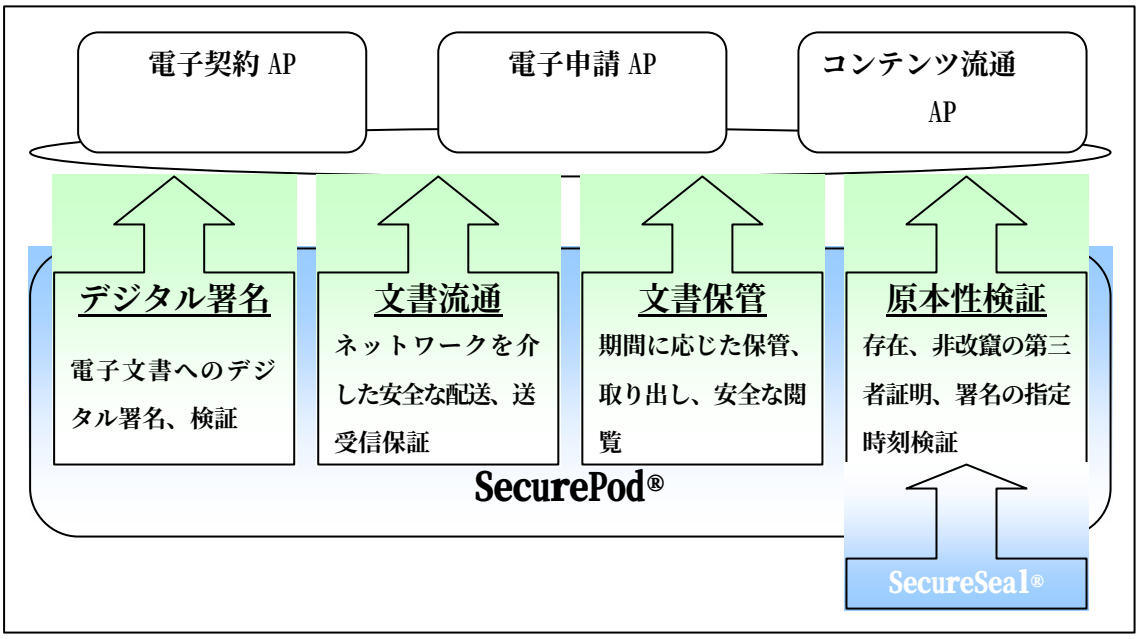

図 3-14 機能概要

SecurePod は J2EE プラットフォーム上に構築されており、各 Web アプリケーションはリモートから SecurePod の各機能を利用できる。SecureSeal への接続も他の機能同様にリモートインタフェースを提供し、Web アプリケーションは SecurePod を介して SecureSeal へ接続できるようになっている。各 Web アプリケーションは、JSP 等による動的コンテンツや、Servlet 等へ接続できる API をユーザへ提供する。

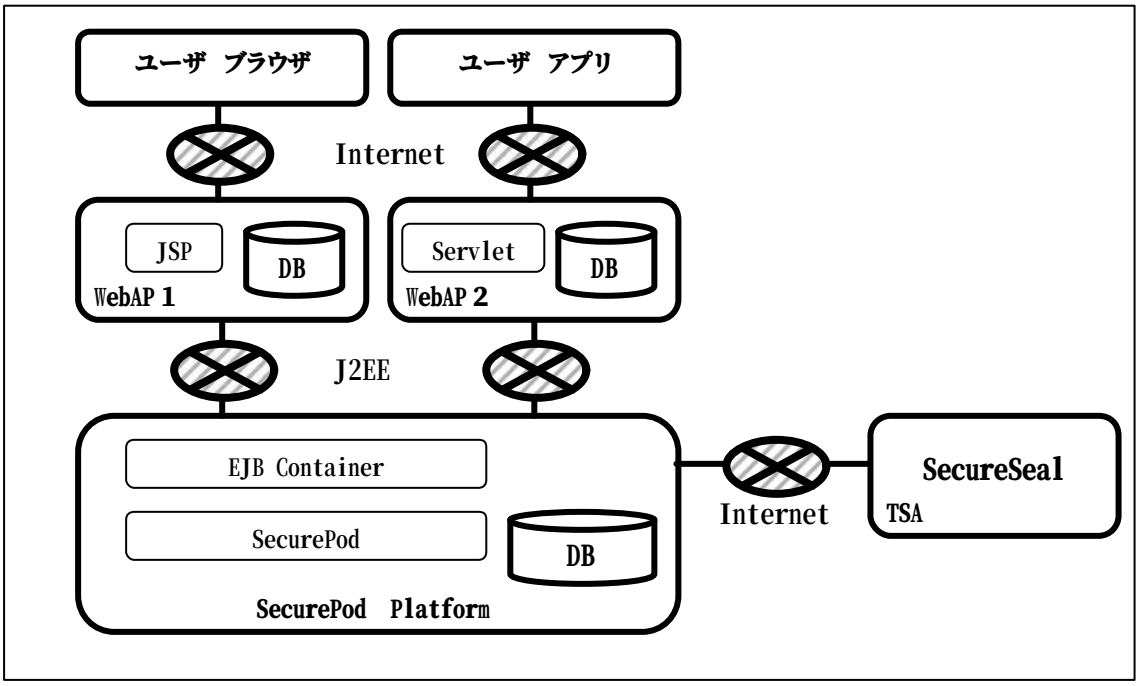

図 3-15 システム構成図

(1) デジタル署名

重要文書の電子化フォーマットとして、長期にわたり見読性を確保できる PDF を使っている。PDF は一つの文書ファイルに複数のデジタル署名を埋め込むことができ、契約文書などの重要文書を電子化し流通させることに適している。SecurePod では、サーバ上にある PDF ファイル内に付与されたデジタル署名データを独自技術により抜き出し、文書の改ざん検証、対応認証局発行の公開鍵証明書かどうか、その公開鍵証明書が有効期間内かつ失効していないかどうかを検証局（VA）へ問い合わせている。また、デジタル署名アプリケーションの開発モジュールとして、WindowsCOM コンポーネントを提供している。このコンポーネントを利用することにより PDF へデジタル署名を行うアプリケーションを独自に開発することが可能となる。

下記に対応製品を記す。

表 3-2 コンポーネント構成

PDF	PDF1.4、1.5
PDFWriter	Acrobat®5、Acrobat®6（Adobe 社）
署名ハンドラ	Document Signer Plug-in 2.0（VeriSign 社）
対応認証局	CECSIGN®（株式会社コンストラクション・イーシー・ドットコム） TDB 電子認証サービス TypeA（株式会社帝国データバンク）
署名アルゴリズム	md5withRSA（1024 ビット） SHA-1withRSA（1024 ビット）
PDF 署名コンポーネント OS	WindowsR98、WindowsRMe、WindowsR2000、WindowsRXP
ブラウザ	IE5 以上

