H13－認証･公証WG－3

# 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

**連絡先**

電子商取引推進協議会（ＥＣＯＭ）

認証･公証ＷＧ

〒 105-0011

東京都港区芝公園 3-5-8　機械振興会館 3 階

Tel. 03-3436-7500

Fax. 03-3436-7570

e-mail： info@ecom.jp

URL： http://www.ecom.jp/

目　次

# 1 はじめに

## 1.1 デジタル署名の有効性を維持する必要性

私たちは日常生活において、ある時点における「ことがら」を紙ベースの文書としてあらわし、「押印」によってその文書を承認している。たとえば、「売買契約書」、「領収書」、「土地の権利書」、「借用書」等の作成・押印がこれにあたる。さて、当事者が文書に押印したこと（押印の有効性）をあとで証明するためには、どのような方法が考えられるだろうか？　一般的には、当事者が押印した文書と、当事者が押印に使用した印鑑の印鑑証明書を一組にして大切に保存する方法があるが、この方法を用いる前提として、印鑑は当事者によって厳正に保管されていること、印鑑証明書は信頼できる第三者機関（役所）から発行されていることがあげられる。この方法は、電子文書への応用を考えると、紙ベースの文書を電子文書に、印鑑を秘密鍵に、そして印鑑証明書を公開鍵証明書におきかえることができそうである。しかし、そこにはクリアしなければならない以下の大きな問題がある。

1. 公開鍵証明書には有効期限がある

   公開鍵証明書には有効期限が設けられている。この有効期限を越えると、デジタル署名を正しく検証できなくなるだけでなく、デジタル署名生成時点でのデジタル署名の有効性の判断ができなくなる。

2. 公開鍵証明書は失効が発生する可能性がある

   公開鍵証明書の失効が発生した場合には、たとえその公開鍵証明書が有効期間内であったとしてもデジタル署名生成時点でのデジタル署名の有効性の判断ができなくなる。

3. デジタル署名に使用されている暗号アルゴリズムは脆弱化する可能性がある

   計算能力の高いコンピュータの出現、あるいは素因数分解の効率的な導出方法の解明等によってデジタル署名に使用されている暗号アルゴリズムが脆弱化すると、公開鍵から秘密鍵が導き出されてデジタル署名が偽造される可能性があるため、デジタル署名生成時点でのデジタル署名の有効性の判断ができなくなる。

本ガイドラインでは、有効な公開鍵証明書とそれに対応する秘密鍵を所持する者が電子文書に署名したこと（デジタル署名の有効性）を長期的に維持するためのモデルシステムの提案を目的とする。

## 1.2 具体的事例

デジタル署名の有効性を維持するためには、前述の問題を解決しなければならない。本節では、これらの問題を具体的な事例で説明する。

実社会において身近で管理される文書の一つとして領収書がある。領収書は長期的な保存が必要とされる可能性があり、その真正性が問われる文書事例としても考えられる。その保存期間は税務調査との関係で最低７年と考えることが一般的である。これは国税徴収権の消滅時間が５年間であることに起因する。また、保存の目的としては、税金の申告用書類、二重払いの回避、権利関係の証明等が挙げられる。図 1-1 は、事業者が物品購入で取得した領収書が、必要経費として税務申請される流れを簡易的に表したものである。[1]

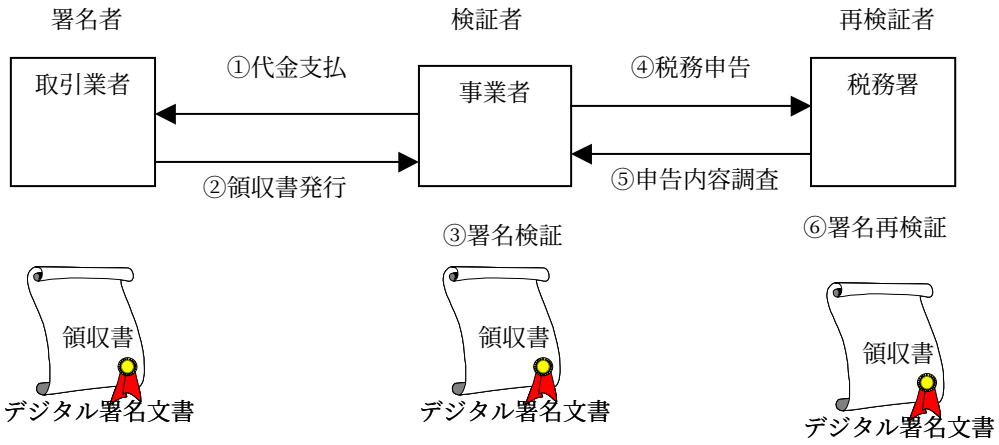



図 1-1 具体的事例

1. 公開鍵証明書の有効期間

   事業者（検証者）は、領収書発行のプロセス③において、デジタル署名の検証に使われる公開鍵証明書が有効期間内であることを確認する必要がある。

2. 公開鍵証明書の失効状態

   事業者（検証者）は、業者から発行された領収書のデジタル署名を検証し、税務申告用として保存する。（プロセス③～④）この時、デジタル署名の検証と同時に、その署名に使われた公開鍵証明書が失効状態でないことを確認する必要がある。これは、公開鍵証明書の所有者が紛失や盗難などの理由により、その公開鍵証明書を失効状態にしている可能性があるからである。その公開鍵証明書が失効

状態であったならば、そのデジタル署名は無効と判断すべきである。

3. 公開鍵の暗号アルゴリズムの脆弱化

デジタル署名生成時点においては、理論上安全であると考えられていた暗号アルゴリズムも、ハッキング可能な技術や理論が確立されてしまうと全く意味を成さなくなってしまう。これは、検証プロセスにおいて有効であると判断されたデジタル署名自身が実は偽造されたデータである疑いが掛けられたり、署名者自身がデジタル署名に対して否認が出来たりすることを意味する。

なお、税務署（再検証者）は申告内容に問題が確認された場合に、事業者の保存する領収書の内容を調査することも考えられる。（プロセス⑤～⑥）場合によっては、公開鍵証明書の有効期間を超えて調査をすることも考えられるため、長期間保存されたデジタル署名文書の検証ができる仕組みが必要となる。

## 参考

領収書とは、代金や手数料などを支払ったときに、受け取った相手方が出す書面（文書）のことを指す。これは、弁済（金品の支払い）もしくは受け取ったことの証拠としての意味を持ち、税務調査の対象としても有効な文書である。このように、領収書は金銭に関するトラブルを事前に回避するための証明書であるとも言える。

領収書の発行時期は、通常は金銭や物品の引渡しと同時に発行される。弁済者は、相手方に受取証（領収書）の交付を請求できる。（民法486条）弁済と領収書の交付とは同時履行の関係にある。言い換えれば、相手方が領収書の交付をしなければ、弁済を拒否することも可能である。これは、二重の支払いの危険を回避するためである。したがって、受け取る側が、領収書を交付しないことに起因する弁済拒否に関しては、遅延利息の支払い義務は発生しない。

領収書作成についての用紙や書式に関しての特別な規定は無いが、必要最低限の記載情報として下記の項目が挙げられる。

・金額とその受け取ったことを示す内容

・受領者（債権者またはその代理人）と支払人の氏名、受領年月日

・金銭を支払った対象（必須では無いが、明確にする際には必要）

■弁済日付の内容

実際の弁済日と異なる日付の領収書でも、その効力は有効である。しかし、偽った日付を利用することで税金を免れたり、還付を受けた場合には脱税対象として刑罰を科せられることになる。また悪意の利用が無くとも、あいまいな日付が判明した場合は、厳格な税務調査の対象となる可能性がある。

■記載事項の訂正（金額の訂正）

官公庁へ提出する領収書、または発行される領収書については訂正を認めていない。これに対し、民間で発行される領収書は訂正が認められている。ただし、訂正した個所には発行権限者の承認のもとに訂正捺印を押印する必要がある。この場合は代理人や使用人が発行権限者の承認のもとに訂正したかどうかでトラブルとなる可能性があるため注意が必要である。

■再発行について

発行者は発行した領収書に関して、紛失などの理由により弁済者から再発行を求められたとしても、その再発行義務は無い。ただし、紛失した領収書に宛名が記入されていない場合は、拾得者により悪用される可能性がある。

■印紙

記載金額が三万円以上の場合に印紙の貼付が印紙税法上要求されている。また、印紙と領収書の効力には関連性は無く、印紙が無い領収書でもその効力に問題は無い。ただし、印紙は納税の一種であり、その支払いを怠ったことに対して処罰されうる可能性がある。

## 1.3 国内外の動向

PKI の普及・拡大にともない、デジタル署名の有効性を長期的に維持するための試みや電子文書の長期的保存に関するシステム開発およびサービスは国内・海外ともに行われている。本節では、その技術・サービス動向と標準化動向について紹介する。

### 1.3.1 技術・サービス動向

#### 1.3.1.1 暗号ブレイク対応電子署名アリバイ実現機構[2,3]

これは、2000 年 3 月に松本らによって提案されたもので、電子データに施されたデジタル署名が正当な署名者によって生成されたものであるか否かを判別するための技術である（図 1-2 に概要を示す）。具体的には図１-3 に示すように、n 回目のデジタル署名生成時に、n-1 回目のデジタル署名のハッシュ値 $H(S_{n-1})$を n 回目の署名対象データと連結させて署名履歴 $S_n$ を生成する。そして、生成された署名履歴 $S_x$ を信頼できる第三者機関に寄託することで、署名者によるデジタル署名生成事実を長期的に保証しようとするものである。n 回目のデジタル署名検証時には、検証者に署名履歴 $S_n$ に加えて直前の署名履歴 $S_{n-1}$ を提出する必要がある。また、$S_n$ 直後の署名履歴 $S_{n+1}$ との整合性を調べる必要がある場合には、$S_{n+1}$ も提出しなければならない。

署名履歴は、その順番が入れ替わったり、一部あるいは全てが消失してしまったりしないように注意して保存しなければならない。なぜならば、署名履歴の系列が途絶えることによって整合性がとれなくなるため、署名者が正しくデジタル署名を生成していたとして

も、その有効性が証明されなくなるからである。

署名者が生成した過去のデジタル署名を偽造するためには、攻撃者は当該デジタル署名以降に生成された署名履歴との整合性を崩さないように偽造しなければならない。署名履歴が信頼できる第三者機関に正しく寄託されているならば、署名の偽造は不可能であるといえる。

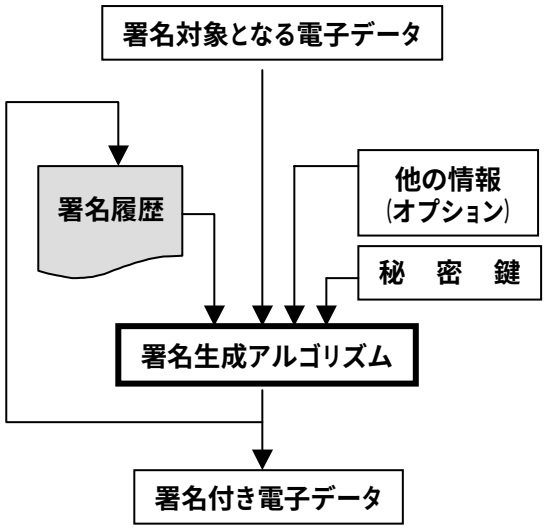
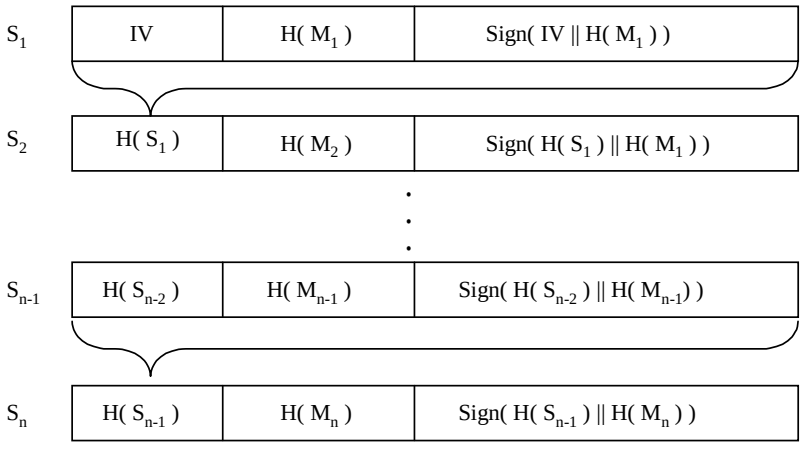



図 1-2　ヒステリシス署名概要




n-1 **回目の署名結果を** n **回目の署名に作用させる**

IV : Initial Data
Sign( A || B ) : **データ** A **と** B **を連結したものの署名値**
H(Mi) : **メッセージ**Mi( i=1,2,…,n ) **のハッシュ値**
Si( i=1,2,…,n) : **署名履歴**

図 1-3　署名履歴の生成

#### 1.3.1.2 長期保存文書のためのデジタル署名期限延長技術開発

IPA（情報処理振興事業協会）は、平成 12 年度電子政府情報セキュリティ基盤技術開発事業として、電子文書長期保存技術検討コンソーシアム（三菱電機、日立製作所、日本電気）が中心となって「長期保存文書のためのデジタル署名期限延長技術開発」を行った。ここでは、デジタル署名の有効性を延長させる「署名延長サーバソフトウェア」の基本機能、および署名延長クライアントソフトウェアが開発されている。

#### 1.3.1.3 セキュアストレージシステム

2000 年 10 月、三菱電機によってデジタル署名の有効性を長期的に維持させるシステムが提案されている[4,5]。この提案の中で同システムは、署名文書を保存するとともに、後述する ETSI TS 101 733 Electronic Signature Formats をベースとしたデジタル署名の有効性長期化のための処理を実行するサーバシステムとして発表されている。

#### 1.3.1.4 Secure Seal（NTT データ）

これは、電子文書の存在性（ある時刻に存在していたこと）と完全性（その時刻以降改ざんされていないこと）を保証するタイムスタンプサービスである（2000 年 4 月より展開。技術的詳細は 2 章を参照）。

利用者は専用の端末ソフトを用いて電子文書のハッシュ値を生成し、「Secure Seal」センタへ登録する。センタは、受け取ったハッシュ値をもとに「Secure Seal 証明書（X.509 証明書ではない）」を利用者に発行する。この証明書には、登録文書のハッシュ値や時刻情報、Secure Seal センタで生成した SHV（Super Hash Value）などが含まれる。電子文書の検証は、登録時と同じ手順でハッシュ値を生成したものが、センタに登録されているものと同値であるかを確認することで実行できる。この仕組みを利用して、電子文書の存在性や完全性の保証を重視する電子カルテシステムやデジタルコンテンツ管理システムに応用されている。

#### 1.3.1.5 電子公証サービス（法務省）

このサービスは、現行の公証制度を基盤として現在紙ベースで提供されている公証サービスを電子文書に適用したものであり、法務大臣によって任命された公証人のなかから、特に指定された公証人（指定公証人）が運用している（2002 年 1 月 15 日開始）。

なお現在このサービスは、法人として登記された企業が対象であり、個人は対象外となっていることに留意されたい。サービス利用の際には、法務省の「商業登記制度に基礎を置く電子認証制度」に基づいた電子証明書を取得と、専用ソフトウェアを準備することが必要である。

法務省では、現在以下の４つのサービスを提供している。

(1) **電子確定日付の付与**

インターネットを介して、嘱託人（クライアント）が作成した電子文書の成立時期及び内容を証明する電子確定日付（日付情報）を公証人が付与する。手数料は、紙文書の場合と同額で１件当たり 700 円。

(2) **電子私署証書の認証**

指定公証人が、嘱託人作成の電子私署証書（電子文書にデジタル署名をしたもの）を認証（文書の内容が違法でないことを審査）する。手数料は、１件当たり 11.000 円（原則）。株式会社、有限会社設立の際に作成する定款については、2002 年 4 月 1 日から電子文書として作成することができる。

(3) **同一性の証明**

電子確定日付文書、電子私署証書が、確かに指定公証人により認証されたものと同一であることを証明してもらうもの。手数料は、１件当たり 700 円。

(4) **同一情報《複製》の取得**

嘱託人は、作成した電子文書の原本保存を公証人にあらかじめ依頼する。その後、指定公証人に保存依頼した電子確定日付文書、電子私署証書の原本の複製作成を依頼し、その結果を取得するもの。電子文書保存の手数料は１件当たり 300 円で、同一情報《複製》の取得の手数料は１件当たり 700 円（原則）。

#### 1.3.1.6　原本性確保支援システム（Trusty Cabinet V1：リコー）

これは、電子文書を証明力の高い「原本」として安全に長期間保存・管理する機能（原本性の確保）を提供するもので、各種業務システムに組み込みやすいソフトウェアモジュールとなっている。以下に、このシステムの機能を示す。

(1) **原本管理機能**

① 原本データと各種履歴については改ざん検知用データを付与し、改ざんの有無を判別可能（保存文書の改ざんチェックにはハッシュ関数を利用。文書と暗号化し

たハッシュ値はセットにして保存。暗号化鍵は、利用者や管理者にもわからないようにプログラム中に格納）。

② 保存データには、作成日時や保存期限などの属性情報を付与して管理する。

③ 保存データに対して更新処理を行うと、自動的に原本の版管理を実施する。

### (2) アクセスコントロール

① 保存装置を利用するクライアントシステムへアカウントを発行し、クライアントシステムを管理・認証する。

② 保存データへのアクセス権は Read Write と Read Only の２つで、エンドユーザごとのアクセスコントロールはクライアントシステムが設定・管理する。

③ 保存データへの操作と装置への操作は全て自動的に履歴を保存する（管理者の操作も履歴に残る）。

### (3) 記憶媒体およびバックアップ

① 保存データを CD-R へ書き出し、媒体識別番号を付与する。

② 保存装置内部データの自動的なミラーリングや、外部記憶装置へのバックアップが可能である。

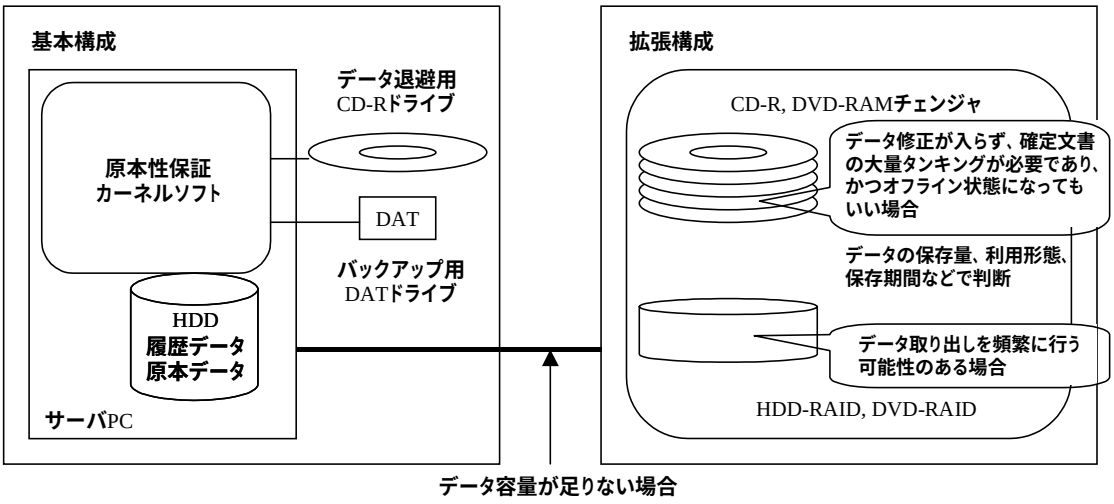



図 1-4 原本性保証システム Trusty Cabinet V1 の構成要素（出典：リコー）

### 1.3.2 標準化の動向

ヨーロッパでは、ETSI(the European Telecommunications Standards Institute：南フランス Sophia Antipolis に本部がある。将来的なヨーロッパにおける情報通信技術の標準化を策定している非営利団体。現在、ヨーロッパ内外 54 カ国・874 人の会員によって構成されている)の Technical Committee Security(SEC) によって、デジタル署名の有効性を維持させるための Technical Specification(TS) が作成されている（2000 年 12 月 8 日にアップデートされており、最新バージョンは 1.2.2。詳細については本ガイドライン 2 章にて述べる)。この TS は ETSI TS 101 733 Electronic Signature Formats[6] と呼ばれており、タイムスタンプ（本ガイドライン 2 章参照）を用いてデジタル署名の有効性を維持させようとするものである。なおこの TS は、IETF(the Internet Engineering Task Force)においても議論され、現在 RFC3125（署名ポリシ）[7]と RFC3126（署名フォーマット）[8]に分かれて登録されている。

## 1.4 デジタル署名の有効性を長期にわたり維持するための要件

これまでにも、デジタル署名の有効性を長期的に維持するための手法として、以下のものが提案されてきた。

① デジタル署名のハッシュ値を公刊メディア（新聞等）を使って公開する。これによりデジタル署名の存在が歴史的事実となるため、その存在が長期的に証明可能となる。

② 電子文書とデジタル署名を、不正な物理的アクセスが困難な耐タンパシステムに格納する。

③ 電子文書とデジタル署名を、社会的に信頼された第三者機関（Trusted Third Party：ＴＴＰ）に預ける。前提として、ＴＴＰはこれらを長期的に保存できるシステムを備えているものとする。

これらの手法は、いずれも有効と判断したデジタル署名の存在を保証することで、デジタル署名の有効性を長期的に維持しようとするものである。しかしながら一方で、デジタル署名が本当に有効であったか、検証過程に誤りがなかったかを確認することについては十分に考慮されていない。そこで、本ガイドラインでは、長期に渡りデジタル署名の有効性を再確認することを可能とすることにより、デジタル署名の有効性を保証するモデルを

検討し、デジタル署名の有効性を維持することを、以下のような事象が発生しても過去に有効性を検証したデジタル署名の再検証を行うことにより、デジタル署名の有効性を確認可能としておくことと定義する。

- 公開鍵証明書の有効期限を過ぎた
- 公開鍵証明書が失効された
- 暗号アルゴリズムが脆弱化した

再検証可能なデジタル署名にはどのような要件が必要とされるであろうか。上記のような事象が発生した後では、過去にデジタル署名を有効であると判断した署名検証と同じ署名検証を行ってもデジタル署名が無効と判断されてしまう。これを防ぐには、これらの事象が発生する前にデジタル署名の検証を行い、検証に利用したあらゆる情報を残しておき、署名再検証時にそれらの情報を利用可能にしておくことが必要となる。よって、次の要件が導かれる。

- 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

デジタル署名の場合、署名検証にはまず公開鍵証明書が必要である。この公開鍵証明書は有効期限の到達や失効によりその有効性を失う可能性があり、署名検証に利用できなくなってしまう。よって、公開鍵証明書が署名検証時には確かに有効であったことを示す情報、つまり失効情報も必要となる。この失効情報は、失効情報を発行した時点での公開鍵証明書の状態を示すものであるから、署名検証時がいつなのかを明確にしておく必要がある。よって次の要件が導かれる。

- 要件２）署名検証時の時刻を明確にしておくこと

署名再検証に必要な情報を明確にしたうえで、それらを残しておいたとしても、署名再検証に必要な情報が改ざんされたらデジタル署名の有効性を確認するための再検証ができなくなってしまう。よって次の要件が導かれる。

●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

署名再検証に必要な情報を明確にしたうえで、それらを残しておいたとしても、署名再検証に必要な情報が消失しては、デジタル署名の有効性を確認するための再検証ができなくなってしまう。よって次の要件が導かれる。

●要件４）署名再検証に必要な情報を保存すること

以上をまとめると、デジタル署名の有効性を維持するための要件は以下の４つとなる。本ガイドラインでは、この 4 要件を満たすことのできるモデルシステムの提案を目的とする。

●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

# 2　デジタル署名長期保存技術

## 2.1　基本コンセプト

前章では、本ガイドラインの立場として、デジタル署名の長期的な有効性の維持を実現することに対し、検証者が過去の特定日時においてデジタル署名文書の有効性を確認した証拠情報を、後日における署名再検証のために署名文書とともに残す方式に着目していることを述べた。本節では、まずその証拠情報生成のための基本的な考え方について述べる。

1.4 で述べたように、本ガイドラインでは、デジタル署名の有効性を長期的に維持するための要件として、次の 4 つを挙げている。

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
2. 署名検証時の時刻を明確にしておくこと
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
4. 署名再検証に必要な情報を保存すること

これらの要件は、図 2-1に示すような基本的な考え方をもとに導出された要件であり、証拠情報の生成においては、これらのうち 1～3 の要件を考慮する必要がある。以下では、これら 3 つの要件を満たすために、証拠情報の生成にどのような技術が必要とされるのかについて述べる。

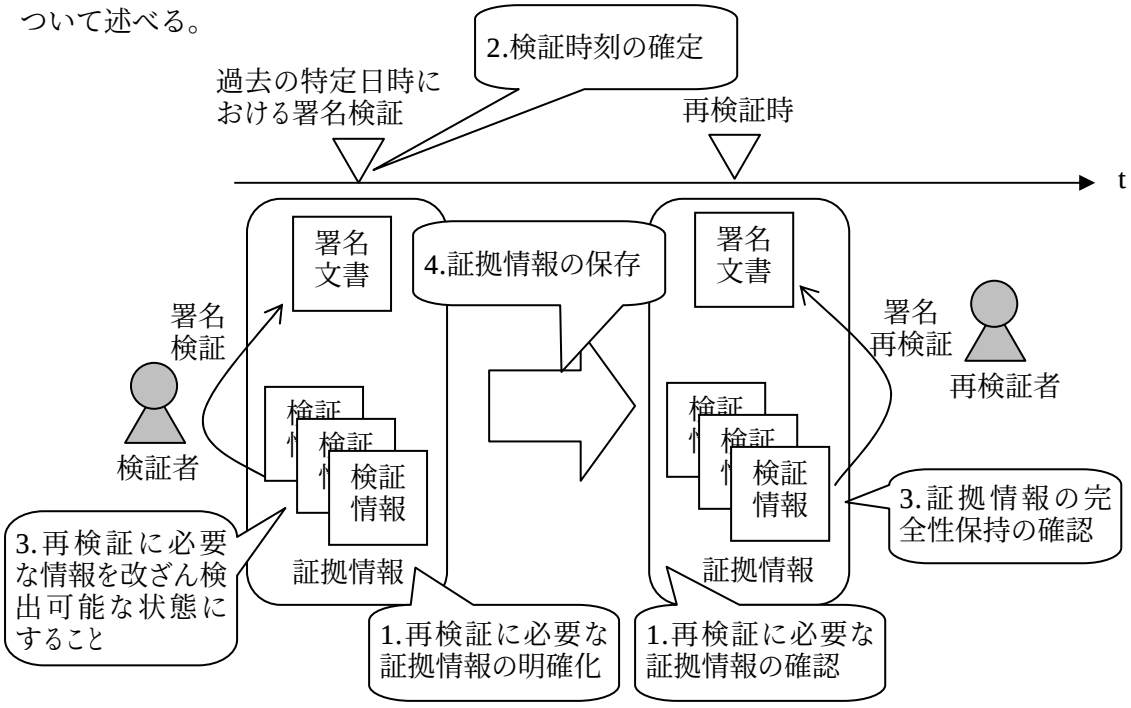



**図 2-1　デジタル署名の長期的な有効性の維持のための基本的な考え方と要件**

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

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ある。本ガイドラインでは、「有効な署名として成立」するためには、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）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と考えており、これらの確認に必要な情報としては、以下のようなものが挙げられる。

●署名の本人性の確認

デジタル署名文書に付与された署名の本人性が確認されるとは、署名検証時点において、有効で信頼できる署名者の公開鍵証明書に含まれる公開鍵を用いて署名検証が正常終了することであり、この確認に必要な情報としては、例えば、デジタル署名を検証するための署名者の公開鍵証明書や、その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）の合意の確認

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を指す。

以上より、署名検証時において、署名規則に基づいた署名検証のために必要な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、図 2-2 に示すように、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。

- 署名規則の合意形成及び確認可能な技術
- 署名規則に基づいて確実に署名検証を行うための技術
- 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
- 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
- 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は、表 2-1 に示す関係で、タイムスタンプ技術、署名検証技術、署名ポリシ合意形成技術、署名フォーマット形成技術が存在しており、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。

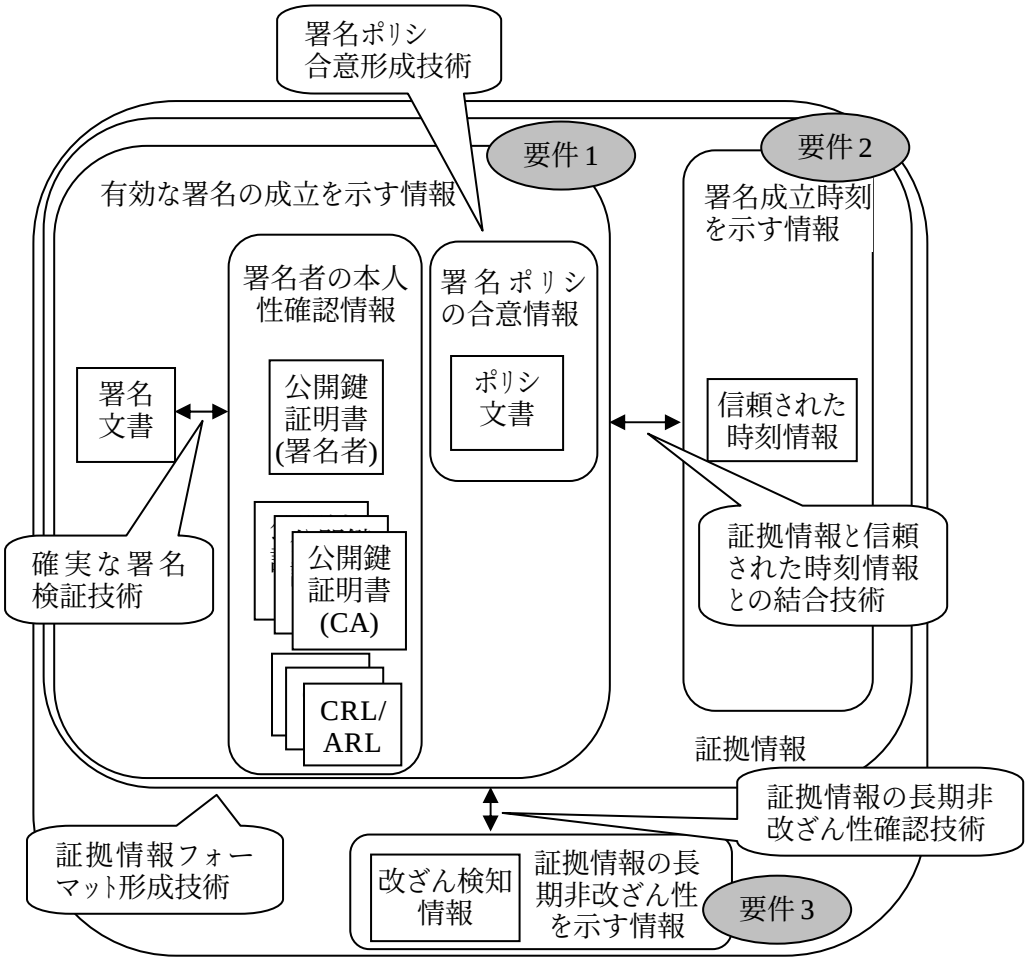



**図 2-2 証拠情報の生成に必要な技術**

**表 2-1 技術要件に対応する既存技術**

| 技術要件 | 既存技術 ( )内は説明される節 |
|---|---|
| 署名規則の合意形成及び確認可能な技術 | 署名ポリシ(2.2.4) |
| 署名規則に基づいて確実に署名検証を行うための技術 | 署名検証技術(2.2.2) |
| 証拠情報をデジタル署名文書とともに保持するための技術 | 長期署名フォーマット(2.2.3) |
| 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 | タイムスタンプ技術(2.2.1) |
| 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 | タイムスタンプ技術(2.2.1) |

## 2.2 基本技術

### 2.2.1 タイムスタンプ

#### 2.2.1.1 概要

タイムスタンプが提供する役割には、下記の二つがある。

・電子文書の存在証明

タイムスタンプが生成された時刻以前に、タイムスタンプが施された電子文書が確かに存在した事を証明するもの。

・電子文書の完全性証明

タイムスタンプを施された電子文書が、その当時の状態を保持している事を証明するもの。

タイムスタンプは、それを生成する技術的な方式と、その技術を用いて安全なタイムスタンプを提供するサービス（ＴＳＡ）に支えられ、上記の役割をはたす。

タイムスタンプ（Time Stamp Token）とは、ＴＳＡ（Time Stamp Authority）が、電子文書のハッシュ値に対して、信頼できる時刻の提供者からの時刻に対して、ある方式で安全に関連付けた状態のものを言う（図 2-3）。


**図 2-3 タイムスタンプの生成例**

#### 2.2.1.2 タイムスタンプの種類

タイムスタンプには、「シンプル・プロトコル」「リンキング・プロトコル」と「分散プロトコル」などの種類がある（表 2-2）。

**表 2-2 タイムスタンプの主なプロトコル**

| プロトコル | 概要 | 長所 | 短所 |
|---|---|---|---|
| シンプル･プロトコル | **デジタル署名を用いる。**<br>一つのＴＳＡが、利用者がタイムスタンプを希望するデータのハッシュ値に時刻情報等を添付してデジタル署名(タイムスタンプ)を生成。 | システムが単純 | ＴＳＡが署名者と結託すると容易にタイムスタンプの改ざんが可能。 |
| リンキング･プロトコル | **ハッシュのみを用いる。**<br>ＴＳＡが過去のハッシュ値を関連付けるリンク情報を生成し、リンク情報からタイムスタンプを生成する。定期的にリンク情報の一部を新聞等に掲載する。 | ＴＳＡに信用が無い場合でもシステム全体として信頼性が確保できる。 | システムの作りが複雑になる。<br>リンク情報を保管するデータベースが必要。 |
| 分散プロトコル | **デジタル署名を用いる。**<br>複数のＴＳＡにタイムスタンプの生成を求める。全てのタイムスタンプが揃ったところで証明できる。 | 同上 | 複数のＴＳＡが必要。システムの作りが複雑になる。 |

「デジタルタイムスタンプ技術の現状と課題」[9] より

### 2.2.1.3 タイムスタンプの実装とサービスの事例

ＲＦＣなどの標準、具体化されている技術概要や実装事例など、それを用いたサービス事例を紹介する（表 2-3）。

**表 2-3 事例一覧**

| | 実装事例･標準等 | サービス事例 |
|---|---|---|
| シンプル･プロトコル | RFC 3161 | なし |
| リンキング･プロトコル | リニア･リンキング･プロトコル<br>ツリー構造のリンキング･プロトコル | Secure Seal(NTT Data) |
| 分散プロトコル | 秘密分散方式の応用 | なし |

#### (1) シンプル・プロトコル： RFC3161

**① 技術概要**

このタイムスタンプは、電子文書のハッシュ値に対し、認証機関（ＣＡ）から公開鍵証明書の発行を受けたＴＳＡが、正確な時刻とともにデジタル署名したものである。電子文書のハッシュ値（**messageImprint**）の他に、要求メッセージにおける各項目を下記に示す。

```
TimeStampReq ::= SEQUENCE  {
   version              INTEGER  { v1(1) },
   messageImprint       MessageImprint,
   reqPolicy            ＴＳＡPolicyID            OPTIONAL,
   nonce                INTEGER                 OPTIONAL,
   certReq              BOOLEAN                 DEFAULT FALSE,
   extensions           [0] IMPLICIT Extensions  OPTIONAL  }

   MessageImprint ::= SEQUENCE  {
        hashAlgorithm               AlgorithmIDentifier,
        hashedMessage               OCTET STRING  }
```

応答メッセージには、タイムスタンプ（**timeStampToken**）の他に、下記の項目がある。

```
TimeStampResp ::= SEQUENCE  {
   status                  PKIStatusInfo,
   timeStampToken          TimeStampToken     OPTIONAL  }

   PKIStatusInfo ::= SEQUENCE {
      status        PKIStatus,
      statusString  PKIFreeText     OPTIONAL,
      failInfo      PKIFailureInfo  OPTIONAL  }
```

上記項目「**PKIStatusInfo**」の詳細は、RFC2510 （Internet X.509 Public Key Infrastructure Certificate Management Protocols）の「3.2.3」に記載されている。

### ② ＴＳＡへの要求事項

このＲＦＣではＴＳＡに対して次のような運用上の要件を示している。

- 信頼できる（正確な、正しい）時刻源を用いて、各タイムスタンプにそれを含める。
- タイムスタンプの発行先のクライアントを特定するＩＤ情報を入手しない。
- クライアントから有効なタイムスタンプの要求情報を受信した場合、早急にタイムスタンプを生成する。
- タイムスタンプをデータに直接付与するのではなく、データのハッシュ値に付与する。
- 利用されるハッシュ関数が十分な安全性を有しているか否か判断する。
- タイムスタンプが付与されるデータについて内容の解読などをしない。
- デジタル署名の生成鍵は、ＴＳＡとして専用のものを用い、公開鍵証明書にその旨明記する。

なお、詳細は RFC3161 を参照のこと。

### ③ 検証方法

タイムスタンプ「TimeStampToken」は、利用者の電子文書のハッシュに対するＴＳＡのデジタル署名であるため、この検証は通常のデジタル署名検証手順と同じ方法で行われる。有効な状態のＴＳＡの証明書や検証パス上の各ＣＡの証明書、失効リストなどが必要となる。

## (2) リンキング・プロトコル

### ① 技術概要

このタイムスタンプには、対象となっている文書のハッシュ値、時刻情報、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）が含まれている。特長として全てのタイムスタンプは、これまで生成されたタイムスタンプに依存するように生成され、その過程で発生したリンク情報（ハッシュ値）を定期的に新聞等に公表することで、システムの安全性を確保している。

主なリンキング・プロトコルとして、リニア・リンキング・プロトコルとツリー構造のリンキング・プロトコルがある。また複数のＴＳＡを利用したリンキング・プロトコルなどもある。

リニア・リンキング・プロトコルでは、利用者がデータのハッシュ値（$H_n$）をＴＳＡに送ると、ＴＳＡでは、①最新のリンク情報（$L_{n-1}$）と送られたハッシュ値を元に次のリンク情報（$L_n$）を生成（$L_n=h(H_n,n,L_{n-1})$）する（図 2-4）。②生成されたリンク情報（$L_n$）がタイムスタンプとして利用者に渡される。その際このリンク情報に時刻（Ｔ）がデジタル署名の形式で付加される場合もある。なおｎは受け付け番号である。このように次々とリンク情報を生成し続け、一定間隔で最新のリンク情報（$L_N$）を新聞などに公開することで、途中全てのリンク情報に改ざんが無いことを証明している。

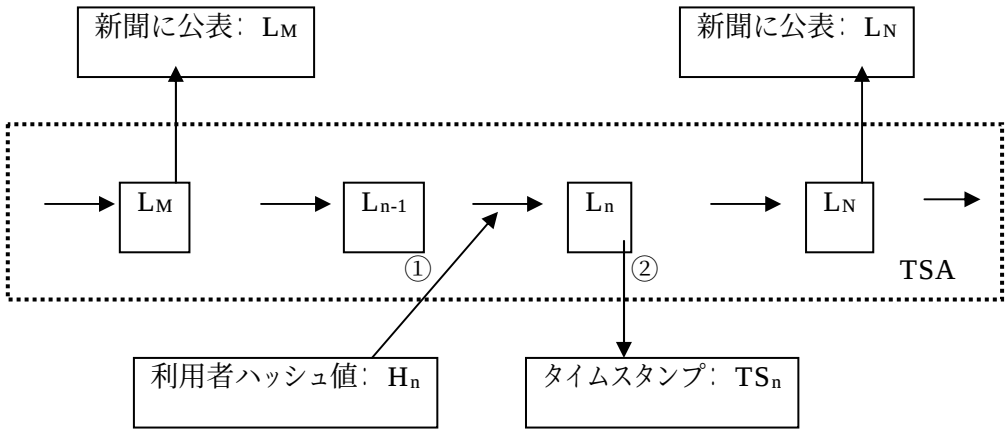



**図 2-4 リニア・リンキング・プロトコル**

このタイムスタンプの検証は、利用者データのハッシュ値をもとに、タイムスタンプ生成と同じ手順により行われる。最終的には新聞等に公開されたリンク情報と比較することで内容の正しさを確認（検証）できる。

一方、ツリー構造のリンキング・プロトコルでは、一定時間（ラウンド）でリンク情報（SRHi）を生成する。このリンク情報は、同一ラウンド内において複数の利用者データのハッシュ値をツリー状に結合・ハッシュ化して生成される（図 2-5）。生成されたリンク情報と、時刻、リンク情報を生成する過程のハッシュ値全てを含む「公証記録（タイムスタンプ）」を生成し利用者に渡される。

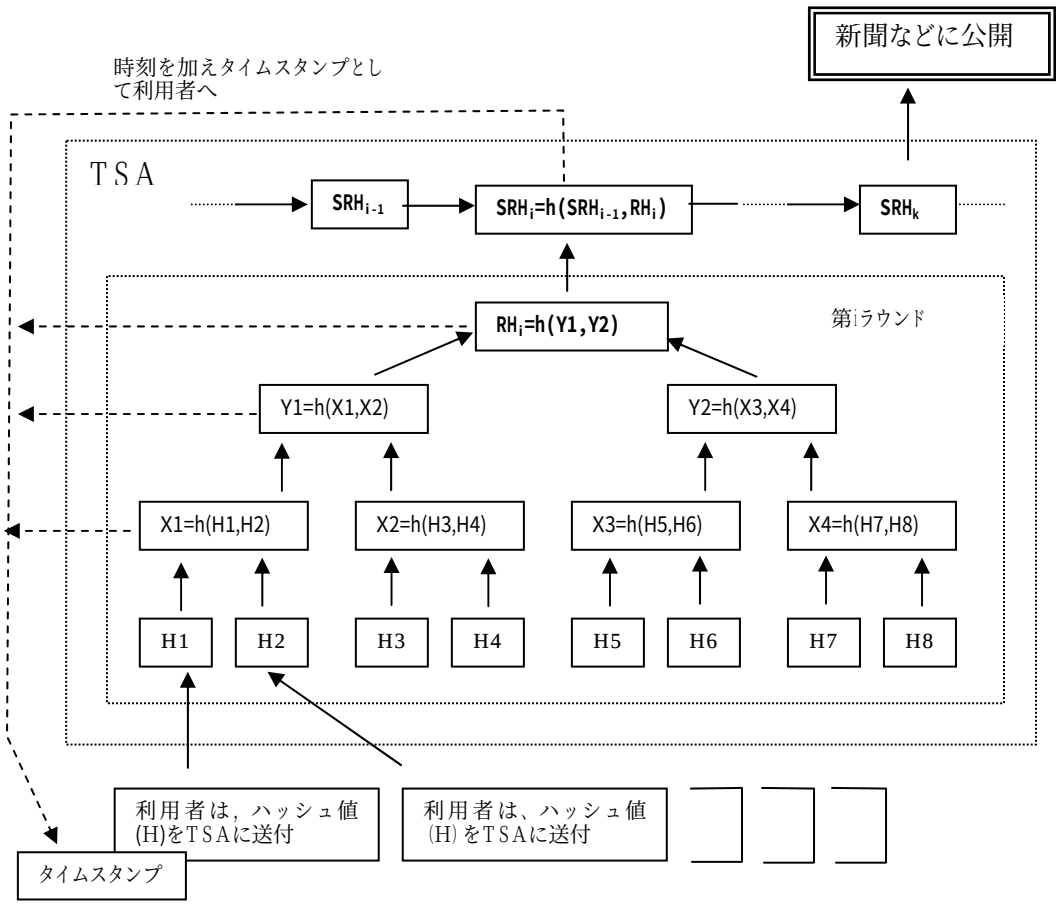



**図 2-5 ツリー構造のリンキング･プロトコル**

### ② サービス事例

ツリー構造のリンキング・プロトコルを利用した「digital Notary」サービス（Surety.com 社：米国）が米国において 1992 年に開始された。日本においては「Secure Seal」（ＮＴＴデータ）として利用することができる。このサービス「digital Notary」におけるリンク情報の一部は、ニューヨーク・タイムズ紙に毎週日曜日に公開され、「Secure Seal」では毎週金曜日の日経産業新聞に掲載されている。

このサービスでは、利用者のデータをもとに再生成されたリンク情報と同一ラウンドのリンク情報がセンタに保存されており、これの比較をもってタイムスタンプの検証としている。専用の端末アプリケーションを使い、元のデータと受領したタイムスタンプなどと共にセンタに問い合わせ、検証結果を受信する。

このタイムスタンプサービスの長期保存対応は、ハッシュの耐衝突を考慮し、ＭＤ５とＳＨＡ１を組み合わせて実現している。ハッシュアルゴリズムの陳腐化後は切り替え（再タイムスタンプ生成）が必要になるが、更新手順などは確立されている。更新時期について具体的な予定は、現在のところない。

### (3) 分散プロトコル

#### ① 技術概要

このタイムスタンプは、複数のＴＳＡによるタイムスタンプを含み、それを一つにまとめたものとなる。タイムスタンプ生成時は複数ＴＳＡが必要となるが、一つにまとめてあるため検証は一度で済む。RFC3161 のセキュリティ考察事項において複数のＴＳＡを利用する記載（RFC3161 4.2 ）があるが、これは利用者の手元に複数のタイムスタンプがあることを示し、分散プロトコルと言わない。分散プロトコルは、あくまで一つのタイムスタンプに結合したものを言う。

#### ② 分散時刻署名システム

ＮＴＴにより研究されたタイムスタンプシステムで、部分秘密鍵を利用した分散時刻署名を生成する（図 2-6）。

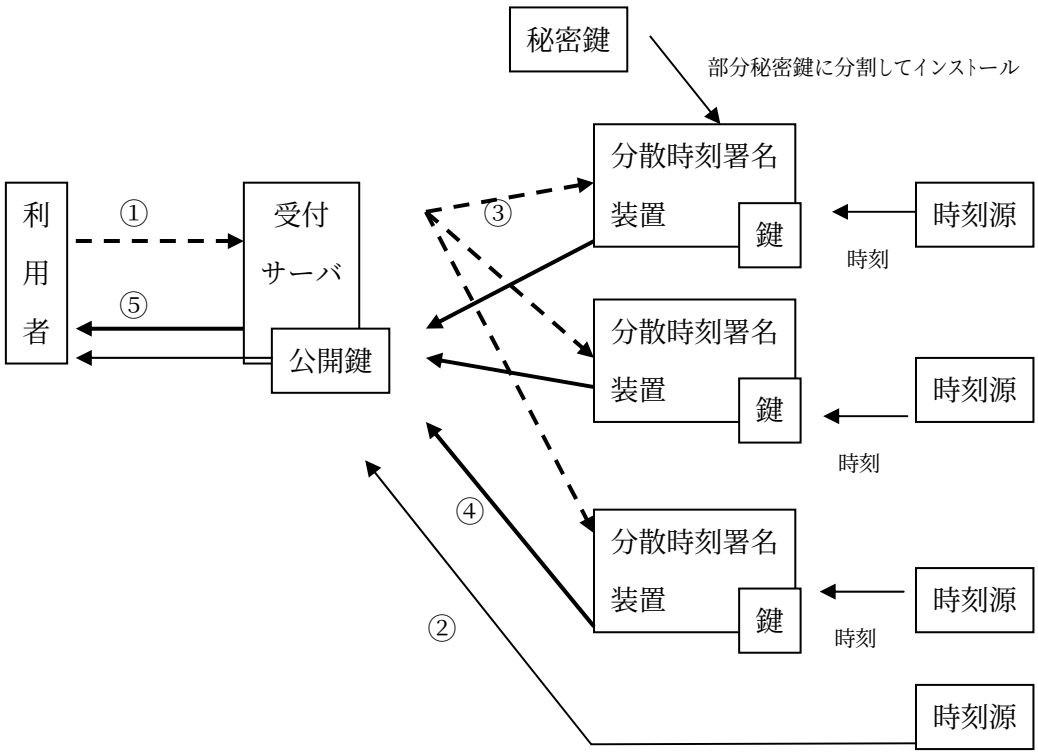



**図 2-6 分散時刻署名システム**

時刻

・①　利用者は電子文書のハッシュを受け付けサーバに送信する。

・②～③　受け付けサーバはハッシュを三つにコピーし時刻を付加して、分散時刻署名装置に送信（タイムスタンプ要求）する。

・④　部分秘密鍵でデジタル署名されたタイムスタンプを受け付けサーバに返す。

・⑤　各署名をマージして利用者に返信する。署名生成の閾値が２であれば最低２つ分散署名装置からの応答があればタイムスタンプの生成ができる。

各々の分散時刻署名装置を独立したＴＳＡとして運用することで、タイムスタンプの安全性を確保できる。

### ③　検証方法

各分散署名装置の秘密鍵に対応するＴＳＡの公開鍵は一つであり、タイムスタンプのデジタル署名を検証することで、タイムスタンプの検証ができる。

### 2.2.2 署名検証技術

#### 2.2.2.1 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もとめられる。

公開鍵証明書の有効性を検証するためには(1)信頼された認証局が存在することを前提とし、(2)証明書の検証を行う。

(1) **信頼された認証局**(Trusted-CA)**の存在**

証明書は信頼された認証局から発行されることが、証明書の検証の前提となる。認証局は通常複数存在して、階層構造や相互認証の構造をとる。

階層構造の場合、上位認証局が下位認証局の公開鍵を署名する(証明書を発行する)ことで信頼関係を構築している。この証明書のチェーンを認証パスと呼ぶ。認証パスの最上位の認証局が信頼された認証局となり、この認証局の証明書は自分自身の秘密鍵で署名する。

相互認証の場合、最上位認証局を頂点とする階層構造を取らず、認証局と認証局が相互に認証する構造で、相互証明書を発行する。さらに、ブリッジ認証局を配置し、相互証明書をブリッジ認証局と認証局間で発行する構造もある。

(2) **証明書の検証方法**

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 ポリシの検証**

証明書に関するポリシ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。

- 署名者の公開鍵証明書
- 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
- 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)
- 検証時刻
- 署名ポリシの合意情報

### 2.2.2.2 署名検証技術の概要

ここでは、2.2.2.1 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。

- 証明書の有効性検証技術
- 署名の有効性検証技術
- データ検証技術
- 認証パス構築技術および認証パス検証技術

**(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 [10]として規定されている。さらに OCSP と同様の考え方で各種ドラフトが検討されている状況である。(SCVP [11]、OCSP を改訂する RFC256bis [12] がインターネットドラフトとなっている。)

### (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) [13]、ＤＳＶ (インターネットドラフト) [14]がある。ＤＶＣＳについては、2.2.2.3(3)で説明する。

### (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ＤＶＣＳ（RFC3029）については、2.2.2.3(3)で説明する。

### (4) パス検証技術およびパス構築技術

署名検証、証明書の検証を行うための、証明書のパスを構築しそれを検証するための技術である。新たにＤＰＶ、ＤＰＤがインターネットドラフトとしてＩＥＴＦで議論されている。ただし、パス検証・パス構築技術は、署名検証、証明書の検証を行う時に利用する技術として重要な技術であるが、本ガイドライン執筆の段階ではＩＥＴ

Ｆでの議論も初期段階であるため、参考にとどめることとする。

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 2-4 にまとめる。

**表 2-4 署名検証関連技術**

| 項番 | 技術分類<br>(標準化状況) | 証明書の<br>有効性検証 | 署名の<br>有効性検証 | データ<br>検証 | 認証パス<br>構築 | 認証パス<br>検証 |
|---|---|---|---|---|---|---|
| 1 | CRL/ARL | ○ | | | | |
| 2 | OCSP(RFC2560) | ○ | | | | |
| 3 | OCSP(RFC2560bis) *1 | ○ | | | | |
| 4 | DVCS(RFC3029) | ○ | ○ | ○ | ○ | ○ |
| 5 | SCVP *1 | ○ | | | ○ | ○ |
| 6 | DSV *1 | | ○ | | | |
| 7 | DPV *1 | ○ | | | | ○ |
| 8 | DPD *1 | | | | ○ | |