デジタル署名に用いる署名鍵と公開鍵証明書は、実社会における印鑑、印鑑登録証に相当し厳重に保管する必要がある。PC の HDD で保管した場合、HDD が壊れた際に署名鍵と公開鍵証明書も消失する危険性がある。そのため SecurePod では、PC から物理的に切り離し、USB ポートに直接接続が可能で、耐タンパー性に優れた eToken を署名鍵と公開鍵証明書の格納デバイスとして推奨している。

(2) セキュア配送

インターネット経由での安全な電子文書交換・配送機能を提供する。SSL による暗号化通信はもちろん、電子文書送受信に関するステータス管理機能を備え、各トランザクションの結果を対象ユーザへ通知する機能を持つ。また、各トランザクションに関してアクセスログを取得し、否認防止を行っている。オプションで分割ダウンロード、レジューム機能、ダウンロード・アップロードの完全性確認機能も提供する。完全性確認機能は、クライアントが送信する前の文書のハッシュ値とサーバへ到達した後の文書のハッシュ値を比較し、一致すれば完全性が保たれていると判断する。

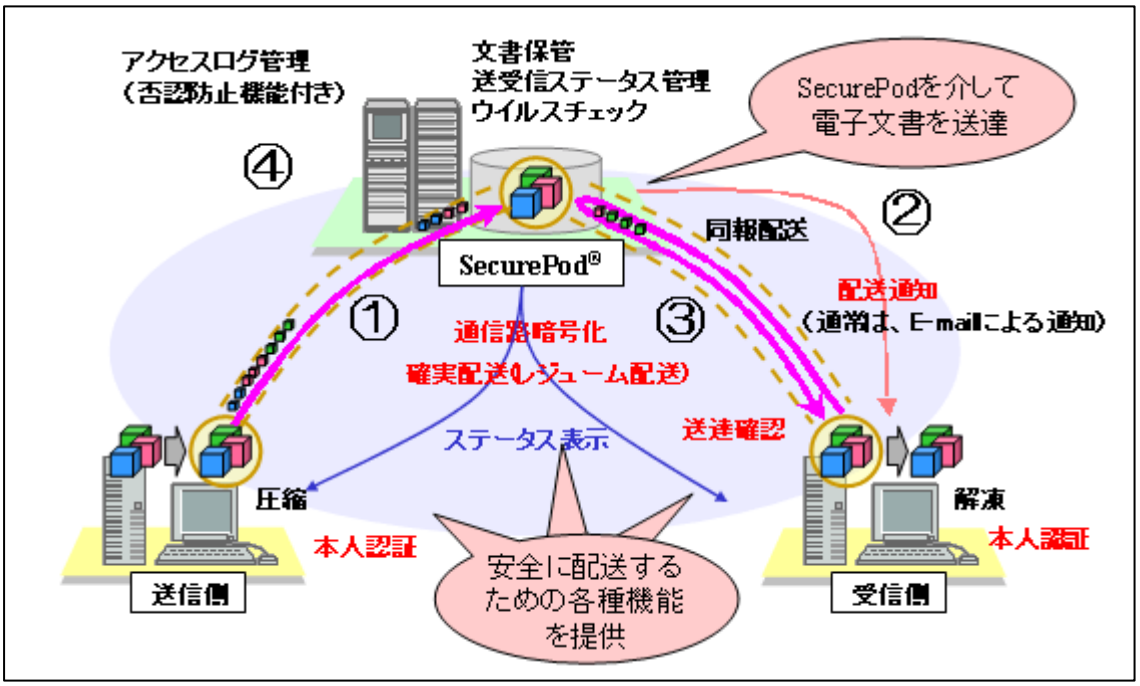

図 3-16 セキュア配送概要図

(3) 電子文書保管

作成された電子文書をユーザと関連付けし、他のユーザから参照できないようにアクセス管理をしている。保管文書は、生体認証による入室管理や不正アクセスに対するセキュリティ監視が徹底されたデータセンタで長期間管理・保管し、毎日バックアップを行い物理的に離れた２拠点でバックアップデータの保管も行っている。保管している電子文書は、Web の検索画面を介していつでも閲覧及びダウンロードが可能である。各トランザクションのログにはデジタル署名を行い、ログの改ざん防止を行っている。

(4) 原本性検証

電子文書証明サービス SecureSeal を利用し、重要な電子文書（甲乙のデジタル署名が添付された電子契約書など）の保管後の非改ざん証明と時刻証明が可能な「原本性保証サービス」を提供している。

SecurePod を利用する大きな特徴として、法廷保存期間に対応したデジタル署名付き文書の長期保管･長期保証がある。契約文書の法廷保存期間は 7 年であることに対し、公開鍵証明書の有効期間は最長でも 5 年間である。現状認証局が設定する有効期限は 2 年程度である。SecurePod では、電子文書と過去の有効性検証に必要な材料（CRL や Root 証明書など）を長期保証が可能な SecureSeal に登録し保管することで、保管されたデジタル署名文書が、使用された公開鍵証明書の有効期間内に署名されたことを検証することができる。この機能を「指定時刻検証サービス」と呼んでいる。この機能は、平成 13 年度の認証・公証 WG-3 において、長期の原本製検証のための要件を定義した「電子署名文書長期保存に関するガイドライン」を満たしている。

原本性保証サービスとデジタル署名の指定時刻検証サービスは、エンドユーザのニーズに

より選択でき、これらの機能を組み込んだ Web アプリケーションのインタフェース（ブラウザ上のボタン等）から利用することが可能である。

下記に SecurePod における長期保証の仕組み（指定時刻検証サービス）を記す。

手順０．準備

認証局（CA）が発行する Root 証明書、CRL（失効リスト）を更新毎に SecureSeal へ登録し保管する。保管文書と利用したユーザの公開鍵証明書を SecureSeal へ登録し保管する。PDF の場合、公開鍵証明書は署名データと共に PDF 内部に埋め込まれている。

ユーザから過去の時刻を指定し、その時点デジタル署名及び公開鍵証明書が有効であったかどうかを検証要求があった場合の手順は下記の通りである。

手順１．保管物（原本、ユーザの公開鍵証明書、CA の Root 証明書、指定時刻における CRL）の原本性検証を行う。必要に応じて CA のリンク証明書の原本性も検証する。

手順２．手順１で原本性を示した Root 証明書を用い、指定時刻における CRL に付与された CA のデジタル署名を検証する。これにより指定時刻における CRL が改ざんされてなく、CA から発行されたことを示す。

手順３．手順１で原本性を示した Root 証明書を用い、ユーザの公開鍵証明書に付与された CA のデジタル署名を検証する。これによりユーザの公開鍵証明書が CA から発行されたことを示す。

手順４．ユーザの公開鍵証明書のシリアル No が CRL に登録されていないことを確認する。これによりユーザの公開鍵証明書が失効していなかったことを示す。

手順５．保管時の SecureSeal タイムスタンプがユーザの公開鍵証明書の有効期間内にあることを確認する。これによりユーザの公開鍵証明書が有効期間内であったことを示す。