*1:'02/2 時点でインターネットドラフトとして議論されている。

### 2.2.2.3 代表的な署名検証技術の詳細

表 2-4 に説明した技術から、次の３つの技術を詳しく説明する。

- ＣＲＬ/ＡＲＬ
- ＯＣＳＰ
- ＤＶＣＳ

また、最近検討が始まったＤＳＶ、ＤＰＶ、ＤＰＤについても参考として簡単に紹介する。

#### (1) ＣＲＬ／ＡＲＬ

認証局が発行した証明書を失効する(無効にする)必要が発生した場合、失効した証明書情報を集めたリスト(ＣＲＬ/ＡＲＬ)を作成し検証者に公開する。公開の手段としてディレクトリサーバに登録したり、検証者に配布したりする等がある。

対象となる証明書によりＣＲＬ/ＡＲＬと区別している。

- ＣＲＬ(Certificate Revocation List)：エンドエンティティの証明書の失効リスト。
- ＡＲＬ(Authority Revocation List)：認証局の証明書の失効リスト。

署名検証の考え方の説明においては両者を区別せず、以降、ＡＲＬも含めてＣＲＬと記述することとする。

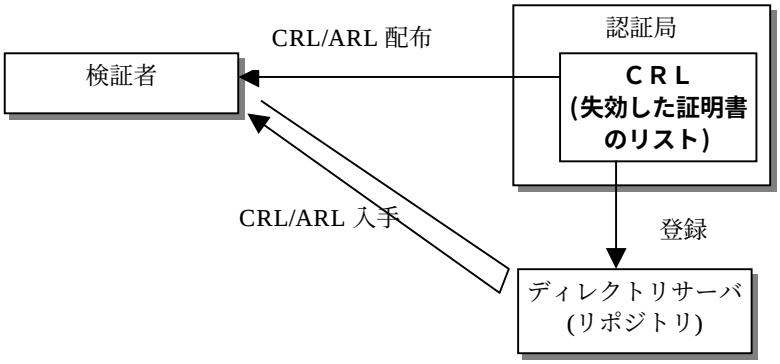



**図 2-7 CRL/ARL 技術の概要**

**① ＣＲＬの目的**

認証局の発行した証明書は、通常規定された有効期間の間、有効な証明書として利用されるが、何らかの要因により有効期間が満了する前に失効する必要が発生することがある。

失効するケースとしては、次のようなことが考えられる。

･秘密鍵を盗まれてしまった(漏洩してしまった)

･秘密鍵を紛失してしまった(格納していたICカードやPCが壊れてしまった)

･証明書を利用することを禁止しなければならない(会社を退職したりサービスを停止したりするとき)

証明書を検証する場合、ＣＲＬを参照して、利用する証明書が失効していないことを確認する必要がある。

**② ＣＲＬの形式**

ＣＲＬには次の情報を含み、認証局により署名されている。(主要な情報のみ示す)

- ＣＲＬを発行した認証局の名称
- ＣＲＬ発行時刻
- 次回ＣＲＬの発行時刻
- 失効した証明書のリスト
- 失効理由

### ③　ＣＲＬの作成と公開

認証局は定期的にＣＲＬを作成し公開する。公開の方法としてはディレクトリサーバに公開することが一般的であるが、証明書を利用者に配布する運用が行われることもある。

### ④　ＣＲＬの利用

証明書利用者が証明書を検証する場合に参照する。検証時には最新のＣＲＬを認証局(ディレクトリサーバ)から入手して参照することが大切である。検証毎にＣＲＬを入手することは効率的でないため、定期的にＣＲＬを入手するか、ＣＲＬに記載された次回ＣＲＬの発行時刻を確認して、入手する契機を決める等の運用を行う。

### ⑤　ＣＲＬの問題点

ＣＲＬを利用する方式は、証明書利用者側にＣＲＬを格納することとなるため、次の懸念がある。

- ＣＲＬを入手するための運用が必要となる
- 証明書の失効から、その失効を反映したＣＲＬを検証者が入手するまでに時間差がある。
- 失効した証明書を特定する情報を含むため、失効した証明書数に比例してＣＲＬが大きくなり、入手するためのダウンロードの効率が悪い。
- 大きなＣＲＬを参照するための処理のオーバヘッドが大きくなることがある

### ⑥　問題点への対応

⑤に示したＣＲＬが大きくなる問題点に対してはデルタＣＲＬを利用する方式がある。デルタＣＲＬは前回作成したＣＲＬに対する差分のみを格納したＣＲＬであり、毎回ＣＲＬ入手時にＣＲＬ全体をダウンロードせずにデルタＣＲＬのみをダウンロードする方式である。

また、その他の問題点に対しては、次に示すＯＣＳＰを利用することで解決することができる。

### (2) ＯＣＳＰ

ＯＣＳＰ(Online Certificate Status Protocol)は証明書検証時に証明書の有効性をオンラインで確認するためのプロトコルである。ＯＣＳＰレスポンダは認証局またはディレクトリサーバからＣＲＬを入手して証明書の検証を行うのが一般的である。このときデルタＣＲＬを利用することも多い。

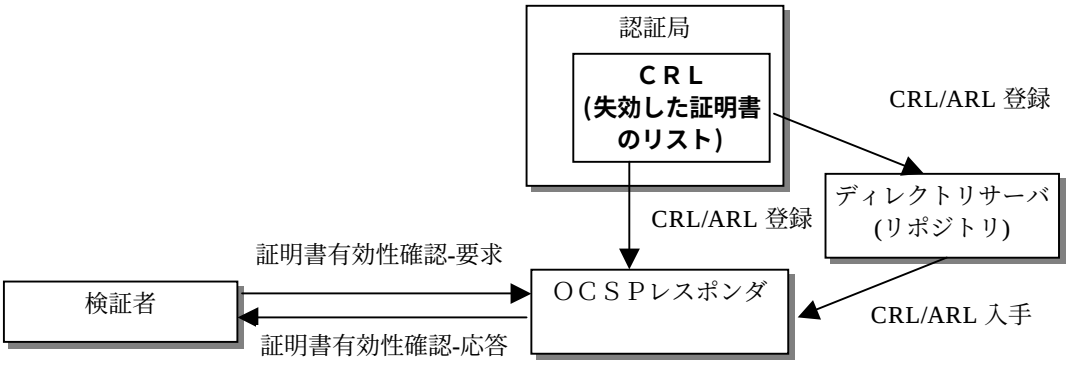



**図 2-8 OCSP技術の概要**

#### ① ＯＣＳＰの目的

ＣＲＬによる証明書の検証が検証者自身でＣＲＬを参照して行うことに対して、証明書の検証サービスを提供するサーバを配置し、オンラインで証明書の有効性を確認するために利用される。

#### ② ＯＣＳＰの構成

ＯＣＳＰレスポンダとＯＣＳＰクライアントで構成される。証明書検証を依頼するＯＣＳＰクライアントから、検証の要求を送信し、ＯＣＳＰレスポンダから検証の結果をクライアントに応答する構成である。

#### ③ ＯＣＳＰのメッセージ形式

ＯＣＳＰの検証要求を行うメッセージには以下の情報が含まれる。

- 検証要求者名
- 検証要求をする証明書

これを受信したＯＣＳＰレスポンダは以下の応答メッセージに以下の情報を含める。

- ＯＣＳＰ要求の応答状態(正常の場合のみ、次の証明書の状態が意味を持つ)
- 証明書の状態(有効、失効、不明等の証明書の状態)
- ＯＣＳＰレスポンダ名

これらの情報をＯＣＳＰレスポンダにより署名している。

### (3) 署名検証,データ検証(DVCS)の概要

#### ① ＤＶＣＳの目的

検証者は CRL やＯＣＳＰを利用することで証明書の有効性検証が可能となり、有効性検証を実行する負担が軽減される利点がある。ただし、デジタル署名された電子文書を長期間保存することに対しては以下のような点も解決しなければならない。

- ＣＲＬやＯＣＳＰでは現時点における証明書の有効性の検証しかできない。(過去の時点の有効性を検証できない)
- 認証パスに存在する認証局の全ての証明書について有効性検証を行わなければならない。

このような点を解決し、さらにいくつかの機能を実現するための技術がＤＶＣＳ(Data Validation and Certification Server Protocols)としてＩＥＴＦで RFC3029 として提案されている。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。

- 過去の時点における証明書の有効性を検証可能となる
- 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。
- その文書を所有していたことを証明可能となる。
- 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)
- 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。

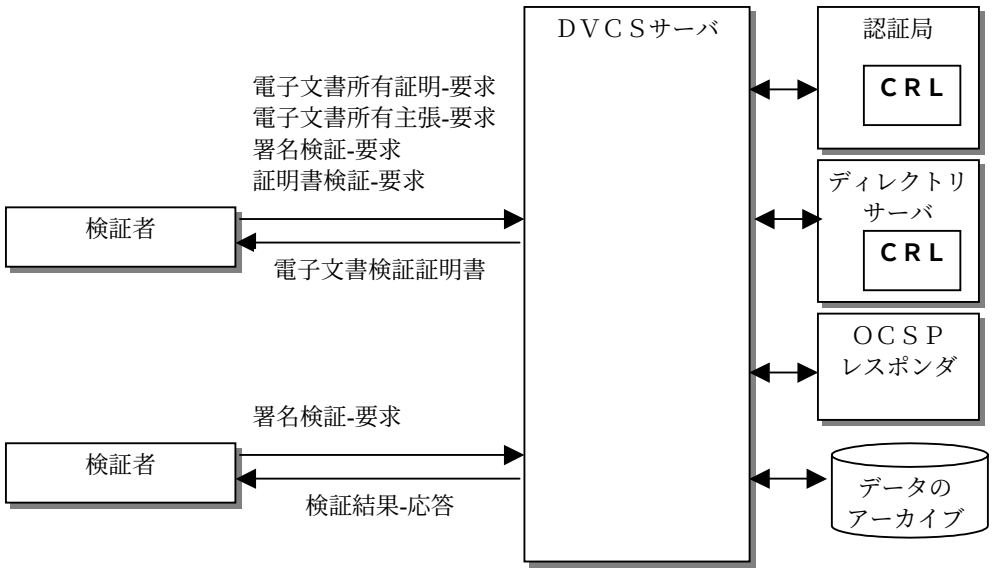



**図 2-9 DVCS技術の概要**

### ② ＤＶＣＳの構成

ＤＶＣＳサーバと検証者(クライアント)で構成される。検証者はＤＶＣＳサーバの提供する機能(サービス)を指定して要求メッセージを送信し、ＤＶＣＳサーバから結果を検証者に応答する構成である。

ＤＶＣＳサーバは各機能を提供するために、ＤＶＣＳサーバ自身が検証者としてＯＣＳＰレスポンダやディレクトリサーバを利用する構成をとることも考えられる。

### ③ ＤＶＣＳのメッセージ形式

ＤＶＣＳに対してサービスの要求を行うメッセージには以下の情報が含まれる。(主要な情報のみ記述。また、要求するサービスに応じて含む情報は異なる。)

- サービス要求者名
- 検証要求をする電子文書
- 所有を主張する電子文書
- 検証を要求する証明書
- 電子文書所有の認証
- 要求時間

これを受信したＤＶＣＳサーバは以下の応答メッセージに以下の情報を含める。

- 電子文書、証明書の検証の結果
- 電子文書検証証明書
- 証書チェーン

これらの情報をＤＶＣＳサーバにより署名している。

#### ④ 長期保存のためのＤＶＣの利用

ＤＶＣにより、ＤＶＣＳの機能を長期に渡り利用するためには、ＤＶＣ自身がＤＶＣＳサーバによりデジタル署名された文書であることを考慮する必要がある。ＤＶＣＳサーバをＴＴＰととらえ、そのデータアーカイブ機能を利用したり、発行されたＤＶＣの有効性が失われる前に、再度ＤＶＣＳに対してＤＶＣの再発行を依頼したりする等が必要となる。

### (4) ＤＳＶ,ＤＰＶ,ＤＰＤ技術

署名の検証のための技術としてＩＥＴＦで議論されている機能にＤＳＶがある。ＤＳＶはＤＰＶ、ＤＰＤと合わせて議論されているもので、表 2-5 に機能概要を示す。

**表 2-5 ＤSV,DPV,DPD の機能概要**

| 項番 | 分類 | 機能 |
|---|---|---|
| 1 | DSV: Delegated Signature Validation | 署名検証機能 |
| 2 | DPV: Delegated Path Validation | 認証パス検証機能 |
| 3 | DPD: Delegated Path Discovery | 認証パス構築機能 |

#### ① 検討の背景・効果

現在検討されている機能は、今後登場するモバイルクライアントでの利用も想定して、次のような効果を狙っている。

A. サーバに各種機能をオフロードすることで複雑な処理をクライアントに実装せず、負荷を軽減できる。
B. クライアントで実装するよりも、サーバで処理する方が処理時間が短くなる。

### ② ＤＳＶ,ＤＰＶ,ＤＰＤプロトコル概要

代表としてＤＳＶ機能の概要を表 2-6 および図 2-10 にしめす。DPV、DPD 機能も同様に、クライアント・サーバ型で実行する機能である。

**表 2-6 DSV 機能の概要**

| 項番 | 機能 | 機能概要 |
|---|---|---|
| 1 | DSV | DSV サーバはある時点(時刻)の署名の有効性を検証する<br>DSV サーバは Time-stamp token を提供する |

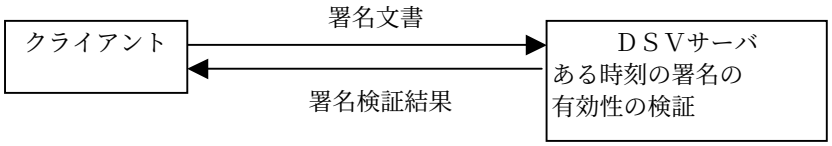



**図 2-10 DSV モデル(概要)**

## 2.2.2.4 各技術の纏め

上記で説明した代表的な３つの技術の関係等を纏める。

**表 2-7 技術比較**

| 機能項目 | | CRL | OCSP | DVCS |
|---|---|---|---|---|
| 機能 | 証明書有効性検証 | ○ | ○ | ○ |
| | 電子文書の署名検証 | × | × | ○ |
| | 電子文書の検証 | × | × | ○ |
| | パス検証 | × | × | ○ |
| | パス構築 | × | × | ○ |
| 証明書検証の有効性<br>(失効が発生した場合に、時間的に遅れなく失効事象が検証者に反映されるかの評価) | | ＣＲＬ更新の間隔により最新の状態を検証できないことがある。<br>また、検証者のＣＲＬ入手の頻度に依存する。 | ＯＣＳＰレスポンダの失効情報更新の頻度に依存する。<br>ＣＲＬを利用する場合には、ＣＲＬと同等。 | ＤＶＣＳサーバの失効情報更新の頻度に依存する。<br>ＣＲＬを利用する場合には、ＣＲＬと同等。 |
| 証明書検証時の検証者の負荷 | | ＣＲＬの定期的な入手が必要<br>ＣＲＬの保管･管理が必要<br>検証処理の実装必要 | 認証パスの構築･検証が必要 | 認証パスの構築･検証が不要 |
| 証明書検証時のﾄﾗﾌｨｯｸ | | なし。ただし、CRL 入手のﾄﾗﾌｨｯｸが発生しうる。 | サーバとの要求･応答ﾄﾗﾌｨｯｸあり | サーバとの要求･応答ﾄﾗﾌｨｯｸあり |
| 証明書検証における、検証時点 | | 現時点 | 現時点 | 過去の時点も検証可能 |

ＰＫＩの実利用の経過を見ると、当初ＣＲＬ方式が使用され、その後ＯＣＳＰ等のオン

ラインによる検証が行われるようになってきた。

上記比較からわかるように、ＤＶＣＳが検証者にとって最も使いやすいサービスを提供することができると考えられる。ＤＶＣＳは実験的範囲での検討であり、この仕様のまま実用化されるかは不明であるが、今後、認証局の増加や証明書利用の場面が増えるにつれて、検証者側に負担の少ない方式が普及すると考えられる。

### 2.2.2.5　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 2-11 を参照。

- 署名時点：署名者がデジタル署名を作成した時刻
- 検証時点：検証者がデジタル署名を検証した時刻
- 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。

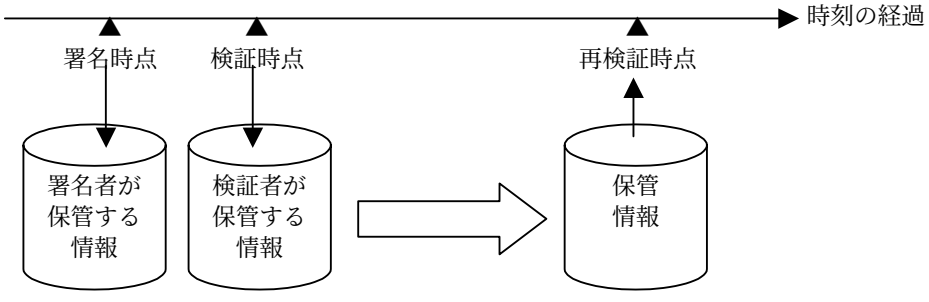



**図 2-11 署名検証に関係する時点**

#### (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、2.2.2.1 では、次の５つを示した。

●署名者の公開鍵証明書

●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

●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)

●検証時刻

●署名ポリシの合意情報

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 2-8 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 2-8 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。<br>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、<br>･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<br>･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<br><br>※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。<br>例えば、<br>･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<br>･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 2-11 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。

### 2.2.3 長期署名フォーマット

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 [15]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 2-12 に示す。

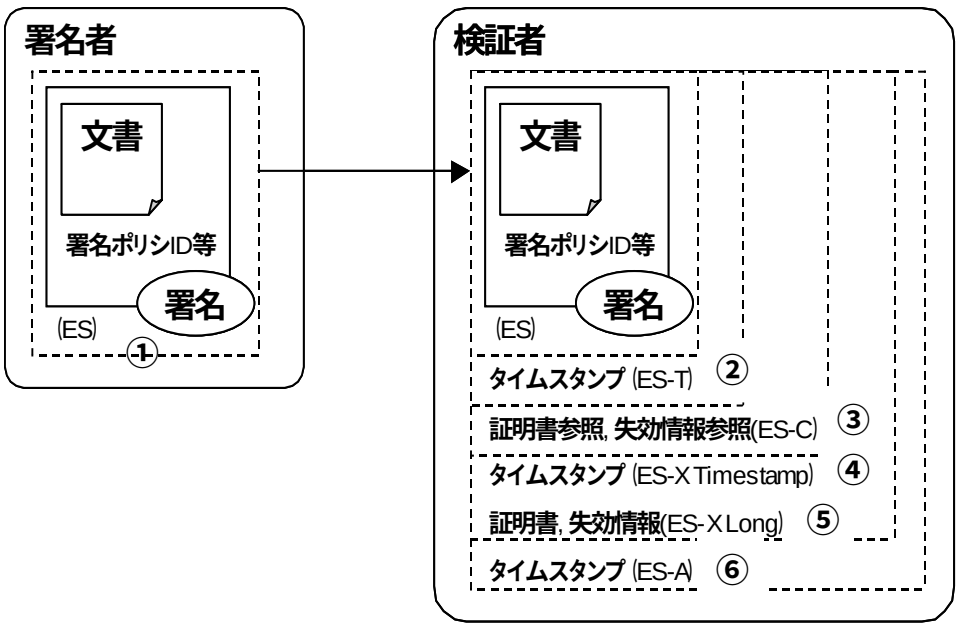



図 2-12 長期署名フォーマット

① ＥＳ

署名者は、デジタル署名を生成する時、以下の情報を含めたデジタル署名を生成する。これらの情報は署名フォーマット内の署名属性に含まれるので改ざんを防ぐことができる。

- 署名ポリシＩＤ
- コミットメントタイプ

- 署名者の証明書ＩＤ
- 署名者の属性
- 署名場所
- 署名時刻
- コンテンツフォーマット

② ＥＳ－Ｔ

署名者もしくは検証者は、署名生成時刻もしくは署名検証時刻を明確にするため、ＥＳの署名値のハッシュ値をタイムスタンプ発行局に送付しタイムスタンプを取得する。そして、取得したタイムスタンプ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。

このタイムスタンプにより、デジタル署名の存在を後で否認されることから保護することが可能となる。また、デジタル署名の検証に必要となる情報が危殆化する前にデジタル署名が存在していたことを保証できるため、危殆化した情報を用いてデジタル署名の有効性を検証することが可能となる。

③ ＥＳ－Ｃ

検証者は、署名再検証に必要な情報を明確にするため、以下の情報をデジタル署名に追加する。これらの情報は署名フォーマット内の非署名属性に含まれる。

- 認証パス上の公開鍵証明書へのリファレンス
- 公開鍵証明書の失効情報へのリファレンス

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。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。

⑤ ＥＳ－Ｘ Long

署名再検証に必要な情報を保存するため、以下の情報をデジタル署名に追加する。これらの情報は署名フォーマット内の非署名属性に含まれる。

●認証パス上の公開鍵証明書

●公開鍵証明書の失効情報

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。

このタイムスタンプは、ＥＳ－Ｃを生成したときに用いられたアルゴリズムや鍵が弱くなるか、危殆化した時や、前回取得したタイムスタンプが有効期限に到達した場合に取得する。

### 2.2.4　署名ポリシ

署名ポリシは、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。署名ポリシは、RFC3125 にもなっている。この署名ポリシの記述形式には以下の２つの形式がある。

●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を可読にした形式。

●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。

署名ポリシを用いたデジタル署名の生成と検証は、図 2-13 に示すようになる。

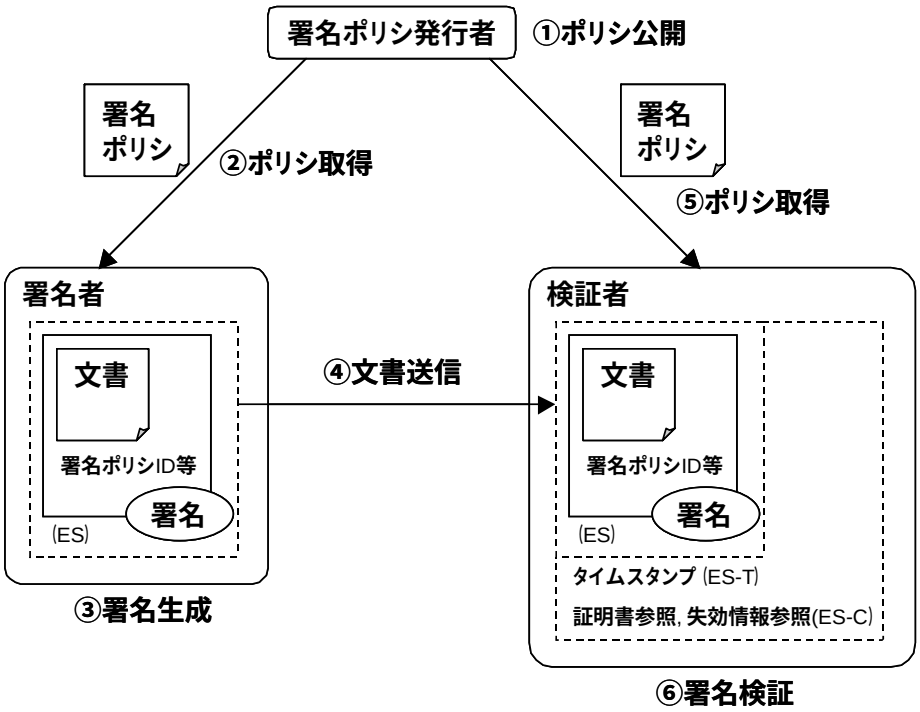



図 2-13 署名ポリシを用いたデジタル署名の生成と検証

① 署名ポリシ発行者は、署名ポリシを公開する。

② 署名者は、要件に合致する署名ポリシを署名ポリシ発行者から入手する。

③ 署名者は、署名ポリシ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、署名ポリシを特定する署名ポリシＩＤ等の情報を含めたデジタル署名を生成する。

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ＩＤを参照し、この署名ポリシを署名ポリシ発行者から入手する。

⑥ 検証者は、署名ポリシ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。

### 2.2.4.1 署名ポリシ（SignaturePolicy）

署名ポリシは、RFC3125 に以下のとおり規定されている。

```
SignaturePolicy ::= SEQUENCE {
    signPolicyHashAlg         AlgorithmIdentifier,
    signPolicyInfo            SignPolicyInfo,
    signPolicyHash            SignPolicyHash           OPTIONAL
    }

SignPolicyInfo ::= SEQUENCE {
    signPolicyIdentifier      SignPolicyId,
    dateOfIssuer              GeneralizedTime,
    policyIssuerName          PolicyIssuerName,
    fieldOfApplication        FieldOfApplication,
    signatureValidationPolicy SignatureValidationPolicy,
    signPolExtensions         SignPolExtensions        OPTIONAL
    }
```

デジタル署名を有効とみなすための規則は signatureValidationPolicy に規定される。

### 2.2.4.2 署名検証ポリシ（SignatureValidationPolicy）

署名検証ポリシは、RFC3125 に以下のとおり規定されている。

```
SignatureValidationPolicy ::= SEQUENCE {
    signingPeriod     SigningPeriod,
    commonRules       CommonRules,
    commitmentRules   CommitmentRules,
    signPolExtensions SignPolExtensions OPTIONAL
    }

CommonRules ::= SEQUENCE {
    signerAndVerifierRules    [0] SignerAndVerifierRules    OPTIONAL,
    signingCertTrustCondition [1] SigningCertTrustCondition OPTIONAL,
    timeStampTrustCondition   [2] TimeStampTrustCondition   OPTIONAL,
    attributeTrustCondition   [3] AttributeTrustCondition   OPTIONAL,
    algorithmConstraintSet    [4] AlgorithmConstraintSet    OPTIONAL,
    signPolExtensions         [5] SignPolExtensions         OPTIONAL
    }

CommitmentRules ::= SEQUENCE OF CommitmentRule

CommitmentRule ::= SEQUENCE {
    selCommitmentTypes            SelectedCommitmentTypes,
    signerAndVerifierRules    [0] SignerAndVerifierRules    OPTIONAL,
    signingCertTrustCondition [1] SigningCertTrustCondition OPTIONAL,
    timeStampTrustCondition   [2] TimeStampTrustCondition   OPTIONAL,
    attributeTrustCondition   [3] AttributeTrustCondition   OPTIONAL,
    algorithmConstraintSet    [4] AlgorithmConstraintSet    OPTIONAL,
    signPolExtensions         [5] SignPolExtensions         OPTIONAL
    }
```