手順６．最後に、保管文書に付与されたユーザのデジタル署名を、手順５で有効性を示したユーザの公開鍵証明書で検証する。これにより保管文書に付与されたデジタル署名が有効な公開鍵証明書を用いて行われたことを示すことができる。

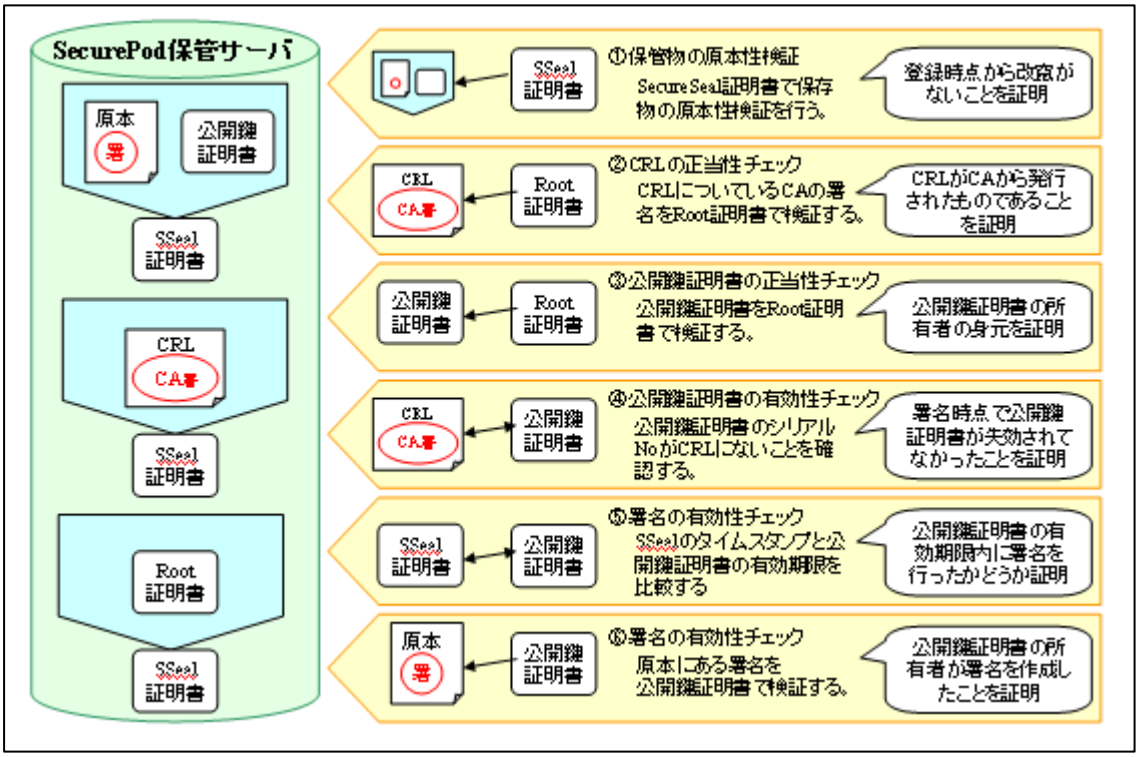

図 3-17 デジタル署名の指定時刻検証処理概要図

3.7.3 使用事例

SecurePod の機能を利用した Web アプリケーションとして「電子契約サービス CECTRUST」がある。CECTRUST は、IT 書面一括法の対象の一つである建設業法と電子署名法の施行をうけ、国土交通省から出されたガイドライン（建設業法施行規則第 13 条の 2 第 2 項に規定する「技術的基準」に係るガイドライン（平成 13 年 3 月 30 日））を満たし、SecurePod をインフラとしたセキュアな電子契約アプリケーションとして構築された。平成 14 年 3 月にサービスを開始し、現在までに、ゼネコン、ガス、通信業といったあらゆる業種のお客様が、取引先との契約書を電子化し、合計で年間 10000 件程度利用されている。

エンドユーザが CECTRUST を利用する大きなメリットの一つとして印紙税が不要になることがあげられる。電子データでやり取りされた契約文書には印紙税が不要であることを財務省に確認しており、契約行為におけるコスト削減に大きく寄与している。また、ASP サービスのため利用するための初期コストが抑えられ短期間で利用開始が可能となっている。その他にも、契約文書が電子化されたことにより、調達業務から契約業務までシームレスに連携することができ、契約行為に関する人件費や契約文書の輸送費なども削減可能になっている。

CECTRUST は、国土交通省の電子入札対応証明書である帝国データバンク発行の Type-A が使用可能になっている。実印に相当する電子証明書を何枚も保持しなくてもよく、電子入札用の証明書を電子契約に使用することができる。

CECTRUST では、ブラウザでのユーザインタフェースのほかに、アプリケーション開発用のコンポーネントも用意している。前述の PDF 署名コンポーネントと組み合わせることにより、大量の電子文書に対して、一括でデジタル署名、送受信が可能となっている。このコンポーネントを利用したアプリケーションも販売され多くのユーザに利用されている。

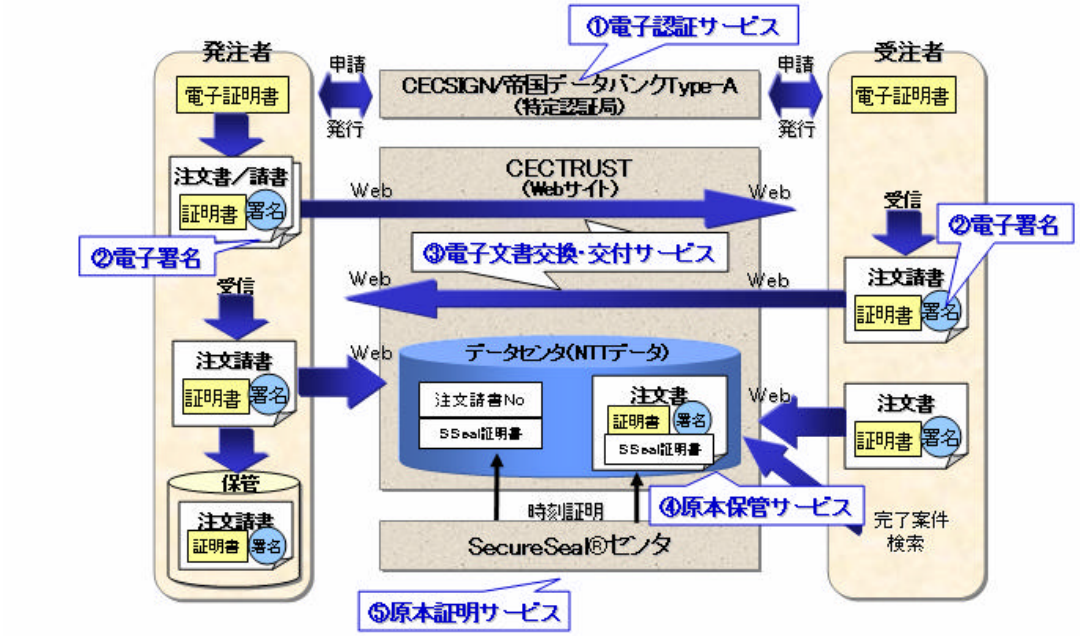

図 3-18 CECTRUST 業務フロー

3.7.4 今後の展開

ビジネス的な展開では、あらゆる電子文書のライフサイクルにあわせたアプリケーションを準備していく。例えば、SecurePod で行っているワークフロー制御を強化し、請求書や明細書などのユーザからユーザへ一方向に流通させるアプリケーションや、ユーザが保管と原本性保証のみを利用するような電子文書保管アプリケーションに対応させる。