署名検証ポリシは、コミットメント毎に異なる規則を commitmentRules に、全コミッ

トメントに共通する規則を commonRules に規定することができる。いずれの規則においても以下の項目を規定することができる。

- 署名者と検証者がデジタル署名に添付しなければならない情報
- 署名者の証明書を信頼してもよい条件
- タイムスタンプを信頼してもよい条件
- 署名者の属性を信頼してもよい条件
- 使用可能なデジタル署名のアルゴリズム

#### 2.2.4.3 署名者・検証者規則（SignerAndVerifierRules）

署名者・検証者規則は、RFC3125 に以下のとおり規定されている。

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

SignerRules ::= SEQUENCE {
    externalSignedData          BOOLEAN           OPTIONAL,
    mandatedSignedAttr          CMSAttrs,
    mandatedUnsignedAttr        CMSAttrs,
    mandatedCertificateRef  [0] CertRefReq        DEFAULT  signerOnly,
    mandatedCertificateInfo [1] CertInfoReq       DEFAULT  none,
    signPolExtensions       [2] SignPolExtensions OPTIONAL
    }

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

署名者・検証者規則は、署名者がデジタル署名を生成する時にデジタル署名に添付しなければならない情報を signerRules に、検証者がデジタル署名を検証した時にデジタル署名に添付しなければならない情報を verifierRules に規定することができる。

#### 2.2.4.4 署名証明書信頼条件（SigninigCertTrustCondition）

署名証明書信頼条件は、RFC3125 に以下のとおり規定されている。

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

CertificateTrustTrees ::= SEQUENCE OF CertificateTrustPoint
```

```
CertificateTrustPoint ::= SEQUENCE {
    trustpoint              Certificate,
    pathLenConstraint   [0] PathLenConstraint   OPTIONAL,
    acceptablePolicySet [1] AcceptablePolicySet OPTIONAL,
    nameConstraints     [2] NameConstraints     OPTIONAL,
    policyConstraints   [3] PolicyConstraints   OPTIONAL
    }

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

署名証明書信頼条件は、デジタル署名の検証に用いる署名者の証明書を信頼してもよい条件を規定することができる。証明書を信頼するには、認証パス検証を行い証明書が有効であると判断しなければならない。この認証パス検証に必要となる入力パラメータを signerTrustTrees に、失効情報を入手する方法を signerRevReq に規定することができる。

### 2.2.4.5 **タイムスタンプ信頼条件**（TimeStampTrustCondition）

タイムスタンプ信頼条件は、RFC3125 に以下のとおり規定されている。

```
TimeStampTrustCondition ::= SEQUENCE {
    ttsCertificateTrustTrees [0] CertificateTrustTrees OPTIONAL,
    ttsRevReq                [1] CertRevReq            OPTIONAL,
    ttsNameConstraints       [2] NameConstraints       OPTIONAL,
    cautionPeriod            [3] DeltaTime             OPTIONAL,
    signatureTimeStampDelay  [4] DeltaTime             OPTIONAL
    }
```

タイムスタンプ信頼条件は、署名検証時刻を特定するタイムスタンプを信頼してもよい条件を規定することができる。署名生成時刻から次回ＣＲＬ発行時刻までの時間を cautionPeriod に、署名生成時刻からタイムスタンプを取得しなければならない期限を signatureTimeStampDelay に規定することができる。時刻の関係を図 2-14 に示す。

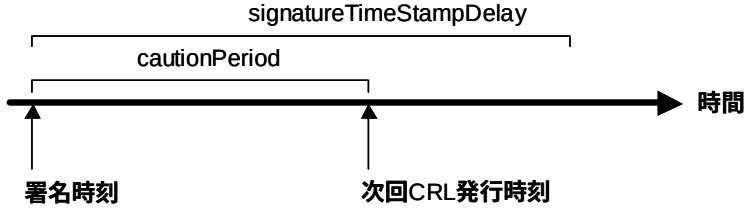



**図 2-14 時刻に関する規則の関係**

### 2.2.4.6 **属性信頼条件**（AttributeTrustCondition）

属性信頼条件は、RFC3125 に以下のとおり規定されている。

```
AttributeTrustCondition ::=  SEQUENCE {
    attributeMandated             BOOLEAN,
    howCertAttribute              HowCertAttribute,
    attrCertificateTrustTrees [0] CertificateTrustTrees OPTIONAL,
    attrRevReq                [1] CertRevReq            OPTIONAL,
    attributeConstraints      [2] AttributeConstraints  OPTIONAL
    }
```

属性信頼条件は、署名者の属性を信頼してもよい条件を規定することができる。

### 2.2.4.7 **アルゴリズム制約**（AlgorithmConstraintSet）

アルゴリズム制約は、RFC3125 に以下のとおり規定されている。

```
AlgorithmConstraintSet ::= SEQUENCE {
    signerAlgorithmConstraints  [0] AlgorithmConstraints OPTIONAL,
    eeCertAlgorithmConstraints  [1] AlgorithmConstraints OPTIONAL,
    caCertAlgorithmConstraints  [2] AlgorithmConstraints OPTIONAL,
    aaCertAlgorithmConstraints  [3] AlgorithmConstraints OPTIONAL,
    tsaCertAlgorithmConstraints [4] AlgorithmConstraints OPTIONAL
    }
```

アルゴリズム制約は、それぞれ以下のデジタル署名に使用可能な署名アルゴリズムを規定することができる。

- 署名者が文書に添付するデジタル署名
- 署名者の証明書のデジタル署名
- 証明書発行者の証明書のデジタル署名
- 属性証明書発行者の証明書のデジタル署名
- タイムスタンプ発行者の証明書のデジタル署名

## 2.3　長期署名フォーマットと署名ポリシのプロファイル

### 2.3.1　長期署名フォーマットのプロファイル

#### 2.3.1.1　推奨署名フォーマット

ETSI TS 101 733（RFC3126）で規定した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定めている。ここでは長期署名保存の観点から、ＥＣＯＭとして推奨する署名フォーマットのタイプを限定することにする。

ＥＴＳＩの署名フォーマットには以下の署名フォーマットのタイプがある。

１）ＥＳ：基本署名フォーマットで署名ポリシ、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（署名時点の確定と署名の否認防止に必要）

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（再検証者は、証明書チェーン上の全ての証明書と、その全ての失効情報[ＣＲＬ、ＡＲＬ、またはＯＣＳＰ応答]にアクセス出来なければならない）

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　１：ＥＳ－Ｃにタイムスタンプをつけたもの

５）ＥＳ－Ｘ　２：ＥＳ－Ｃにある全ての証明書と失効情報の参照値にタイムスタンプを付けたもの

６）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

７）ＥＳ－Ａ：ＥＳ－Ｘにタイムスタンプを付加したもの

ここでは、数年、数十年後にも署名時点の署名の有効性を再検証できるような署名フォーマットを定めることにする。この数年、数十年後には以下のような状態がありそうである。

・再検証者は、もはや署名検証に必要な署名者の証明書や失効情報および認証パス上の全ての証明書（証明書チェーン）や失効情報を収集できないかもしれない。

・証明書チェーン上の証明書の署名鍵（ＣＡ鍵）が危殆化しているかもしれない。
・ＥＳ－Ｃで使用している暗号アルゴリズムが破られているかもしれない。

このような状態においても署名の最初の検証で有効であったことを再検証できるようにするためには、署名ポリシのコピーを保持し、最初の署名検証時点で検証に用いた証明書チェーン上の全ての証明書と、それぞれの証明書の失効情報（ＣＲＬ、ＡＲＬまたはＯＣＳＰ応答）を添付しておかなければならない。したがって、このような用途で長期署名保存を行うためにはＥＳ－Ｘ Long の署名フォーマットが必要になる。従って全ての証拠としての証明書と失効情報を備えた自己完結型の署名フォーマットが必要になる。

またＥＳ－Ｘで使用している署名アルゴリズムが危殆化する怖れがある場合、ＥＳ－Ｘにタイムスタンプを付加する必要がある。したがって、署名フォーマットはＥＳ－Ｘ Long Time-Stamped になる。このフォーマットを用いれば少なくともタイムスタンプの証明書の有効期限までは署名の再検証を可能とする。

このタイムスタンプの有効期限が切れるか、署名アルゴリズムがさらに危殆化する怖れがある場合は、さらに新たなタイムスタンプでカプセルすることで署名の有効性を延長することも必要になる。すなわち署名フォーマットはＥＳ－Ａとして保存しなければならない。

以上の理由から本プロファイルではＥＳ－Ｘ Long Time-Stamped を長期署名検証用の推奨フォーマットにすることにする。最初の署名検証者は署名の検証過程でＥＳ－Ｘ Long Time-Stamped を作成しておかなければならない。

#### 2.3.1.2　署名フォーマットの構文

ＥＴＳＩ署名フォーマットはＣＭＳ（RFC2630）の SinedData をベースとしている。ここでは推奨するＥＳ－Ｘ Long Time-Stamped の構文について概説する。詳しい構文は ETSI TS 101 733 （RFC3126）を参照のこと。

ＥＳ－Ｘ Long Time-Stamped のフォーマットは以下に示される。

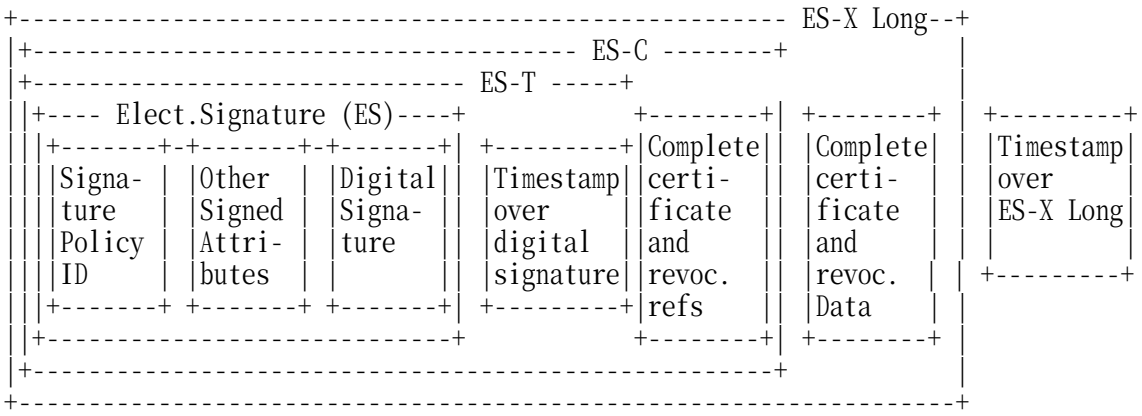



図 2-15 ＥＳ－Ｘ Long Time-Stamped のフォーマット

署名文書自体はＣＭＳの Data Content Type などの任意のオクテッド･ストリングやその他の Signed Content Type などＣＭＳで指定した Content Type が対象である。

ＥＴＳＩの長期署名フォーマットの構文はＣＭＳ SignedData で構成されるために、ここに参考のためにＣＭＳ SinedData 構文を示しておくことにする（RFC2630）。

```
SignedData ::= SEQUENCE {
       version CMSVersion,
       digestAlgorithms DigestAlgorithmIdentifiers,
       encapContentInfo EncapsulatedContentInfo,
                          --カプセル化コンテント情報
       certificates [0] IMPLICIT CertificateSet OPTIONAL,
       crls [1] IMPLICIT CertificateRevocationLists OPTIONAL,
       signerInfos SignerInfos  --署名者情報 }

  DigestAlgorithmIdentifiers ::= SET OF DigestAlgorithmIdentifier

  SignerInfos ::= SET OF SignerInfo
```

カプセル化コンテント情報（EncapsulatedContentInfo）は以下に定義される。

```
EncapsulatedContentInfo ::= SEQUENCE {
       eContentType ContentType,
       eContent [0] EXPLICIT OCTET STRING OPTIONAL }

  ContentType ::= OBJECT IDENTIFIER --署名対象データタイプ
```

署名者情報（SignerInfo）は以下のように定義される。

```
SignerInfo ::= SEQUENCE {
       version CMSVersion,   --CMSバージョン番号(v3)
       sid SignerIdentifier, --署名者識別子
       digestAlgorithm DigestAlgorithmIdentifier,--ハッシュアルゴリズム
       signedAttrs [0] IMPLICIT SignedAttributes OPTIONAL, --署名属性
       signatureAlgorithm SignatureAlgorithmIdentifier,
                                --署名アルゴリズム識別子
       signature SignatureValue, --署名値
       unsignedAttrs [1] IMPLICIT UnsignedAttributes OPTIONAL
                                     --非署名属性 }

  SignerIdentifier ::= CHOICE {
       issuerAndSerialNumber IssuerAndSerialNumber,
       subjectKeyIdentifier [0] SubjectKeyIdentifier }

  SignedAttributes ::= SET SIZE (1..MAX) OF Attribute

  UnsignedAttributes ::= SET SIZE (1..MAX) OF Attribute
```

本ガイドラインで推奨するＥＳ－Ｘ Long Time-Stamped は以下のＥＳ、ＥＳ－Ｔ、ＥＳ－Ｃ、ＥＳ－Ｘ Long、ＥＳ－Ｘ Long Time-Stamped の全てを含むことになる。ここでは本ガイドラインで推奨する各フォーマットで指定すべき項目を述べる。

### (1) 基本署名フォーマット（ＥＳ）

ＥＳは上図に示すように、署名ポリシＩＤ（Signature policy ＩＤ）、他の署名属性（Other Signed Attribute）、署名値（Digital Signature）からなる。

- 署名ポリシＩＤ（Signature policy ＩＤ）
  署名ポリシは署名者と検証者が合意する署名ポリシで（2. 3. 2 参照）、ＯＩＤで規定しＣＭＳの SigndData の SignerInfo type の SignedAttribute（署名属性）に置かれる。
- 他の署名属性（Other Signed Attribute）
  署名属性はＣＭＳからインポートした属性とＥＴＳＩで定めたものがある。これらの属性はすべてＣＭＳの SigndData の SignerInfo type の SignedAttribute（署名属性）に置かれる。最低必要な署名属性には以下のものがある。

＊ Content Type（Data Content Type など）

＊ MessageDigest（署名対象のハッシュ値）

＊ SiningTime（署名者が主張する署名時間：タイムスタンプではない）

＊ SigningCertificate（署名者の証明書ＩＤで ESS Signing Certificate Attribute で定義した ESSCertID（RFC2634））

＊ SignaturePolicyID（署名ポリシＩＤ）

・ 署名値（Digital Signature）

ＣＭＳの SigndData の Signer Info type の SignatureValue で署名対象＋署名属性に対して署名者の署名鍵で署名した値

### (2) ＥＳ－Ｔ

ＥＳ－Ｔのタイムスタンプは、ＴＳＡによってＥＳの署名値（ＣＭＳの Signer Info の Signature Value）に対してタイムスタンプを付けたものである（ＥＳフォーマット全体に対するタイムスタンプではない）。ＥＳの署名値のハッシュ値に対してＴＳＡの SignatureTimeStampToken を得る。この TimeStampToken はＣＭＳの SignerInfo の非署名属性に置かれる。Time Stamp のプロトコルは RFC3161 を使用する。

### (3) ＥＳ－Ｃ（完全証明書および失効情報参照データ）

証明書および失効情報の参照値をＥＳ－Ｔに付加したものである。

・ 完全な証明書参照値

証明書は検証者のトラストアンカーのＣＡ証明書から署名者の証明書を発行するＣＡ証明書（署名者証明書は含まない）までの証明書チェーン上にある全ての完全な証明書の参照値（CompleteCertificateRefs）を収集する。

```
CompleteCertificateRefs ::= SEQUENCE OF OTHERCertID
```

参照値である個々の OTHERCertID は、ＥＳＳ（RFC2364）の SigningCertificate と同じ属性で、証明書のＳＨＡ-1によるハッシュ値である。完全な証明書の参照値はＣＭＳ の SignedData の SinerInfo の非署名属性に置かれる。

・ 完全な失効情報参照値

完全な失効情報参照値（CompleteRevocationRefs）は、上記の証明書チェーン上の全

ての証明書のＡＲＬおよび署名者証明書のＣＲＬ、あるいはそれに対応するＯＣＳＰレスポンスのハッシュ値である。

```
CompleteRevocationRefs ::= SEQUENCE OF CrlOcspRef
```

完全な失効情報の参照値はＣＭＳの SignedData の SinerInfo の非署名属性に置かれる。

### (4) ＥＳ－Ｘ Long　完全な証明書データおよび失効情報データ

ＥＳ－Ｘ Long を構成するには上記ＥＳ－Ｃで参照した全ての証明書（署名者の証明書も含む）および失効情報（ＣＲＬ、ＡＲＬｓまたはＯＣＳＰレスポンス）を収集し、ＣＭＳの SignedData の SinerInfo の非署名属性に置くことにする。

### (5) ＥＳ－Ｘ Long Time-Stamped

ＥＳ－Ｘ Long Time-Stamped のフォーマットは、初めて構成するＥＳ－Ａと同じものである。ＥＳ－Ｘ Long Time-Stamped は、以下のデータオブジェクトを結合したもののハッシュ値に対してタイムスタンプされ、このタイムスタンプ･トークンはＣＭＳの SignedData の SinerInfo の非署名属性に置くことにする。

＊ カプセル化情報の eContent OCTED STRING

＊ 署名属性（signedAttributes）

＊ 署名フィールド

＊ CompleteCertificateRefs 属性

＊ CompleteRevocationRefs 属性

＊ CertifivateValue 属性

＊ RevocationValue 属性

＊ （以前の ArchiveTimeStampToken があれば、これも含む）

上記のようなＣＭＳ構文を構成し、ＥＳ－Ｘ Long Time-Stamped を長期署名保存のフォーマットとすることにする。

このフォーマットはタイムスタンプサーバ（ＴＳＡ）の証明書の有効期限まで有効であるが、この長期署名フォーマットの寿命をさらに延長させるためにはタイムスタンプサーバ（ＴＳＡ）の証明書の有効期限が切れる前に新しいＴＳＡのタイムスタン

プ・トークンを重ねることにしなければならない。

このような長期署名保存に用いるタイムスタンプは十分な強度をもつタイムスタンプ署名鍵でなければならない。ＴＳＡの証明書の有効期限は、強度のある鍵を用いて十分に長い期間有効であるようにすべきである。

例えば、ＲＳＡ鍵相当で 2048 ビット以上を用いＴＳＡ証明書の有効期限を 10 年以上とすることが望まれる。ＴＳＡ証明書の有効期限を 10 年とすれば、最大 10 年の期間署名の有効性が保証される。さらに長期に署名を保証するには、ＴＳＡの有効期限が切れる前にその時点で新しいＴＳＡの署名鍵を使ってタイムスタンプした新しいＥＳ－Ａを作成して署名の有効性を延長させておかなければならない。

### 2.3.2　署名ポリシのプロファイル

ＥＴＳＩの署名フォーマットには署名ポリシを付与することを必須としている。

「署名ポリシ」とは、このポリシの元に署名の有効性を決定するための、署名の生成と有効性検証の一連の規則である。署名ポリシは署名者と検証者が参照する規則でＯＩＤで指定される。

#### 2.3.2.1　署名ポリシの規定法

1)　フリーフォーマットの文書で記述した規則、または署名文書に明示的または暗示的に含まれる規則、または

2)　形式構文(ASN.1)で電子的処理が可能な構文

で規定される。ここでは２）の形式構文のプロファイルを定めることにする。

#### 2.3.2.2　署名ポリシでの関連者

・署名ポリシ発行者

署名の生成と検証のための技術的、手続的規則である署名ポリシを発効する者(TSP)

・署名者

署名ポリシに示された規則に沿って署名したことを約束する者

・検証者

署名ポリシの規則に沿って署名の有効性を検証しなければならない

・再検証者

検証者が行った同じ条件で検証する

#### 2.3.2.3 本署名ポリシプロファイルの方針

ＥＴＳＩの規定した２）のＡＳＮ.１構文の署名ポリシは、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として最低限必要な署名ポリシのプロファイルを規定する。

この署名ポリシ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、署名者と検証者の合意ポリシとして指定するために使われるが、実際は署名アルゴリズムは公開鍵証明書のアルゴリズムで指定されたものが用いられており、検証時点でこのアルゴリズムが確かめられており、ＣＭＳ署名フォーマットにも指定されているため、ここでは特に指定しないことにした。

#### 2.3.2.4 署名ポリシ（SignaturePolicy）プロファイル

(1) **署名ポリシ**

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

署名ポリシを保護するために、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, --この署名ポリシのOID
       dateOfIssue               GeneralizedTime, --ポリシ発行日
       policyIssuerName          PolicyIssuerName, --ポリシ発行者名
       fieldOfApplication        FieldOfApplication, --適用分野
       signatureValidationPolicy    SignatureValidationPolicy,
                                --署名有効性検証ポリシ
       signPolExtensions         SignPolExtensions  OPTIONAL }

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

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

(2) **署名有効性検証ポリシ（**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,   --署名ポリシの適用日
       commonRules           CommonRules,    --署名ポリシの共通規則
       commitmentRules       CommitmentRules, --署名者が約束する規則
       signPolExtensions     SignPolExtensions   OPTIONAL}

SigningPeriod ::= SEQUENCE {
       notBefore    GeneralizedTime, ポリシ適用開始日
       notAfter     GeneralizedTime OPTIONAL–ポリシ適用終了日 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。

① **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（algorithm ConstraintSet）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.

② **コミットメント規則**（CommitmentRules）

CommitmentRules は、この署名ポリシで合意した規則である。

CommitmentRules は、予め定めたコミットメントタイプ（SelectedCommitment Types）を選択する。このコミットメントタイプが特になければ（NULL の場合）、以下の３つのオプションをコミットメントの規則を明確にするために指定しておくことが望ましい。

・SignerAndVerifierRules

・SigningCertTrustCondition

・TimestampTrustCondition

属性証明書の AttributeTrustCondition や署名アルゴリズムの制約はこの推奨ポリシでは指定しないことにする。

```
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 }
```

A. **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 }
```

a) **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 に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    }
```

b) **検証者規則**（VerifierRules）

検証者規則は、このポリシ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
```

B. **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  }
```

a) **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。

b) **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) –署名ポリシ拡張で定義した方法でチェック、 使用しない   }
```

C. **タイムスタンプ信頼点条件**（Time Stump Trust Condition）

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、次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。

### (3) 署名ポリシ拡張

署名ポリシ拡張は、本署名ポリシの各所で独自に定義できるようになっており、ＯＩＤを指定して用いることが出来る。

```
SignPolExtensions ::= SEQUENCE OF SignPolExtn

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

# 3　電子署名文書長期保存システムの実装モデル

## 3.1　基本モデル

### 3.1.1　システム要件

「1.4 有効性を維持するための要件」で、デジタル署名の有効性を維持するための要件として次の 4 項目を挙げた。

（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。

（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、1.4 で述べられた「署名検証時に、署名再検証に必要な情報を明確にする」だけでなく、それらの情報を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

### 3.1.2 モデル構成

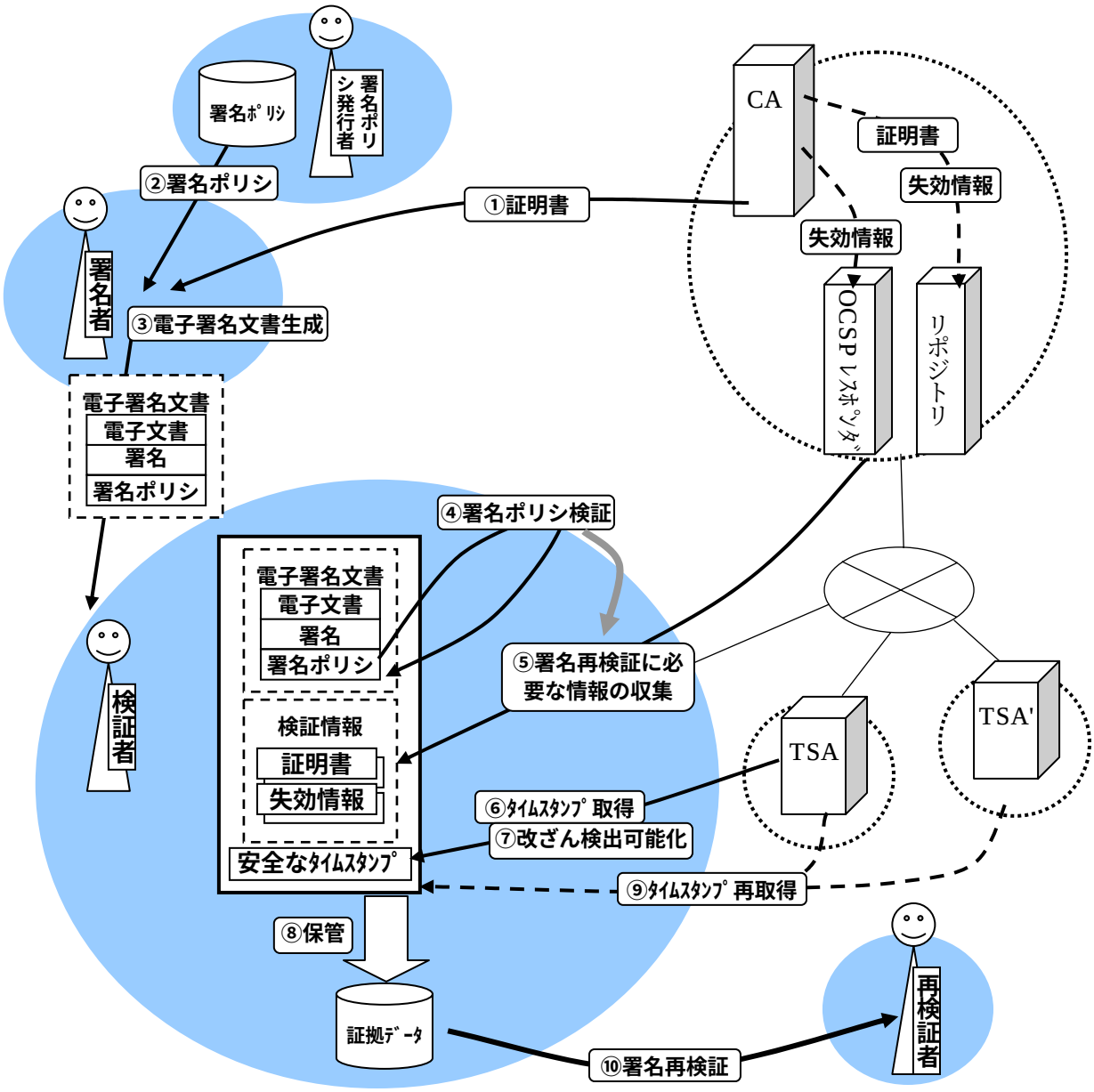



図 3-1 モデル構成

各構成要素の機能は次の通りである.