技術的な展開では、既存システムや周辺技術との連携を強化することが考えられている。例えば、既に社内システム等が整っているユーザが社内システムから SecurePod へシームレスに接続できるように、接続インターフェイスの違いを吸収するモジュールを用意することである。XML 文書の流通も検討しており、システム間連携を強化していく。その他、SingleSignOn やコンテンツ保護技術（Digital Rights Management）を取り入れ、SecurePod 上によりセキュアに、重要情報保護を目的としたアプリケーションを構築できる。

付属書A　参考文献

今回検討を進めるにおいて参考とした文献を以下に示す。文書によってはバージョンが更新されることがあるので、注意を要する。(URL は 2004 年 3 月現在)

1. A. Shamir, “How to Share a Secret”, Communications of the Association for Computing Machinery, vol.22, no.11, pp.612-613 (Nov. 1979)
2. “Time-stamping services Part 1: Framework” ,ISO/IEC 18014-1:2002
3. “Time-stamping services Part 2: Mechanisms producing independent tokens”, ISO/IEC 18014-2:2002
4. “Time-stamping services Part 3: Mechanisms producing linked tokens”, ISO/IEC 18014-3:FCD
5. “ Internet X.509 Public Key Infrastructure Time-Stamp Protocol (TSP) ”, C.Adams, P.Cain, D.pinkas, R.Zuccherato, RFC3161,Aug2001, http://www.ietf.org/rfc/rfc3161.txt
6. “ Internet X.509 Public Key Infrastructure Time-Stamp Protocol (TSP) ”, C.Adams,P.Cain, D.pinkas, R.Zuccherato, RFC3161,Apr2002
7. “Policy Requirements for Time-Stamping Authorities (TSAs)”, RFC3628, D.Pinks, N.Pope, J.Ross, Nov2003, http://www.ietf.org/rfc/rfc3628.txt
8. “Time Stamping Profile”, ETSI TS 101 861,Mar2003
9. “Tokens and Protocol for the Temporal Integrity Markup Language (TIML) ”,OASIS draft
10. “暗号技術評価報告書 (2002年度) ”,CRYPTREC (Cryptography Research and Evaluation Committees) ,http://www.shiba.tao.go.jp/kenkyu/CRYPTREC/fy15/cryptrec20030512_report044.htm
11. “Selecting Cryptographic Key Sizes”, A.K. Lenstra, E.R. Verheul, 1999
12. “時刻認証基盤ガイドライン”,タイムビジネス推進協議会,2002
13. “デジタルタイムスタンプ技術の現状と課題”,宇根正志、松浦幹太、田倉昭: 日本銀行金融研究所,2000
14. “Electronic Signature Formats for long term electronic signatures“ ,IETF RFC3126、http://www.ietf.org/rfc/rfc3126.txt
15. “Internet X.509 Public Key Infrastructure Online Certificate Status Protocol-OCSP”,IETF RFC2560, http://www.ietf.org/rfc/rfc2560.txt
16. “Internet X.509 Public Key Infrastructure Data Validation and Certification Server Protocols”, IETF RFC3029, http://www.ietf.org/rfc/rfc3029.txt
17. “Cryptographic Message Syntax”,IETF RFC2630, http://www.ietf.org/rfc/rfc2630.txt
18. “Electronic Signature Formats”,ESTI TS 101 733
19. “Electronic Signature Formats for long term electronic signatures”, IETF RFC3126, http://www.ietf.org/rfc/rfc3126.txt
20. “Electronic Signature Policies”, IETF RFC3125, http://www.ietf.org/rfc/rfc3125.txt

21. 暗号ブレイク対応電子署名アリバイ実現機構（その１）－コンセプト概要－”,松本勉 他, 情報処理学会 CSEC 研究会, ２０００年３月
22. 暗号ブレイク対応電子署名アリバイ実現機構（その２）－詳細方式－”,洲崎誠一 他, 情報処理学会 CSEC 研究会, ２０００年３月
23. ヒステリシス署名と証拠性基盤”, 洲崎誠一, 電子情報通信学会 ２０００年ソサイエティ大会パネル討論「量子・物理暗号とソフトウェア暗号の協調と未来展望」, ２０００年１０月
24. 松本勉、岩村充、佐々木良一、松木武：暗号ブレイク対応電子署名アリバイ実現機構（その１）－コンセプトと概要－、情報処理学会コンピュータセキュリティ研究会第 8 回研究発表会、2000
25. 洲崎誠一、宮崎邦彦、宝木和夫、松本勉：暗号ブレイク対応電子署名アリバイ実現機構（その２）－詳細方式－、情報処理学会コンピュータセキュリティ研究会第 8 回研究発表会、2000
26. 宮崎邦彦、吉浦裕、佐々木良一、洲崎誠一、松木武：連鎖構造を用いたディジタル署名の安全性強化に関する一考察、情報処理学会第６２回全国大会、2001
27. 伊藤信治、宮崎邦彦、本多義則、谷川嘉伸：電子署名の長期保証に関する一考察、電子情報通信学会情報セキュリティ研究専門委員会　暗号と情報セキュリティシンポジウム、2004
28. 電子文書証明　e ドキュメントの原本性確保　NTT データ経営研究所 編著　NTT 出版
29. タイムビジネス　ネット時代の時刻認証サービス　大橋正和 監修　タイムビジネス推進協議会 編著　NTT 出版
30. 電子署名文書長期保存に関するガイドライン　平成 14 年 3 月　電子商取引推進協議会 認証・公証 WG
31. タイムスタンプサービス調査報告書　平成 15 年 3 月　電子商取引推進協議会
32. 日経インターネットテクノロジー　2002 年 4 月号　日経 BP 社

IETF：Internet Engineering Task Force
ETSI：European Telecommunications Standards Institute
EESSI：European Electronic Signature Standardization Initiative

付属書Ｂ　商標関連

・Solaris8、Sun Fire V120 は米国サンマイクロシステムズの登録商標です。
・JRUN は米国マクロメディアの登録商標です。
・JP1/Cm2/Operations Assist Agent、JP1/Extensible SNMP Agent、JP1/Agent for Process Management は株式会社　日立製作所の登録商標です。
・LUNA C3,LUNA Xpplus は米国 chrysalis-its の登録商品です。
・Acrobat®5、Acrobat®6 は米国アドビ社の登録商標です。
・Document Signer Plug-in 2.0 は米国ベリサイン社の登録商標です。
・CECSIGN、CECTRUST は株式会社コンストラクション・イーシー・ドットコムの登録商標です。
・TDB 電子認証サービス TypeA は株式会社帝国データバンクの登録商標です。
・WindowsR98、WindowsRMe、WindowsR2000、WindowsRXP は米国マイクロソフト社の登録商標です。
・InternetExplorer は米国マイクロソフト社の登録商標です。
・Java は米国サン・マイクロシステムズ社の登録商標です。