(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) 署名ポリシ発行者

署名ポリシを発行する。

(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。

### 3.1.3 処理内容

前節であげたモデルシステムの処理内容を次に示す。ただし、処理の順序は以下に示す通りとは限らない。

**表 3-1 基本モデルシステムの処理内容**

| 項目 | 内容 | 対応する要件 |
|---|---|---|
| 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署名ポリシの発行 | 署名ポリシ発行者が署名者に対して署名ポリシを発行する。ただし、署名ポリシ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発行者から発行を受けた署名ポリシに基づき、電子文書に対するデジタル署名を生成する。更にデジタル署名を電子文書及び署名ポリシと結合し、電子署名文書として検証者に転送する。ただし、署名ポリシがデジタル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署名ポリシ検証 | 検証者が、署名ポリシの内容を検証する。署名ポリシの内容により、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 | システム要件 1 |

| | | |
|---|---|---|
| | 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。 | |
| 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。次節以降で、代表的な２例を示す。

## 3.2　セキュアストレージ型モデル

セキュアストレージは、ＰＫＩ（公開鍵基盤）に基づく電子認証・公証技術により、電子署名文書を原本として安全に長期間保存することを可能としたシステムである。

### 3.2.1　特徴

・署名者あるいは検証者の委託を受け、第三者であるエージェントが電子署名文書の長期保存を実施
・検証をオーソリティに委託する必要がなく、クライアント（署名者、検証者、再検証者）自身で実施することが可能
・RFC3161 準拠のタイムスタンプを利用
・署名再検証のための証拠データの格納形式として、ETSI TS 101 733 Electronic Signature Formats を利用

### 3.2.2　構成

(1) 署名者

電子署名文書を生成する。電子署名文書長期保存エージェントへの委託が可能である。

(2) 検証者

電子署名文書の有効性を検証する。電子署名文書長期保存エージェントへの委託が可能である。

(3) 再検証者

電子署名文書の有効性を再検証する。標準的な検証手段を利用することにより、証拠情報を単独で実施できる。

(4) 署名ポリシ発行者

署名ポリシを発行する。

(5) 電子署名文書長期保存エージェント

署名者あるいは検証者の委託を受け、電子署名長期保存システムの４要件の処理を実施する。ただし、内２要件（署名再検証に必要な情報を収集した時刻の添付、署名再検証に必要な情報の改ざん検出可能化）のためのデータを実際に生成するのは、(7)ＴＳＡ／ＴＳＡ’である。データは ETSI TS 101 733 Electronic Signature Formats 準拠の形式で保存される。

(6) 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。証明書失効情報として、ＣＲＬあるいはＯＣＳＰレスポンスを用いる。

(7) ＴＳＡ／ＴＳＡ’

RFC3161 準拠の安全なタイムスタンプを発行する。２要件（署名再検証に必要な情報を収集した時刻の添付、署名再検証に必要な情報の改ざん検出可能化）のためのデータとして利用される。

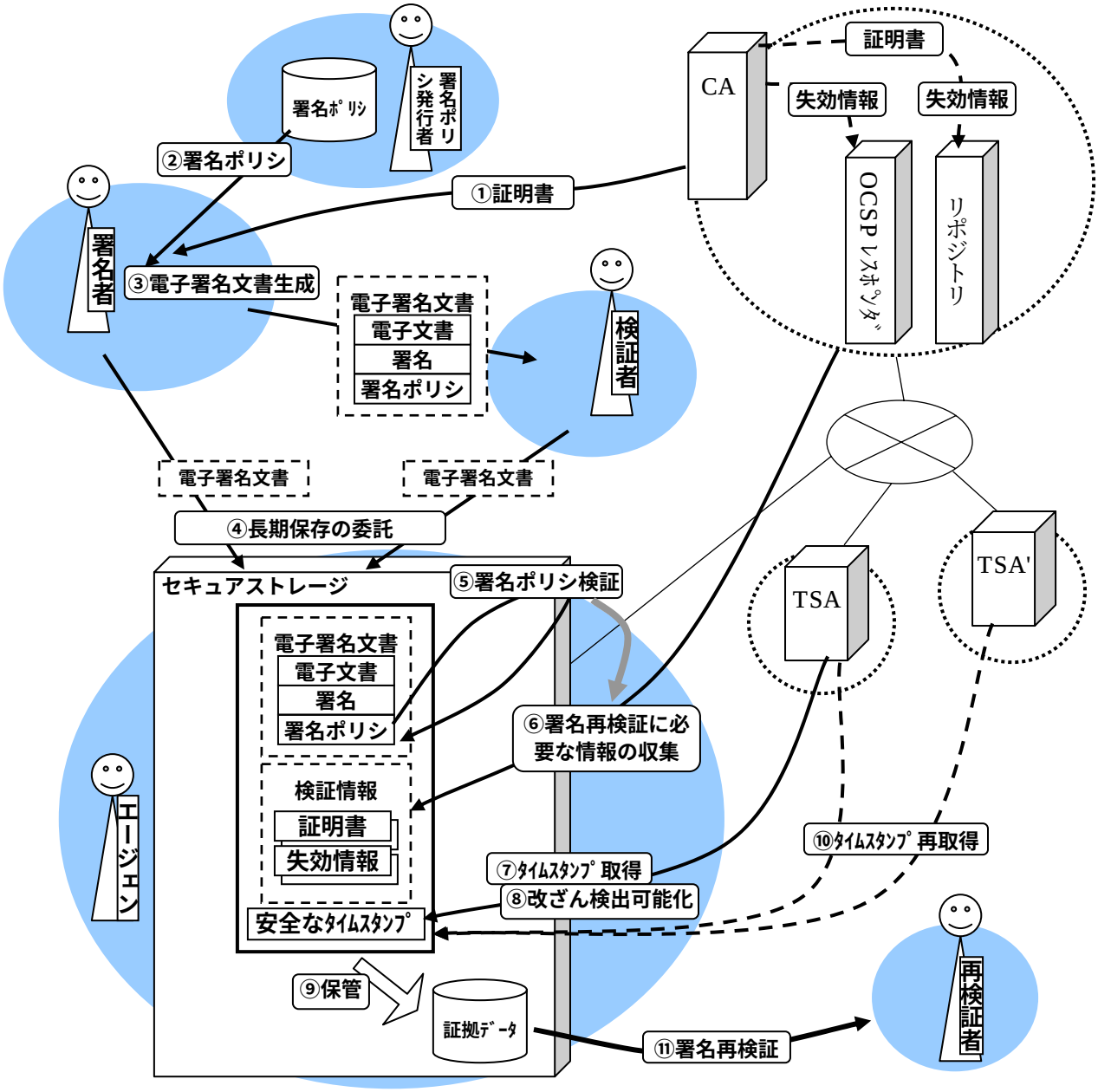



図 3-2　セキュアストレージ型モデルの構成

### 3.2.3 処理

(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の発行を行う。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（ＣＲＬ及びＯＣＳＰレスポンス）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。

(2) 署名ポリシの発行

署名ポリシ発行者が署名者に対して署名ポリシを発行する。

(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発行者から発行を受けた署名ポリシに基づき、電子文書に対するデジタル署名を生成する。更にデジタル署名を電子文書及び署名ポリシと結合し、電子署名文書として検証者に転送する。

(4) 長期保存の委託

署名者あるいは検証者が、電子署名文書長期保存エージェントに対して長期保存を委託する。委託に関わるリクエスト及びレスポンスは、ＨＴＴＰ上に定義されたセキュアストレージサーバの提供するプロトコルを用いる。

(5) 署名ポリシ検証

電子署名文書長期保存エージェントが署名ポリシの内容を検証し、署名再検証に必要な情報を認識する。

(6) 署名再検証に必要な情報の収集

電子署名文書長期保存エージェントが、署名ポリシに基づき、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（ＣＲＬあるいはＯＣＳＰレスポンス）である。

(7) 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電子署名文書長期保存エージェントがＴＳＡよりタイムスタンプを取得する。タイムスタンプ及びタイムスタンプ発行に関わるリクエスト／レスポンスは RFC3161 に準拠している。リクエスト／レスポンスの送受はＨＴＴＰを利用する。

(8) 改竄検出可能化

RFC3161 に準拠する安全なタイムスタンプを利用しているため、タイムスタンプの有

効期限までの間は、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非改竄性を保証できる。

(9) 保管

電子署名文書長期保存エージェントは，前ステップで得た情報を、ETSI TS 101 733 Electronic Signature Formats 準拠の形式（最初のステップで、ＥＳ－Ａまでを作成する。図 3-3 参照。）で保存する。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。

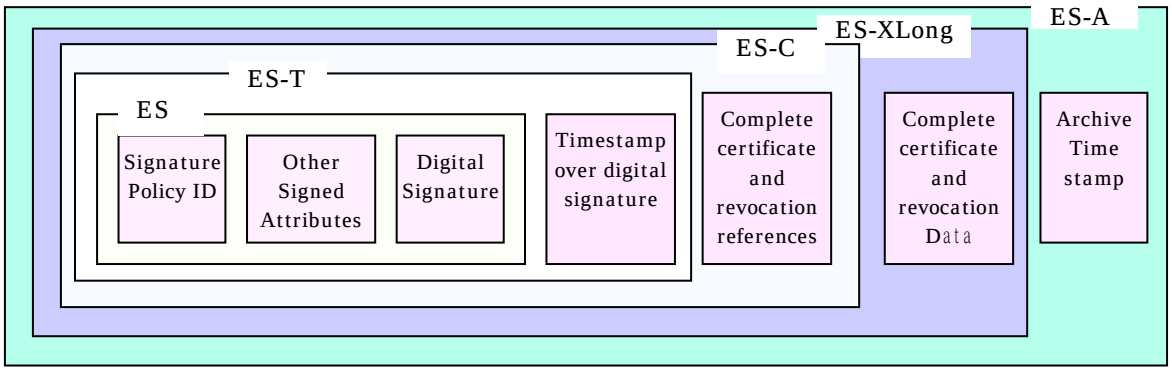



図 3-3 証拠データの形式

(10) タイムスタンプ再取得

RFC3161 に準拠するタイムスタンプはデジタル署名に基づいており、有効期限が存在する。従って、期限到来以前に、電子署名文書長期保存エージェントは、全データ（電子署名文書、前々ステップで収集した情報、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。新たに取得したタイムスタンプを含むデータ全体を ETSI TS 101 733 Electronic Signature Formats 準拠の形式（Archive Timestamp を追加）で保存する。

(11) 署名再検証

再検証者は、図 3-3 に示す形式の証拠データを受取り、図 3-4 に示す手順にしたがって、署名の有効性を検証する。各ステップの検証処理はデジタル署名の検証やタイムスタンプの検証など、標準的な手段で可能である。

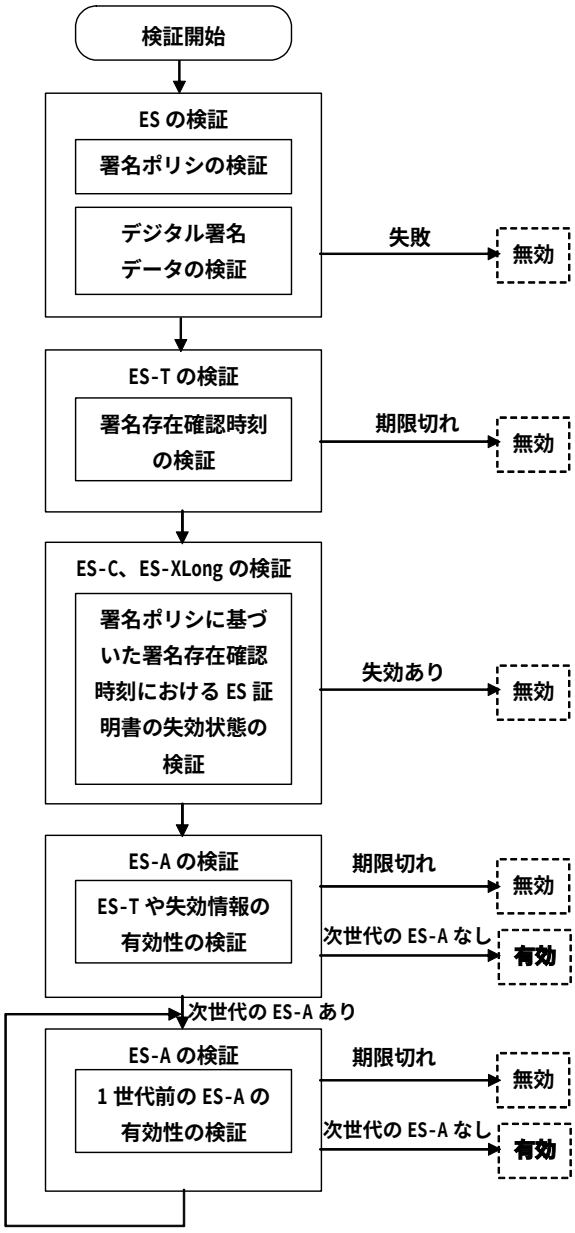



図 3-4 証拠データに基づく署名有効性検証手順概要

### 3.2.4 電子署名長期保存システムの要件との対応関係

電子署名文書長期保存システムの４要件、 (システム要件 1) 署名再検証情報の収集、(システム要件 2) 署名再検証情報の収集時刻の付加、 (システム要件 3) 署名再検証情報の改ざん検出可能化、(システム要件 4) 署名再検証情報を保存、の全てをセキュアストレージ及びタイムスタンプ局が行う。セキュアストレージは、上記(システム要件 1)を実施する署名延長サーバと(システム要件 4)を実施するストレージサーバで構成される。従って、署名者あるいは検証者自身が署名延長サーバやストレージサーバを運用することにより、(システム要件 1)、(システム要件 4)、あるいは(システム要件 1)及び(システム要件 4)を自らが実施することも可能である（表 3-2 参照）。

**表 3-2 電子署名長期保存システム要件との対応**

| パターン | 署名者あるいは検証者 | エージェント | タイムスタンプ局 |
|---|---|---|---|
| 1 | 全て委託 | (システム要件 1),( システム要件 4)を実施 | (システム要件 2),( システム要件 3)を実施 |
| 2 | (システム要件 4)を実施 | (システム要件 1)を実施 | (システム要件 2),( システム要件 3)を実施 |
| 3 | (システム要件 1)を実施 | (システム要件 4)を実施 | (システム要件 2),( システム要件 3)を実施 |
| 4 | (システム要件 1),( システム要件 4)を実施 | 実施せず | (システム要件 2),( システム要件 3)を実施 |

## 3.3　DLMS(Document Lifecycle Management Service)型モデル

Document Lifecycle Management Service(以下ＤＬＭＳ)とは、ＮＴＴデータが提供する「電子文書の安全な流通・保管のためのプラットホーム」の総称である。このサービスでは、電子認証・公証の技術を利用することで、電子署名文書の有効性を長期に渡って保証している。本サービスは、まず建設業界における請負契約の電子的な締結を実現するために、2002 年 2 月から提供を予定している。

### 3.3.1　特徴

・文書作成者が電子署名文書をＤＬＭＳ上のリポジトリに保管し、かつ必要な場合別のユーザ（受信者）に安全な経路を通じて電子署名文書を送信
・予め定義されたユーザの権限に基づき、ＤＬＭＳ上に保管された文書に対して厳密なアクセスコントロールを適用
・一電子文書に対する複数署名をサポートすることで、電子的な契約の成立を実現
・電子署名文書をＤＬＭＳ上のリポジトリに保管すると、SecureSeal（ＮＴＴデータによるタイムスタンプサービス）により自動的にタイムスタンプの取得とハッシュ値の登録を実施
・タイムスタンプの方式としてリンキング・プロトコルを採用
・署名の検証はＤＬＭＳ上で自動的に行われる他、送信者・受信者が実施する事も可能

### 3.3.2　構成

(1) 署名者（文書作成者）

電子的な文書を作成し、ＤＬＭＳ上のリポジトリに保管する。通常、文書には文書作成者のデジタル署名と証明書が添付される。

(2) 検証者（受信者）

ＤＬＭＳから電子署名文書を受信する。ＤＬＭＳ上で電子契約を行う場合は、文書作成者の署名を検証する（受信者が検証する他に、ＤＬＭＳに検証を委託することも可）とともに、受信者の署名・証明書を同文書に付加し、ＤＬＭＳに送り返す。

(3) 仲裁者

電子文書署名の有効性を再検証する。

(4) 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。証明書失効情報として OCSP レスポンスを用いる。

(5) ＴＳＡ

リンキング・プロトコルに基づいたタイムスタンプを発行する。

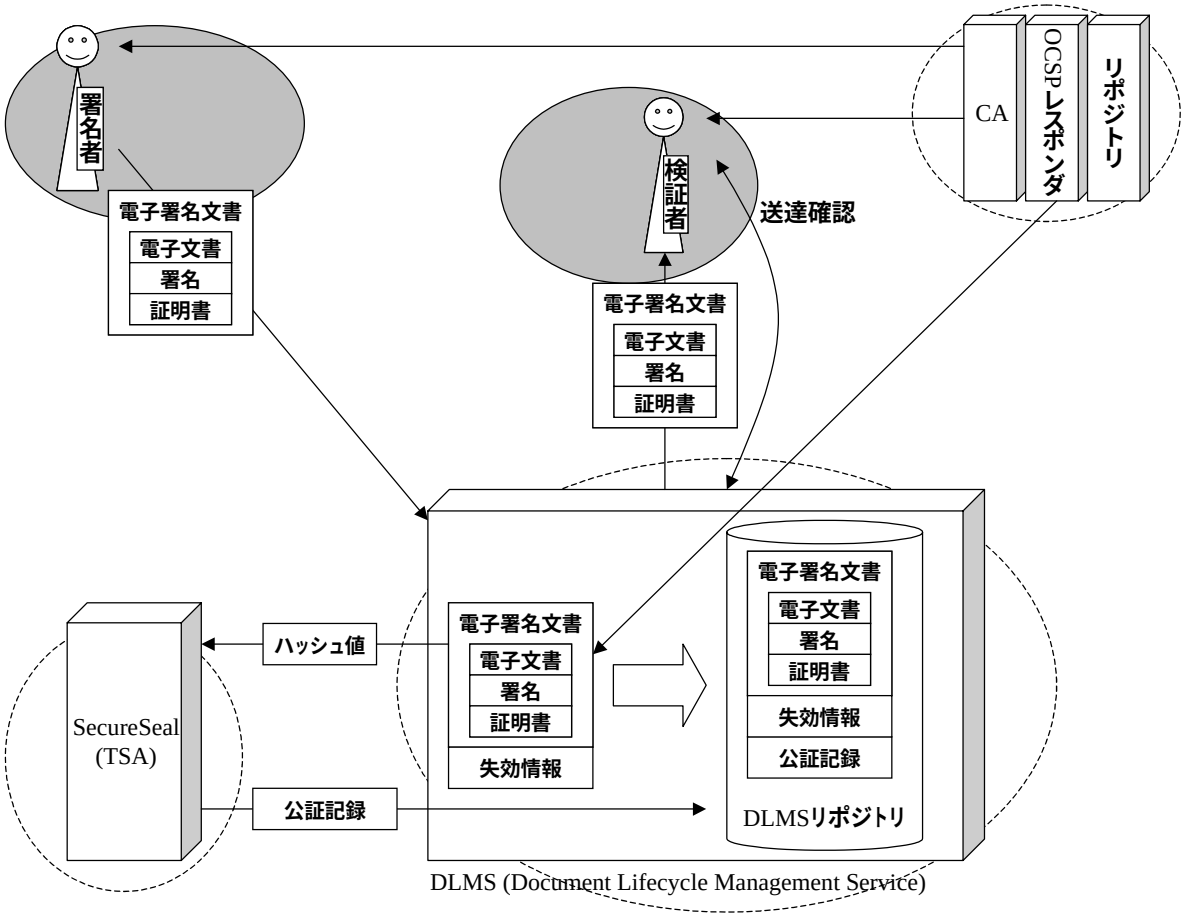



**図 3-5 DLMS 型モデルの構成**

### 3.3.3 基本モデルとの関係

ＤＬＭＳ型モデルは前述の基本モデルと同じ要件を達成するための仕組みであるが、その実現方式や機能分担の考え方にいくつか違いが存在する。ＤＬＭＳ型モデルと本ガイドラインとの関係を、以下に記す。

・電子認証について、ＤＬＭＳが提供する機能は基本的にＯＣＳＰによる署名の検証のみであり、それ以外の認証にまつわる機能は全て外部ＣＡに依存している。署名ポリシや ETSI TS　101 733 Electronic Signature Formats のサポートは、全て外部ＣＡ

がそれらをサポートしているか否かによって決まる。

・本ガイドラインは電子署名文書の長期保存について検討を行ったものであるが、実際に電子署名文書を流通・交換・保存するためには「デジタル署名が長期に渡って検証可能であること」以外の機能要件も、当然必要となる。具体的には、文書流通のためのワークフロー管理、アクセスコントロール、送達確認などが必要であり、ＤＬＭＳ型モデルではそれらの機能をサポートしている。

### 3.3.4 処理

(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の発行を行う。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。Ｘ．５０９かつＯＣＳＰレスポンスをサポートした認証局が利用可能である。

(2) 電子署名文書作成

ＣＡから発行された証明書に基づき、署名者が電子文書に対するデジタル署名を生成する。デジタル署名は、署名生成機能を持つ各種アプリケーションを使用して作成する他に、ＤＬＭＳが提供するプラグインを用いて作成することも可能である。

(3) ＤＬＭＳへの電子署名文書送信

署名者がＤＬＭＳに対して、電子署名文書を送信する。ＤＬＭＳへの接続は、証明書に基づくクライアント認証によりアクセス制御が行われる（他の方式も選択可能）。

(4) ＤＬＭＳによる検証

電子署名文書を受信すると、ＤＬＭＳは署名および証明書のチェックを自動的に行う（ＯＣＳＰレスポンスを使用）。

(5) タイムスタンプ及び監査記録の作成（改ざん検出可能化）

電子署名文書からハッシュ値を作成し、ＴＳＡ（SecureSeal）に登録する。ＴＳＡから公証記録を取得し、電子署名文書と併せて管理することにより、署名生成時刻・文書作成時刻を特定し、かつ署名生成時に証明書が有効であったことを将来に渡って保証する。また、文書の保管（作成）に限らずＤＬＭＳ上で行なわれる全ての操作は、常に監査記録が自動的に作成される。これにより、「いつ/誰が/何をしたのか」特定可能である。

(6) 文書の保管と受信者への通知

電子文書、署名、公証記録と失効情報を結合し、ＤＬＭＳリポジトリ上に保管する。また、その文書の送信先が指定されている場合、受信者に対してＤＬＭＳ上に受信者宛て電子署名文書が保管されていることを通知する。

(7) 受信者による電子署名文書の受信

上記通知を受信した後、受信者はＤＬＭＳから電子署名文書を取得する。受信者が電子署名文書を取得したかどうかは、ＤＬＭＳ上でステータスが管理されており、これにより受信者による否認を防止する。

### 3.3.5 電子署名長期保存システムの要件との対応関係

電子署名長期保存システムの要件として挙げられている４項目（3.1.1 参照）について、ＤＬＭＳ型モデルでの対応関係を以下に記す。

1. 署名再検証情報の収集：ＣＲＬはＤＬＭＳで自動収集（それ以外の情報収集は外部ＣＡと署名者に依存）
2. 署名再検証情報の収集時刻の付加：ＤＬＭＳで実施
3. 署名再検証情報の改ざん検出可能化：ＤＬＭＳで実施
4. 署名再検証情報の保存：ＤＬＭＳで実施

# 4 ケーススタディ（生命保険）

本章では、ケーススタディとして文書を長期に渡り保存する分野の一つである生命保険をとりあげ、３章までで検討した長期保存に向けてのデジタル署名の適用を検討する。4.1 節ではまず生命保険業界での電子化の現状、そして取り扱っている文書について述べる。4.2 節でデジタル署名の導入について検討する。

## 4.1 現状 [16-23]

### 4.1.1 電子化の現状

現状の生命保険に関するサービスでオンラインで提供されているものは、情報提供、申込書請求、保険料試算、保険設計、契約者貸付、配当金引出し、据え置き金引出し、初回保険料の口座振替等である。申し込み用紙には自署・押印が必要になっており、完全な電子化はなされていない。日本だけではなく、海外も含めて生命保険は、電子化がもっとも遅れている分野である。その理由としては、・仕組み（解約返戻金に関する事項や免責事項、予定利率に関する事項等）の理解が難しい、・契約者および被保険者確認、診査等の手続きが必要で非対面が難しい等が挙げられる。また、米国では、日本で挙げた理由の他、・保険に関する規制が州別、・E－Signature Act（連邦法）では、契約の解約は、デジタル署名の効果が否定される例外であるが、解約が比較的多いため紙での管理が増え、保険会社としては二重管理になりコストも増えるといった点が挙げられる。