付属書Ｃ　TSA 証明書を発行する CA の証明書ポリシー・運用規程（CP/CPS）を策定するためのガイドライン

はじめに

文書の電子化を実現するにあたり、電子化された文書を長期に渡り安全に保管するには、重要な課題がある。この課題に対し、ECOM では 2000 年度から電子署名文書の長期保存に関する検討を行ってきた。その中で、電子文書がある時刻以前に確かに存在した事を証明（存在証明）する共に、その当時の状態を保持していることを証明（完全性証明）するタイムスタンプの必要性を確認した。そこで、電子文書の日時を特定するタイムスタンプを活用した電子署名文書の長期保存システムの確立が必要不可欠とのことから、2002 年度の活動より、タイムスタンプのサービスに関する調査報告書と、タイムスタンプサービスの利用ガイドラインおよびタイムスタンプサービスのガイドラインを策定し、これを公開した。

ここで、PKI 方式のタイムスタンプでは、タイムスタンプを行う TSA に対し、TSA の証明書を発行する認証局（以下、CA）が存在することが前提となる。しかし、現在、個人用の証明書やサーバ証明書などを発行する CA は広く存在するが、TSA の証明書を発行する CA はほとんど存在しない。その原因の一つとして、一般的な証明書のポリシー・運用に対し TSA の証明書のポリシー・運用をどう規定すべきか議論が十分に行われておらず、CA の構築が困難である、という点がある。

TSA の証明書を発行する CA が、タイムスタンプとそれを必要とする電子署名文書長期保存の基盤となっているだけに、TSA の証明書を発行する CA の課題、疑問点を明らかにするとともに、CA がどうあるべきか、どう対応すべきかのガイドラインを示し、CA 構築の助けとしたい。

【用語の定義】

本ガイドラインで使用される用語について定義する。

・証明書ポリシー（CP）

公開鍵証明書の発行に関する CA のポリシー。

・タイムスタンプポリシー

タイムスタンプの発行に関する TSA のポリシー。

ガイドラインの構成

TSA の証明書を発行する CA のポリシー・運用を考える上で、最も重要な点は、CA と TSA の役割の規定である。CA・TSA が、それぞれの発行する証明書・タイムスタンプに対して、どのような責務を負うべきなのかを明確にすることで、互いの証明書ポリシーとタイムスタンプポリシーを適切に分けることが可能となる。

ここで重要なのは、CA の証明書ポリシーでは、TSA の存在のみを証明し、TSA の署名行為に関する責務を負うべきではない。また、TSA の署名行為に関する責務は、CA の証明書ポリシーに反しない上で、TSA のポリシーで決めるべきである。そこで、CA の証明書ポリシーと TSA の署名ポリシーに関し、以下の提言を行う。

【CA の証明書ポリシーと TSA のタイムスタンプポリシーに関する提言】

CA が発行する証明書と TSA が発行するタイムスタンプの目的と責任範囲を分離・区別し、CA と TSA の責務を明確にする。

①CA の証明書ポリシーでは、TSA の存在のみを証明し、TSA のタイムスタンプ行為に関する責務は負わない

②TSA のタイムスタンプポリシーでは、秘密鍵で署名したタイムスタンプの発行に関する責務を負う。

本ガイドラインでは、この提言に基づいたガイドラインを以下の形式で記載する。

1．TSA 証明書を発行する CA の証明書ポリシー・運用規程（CP/CPS）を策定するためのガイドライン

ここでは、CP/CPS の項目にあわせた形式で、TSA 証明書を発行するための CA のガイドラインを記載する。基本的な CA の CP/CPS で十分な部分が多いため、一般的な内容については ECOM 平成９年度成果報告書である「認証局運用ガイドライン」を参照いただくこととし、TSA 証明書を発行する CA の特徴的な部分についてのみ述べる。

2．TSA 証明書を発行する CA の運用に関する留意事項

ここでは、CA の運用に関するその他の留意事項について記載する。

3．TSA のタイムスタンプポリシー

ここでは、TSA 証明書を発行する CA の CP/CPS に対する、TSA のポリシー・運用について記載する。

なお、TSA 証明書はタイムスタンプの有効性検証に用いられるため、より有効期間の長い証明書が求められる。このため、本ガイドラインにおいても、証明書の有効期間を数年間として説明を行っている部分があることをご了承いただきたい。

1. TSA 証明書を発行する CA の証明書ポリシー・運用規程（CP/CPS）を策定するためのガイドライン

1.1. TSA 証明書の発行・利用に関わる関係者

本ガイドラインでは、TSA 証明書の発行・タイムスタンプへの利用に関して、以下の関係者を定義する。

（1）認証局（CA）
タイムスタンプ局のタイムスタンプサーバが使用する公開鍵証明書（TSA 証明書）を発行する認証局。

（2）タイムスタンプ局（TSA）
信頼される時刻ソースから時刻の提供を受けて、RFC3161 に基づくタイムスタンプ・プロトコルに準拠したタイムスタンプトークンを発行する事業者。

（3）依存者
タイムスタンプ局が発行したタイムスタンプトークンを信頼して利用、または検証する者。

1.2. 義務

1.2.1. CA の義務

（省略）

1.2.2. TSA の義務

・秘密鍵の使用の制限

TSA は、CA の定めた期間内でのみ秘密鍵を使用可能とし、CA が証明書を発行する前にタイムスタンプを発行したり、CA の定めた期間を越えてタイムスタンプを発行してはいけない。また、TSA は CA の定めた期間を越える前に、秘密鍵を確実に廃棄しなくてはならない。

例）TSA は、証明書の有効期間が６年間（例）であっても、秘密鍵の使用期間が証明書発行後１年間（例）と定められている場合、１年毎に鍵の更新を行い、CA から審査・証明書の発行を受けなくてはならない。
なお、TSA の鍵を更新しても古い証明書が失効するわけではないため、TSA が過去に発行したタイムスタンプへの影響はない

注）TSA の一般的な義務については、ECOM 平成１４年度成果報告書「タイムスタンプサービスの運用ガイドライン」を参照頂きたい。

1.2.3. 依存者の義務

依存者は、証明書の有効期間中に、タイムスタンプ・TSA 証明書の検証を実施する義務がある。タイムスタンプの有効性は TSA により所定の有効期間保証されているが、依存者は証明書・タイムスタンプが途中で失効している場合があることを前提に利用しなければならない。