こうした中、日本では電子化に向けての提言やガイドラインとして、金融庁事務ガイドラインでは、インターネットによる商品販売の取扱として、留意点を出している。

- 申込者の確認（必要に応じて、被保険者の身体の状況に係る告知、診査又は同意）
- 契約申込み他情報の不備及び変質の防止、契約者の保護
- 情報の漏出の防止
- 契約の申込み他契約関係の手続の内容、契約内容及び重要事項の確認、保存措置
- 契約に関し申込者の保険会社との間の事後の行為に対する制約とならない為の措置

また、金融庁のホームページでは、電子金融取引を行うにあたっての注意点として、免許・登録を受けている業者一覧や 外国金融サービス業者が日本でサービスを行う場合の法規制、セキュリティ上の注意点などを掲載している(2001 年 10 月現在)。

課題としては、一般的な電子化する際の考察事項である、セキュリティ、システムダウ

ンに備えて非電子的手段による連絡手段の準備、利用者への電子化の周知、アフターサービスの充実、サービスの使いやすさの配慮（入力ミス、クリックミスを防止させる）の他に、以下が挙げられる。

- 説明義務（保険契約者に当該保険商品の内容を説明する義務）、告知義務（被保険者から現在の健康状態、過去の病歴および職業など必要な情報の告知を受ける義務）をどう確実に行うか、説明を行った（受けた）という事をどうやって確認するか
- 被保険者の同意を確認する方法（特に他人の生命保険契約の場合）
- 売り手（インターネット上の生命保険会社）が適切な販売資格を有しているかの確認方法
- 募集を行っている者がどこから行っているかの特定（海外の場合の規制等）

対象文書図 4-1 に、生命保険に関わる処理のフローと、関連する文書について述べる。なお、ここで示すのは、生命保険用に一般的に使われている文書でありこれが全てではない。また、文書の名称も保険会社によって異なる事がある。

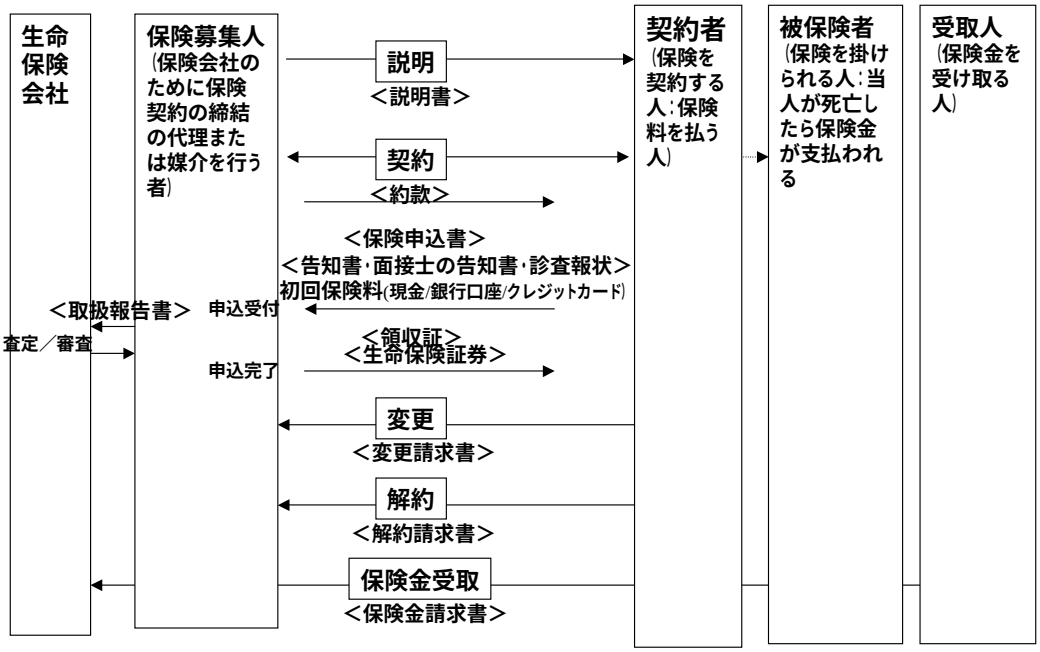



**図 4-1生命保険契約フロー**

なお、個人データに関しては、1987 年 3 月 財団法人金融情報システムセンター（ＦＩＳＣ：The Center for Financial Industry Information Systems）を中心に「金融機関等における個人データ保護のための取扱指針」を策定し、生命保険会社が業務上収集する個人に関するデータの利用・提供・管理および開示等個人データの取扱についてのガイドラインを示している。表 4-1 にそれぞれの文書の特長とガイドラインでの指針を示す。

**表 4-1 文書一覧･FISC での管理指針**

| フェーズ | 文書 | 種別 | 概要 | 保存 | FISC 指針 |
|---|---|---|---|---|---|
| 説明時 | 説明書 | 募集データ | 申込前に保障内容や保険料見積などを説明した資料 | 長期保存しない | ･管理･廃棄手続きを明確<br>･適切な手段により取扱を制限 |
| 契約時[1] | 約款 | 契約データ | 約款を渡す際に、契約者と同じ印鑑を申込書に押してもらう。<br>これを以って契約者は約款を受領したことになる。 | 長期（保険期間･契約期間）[2] | 目的に応じた適切な保存期間を別途定める。<br>基礎統計データとする場合個人データを匿名化し必要な保存期間、保存するものとする。<br>職務権限を有する者がアクセスしているかを判断し得る適切な手段により取扱を制限する。 |
| | 保険契約申込書 | 契約データ | 申込時に契約者に書いてもらう。<br>契約者の署名･印鑑、被保険者の署名･印鑑、保険金受取人を指定してもらう。 | 長期（保険期間･契約期間） | |
| | 保険料支払いに伴う口座振替用紙等 | 契約データ | 支払い方法によって異なるが、銀行振替の場合は契約者の署名と銀行印を押した銀行の振替用紙となる。 | 長期（保険期間･契約期間） | |
| | 領収証 | 契約データ | 第一回保険料領収証を保険料と引き換えに契約者に渡す。 | 長期（保険期間･契約期間） | |

[1] これ以外にも、保険金額が高額の場合には、高額契約取扱報告書や保険金受取人が第三者（一定範囲の親族外など）の場合はチェックリストや面接報告書等を要する場合もある
[2]保険期間終了後も取引履歴の確認や保険計理面、問い合わせなどのため一部を長期保存することもある

| フェーズ | 文書 | 種別 | 概要 | 保存 | FISC 指針 |
|---|---|---|---|---|---|
| | 保険証券 | 契約データ | 保険募集人が契約者に渡す場合、保険会社から直接契約者に渡す場合がある。<br>前者の場合は契約者の受取印が必要。 | 長期（保険期間･契約期間） | |
| | 告知書[3] | 審査データ | 保険募集人または面接士が被保険者に聞いて、被保険者が記入する。 | 長期（保険期間･契約期間） | 審査方針の相違により、管理･保存の取扱が異なるため、各社において適切な保存を定めること<br>･統計調査に関わる保存期間については別途定めること<br>･個人データを含むため最も高い保護水準で管理が必要<br>･情報端末からのアクセスの際には、端末確認／端末利用者の本人確認／設置場所別の端末操作制限<br>･個人データを必要とする業務を第三者に委託する場合、委託契約において、委託先に対して委託元と同水準の個人データ安全保護措置をとることを定める |
| | 面接士の診断書 | 審査データ | 面接士が被保険者に聞いて記入する。 | 長期（保険期間･契約期間） | |
| | 診査報状 | 審査データ | 医師の診断が必要 | 長期（保険期間･契約期間） | |
| | 民間信用情報調査機関の企業データ | 審査データ | 企業保険の場合のみ必要 | 長期（保険期間･契約期間） | |

[3]告知書・面接士の診断書・診査報状どれを以って査定するかは保険金額に応じて定まる

| フェーズ | 文書 | 種別 | 概要 | 保存 | FISC 指針 |
|---|---|---|---|---|---|
| 契約内容変更時 | 変更請求書 | 契約データ | 契約者の署名と保険会社に届け出た印鑑が必要。その他、契約者の印鑑証明書、保険料領収証、保険証券が必要。保険会社は、証書の裏面に変更があった旨を記述することもある。 | 長期（保険期間･契約期間） | |
| 解約時 | 解約請求書 | 契約データ | 契約者の署名と保険会社に届け出た印鑑が必要。契約者の印鑑証明書、保険料領収証、保険証券が必要。 | 長期（保険期間･契約期間） | |
| 保険金受け取り時 | 保険金請求書 | | 保険金受取人の署名と印鑑が必要。保険証券、保険金受取人の印鑑証明書が必要。 | 長期（保険期間･契約期間） | |

その他、契約の諸手続には、契約者の印鑑証明書、戸籍謄本（抄本）、住民票、運転免許証、旅券、健康保険証など役所発行の公的書類が必要となるが、本検討では対象外とする。

表 4-2 に、各文書の概要と、記名、署名、押印の必要性を示す。

**表 4-2 各文書概要**

| 文書種類 | 記述者 | 記述内容（主なもの） | 署名･記名･押印 |
|---|---|---|---|
| 説明書 | 生命保険会社 | ･保険に対する説明 | 生命保険募集人名 |
| 約款 | 生命保険会社 | ･生命保険の保障に関する各条項 | 生命保険会社 |
| 保険契約申込書 | 契約者 | ･契約者名、年齢、性別<br>･被保険者名、年齢、性別<br>･被保険者と契約者の関係<br>･被保険者の職業、国籍<br>･保険金受取人名<br>･保険内容 | 契約者署名、押印<br>被保険者署名、押印 |

| 文書種類 | 記述者 | 記述内容（主なもの） | 署名･記名･押印 |
|---|---|---|---|
| | | ･契約者連絡先 | |
| 告知書 | 被保険者 | ･健康状態、（配偶者･子に対する健康状態） | 被保険者署名<br>生命保険募集人署名、押印 |
| 取扱報告書 | 生命保険募集人 | ･生命保険募集人の署名、押印<br>･被保険者の情報<br>･契約者の情報<br>･保険会社事務員の各項目確認チェック | 生命保険募集人署名、押印<br>生命保険会社担当、責任者受付印 |
| 第一回保険料領収証 | 生命保険募集人 | ･保険種類、金額 | 生命保険募集人署名、押印、社印 |
| 生命保険証券 | | ･証券番号<br>･契約日<br>･契約者名、被保険者名、保険金受取人名<br>･保険種類、保険金額、保険期間、保険料払込期間、保険料支払い方法 | 社印･社長署名、社長印<br>契約者印<br>被保険者印 |
| 変更請求書 | 契約者 | ･証券番号<br>･契約者名<br>･契約内容の変更･訂正事項 | 契約者署名、押印<br>（被保険者署名、押印[4]）<br>生命保険会社担当、責任者受付印 |
| 解約請求書 | 契約者 | ･証券番号<br>･契約者名 | 契約者署名、押印<br>生命保険会社担当者記名、担当者受け付け印、責任者受付印 |
| 保険金請求書 | 保険金受取人 | ･証券番号<br>･保険金受取人氏名 | 保険金受取人署名、押印、<br>生命保険会社担当、責任者受付印 |

### 4.1.2 保険契約における署名・押印

本節では、保険契約における署名・押印の考え方（解釈）を示す。

#### (1) 契約者・被保険者・保険金受取人の署名・押印

３者が何らかの行為を行う際に、保険会社（＝取引の相手方）に対して自署・押印する。

**① 契約者・保険金受取人の場合**

[4] 変更内容によって必要な場合がある

- 何らかの行為の申し出者が、契約者・保険金受取人本人であることを保険会社に対して証明する
- 保険契約申込・契約内容変更請求・保険金請求等の保険会社に対する形成権行使・請求権行使としての確定的な意思表示と解する
- 申込書、請求書等書面上の申し出内容（意思表示の具体的内容）に相違がないことの加重追認と解する

**② 被保険者の場合**

- 契約者が行う保険契約申込や契約内容変更請求を了知･承諾していることの確定的な意思表示と解する
- 契約者が行う保険契約申込や契約内容変更請求等を了知･承諾している者が、被保険者本人であることを保険会社に対して証明する

### (2) 保険募集人・保険会社の署名・押印

保険事務の大量処理や処理効率の観点から、契約者・被保険者・保険金受取人（＝取引の相手方）に対し、行為の都度に自署・押印することはない（一部に例外あり）。契約申込や諸請求を承諾・履行後に、取引の相手方への通知等の施策をもって承諾等の意思表示とする方法が一般的。

**① 保険募集人の場合**

- 契約に関する各種手続きについて、当該保険募集人本人が正当なる権限の基に媒介した事実を保険会社に対して表示していると解するのが妥当
- 領収証における署名・押印は、正当なる権限を有する当該保険募集人が正当なる領収証を用いて金銭収受を履行したことを、契約者に対して明示するとともに、保険会社に対して表示していると解するのが妥当

**② 保険会社の場合**

- 一般的には、契約者・被保険者・保険金受取人（＝取引の相手方）に対する自署・押印行為はない。予め書面上に印刷された記名押印は多数あり
- 各種申込書・請求書上に押印されている印鑑は以下の意味をもつと解釈するのが妥当

A. 書類受付や処理実行等の事務記録としての押印

B. 事務分掌や決裁等の社内規程に基づく承認行為の記録としての押印

表 4-3 に、文書毎に必要な署名、記名、押印をまとめる。

**表 4-3　文書と署名･押印一覧**

| 対象 / 文書 | 契約者 | | 被保険者 | | 保険金受取人 | | 保険募集人 | | 保険会社 |
|---|---|---|---|---|---|---|---|---|---|
| | 署名 | 押印 | 署名 | 押印 | 署名 | 押印 | 署名 | 押印 | 記名押印 |
| 説明書 | | | | | | | ○ | ○ | |
| 約款 | *1 | | | | | | | | ○ |
| 保険契約申込書 | ○ | ○ | ○ | ○ | | | | | |
| 告知書 | | | ○ | | | | | | |
| 取扱報告書 | | | | | | | ○ | ○ | ○*2 |
| 領収証 | | | | | | | ○ | ○ | ○ |
| 生命保険証券 | | | | | | | | | ○ |
| 変更請求書 | ○ | ○ | △*3 | △*3 | | | | | ○*2 |
| 解約請求書 | ○ | ○ | | | | | | | ○*2 |
| 保険金請求書 | | | | | ○ | ○ | | | ○*2 |

*1 保険契約申込書にて受取確認印
*2 受付者、所属長の印（事務記録としての自署･押印であり契約者等に対する意思表示ではない）
*3 変更内容によって必要なものがある

図 4-2 に、主な文書について、署名、検証･確認の流れを示す。

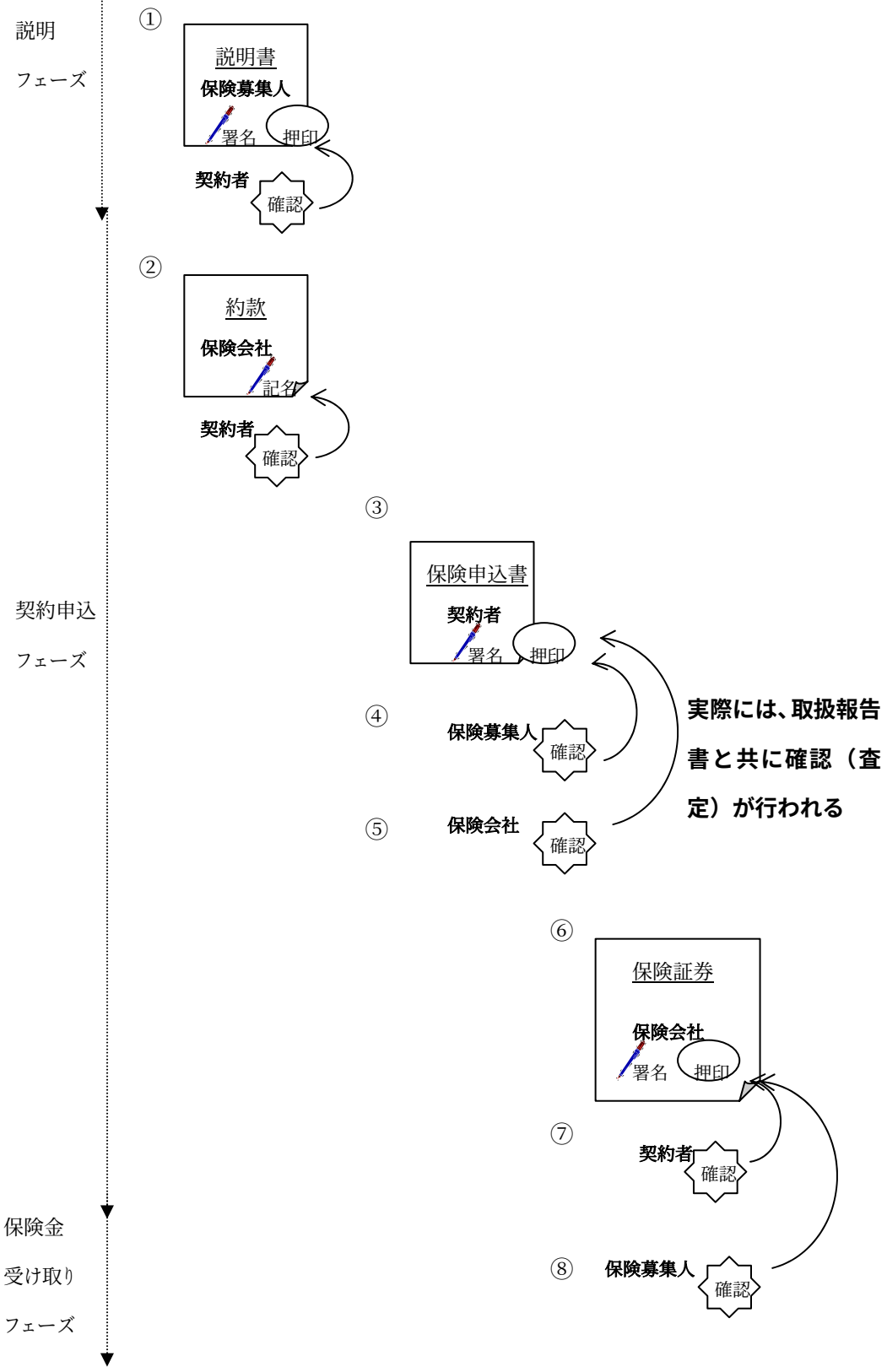



**図 4-2　主な文書の署名･検証の流れ**

### 4.1.3 文書管理

本節では、現状の生命保険会社での、一般的な文書管理方法について述べる。

#### (1) 更新反映方法

変更請求書を契約者から提出してもらう。これには、保険契約申込時と同様に契約者の署名・押印を必要とする。保険会社では、新旧の契約情報を保管する。

また、変更した旨を保険証券の裏面に記入することにより、変更を記録することもある。

#### (2) 保存期間

4.1.1 で述べたように保存期間は原則として保険期間（保険金等の措置期間、年金等の受給期間を含む）である。ただし保険期間満了以前に解約・失効等により消滅する契約については、保険契約の履行に必要な期間の経過により保存期間が終了するものとする。また、保険期間以外についても所要の期間、データを保存することがあるが、これについては目的に応じた適切な保存期間を別途定めることとする。実際には、後で必要になり探す際の手間等を考え、全てのデータを永久保存としている例もある。

#### (3) 保管方法

データの種類により、複数の媒体にて管理したり、正副の複数ストックし分散保管などを行ったりしている。保管データのアクセス時には、不正行為防止のため、複数の管理者のチェックを必要とする場合もある。

その他、内部管理体制を敷いてデータの真正性を確保する、操作記録の管理の徹底などを行っている。

### 4.1.4 認証方法

現状の生命保険で、金融庁事務ガイドラインにより定められているそれぞれの登場人物の認証方法を示す。

#### (1) 生命保険会社

生命保険業免許を保持。保険業法で定められる免許申請手続に基づき、商号、資本、

取締役名、所在地、定款等を含む免許申請書を内閣総理大臣に提出し、保険業法で定められる免許審査基準により審査を受ける

(2) **生命保険募集人**

保険業法で定められる登録の申請に基づき、名称、事務所の名称および所在地、所属保険会社の商号等を含む登録申請書を内閣総理大臣に提出し、生命保険募集人登録簿に登録される

(3) **契約者・被保険者**

- 運転免許証やパスポート等の本人を特定し得る書類による直接的な確認
- 企業等の法人の存在が確認できる書類による直接的な確認
- 保険証券を郵送により郵便物が返送されなかった事の間接的な確認
- 本人確認を行った保険料収納機関からの間接的な確認
- 生命保険募集人の訪問や保険会社が電話等を利用して保険契約者と交信することによる直接的な確認
- 診査において、医師による運転免許証やパスポート等の本人を特定し得る書類による間接的な確認
- 保険会社や生命保険募集人による直接面接

他人の生命の保険契約については、

- 個人保険の場合､保険契約申込書の被保険者同意欄に被保険者本人が署名・押印
- 全員加入の団体保険の場合、被保険者となることに同意した者全員の署名・押印のある名簿
- 事業保険の場合､当該保険契約の目的･趣旨に沿った業務運営が行われているか、保険の目的となる遺族補償規定等の書類

### 4.1.5 起こり得る不法行為

本節では、生命保険において起こり得る不法行為について記述する。

(1) **保険契約に関わる不法行為**

**①　被保険者が承諾しないうちに保険に申し込まれてしまった**

A.　契約者が善意で（事の重大さをしらずに安易に）申し込んでしまう

B.　保険金を目当てに被保険者に内緒で保険に申し込み、保険金をもらう

**②　告知義務違反**

A.　現在、傷病の治療中であるのに、治療中ではないと嘘をついて保険に申し込む

B.　過去に傷病の既往歴があるのに、そのことを内緒で保険に申し込む

C.　現在、危険度の高い職業についているのに他の職業といつわる

**③　身代わり診査**

A.　被保険者と別人の健康な人が診査をうけて被保険者本人になりすます

**④　虚偽の診断書を書かせる**

例：高度障害の場合、死亡した際と同額の保険金が先に受け取れるため、高度障害と偽りの診断書を作成し、保険金を請求する

**⑤　保険契約の虚偽作成**

生命保険業の従事者が成績の為に、契約者・被保険者に内緒で架空の保険契約を作成する

これらの対策としては、

- 契約者に対して説明周知の徹底
- 診査時と申込書の筆跡確認
- 生命保険会社間の情報網で高額保険に入っている場合の情報を共有する
- 保険金受取人を第何親等までに制限する（一定範囲の親族以外の第三者は保険金受取人にできない）
- 生命保険業の従事者に対する検査・法令遵守教育の徹底

などが挙げられる。

### (2) 保険事業者以外のものが保険事業者を装って詐欺的取引

対策としては、オンラインの場合は、登録番号を表記し必要に応じて顧客（契約者）が資格を確認できるようにする、サーバ証明書を掲載し確認させるなどの処置が考えられる（例：ベリサイン社のセキュアサイトシール）。従来のオフラインの場合は、生命保険募集人訪問時に認定書を保持することが義務付けられているため、それを提示し確認させる事が可能である。

その他にも、サービス提供者側の不正として、生命保険の従事者による保険料の横領（保険契約の申込内容を改ざんし、保険料を搾取する）、保障内容に関する不完全な説明（保険金が変動するにもかかわらず定額であると説明したり、解約時の返戻金に最低保証がないにもかかわらず、一定保証があるかのごとく説明したりするなど）も考えられる。

電子化する場合においても当然これらの被害は防止できるように考慮されなければならない。

## 4.2　デジタル署名の利用

本節では、生命保険業界において文書を電子化する場合のデジタル署名の導入、そしてそれを長期保存する場合の分析を行う。

3 章では、2 章で述べたデジタル署名文書を長期に渡り保存しその有効性を保つための技術を元に具体的モデルを提案した。本節では、3.2 セキュアストレージ型モデル、3.3 ＤＬＭＳ(Document Lifecycle Management Service)型モデルを生命保険業務に適用する。

### 4.2.1　セキュアストレージ型モデルの適用

#### 4.2.1.1　エンティティ

- 保険会社
- 生命保険募集人
- 契約者
- 被保険者
- 保険金受取人
- ＣＡ
- 電子署名文書長期保存エージェント
- ＴＳＡ
- 署名ポリシ発行者
- 再検証者

生命保険業界の場合、各生命保険会社で保険業務を運用している現状から、電子署名文書長期保存エージェントは生命保険会社が持つあるいは委託する機関になるであろうと思われる。