1.3. 責任

1.3.1. CA の責任

CA は、TSA の存在のみを証明し、TSA のタイムスタンプ行為に関する責務は負わない。

1.3.2. TSA の責任

TSA は、タイムスタンプの時刻・タイムスタンプの有効期間等を保証する。

1.3.3. 依存者の責任

（省略）

1.4. 免責事項

CA が失効処理をしていたにも係わらず、依存者が証明書・タイムスタンプの有効性検証を怠った場合のトラブルについては、CA は一切責務を負わない。

1.5. 識別と認証（TSA の審査）

CA は、証明書の有効期間とは別に、CA の定めた期間毎に TSA の存在を審査し、証明書の更新処理を行う。

CA は、有効期間６年間（例）の証明書を発行した場合も、１年（例）ごとに TSA の審査を行う（＝TSA の存在証明は１年間のみ有効とする）ことで、TSA の存在証明に関するリスクを軽減することができる。また、依存者にとっても、TSA の存在が CA によって１年ごとに審査されていることで信頼性を確認することができる。

CA が TSA を審査する内容については、基本的な CA の CP/CPS を変更する必要は無い。もし TSA が、自身の信頼性を明らかにするため、時刻の認証やセキュリティ監査等を必要とする場合は、専門の知識を持つ TA や監査法人等を通じて審査を受けるべきであり、CA が審査しなければならないものではない。

1.6. 鍵更新

CA は、証明書の有効期間とは別に、TSA の秘密鍵が証明書発行後１年間（例）のみ署名に利用可能（＝秘密鍵の活性化が１年間のみ可能）となるように制限する。すなわち、証明書の有効期間が６年間でも、TSA は１年間ごとに秘密鍵を破棄し、証明書を更新する必要がある。
これにより、CA は自ら定めた運用期間で TSA の存在証明を保証することが可能となる。

注）TSA 側が確実に１年後署名できなくなる仕組み、もしくは署名できないことを証明する仕組みが必要。

例）

- TSA が使用するタイムスタンプサーバーは、秘密鍵のバックアップ機能が削除された安全な仕組みを必須とする。
- TSA が証明書の更新手続きを行うには、秘密鍵を破棄し、CA の審査手続きを必須とす

る。

・TSA の秘密鍵を更新後 CA は、reasonCode を付けて TSA の証明書を失効する。RFC3161 の規定によると reasonCode に superseded（4）を付与すれば、TSA の証明書が失効した場合であっても、失効時以前に生成されたトークンは、すべて有効とみなされる。このような reasonCode を発行した運用により、TSA の秘密鍵の活性化期間については、CA 側としてもコントロールすることが可能になる。ただし、CA の運用規定にて、reasonCode の説明が必要になる。

1.7. 証明書の失効

1.7.1. CRL の発行間隔

CA が TSA 証明書の CRL を定期的に発行する場合、TSA 証明書の発行枚数は
少なく、失効の可能性もほとんど無い。しかし、危殆化時の影響が大きいため、CA は 1 日間程度の間隔で CRL を更新すべきである。

1.7.2. TSA 閉局時の証明書の失効について

TSA が事業を閉鎖する場合、CA は証明書を失効するべきであろうか。また、失効を行う場合、CA による証明書の失効が、TSA が過去に発行したタイムスタンプまで失効してしまうことになるのだろうか。

RFC3161 においては、「TSA がこれ以上使われるべきではない状態であるが、TSA の秘密鍵がまだ危険にさらされていないと信用できる場合、この認証局の証明書は失効するものとする。」とされている。同様に、TSA 閉鎖時の証明書の失効については、“ETSI TS 102 023 タイムスタンプ局のポリシー要件”の「7.4.9 TSA の閉鎖」において「TSA はその証明書を失効するステップを実行しなければならない。」とされている。

これに基づいて TSA 証明書の失効を実行する場合、失効後に発行されたタイムスタンプトークンは無効であるが、失効以前に発行されたタイムスタンプトークンの有効性には影響を与えないことに注意しなくてはならない。タイムスタンプにとっては、タイムスタンプ発行時点でその TSA が存在したかどうかが重要であり、TSA が事業を行っていた時点で発行したタイムスタンプを失効する必要はない。

そこで、TSA 証明書が失効していても、TSA がその秘密鍵で過去に発行したタイムスタンプは有効であることを検証するための情報が必要である。RFC3161 においては、「CRL エントリ拡張の理由コード(reasonCode)を使用し、unspecified(0)、affiliationChanged(3)、superseded（4)、cessationOfOperation（5）のいずれかが設定されるものとされる」とし、理由コードを元に有効性を判断することを推奨している。（TSA 閉局の場合の理由コードとしては、cessationOfOperation（5）が適切である。）

また、理由コードの確認とともに、CRL に記載されている証明書の失効日（revocationDate）（もしくは無効日時（invalidityDate））をチェックし、タイムスタンプの発行日時が失効日（無効日時）以前であるかどうかについても、確認する必要がある。失効日時以降に発行されたタイムスタンプは、無効である。

このように、理由コードと失効日を利用すれば、証明書を失効してもタイムスタンプの有効

性を確認することが可能である。しかし、この場合タイムスタンプを検証する依存者・アプリケーションが、証明書の失効理由・失効日を解釈してタイムスタンプの有効性を判断できる必要がある。タイムスタンプのアプリケーションを作成する場合、この点に注意しなくてはならない。
また CA は、どのような失効要件に対し、どの理由コードが対応するのかが証明書利用者および依存者に確認できるよう、CP/CPS などに判断基準を記載し、公開しておくことが望ましい。

なお、「1.6.鍵更新」のように、TSA 証明書が 1 年間ごとに更新される場合、CA は TSA 閉局の年度に発行された TSA 証明書のみを失効すればよい。前年度に発行された TSA 証明書は、既に秘密鍵が使用されておらず、かつ TSA 自身の存在証明についても、新しい年度の審査を通過していることが確認できているため、失効する必要はない。

※CA が何らかの理由で過去の TSA 証明書を失効する必要がある場合など、特別なケースは除く。（例：過去に行われた TSA の不正が発覚した場合）

また、TSA 証明書の失効において、理由コードが無い場合、RFC3161 では、「該当する鍵で署名された全てのトークンは無効と考えられる。」としている。これは、理由コードを使用しない場合、失効の理由がタイムスタンプの有効性に影響を及ぼさないものなのか、鍵の危殆化などのようにタイムスタンプを無効と判断すべきものなのか判別できないためである。
そこで、（RFC や ETSI で推奨された方法ではないが、）CA やアプリケーションが理由コードを利用できない場合は、CA・TSA 側で TSA 証明書の失効理由を確認できる手段を用意するなどの対応が考えられる。
もしくは、TSA が事業を閉鎖する場合であっても、TSA の秘密鍵がバックアップされておらず、かつ確実に削除されたことを確認可能な場合、CA は証明書を失効しない、という運用も考えられる。