#### 4.2.1.2 検討モデル

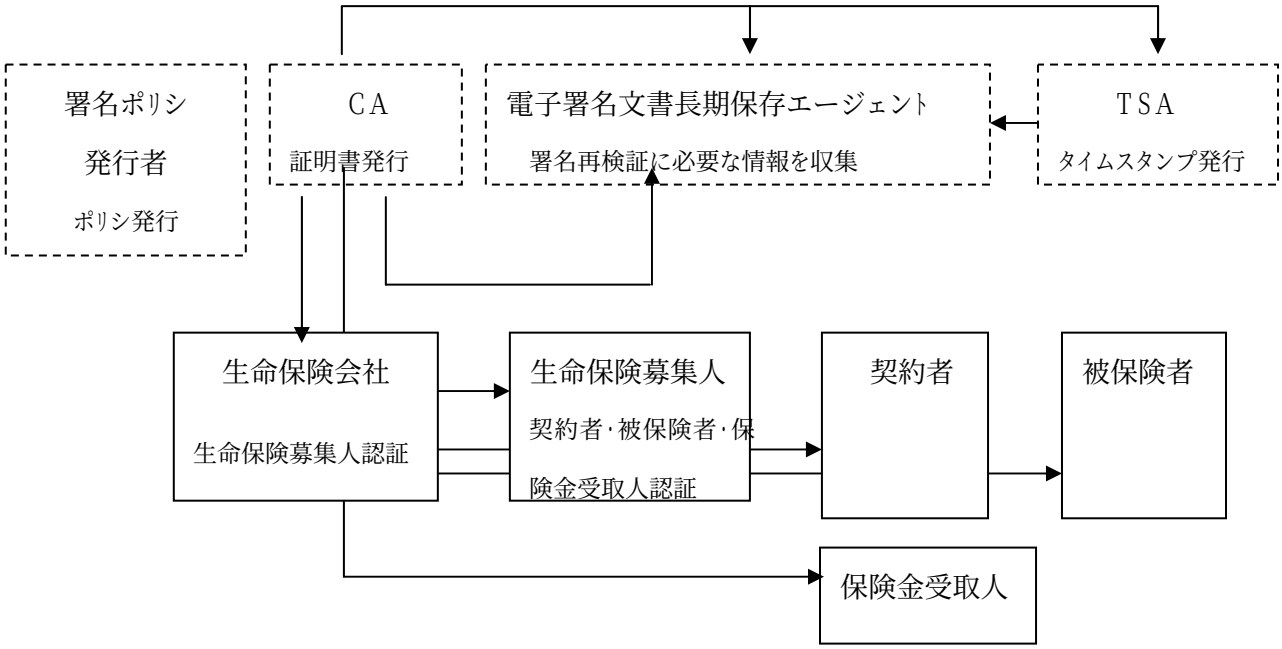



**図 4-3 証明書関連の流れ**

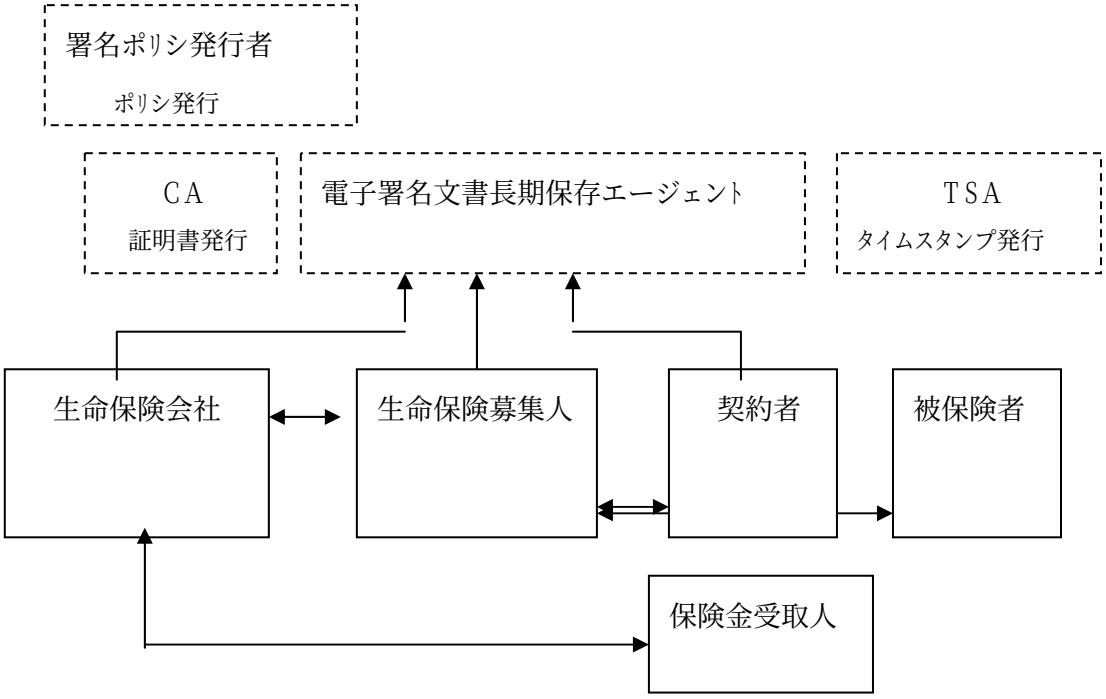



**図 4-4 文書の流れ**

#### 4.2.1.3 検討対象フェーズ

ＣＡ、 電子署名文書長期保存エージェント、 ＴＳＡは既に存在し、生命保険会社、 生命保険募集人はオンライン上に存在する段階で、契約者となり得る人が生命保険への加入を行おうとするフェーズから保険金受取人となった人が生命保険金を受け取るまでのフェーズを検討する

#### 4.2.1.4 認証方法

**表 4-4 認証方法**

| 対象 | 認証タイミング | 認証者 | 認証方法 |
|---|---|---|---|
| CA | 生命保険会社設立前 | 本節では検討の対象外とする | |
| 電子署名文書長期保存エージェント、 TSA | 生命保険会社設立前 | 本節では検討の対象外とする | |
| 生命保険会社 | 生命保険会社設立時 | 本節では検討の対象外とする | |
| 生命保険募集人 | 生命保険募集人登録時 | 本節では検討の対象外とする | |
| 契約者 | 生命保険加入時 | 生命保険募集人、生命保険会社 | 現行の認証基準・書類に基づきオフラインにて認証 |
| 被保険者 | 生命保険加入時 | 生命保険募集人、生命保険会社 | 現行の認証基準・書類に基づきオフラインにて認証 |
| 保険金受取人 | 生命保険加入時 | 生命保険募集人、生命保険会社 | 現行の認証基準・書類に基づきオフラインにて認証 |

#### 4.2.1.5 デジタル署名長期保存技術適用例

以下に、各フェーズでのエンティティの処理概要を述べる。ポリシ発行者については特に規定しないが、生命保険会社毎に設置し、一生命保険会社で 2.2.4 節、2.3 節で述べたような方式でポリシを策定し署名者、検証者間で利用させる方式、生命保険業界で統一的にポリシを策定し、各生命保険会社がそれに従い、署名者、検証者間で利用させる方式が考えられる。

### (1) 登録フェーズ

前述のとおり、本節では、ＣＡ、電子署名文書長期保存エージェント、生命保険会社、生命保険募集人は認証登録済み、ＣＡからはそれぞれのエンティティに証明書を発行してあると仮定する。なお、以下の(2)生命保険募集フェーズをはじめとし、各エンティティでは、通信相手の認証の目的で使う証明書と、扱うデータに対する署名付けを行う際の鍵に対する証明書が双方存在する場合がある。本節の目的である長期保存に向けての検討となるのは後者である。

### (2) 生命保険募集フェーズ

生命保険会社は、ＷＥＢサイトに募集広告を出す事になると思われるが、サーバ証明書を用いたＳＳＬ通信を行うなどにより、生命保険会社の正当性を確認できるようにする事が必要である。この方法については、本節での本質ではないため、特に述べない。なんらかの形で相手認証を行う必要があるという記述に留める。

### (3) 生命保険説明フェーズ

● 説明データ（生命保険募集人→顧客）

契約者となる人（顧客）が生命保険募集人に対して加入に向けての説明を要求した場合、生命保険募集人は顧客に説明を行う（ここでは説明の手段・内容は議論しない）。生命保険募集人は説明を行った旨を示すデータ（以下では「説明データ」と言う）を生成し、そのデータへの署名付けを行う。本データは長期保存の対象ではない為、署名の方法は 3.2.3 節で述べたものでなくても良い。

### (4) 生命保険加入フェーズ

契約者は、生命保険募集人に登録手続きを行いＣＡから証明書を発行してもらう。被保険者、保険金受取人についても登録手続きを行いＣＡから証明書を発行してもらう。

● 説明データ（生命保険募集人→契約者）

契約者は、生命保険に加入する決定をした後、(3)生命保険説明フェーズにて生命保険募集人が説明を行った内容を理解・承諾した上で説明データ（生命保険募集人の署名付き）に署名付けを行う。説明内容に対して内容を理解・承諾できな

い場合は、生命保険会社に対して再度説明を要求し、説明データを生成させる事になる。いずれにしても本データは長期保存の対象ではない為、署名の方法は 3.2.3 節で述べたものでなくても良い。

●約款（保険会社→契約者）

保険会社は、約款に対し署名付けを行う。その詳細手順は、3.2.3 (2)から(4)に従う。この際の検証者は契約者になる。電子署名文書長期保存エージェントは、3.2.3(5)から(10)の手順に従い電子署名を保存する。保険会社は署名添付済みの約款を契約者に送付する。

契約者は、約款を受け取ったら、保険会社の署名を検証後、自身の署名を添付する。その詳細手順は、3.2.3 (2)から(4)に従う。また、署名ポリシは保険会社が付けた署名のものと同一である。電子署名文書長期保存エージェントは、3.2.3(5)から(10)の手順に従い電子署名を保存する。

●保険申込書（契約者、被保険者→生命保険募集人）

契約者は保険申込書に必要情報を入力し、署名付けを行う。その際署名ポリシを入手しそれに従う。その後、署名済みデータを被保険者に送付する。現状の世界で、契約者と被保険者は面識があるものという前提であるため、この二者間は双方の公開鍵証明書を電子メール等で交換しセキュア通信を行っても良いし媒体による電子データの受け渡しでも良い。被保険者は受け取った保険申込書にさらに必要情報を入力し、署名付けを行う。その詳細手順は、3.2.3 (2)から(4)に従う。この際の検証者は生命保険募集人になる。電子署名文書長期保存エージェントは、3.2.3(5)から(10)の手順に従い電子署名を保存する。被保険者は署名済みのデータを契約者に渡し、契約者はそれを生命保険募集人に送付する。

●告知書（被保険者→生命保険募集人）

被保険者は、必要事項を入力後告知書に対し署名付けを行う。詳細手順は、3.2.3 (2)から(4)に従う。この際の検証者は生命保険募集人になる。電子署名文書長期保存エージェントは、3.2.3(5)から(10)の手順に従い電子署名を保存する。被保険者は署名添付済みの告知書を生命保険募集人に送付する。なお、告知書に医師の診断が必要な場合その結果情報の組み込み方法としては、・紙ベースのものの電子化書類と共に別手段による医師への確認が必要になると思われるが詳細はここでは省略する。

この際、申込書と告知書の結びつきが必要なため、電子データ上にリンク情報を設けるか、あるいは、双方の文書のハッシュ値を結合したものへの署名付けを行う二重署名を生成する事も考えられる。

● 初回保険料

契約者は生命保険募集人に対して初回保険料の振込みを行う。手段としては銀行振込、クレジットカード支払い等が考えられるが、いずれにしても金融機関での確認が取れるため、特にここでは電子データは導入検討しない。

● 領収証（生命保険募集人、保険会社→契約者）

生命保険募集人は初回保険料の受け取りが確認できたら、領収証を作成しそれに対し署名付けを行う。詳細手順は、3.2.3 (2)から(4)に従う。この際の検証者は契約者になる。電子署名文書長期保存エージェントは、3.2.3(5)から(10)の手順に従い電子署名を保存する。生命保険募集人は署名添付済みの領収証を契約者に送付する。

● 取扱報告書（生命保険募集人→保険会社）

生命保険募集人は、契約者からの申込書、告知書、初回保険料の確認（つまり領収証データの作成送付が完了）ができると取扱報告書を作成しそれに対し署名付けを行う。詳細手順は、3.2.3 (2)から(4)に従う。この際の検証者は保険会社になる。電子署名文書長期保存エージェントは、3.2.3(5)から(10)の手順に従い電子署名を保存する。生命保険募集人は署名済みの取扱報告書、上述までで受け取っている契約者からの書類を保険会社に送付する。

● 生命保険証券（保険会社→契約者、被保険者）

保険会社は、上述の保険募集人から受け取った書類の検証をすると、保険証券に署名付けを行う。保険会社は契約者に署名済みの保険証券を送付する。契約者は保険証券の保険会社の署名を検証後、自身の署名を添付する。詳細の手順は3.2.3 (2)から(4)に従う。この際の検証者は任意の人（生命保険募集人又は再検証者）になる。電子署名文書長期保存エージェントは、3.2.3(5)から(10)の手順に従い電子署名を保存する。

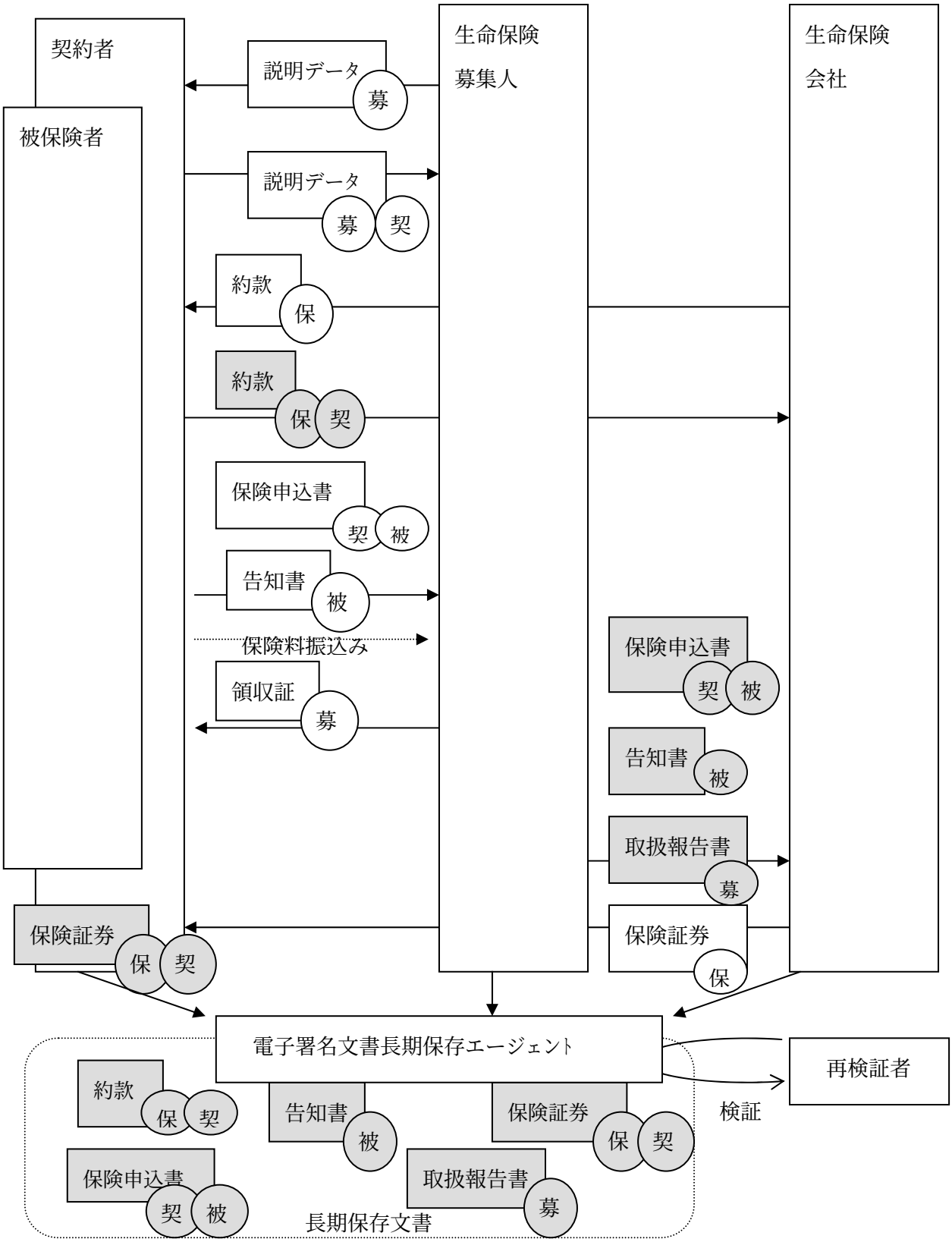



**図 4-5 処理の流れ**

各文書の〇は署名を示す（中の文字は、契は契約者、保は生命保険会社、募は生命保険募集人、被は被保険者を指す）

### (5) 生命保険保険期間

電子署名文書長期保存エージェントは、ETSI TS 101 733 Electronic Signature Formats 準拠の形式で(4)で述べた文書を保存する。

また、生命保険会社から契約者へは定期的に通知文書等を電子メール等を用いて配信する事により電子メールアドレスの有効性の確認を行う事ができる。その際には、生命保険会社のデジタル署名添付があった方が良いと思われるが、長期保存する対象のものではない為、必ずしも 3.2.3 節で述べた方法ではなくても良い。また、現状のとおり郵送にて文書を送付する事による住所の確認とも併せて行う事により、より確認の確実性が増す。

### (6) 生命保険内容変更時

- 変更請求書（契約者→保険会社）

  契約者は契約内容に関して変更の必要性が出た場合、契約変更請求書に必要事項を記入し、署名を添付する。詳細の手順は 3.2.3 (2)から(4)に従う。この際の検証者は生命保険会社になる。電子署名文書長期保存エージェントは、3.2.3(5)から(10)の手順に従い電子署名を保存する。契約者は生命保険会社に署名済みの変更請求書、保持している保険証券を送付する。

- 生命保険会社は変更請求書を受け取ると契約者の署名を検証後、内容を確認し更新処理を行う。

- 生命保険会社は保険証券に変更事項を入力し、署名を添付する。詳細の手順は 3.2.3 (2)から(4)に従う。この際の検証者は契約者になる。電子署名文書長期保存エージェントは、3.2.3(5)から(10)の手順に従い電子署名を保存する。生命保険会社は契約者に更新済みの署名付き保険証券を送付する。

### (7) 生命保険解約時

- 解約請求書（契約者→保険会社）

  契約者は契約を解除したい場合、解約請求書に必要事項を記入し、署名を添付する。詳細の手順は 3.2.3 (2)から(4)に従う。この際の検証者は生命保険会社になる。電子署名文書長期保存エージェントは、3.2.3(5)から(10)の手順に従い電子署名を保存する。契約者は生命保険会社に署名済みの解約書、保持している保険証

券、その他必要書類を送付する。

- 生命保険会社は解約請求書を受け取ると契約者の署名を検証後、内容を確認し解約処理を行う。
- 生命保険会社は解約した旨を記した電子データに、署名を添付する。詳細の手順は 3.2.3 (2)から(4)に従う。この際の検証者は契約者になる。電子署名文書長期保存エージェントは、3.2.3(5)から(10)の手順に従い電子署名を保存する。生命保険会社は契約者に署名付きデータを送付する。

### (8) 生命保険金支払い時

- 支払依頼書（保険金受取人→保険会社）

  保険金受取人は、保険金の支払いを受ける場合、支払依頼書に必要事項を記入し、署名を添付する。詳細の手順は 3.2.3 (2)から(4)に従う。この際の検証者は生命保険会社になる。電子署名文書長期保存エージェントは、3.2.3(5)から(10)の手順に従い電子署名を保存する。契約者は生命保険会社に署名済みの支払依頼書、保持している保険証券、その他必要書類を送付する。
- 生命保険会社は支払依頼書を受け取ると契約者の署名を検証後、内容を確認し支払処理を行う。
- 生命保険会社は支払いをした旨を記した電子データに、署名を添付する。詳細の手順は 3.2.3 (2)から(4)に従う。この際の検証者は保険金受取人になる。電子署名文書長期保存エージェントは、3.2.3(5)から(10)の手順に従い電子署名を保存する。生命保険会社は保険金受取人に署名付きデータを送付する。

### (9) 問題発生時

以上のような処理を行う事により、保険金支払い時およびその他のフェーズにおいて、契約者・被保険者・生命保険募集人・生命保険会社間でのトラブルが発生した場合、再検証者が電子署名文書長期保存エージェントに保存データを照会し、その署名を検証する事により、長期に渡り保存されたデータであっても、各書類の有効性を確認する事ができる。

### 4.2.2 DLMS(Document Lifecycle Management Service)型モデルの適用

#### 4.2.2.1 エンティティ

- 保険会社
- 生命保険募集人
- 契約者
- 被保険者
- 保険金受取人
- ＣＡ
- ＤＬＭＳ(Document Lifecycle Management Service)
- ＴＳＡ
- 再検証者

生命保険業界の場合、各生命保険会社で保険業務を運用している現状から、DLMS は生命保険会社が持つあるいは委託する機関になるであろうと思われる。

#### 4.2.2.2 検討モデル

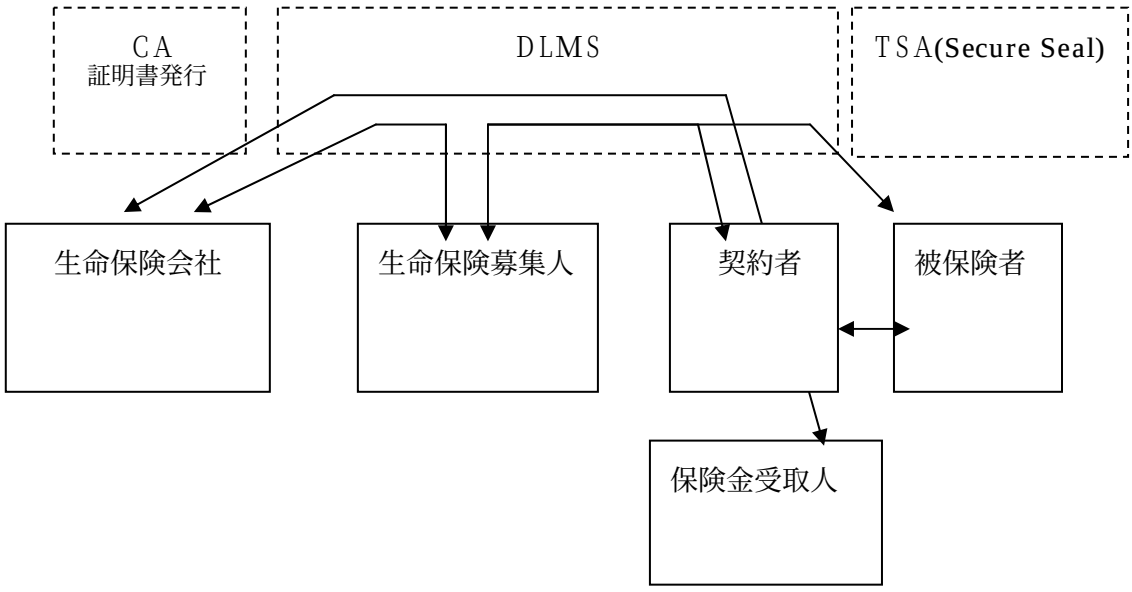



**図 4-6文書の流れ**

「証明書関連の流れ」については、4.2.1.1 と類似なので省略する。

#### 4.2.2.3　検証対象フェーズ

4.2.1.3 と同様方式とする

#### 4.2.2.4　認証方法

4.2.1.4 と同様方式とする

#### 4.2.2.5　デジタル署名長期保存技術適用例

(1) **登録フェーズ**

前述のとおり、本節では、ＣＡ、 ＤＬＭＳ、生命保険会社、生命保険募集人は認証登録済み、ＣＡからは証明書を発行してあると仮定する。

(2) **生命保険募集フェーズ**

生命保険会社は、ＷＥＢサイトに募集広告を出す事になると思われるが、サーバ証明書を用いたＳＳＬ通信を行うなどにより、生命保険会社の正当性を確認できるようにする事が必要である。

(3) **生命保険説明フェーズ**

- 説明データ（生命保険募集人→顧客）

契約者となる人（顧客）が生命保険募集人に対して加入に向けての説明を要求した場合、生命保険募集人は顧客に説明を行う（ここでは説明の手段・内容は議論しない）。生命保険募集人は説明を行った旨を示すデータ（以下では「説明データ」と言う）を生成し、そのデータへの署名付けを行う。本データは長期保存の対象ではない為、署名の方法は特に問わない。

(4) **生命保険加入フェーズ**

契約者は、生命保険募集人に登録手続きを行いＣＡから証明書を発行してもらう。被保険者、保険金受取人についても登録手続きを行いＣＡから証明書を発行してもらう。

- 説明データ（生命保険募集人→契約者）

契約者は、生命保険に加入する決定をした後、(3)生命保険説明フェーズにて生命保険募集人が説明を行った内容を理解・承諾した上で説明データ（生命保険募集人の署名付き）に署名付けを行う。説明内容に対して内容を理解・承諾できない場合は、生命保険会社に対して再度説明を要求し、説明データを生成させる事

になる。いずれにしても本データは長期保存の対象ではない為、署名の方法は特に問わない。

●約款（保険会社→契約者）

保険会社は、約款に対し署名付けを行う。その詳細手順は、3.3.4(2)から(4)に従う。この際の受信者は契約者になる。ＤＬＭＳは、3.3.4(5)の手順に従い、電子署名を保存する。これらは、SecureSeal を介してタイムスタンプの登録およびハッシュ値の登録が行われる。契約者は 3.3.4(6),(7)の手順に従いＤＬＭＳから約款を受け取ると検証した後、自身の署名、証明書を約款に付加しＤＬＭＳに送付する。

● 保険申込書（契約者、被保険者→生命保険募集人）

契約者は保険申込書に必要情報を入力し、署名付けを行う。その後、署名済みデータを被保険者に送付する。現状の世界で、契約者と被保険者は面識があるものという前提であるため、この二者間は双方の公開鍵証明書を電子メール等で交換しセキュア通信を行っても良いし媒体による電子データの受け渡しでも良い。被保険者は受け取った保険申込書にさらに必要情報を入力し、署名付けを行う。その詳細手順は、3.3.4(2)から(4)に従う。この際の受信者は生命保険募集人になる。ＤＬＭＳは、3.3.4(5)の手順に従い、電子署名を保存する。これらは、SecureSeal を介してタイムスタンプの登録およびハッシュ値の登録が行われる。生命保険募集人は 3.3.4(6),(7)の手順に従いＤＬＭＳから保険申込書を受け取ると検証した後、自身の署名、証明書を保険申込書に付加しＤＬＭＳに送付する。

●告知書（被保険者→生命保険募集人）

被保険者は、必要事項を入力後告知書に対し署名付けを行う。詳細手順は、3.3.4(2)から(4)に従う。この際の受信者は生命保険募集人になる。DLMS は、3.3.4(5)の手順に従い、電子署名を保存する。これらは、SecureSeal を介してタイムスタンプの登録およびハッシュ値の登録が行われる。生命保険募集人は 3.3.4(6),(7)の手順に従いＤＬＭＳから告知書を受け取ると検証した後、自身の署名、証明書を告知書に付加しＤＬＭＳに送付する。