TSA 閉局時の証明書の失効に関しては、上記のような項目を検討し、CA の対応可能な運用を選択する必要がある。
できれば、以下の対応を行うことが望ましい。

- CA は TSA 証明書の失効には必ず理由コードを記載する。
  また、失効の要件と理由コードの対応を CP/CPS などに記載して公開する。
- タイムスタンプのアプリケーションは、必ず理由コードと失効日（もしくは無効日時）を確認する。
  ケースにもよるが、理由コードが affiliationChanged（3）、superseded（4）、cessationOfOperation（5）の場合、証明書の失効はタイムスタンプの有効性に影響を与えないと判断する。
  理由コードが無い場合は、CA・アプリケーションを考慮した上で、タイムスタンプを無効と判断すべきかどうか対応する。必要があれば、CA に失効理由を確認する。
  ※unspecified（0）は、理由コードが無い場合と同様の対応とする。

※なお、TSA が閉局を行う際、タイムスタンプの要求者・依存者は様々な影響を受ける。このため TSA は、”TSA 閉局の予告”、”タイムスタンプの発行停止”、”TSA の閉局”を段階的に行い、タイムスタンプの利用者が新たな TSA からタイムスタンプを取得するための十分な移行期間（例：半年間）を取った上で、TSA を閉局するべきである。

【TSA の閉局によるタイムスタンプ要求者・依存者への影響】

・TSA 閉局の予告～タイムスタンプの発行停止

タイムスタンプ要求者は、タイムスタンプの発行が停止される前に、新たな TSA からサービスを受ける環境を用意しなくてはならない。

タイムスタンプの発行停止までの期間が短いと、タイムスタンプ要求者がタイムスタンプを利用できない状況が発生し、損害を与えてしまう危険性がある。

・タイムスタンプの発行停止～TSA の閉局

CRL に理由コードを記載しない場合や、タイムスタンプのアプリケーションが理由コードに対応していない場合、依存者は、閉局する TSA から発行されたタイムスタンプに対して、TSA 証明書が失効する前に新たな TSA からアーカイブタイムスタンプを取得する必要がある。

TSA の閉局までの期間が短いと、依存者が新しい TSA からアーカイブタイムスタンプを取得できず、閉局する TSA から発行されたタイムスタンプが全て無効と判断されてしまい、損害を与えてしまう危険性がある。

1.8. 秘密鍵が危殆化した場合の対処

TSA 証明書の有効期間内に CA の証明書が危殆化した場合、CA は直ちに失効処理を行うとともに、TSA に通知を行う。この際に発生したトラブルについては、CA が責務を負う。

また、TSA の鍵が危殆化し、TSA から証明書の失効申請を受けた場合、CA は直ちに失効処理を行う。発生したトラブルについては、TSA が責務を負う。

1.9. 業務の終了

CA が業務を終了する場合、CA は依存者がタイムスタンプを検証するのに必要な情報（TSA の証明書、ルート CA・サブ CA の証明書、CRL など）を公開しなくてはならない。（また、TSA の審査記録や証明書の発行記録などは、TSA の信頼性を確認するために後日必要となる可能性があるので、保管しておくことが望ましい。）

※なお、TSA は、TSA 証明書を取得する際、ルート証明書が安全な方法で公開されている CA を選択するべきである。

1.10. 公開鍵と秘密鍵の有効期間

CA の発行する証明書の有効期間は、“ETSI TS 102 023 タイムスタンプ局のポリシー要件”の「7.2.4 TSA 鍵の再発行」においても言及されているように、選択されたアルゴリズムと鍵の長さが目的に合致したものとして認定されている期間より長くしてはならない。安全な鍵サイ

ズについては、ECOM の平成１１年度成果報告書「暗号利用技術ハンドブック　第２版」に記載があるので、参照いただきたい。

※例として、1024bit RSA 相当の場合、2007 年における国家レベルの攻撃者であっても、解読に６年間かかると記載されている。企業レベルの攻撃者であれば、6×10^3 年間である。

なお、証明書の利用面から見た場合、TSA 証明書はタイムスタンプの有効性検証に用いられるため、より有効期間の長い証明書であることが重要となる。これは後述のアーカイブタイムスタンプにおいても同様で、有効期間が長いほどタイムスタンプの回数を少なくすることができ、文書の管理上都合がよい。

表付-1 秘密鍵長と有効期間に関する推奨例

秘密鍵長	証明書有効期間	秘密鍵活性化期間	証明書検証可能期間
1024 ビット	６年	１年	５年
2048 ビット	１１年	１年	１０年

※推奨例における有効期間の根拠性について

証明書の有効期間については､暗号ハンドブックに記載されている数値を基準とした｡ただし、２０４８ビット長については、国家レベルの攻撃者であっても２０年以上安全と記載されているが、運用上の問題から１１年とした。これは、ビジネス的に１０年以上の検証可能な証明書が求められていること、安全性を考慮した理由により決定した。また、推奨例の有効期間については、あくまでも２００４年度時点における有効期間であり、適宜暗号強度に配慮した見直しが必要である。

1.11. 証明書のプロファイル

TSA の証明書は､その証明書がタイムスタンプ用のものであることを示す情報として､extended key usage の KeyPurposeID に、id-kp-timeStamping を設定しなくてはならない。また、この拡張はクリティカルでなければならない。これは RFC3161 で規定されている。

表付-2 TSA 証明書プロファイル

Field	Type	TSA 証明書の例	備考
証明書拡張			
ExtendedKeyUsage		critical	
KeyPurposeId	OBJECT IDENTIFIER	１３６１５５７３８ (timeStamping)	

2． TSA 証明書を発行する CA の運用に関する留意事項

2.1. アーカイブタイムスタンプを行う TSA に証明書を発行する CA のポリシー

TSA の発行するタイムスタンプには、署名時に付与するタイムスタンプと、長期保存のためのアーカイブタイムスタンプの２種類がある。

TSA 証明書を発行する CA のポリシーを検討する際、どちらのタイムスタンプを行う TSA に証明書を発行するかで、CA・TSA の責務が変わってくることに注意しなくてはならない。

（１）CA のアーカイブするデータ

CA がアーカイブタイムスタンプを行う TSA への証明書を発行する場合、過去のトラストポイントの情報を長期間保管するべきである。具体的には、発行した全ての TSA の証明書、ルート CA・サブ CA の証明書、CRL、審査記録・発行記録などである（有効期間の切れた証明書・CRL も含む)。CA はこれらの情報の信憑性を証明できなくてはならない。また、CA の事業が閉鎖される場合、CA はこれらの情報を他の CA に引き継がなくてはならない。

依存者は、これらの情報を CA から入手することにより、有効期間の切れた古いタイムスタンプの検証が可能となるとともに､「当時その TSA が確かに信頼できるものであったかどうか」ということを確認できる。

【参考】

アーカイブタイムスタンプの検証では、以下の条件が必要となる。

①検証者は、すべての署名・タイムスタンプの検証を行う。

②検証者はアーカイブタイムスタンプの検証に必要な証明書を発行したすべての CA を信頼できることが前提である。

（信頼できない CA・TSA を排除するため。）

ここで問題となるのは、

・検証者にとって、すべての CA が「信頼できる CA」であるかどうかは、判断が難しい。

・そのすべての CA から、検証に必要なすべての情報を入手するのは難しい。

古い CA では CRL 配布点等が変わっているかもしれない。

という点である。

そこで、アーカイブタイムスタンプには、以下のような登録機関・データベースが求められている。

・当時の「信頼できる CA」が登録されている。

・登録された CA の発行したすべての証明書・CRL を保管する。

・保管している情報の信憑性を証明できる。

・検証者からの要求に対し、必要な検証情報を検索し提供する。

・登録された CA に対し、第三者的な立場にある。

上記を満たす登録機関・データベースは、アーカイブタイムスタンプの根幹をなすものであり、事業を閉鎖する場合は、必ずそれを他の事業者に引き継がなければならない。
これらの条件を満たすものとして、国家・業界団体等による統一された登録機関・データベースの設置・運営が最も望ましい。それ以外にも社会的に信頼できる情報源・媒体・事業者等を複合的に利用することで、同等の機能を実現することも可能である。
具体的な保管方法として

- 紙文書による長期保管
  長期保管が必要な電子情報を紙文書形式に変換・印刷し、公証役場の確定日付を取得することで、紙文書として長期間保管する。
- 非改竄と存在時刻が証明できる措置を講じた長期保管
  長期保管が必要な電子文書に対し、保管期間中公知に安全と認められたタイムスタンプ事業者が提供するタイムスタンプサービスを利用するか、又はそれに相当する措置を講じて重要な情報を長期保管する。

2.2. CA を運営する事業者について

TSA 証明書を発行する CA を運営する事業者として、TSA が自社で CA を構築することは問題ないか。もしくは CA が自社で TSA 事業を運営することは問題ないか。“ETSI TS 102 023 タイムスタンプ局のポリシー要件”の「7.2.3 TSA 公開鍵の配
布」においては、「注記 ：例えば、TSA の証明書は、TSA と同じ組織によって運営される認証局、またはほかの機関が発行できるものとする。」といった記述があり、TSA 事業者が自社の CA を用いることは特に問題視されていない。

しかし、実際どういった脅威が起こり得るか考えてみるべきである。CA と TSA 事業者が結託した場合の脅威は、時刻が改ざん可能なサーバにタイムスタンプ証明書が発行されてしまい、信頼できないタイムスタンプが発行されることである。
従って、TSA 事業者の発行するタイムスタンプが、TSA 事業者自身では時刻を改ざんできないものであることが証明可能であれば、TSA 事業者が自社の CA の TSA 証明書を使用しても問題ないと言える。
これを証明する方法としては、例えば以下のような条件が挙げられる。

- TSU の鍵がバックアップできないこと。
- TSU の時刻を TSA 事業者自身では設定できないこと。
- 発行されたタイムスタンプに、時刻が改ざんされていないことを
  証明可能な情報（例：時刻監査証明書）が含まれていること。
- TSU の時刻に関する運用ログが厳正に管理されていること。
  また、第三者の時刻監査を受けていること。

3． TSA のタイムスタンプポリシー

TSA 証明書を発行する CA の CP/CPS に対し、TSA のポリシー・運用は以下のようになる。

・タイムスタンプの有効期間は、TSA 事業者が決定する。ただし、証明書の有効期間を越えて保証することはできない。

・タイムスタンプの有効期間を６年間（例）とする場合、
TSA は秘密鍵で署名したタイムスタンプの有効期間を維持し、事業を継続するためには、毎年少なくとも５年間は、CA からの審査を受け、新たな証明書の発行を受けなくてはならない。

・TSA が発行するタイムスタンプについての責務は、タイムスタンプの有効期間に限定し、失効処理後に発生したトラブルについては、すべて免責されるものとする。

・TSA 証明書の失効の可能性を抑えるため、秘密鍵はバックアップできない仕組みとすべきである。

・TSA のタイムスタンプポリシーには、それとリンクする CA の証明書ポリシーについて明記する。

おわりに

本稿では、CA・TSA の責務に関する提言を行うとともに、それに基づいた TSA 証明書を発行する CA のポリシー・運用に関するガイドラインと運用に関する留意事項、およびそれに関わる TSA のタイムスタンプポリシーを記載した。一般的な CA のガイドラインを前提とした上で、TSA 証明書を発行する CA に特徴的な部分をご理解頂けたかと思う。
本ガイドラインを参考とし、電子署名文書の長期保存に必要とされる TSA 証明書を発行する CA が構築されることを期待したい。

以上

メンバーリスト

事務局

川松　和成　電子商取引推進協議会　主席研究員
前田　陽二　電子商取引推進協議会　主席研究員
松山　博美　電子商取引推進協議会　主席研究員

顧問

松本　　勉　横浜国立大学大学院
平田　健治　大阪大学大学院

リーダー

櫻井　徹　　株式会社ＮＴＴデータ
宮崎　一哉　三菱電機株式会社
木村　道弘　日本電気株式会社

TF5 メンバー（編集メンバー）

氏名	会社名
磐城　洋介	NTT コムウェア株式会社
野村　　進	NTT コミュニケーションズ株式会社
手塚　優	エントラストジャパン株式会社
上畑　正和	セイコーインスツルメンツ株式会社
雨宮　隆征	セイコーインスツルメンツ株式会社
秋山　将	日本電信電話株式会社
高村　昌興	株式会社 NTT データ
本多　義則	株式会社日立製作所
久保寺　範和	日本電気株式会社
吉川　信雄	富士通株式会社

SWG3 メンバー（参加メンバー）

氏名	会社名
鈴木　優一 ＊	エントラストジャパン株式会社
溝上　卓也	日立ｿﾌﾄｳｪｱｴﾝｼﾞﾆｱﾘﾝｸﾞ株式会社
河田　悦生	株式会社 NTT ドコモ
関野　公彦 ＊	株式会社 NTT ドコモ
西田　真	大日本印刷株式会社
近藤　哲生	シャチハタ株式会社
植木　格郎	東京電力株式会社
藤川　真樹	綜合警備保障株式会社
小林　太	株式会社帝国データバンク
相原　敬雄	日本ベリサイン株式会社
尾中　壱行	日本信販株式会社
野口　雄治	日本認証サービス株式会社
柿崎　竜人	NTT コムウェア株式会社
川城　三治	グローバルフレンドシップ株式会社
日向　一人	富士電機株式会社
石原　達也	株式会社東芝

（注）＊はオブザーバー

平成 15 年度　経済産業省　委託事業
ＥＣ技術基盤の相互運用性に関する調査研究事業
（電子署名生成・検証システムのセキュリティ環境の
標準化等調査）
電子署名文書長期保存に関する実用化動向調査報告書
平成 16 年３月発行

発行所　財団法人　日本情報処理開発協会
電子商取引推進センター
東京都港区芝公園３丁目５番８号
機械振興会館３階
TEL：03(3436)7500
印刷所　新高速印刷株式会社
東京都港区新橋 5-8-4
TEL：03(3437)6365

この資料は再生紙を使用しています。