なお、告知書に医師の診断が必要な場合その結果情報の組み込み方法としては、紙ベースのものの電子化書類と共に別手段による医師への確認が必要になると思われるが詳細はここでは省略する。

この際、申込書と告知書の結びつきが必要なため、電子データ上にリンク情報を設けるか、あるいは、双方の文書のハッシュ値を結合したものへの署名付けを行う二重署名を生成する事も考えられる。

● 初回保険料

契約者は生命保険募集人に対して初回保険料の振込みを行う。手段としては銀行振込、クレジットカード支払い等が考えられるが、いずれにしても金融機関での確認が取れるため、特にここでは電子データは導入検討しない。

● 領収証（生命保険募集人、保険会社→契約者）

生命保険募集人は初回保険料の受け取りが確認できたら、領収証を作成しそれに対し署名付けを行う。詳細手順は、3.3.4(2)から(4)に従う。この際の受信者は契約者になる。ＤＬＭＳは、3.3.4(5)の手順に従い、電子署名を保存する。これらは、SecureSeal を介してタイムスタンプの登録およびハッシュ値の登録が行われる。契約者は 3.3.4(6),(7)の手順に従いＤＬＭＳから領収書を受け取ると検証した後、自身の署名、証明書を領収書に付加しＤＬＭＳに送付する。

● 取扱報告書（生命保険募集人→保険会社）

生命保険募集人は、契約者からの申込書、告知書、初回保険料の確認（つまり領収証データの作成送付が完了）ができると取扱報告書を作成しそれに対し署名付けを行う。詳細手順は、3.3.4(2)から(4)に従う。この際の受信者は保険会社になる。ＤＬＭＳは、3.3.4(5)の手順に従い、電子署名を保存する。これらは、SecureSeal を介してタイムスタンプの登録およびハッシュ値の登録が行われる。保険会社は 3.3.4(6),(7)の手順に従いＤＬＭＳから取扱報告書を受け取ると検証した後、自身の署名、証明書を取扱報告書に付加しＤＬＭＳに送付する。

● 生命保険証券（保険会社→契約者、被保険者）

保険会社は、上述の保険募集人から受け取った書類の検証をすると、保険証券に署名付けを行う。ＤＬＭＳは、3.3.4(5)の手順に従い、電子署名を保存する。これらは、SecureSeal を介してタイムスタンプの登録およびハッシュ値の登録が行われる。契約者はＤＬＭＳから生命保険証券を受け取ると検証した後、自身の署名、証明書を生命保険証券に付加しＤＬＭＳに送付する。この際の検証者は任意の人（生命保険募集人又は再検証者）になる。

### (5) 生命保険保険期間

ＤＬＭＳは 3.3.4(5)で述べた方式で文書を長期保存する。また、生命保険会社から契約者へは定期的に通知文書等を電子メール等を用いて配信する事により電子メールアドレスの有効性の確認を行う事ができる。その際には、生命保険会社のデジタル署名添付があった方が良いと思われるが、長期保存する対象のものではない為、必ずしも 3.3.4 節で述べた方法ではなくても良い。また、現状のとおり郵送にて文書を送付する事による住所の確認とも併せて行う事により、より確認の確実性が増す。

### (6) 生命保険内容変更時

- 変更請求書（契約者→保険会社）

  契約者は契約内容に関して変更の必要性が出た場合、契約変更請求書に必要事項を記入し、署名を添付する。詳細の手順は 3.3.4(2)から(4)に従う。この際の受信者は生命保険会社になる。

  ＤＬＭＳは、3.3.4(5)の手順に従い、電子署名を保存する。これらは、SecureSeal を介してタイムスタンプの登録およびハッシュ値の登録が行われる。保険会社は 3.3.4(6),(7)に従いＤＬＭＳから変更請求書と契約者から保持している保険証券を受け取ると検証した後、自身の署名、証明書を変更請求書に付加しＤＬＭＳに送付する。

- 生命保険会社は変更請求書を受け取ると契約者の署名を検証後、内容を確認し更新処理を行う。生命保険会社は保険証券に変更事項を入力し、署名を添付する。詳細の手順は 3.3.4(2)から(4)に従う。この際の受信者は契約者になる。契約者は 3.3.4(6),(7)に従いＤＬＭＳから変更済み保険証券を受け取ると検証した後、自身の署名、証明書を保険証券に付加しＤＬＭＳに送付する。

### (7) 生命保険解約時

- 解約請求書（契約者→保険会社）

  契約者は契約を解除したい場合、解約請求書に必要事項を記入し、署名を添付する。詳細の手順は 3.3.4(2)から(4)に従う。この際の受信者は生命保険会社になる。ＤＬＭＳは、3.3.4(5)の手順に従い、電子署名を保存する。これらは、SecureSeal を介してタイムスタンプの登録およびハッシュ値の登録が行われる。保険会社は

ＤＬＭＳから解約請求書と契約者から保持している保険証券、その他必要書類を受け取ると検証した後、自身の署名、証明書を解約請求書に付加しＤＬＭＳに送付する。

- 生命保険会社は解約請求書を受け取ると契約者の署名を検証後、内容を確認し解約処理を行う
- 生命保険会社は解約した旨を記した電子データに、署名を添付する。詳細の手順は 3.3.4(2)から(4)に従う。この際の受信者は契約者になる。契約者は 3.3.4(6),(7)に従いＤＬＭＳから本電子データを受け取ると検証した後、自身の署名、証明書を本電子データに付加しＤＬＭＳに送付する。

### (8) 生命保険金支払い時

- 支払依頼書（保険金受取人→保険会社）

  保険金受取人は、保険金の支払いを受ける場合、支払依頼書に必要事項を記入し、署名を添付する。詳細の手順は 3.3.4(2)から(4)に従う。この際の受信者は生命保険会社になる。ＤＬＭＳは、3.3.4(5)の手順に従い、電子署名を保存する。これらは、SecureSeal を介してタイムスタンプの登録およびハッシュ値の登録が行われる。保険会社は 3.3.4(6),(7)に従いＤＬＭＳから支払依頼書と契約者から保持している保険証券、その他必要書類を受け取ると検証した後、自身の署名、証明書を支払依頼書に付加しＤＬＭＳに送付する。
- 生命保険会社は支払依頼書を受け取ると契約者の署名を検証後、内容を確認し支払処理を行う
- 生命保険会社は支払いをした旨を記した電子データに、署名を添付する。詳細の手順は 3.3.4(2)から(4)に従う。この際の受信者は保険金受取人になる。保険金受取人は 3.3.4(6),(7)に従いＤＬＭＳから本電子データを受け取ると検証した後、自身の署名、証明書を本電子データに付加しＤＬＭＳに送付する。

### (9) 問題発生時

以上のような処理を行う事により、保険金支払い時およびその他のフェーズにおいて、契約者・被保険者・生命保険募集人・生命保険会社間でのトラブルが発生した場合、再検証者がＤＬＭＳに保存データを照会し、その署名を検証する事により、長期

に渡り保存されたデータであっても、各書類の有効性を確認する事ができる。

### 4.2.3 デジタル署名長期保存技術適用例考察

前節までの検討を踏まえ、電子化しデジタル署名を用いて長期保存する場合の利点および、問題がある点、従来の課題の克服、起こり得る被害に対する対策についてまとめる。

#### 4.2.3.1 利点・課題

利点としては、従来の人による確認等で人の処理に頼っていた点については、電子化しルール化することで確実性が増すと共に効率化が図れる。またデジタル署名を添付することにより、当事者以外の者も、データ（書類）作成時点、作成後もその正当性の検証が可能になり、問題発生時の解決に役立つ。さらに長期保存の技術を導入する事により、データ（書類）作成から時間が経過した後に問題が発生しても確実に検証が可能になる。

課題としては、対面でない分、デジタル署名を行った者が意図した人かどうかの個人の特定が困難であることが挙げられる。IC カードに署名鍵を格納する方式や、指紋等の生体認証と組み合わせて署名鍵を用いるなどの方式を考慮する必要がある。また、電子化を行う際のシステムの操作性については十分考慮する必要がある。

#### 4.2.3.2 起こり得る不法行為に対する対策

表 4-5 に、4.1.5 節で述べた起こり得る不法行為に対する対策をまとめる。

**表 4-5 被害に対する対策**

| 不法行為 | | 悪意のある対象 | 対策 |
|---|---|---|---|
| 保険金詐欺 | 契約者の不正（故意）による保険金詐欺 | 契約者 | 被保険者に関してもデジタル署名を添付し、加入の意図があることを確認する事で回避できる |
| | 告知義務違反(傷病歴、職業) | 被保険者 | 従来（非電子化）と同様 |
| | 身代わり診査 | 被保険者 | 従来（非電子化）と同様 |
| | 虚偽の診断書 | 契約者、被保険者、保険金受取人、医師 | 従来（非電子化）と同様 |
| 保険契約の虚偽作成 | 生命保険業の従事者の不正による保険契約の作成 | 生命保険業の従事者 | デジタル署名を添付し、データが意図した者によって作成された事を当事者以外のものも検証 |

| 不法行為 | | 悪意のある対象 | 対策 |
|---|---|---|---|
| | | | することが可能とする事で回避できる |
| 保険事業者になりすまし | | 任意の人 | デジタル署名を添付し、正当な業者である事を当事者以外のものも検証することが可能とする事で回避できる |
| 生命保険業の従事者による保険料の横領 | 生命保険業の従事者の不正による保険契約の改ざん | 生命保険業の従事者 | デジタル署名を添付し、当事者以外のものもデータが改ざんされていない事を検証することが可能とする事で回避できる |
| 保障内容に関する不完全な説明 | 契約者にリスクをもたらす要素に関して十分な説明を行わない | 生命保険業の従事者 | 説明データに関しても内容を後日確認できるようにデジタル署名をつける事で回避できる |

# 5 提　言

## 5.1　当面の課題

本ガイドラインでは、デジタル署名の有効性を長期に亙り維持するための要件として、署名検証時における“署名再検証に必要な情報”の収集、署名検証時刻の特定、“署名再検証に必要な情報”への改竄検知措置、“署名再検証に必要な情報”の保管を挙げ、“署名再検証に必要な情報”の内容については、署名の本人性の確認ができること、また署名ポリシとして予め署名者と検証者との間で同意しておくこと、署名検証時刻の特定については、タイムスタンプを取得することを述べた。

デジタル署名文書を従来の紙と印鑑の世界と同様に長期保存に耐えるものとするには、技術面および制度面から、次の課題を解決する必要がある。

### 5.1.1　署名ポリシの普及

従来の紙と印鑑の世界で、予め双方の合意により信頼の拠り所を印鑑証明に求め、押印者が印鑑証明を提示するのと同様、デジタル署名文書においても、署名者と検証者との間で信頼の拠り所とする認証局やタイムスタンプサービス、署名再検証に必要な情報の収集に関する署名者と検証者それぞれの義務などを予め署名ポリシとして合意しておく必要があり、本ガイドラインの２章では、汎用的な署名ポリシのプロファイルを提案した。デジタル署名文書の長期保存をはかるには、契約書をはじめとするデジタル署名文書の基本フレームに署名ポリシを組み込んでおく必要があり、今後手戻りを発生させないためにも、署名ポリシの必要性を広く理解してもらいこれを普及させることは急務である。

### 5.1.2　オブジェクト登録体制の整備

署名ポリシは、アプリケーションの運営主体が自由に発行可能である。署名ポリシはユニークな識別子（オブジェクト識別子）を持つ必要があることから、署名ポリシの発行者はオブジェクト登録機関に署名ポリシ（＝オブジェクト）を登録し、ユニークな識別子（＝オブジェクト識別子）を割り付けてもらう必要があるが、現在、署名ポリシの登録に関する国内共通的な仕組みもコンセンサスもない。また、登録済みの署名ポリシの有効活用と幅広い普及のための、登録済みオブジェクトの検索システムも整備されておらず、これらの体制整備は急務である。

### 5.1.3　信頼に足るタイムスタンプサービスの提供

タイムスタンプサービスはデジタル署名の有効性を長期に亙り維持するための要であるが、現在提供されているあるいは今後提供されるタイムスタンプサービスが、将来とも時刻の正しさや事業継続の観点で安心して利用できるか否かの判断の拠り所がなく、信頼に足るタイムスタンプサービスの提供は不可欠である。

## 5.2　課題を解決するための提言

### 5.2.1　オブジェクト登録体制の拡充

署名ポリシなどのオブジェクトの登録方法には、登録機関に直接オブジェクトを登録する方法と、法人格を有するなど実体のある組織を登録機関に登録し、その組織が下位の登録機関としてオブジェクト登録を代行する方法とがある。各々の組織が個別にオブジェクトを管理するには、オブジェクトの登録管理や公開など運用面の負荷が大きく、登録機関に直接オブジェクトを登録できることが望ましいが、現在、国内の登録機関はＯＳＩオブジェクト以外のオブジェクトに順応できる体制にはなく、また、国内外を問わず登録機関に登録されたオブジェクトの公開や検索サービスも未整備である。署名ポリシなどの非OSIオブジェクトの登録や公開のため体制整備は急務であり、既存の登録機関の拡充、あるいは既存の登録機関の統廃合も視野に入れた新たな登録機関の整備が望まれる。これによりはじめて、署名ポリシの共通化による効率化と普及、似て非なるオブジェクトの氾濫回避をはかることが可能となる。

### 5.2.2　タイムスタンプサービス監査制度の確立

タイムスタンプサービスが、高い信頼を得て長期に亙りそのサービスを継続し、社会インフラとして広く認められるためには、

① タイムスタンプサービスとしての技術的および運用的な要件を満たしていること

② 長期に亙る事業運営に問題ないこと

に関し、何等かの確証が必要である。

このためには、タイムスタンプサービスの監査制度を確立し、また、監査にパスした事業体には長期に亙る運営の一助として税制面等での考慮なども必要である。

タイムスタンプサービス事業者が、そのタイムスタンプサービスがスタンプする時刻の

正しさについて、そのサービスが保証する精度を維持するための方法と運用（どの時刻ソースに合わせるためにどういう手段を用いてどう運用しているか）や財政基盤について監査を受けその結果を公表することが広く信頼を得る最善の策となる。

タイムスタンプサービスとしての技術的および運用的要件に関しては、２章で述べた様に既にＥＴＳＩ等でも検討が進められており、早急に監査制度の確立に向けたアクションをとる必要がある。

## 5.3　企業間取引以外の電子署名文書長期保存

本報告で述べた内容は、４章のケーススタディに挙げた企業と個人（ＢｔｏＣ）の間、企業間取引（ＢｔｏＢ）や個人間の取引（ＣｔｏＣ）の電子署名文書だけでなく、政府・自治体間（ＧｔｏＧ）、政府・自治体と企業・個人間（ＧｔｏＢ／Ｃ）の電子署名文書にも適用可能である。しかしながら、政府・自治体自身が公証サービスの役目を果たしていることに加え、原本性保証システムとしての役目も果たしていることから、本ガイドラインに述べた基本モデルの適用以外に次の対応も可能である。

① 政府・自治体間の電子署名文書

決裁文書、報告書などが該当する。その文書の原本が保管されている限り、公開鍵証明書の有効期限が切れた後も、原本と照らし合わせることにより、その文書が存在し改竄されていないものであることを確認することができる。

② 企業／個人から政府・自治体への電子署名文書

申請書や届出書などが該当する。一旦受け付けた後は、それが政府・自治体に保管されている限り、上記①政府・自治体間の電子署名文書と同様の方法で確認ができる。時系列の明確化と証拠保全のために受付時にタイムスタンプを付け受け取りを発行することが望ましい。

③政府・自治体から企業／個人への電子署名文書

許可証、決定書などが該当する。扱いは上記①政府・自治体間の電子署名文書と同様である。企業／個人は、政府・自治体にその文書の原本が保管されている限り、公開鍵証明書の有効期限が切れた後も、必要であれば政府・自治体に照会することにより、その文書が存在し改竄されていないものであることを確認することができる。

# 6 付録

## 6.1 用語集

本ガイドラインで使用されている主要な用語の一部を、以下に概説する。

1. 公開鍵暗号システム（Public Key Cryptosystem）
   関連した２つの鍵(公開鍵と秘密鍵)を使用する非対称暗号方式(asymmetric cryptographic algorithm)の一つであり、一方の鍵(公開鍵)で暗号化したデータは他方の鍵(秘密鍵)でのみ復号化できるようになっているシステム。２つの鍵は、公開鍵が与えられても、秘密鍵を導き出す事が計算上不可能な特性を持つ。
2. 公開鍵（Public Key）
   公開鍵暗号システムにおける鍵ペアのうちの一つで、通信相手等の他人に知らせて利用してもらうための鍵。
3. 秘密鍵（Private Key）
   公開鍵暗号システムにおける鍵ペアのうちの一つで、他人には知られないように秘密にしておく鍵。
4. 鍵ペア（Key Pair）
   公開鍵暗号システムにおける公開鍵及びそれに対応する秘密鍵。
5. 共通鍵（Secret Key）
   発信者と受信者が同一の暗号鍵を使用してデータの暗号化と復号化を行う対称暗号方式(symmetric cryptographic algorithm)における鍵。
6. 公開鍵基盤（ＰＫＩ：Public Key Infrastructure）
   公開鍵暗号システムを用いて情報システムやコミュニケーションシステムのセキュリティを確保するための一連の技術およびサービス。
7. 認証
   人、物、情報の真正性を確認すること（Authentication）を意味し、「人の認証」、「電子認証」といった使い方がされる。また、証明書（Certificate）を発行して公開鍵の持ち主に関する証明を行うこと（Certification）を意味し、「認証局」、「相互認証」といった使い方がされる。これ以外に、あるシステム資源に対してアクセスする権限があるか否か確認し、その使用する権限を与えること（Authorization）も意味

する。

8. 証明書（Certificate）

認証対象者の識別情報と公開鍵とが対応していることを証明するデジタル文書。認証対象者の識別情報、その公開鍵、鍵の利用目的・範囲、発行認証局名などが含む一連の情報に、認証局のデジタル署名を付加したもの。従って、厳密には公開鍵証明書であるが、本ガイドラインでは曖昧さがない限り単に証明書という。

9. 証明書の発行（Certificate Issuance）

証明書を生成し、証明書に登録された申請者に対し、その内容を通知する行為。

10. 証明書の失効（Certificate Revocation）

証明書の有効期間内に、秘密鍵が危殆化した場合、あるいは氏名等の重要な属性情報に変更が生じた場合に証明書を無効にする行為。

11. 証明書の一時失効（Certificate Suspension）

証明書の有効期間中に一時的に証明書を失効させる行為。

12. 失効リスト（Certificate Revocation List＝CRL）

失効した証明書のリスト。通常認証局によるデジタル署名が付される。

13. 認証局（Certification Authority＝CA）

証明書の発行、開示、失効もしくは一時失効等のサービスを行う信頼された個人または法人。

14. 登録局（Registration Authority＝RA）

証明書の発行や失効のプロセスにおいて、本人確認などの一部機能を認証局の承認を受けて行う個人または法人。登録局は、証明書や失効リストの生成は行わない。

15. リポジトリ（Repository）

証明書や失効リスト等を保管し、証明書利用者等に対してこれらの開示や配布もしくは検索等のサービスを提供するシステム。

16. 証明書加入者（Certificate Subscriber）

認証局から証明書の発行を受けた者。本ガイドラインでは特に区別が必要な場合を除いて、単に「加入者」という。

17. 信頼者・リライングパーティー（Relying Party）

取引等において証明書を利用する場合、証明書を受け取って、それを信頼して行動する者。加入者ばかりでなく非加入者も含まれる。本ガイドラインでは特に区別が必要

な場合を除いて、単に「信頼者」という。

18. 証明書利用者（Certificate User）

証明書加入者及び証明書信頼者などの証明書を利用する者。本ガイドラインでは特に区別が必要な場合を除いて、単に「利用者」という。

19. 証明書ポリシ（Certificate　Policy）

認証局のサービス・運用等に関する方針や規定、基準。

20. 認証局運用規定（ＣＰＳ：Certification Practice Statement）

証明書ポリシに基づいて、認証の実施における手続き、遵守事項などを文書化したもの。利用者等の認証局の外部者に開示されるもの。

21. 事務取扱要領（Operation Manuals）

認証局運用規定に基づいて、認証局内部における実務を詳細に規定したもの。

22. 危殆化（Compromise）

秘密鍵や関連機密情報等が、盗難や漏洩、第三者による解読等によってその機密性を失ったか、あるいはその可能性があること。

23. デジタル署名（Digital Signature）

署名対象データのハッシュ値(データを数学的な操作によって一定の長さに縮小させたもの。ハッシュ値から元のデータは再現不可能)に対して、秘密鍵で暗号化したもの。デジタル署名の検証は、デジタル署名を公開鍵で復号化した値と元のデータのハッシュ値とを照合することで可能。デジタル署名は、当該秘密鍵の保有者のみが生成できることから文字による署名と同等の効果が推定される。

24. 電子署名（Electronic　Signature）

間違い無く本人である事を証明する電子的なデータ。デジタル署名と同義で使われる事が多いが、広義ではアナログ署名を電子データにしたものも含む。

25. 暗号モジュール（Cryptographic Module）

暗号鍵等の生成、保管、利用などにおいて、セキュリティを確保する目的で使用されるソフトウェア、ファームウエア、ハードウェアあるいはそれらを組み合わせた装置。

26. デルタＣＲＬ

前回発行した最新のＣＲＬ(完全ＣＲＬ)との差分情報のみを含むＣＲＬ。前回の完全ＣＲＬとデルタＣＲＬをあわせることで、最新の完全ＣＲＬを生成できる

27. 再検証

電子署名は宛先である最初の検証者がその有効性を検証するが、その数年、数十年後に、第三者が再び検証する必要性が生じる。それは監査人であったり、署名者が署名を否認し係争が生じたりした場合など仲裁者が署名を再検証する事の必要性が生じる。再検証の結果は、署名の最初の検証と同一の結果をもたらされなければならない。この事を可能にする為に最初の検証時に署名の有効性を確認した時に署名検証時点以前に署名が存在していた事を証明するタイムスタンプを付け、検証に用いた全ての証拠としての証明書、失効情報を保存しておく長期署名フォーマットが必要になる。

28. 耐タンパ性

装置を分解するなどして、中にある秘密情報を不正に入手しようとする行為（Tamper）に対する耐性。

## 6.2 参考文献

今回の検討を進める過程で参考とした文献を以下に示す。特にＷＥＢ上の文書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。


１．「だれも教えなかった 印鑑 領収書 内容証明 契約書」総合法令編
1998 年 10 月 6 日 総合法令出版株式会社

２．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、情報処理学会研究報告 2000-CSEC-8、pp.13--17(2000)

３．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、情報処理学会研究報告 2000-CSEC-8、pp.19--24(2000)

４．宮崎一哉、鴨志田昭輝、中川路哲男：セキュアストレージシステムの開発（１）－電子データの長期保存における原本保証、秘匿、公証技術－、情報処理学会第 61 回全国大会公演論文集 CD-ROM(2000)

５．鴨志田昭輝、宮崎一哉、中川路哲男：セキュアストレージシステムの開発（２）－デジタル署名有効期限延長方式の提案－、情報処理学会第 61 回全国大会公演論文集 CD-ROM(2000)

６．ESTI TS 101 733 V1.2.2 「Electronic Signature Formats」 http://portal.esti.org/sec/ts_101733v010202p.pdf

７．IETF RFC3125「Electronic Signature Policies」http://www.ietf.org/rfc/rfc3125.txt

８．IETF RFC3126 「Electronic Signature Formats for long term electronic signatures」http://www.ietf.org/rfc/rfc3126.txt

９．宇根正志、松浦幹太、田倉昭：デジタルタイムスタンプ技術の現状と課題、日本銀行金融研究所 2000 年４月

１０．IETF RFC2560 http://www.ietf.org/rfc/rfc2560.txt

１１．ＳＣＶＰ http://www.ietf.org/internet-drafts/draft-ietf-pkix-scvp-07.txt

１２．IETF RFC2560bis
http://www.ietf.org/internet-drafts/draft-ietf-pkix-rfc2560bis-01.txt

１３．IETF RFC3029 http://www.ietf.org/rfc/rfc3029.txt

１４．ＤＳＶ，ＤＰＶ，ＤＰＤ
http://www.ietf.org/internet-drafts/draft-ietf-pkix-dsv-dpv-dpd-req-00.txt

１５．IETF RFC2630「CryptographicMessageSyntax」http://www.ietf.org/rfc/rfc2630.txt
１６．金融情報システムセンター「平成１３年度金融情報システム白書」
(財) 金融情報システムセンター編　財経詳報社
１７．金融情報システムセンター「電子保険取引における規制の現状と今後についての研究会報告書」平成 11 年度　(財) 金融情報システムセンター
１８．生命保険協会「生命保険業における個人データ保護の取扱指針」
http://www.seiho.or.jp/sisin/mokuji.html（2000 年 10 月時点）
１９．金融庁事務ガイドライン第二分冊：保険会社関係（平成 13 年 7 月 26 日改正）
http://www.fsa.go.jp/guide/guide.html（2000 年 10 月時点）
２０．金融商品の販売等に関する法律(平成 12 年法律第 101 号)
http://isweb4.infoseek.co.jp/business/gyoho/hanbai/index.html（2000 年 10 月時点）
２１．保険業法 (平成 7 年法律第 105 号)(最終改正　平成 12 年 5 月 31 日)
http://isweb4.infoseek.co.jp/business/gyoho/hou/index.html（2000 年 10 月時点）
２２．金融サービスの電子取引等と監督行政に関する研究会「金融サービスの電子取引の進展と監督行政」(平成 12 年 4 月 18 日版) http://www.fsa.go.jp/（2000 年 10 月時点）
２３．生命保険協会ホームページhttp://www.seiho.or.jp/（2000 年 10 月時点）

# 7 メンバーリスト

事務局

米倉　早織　電子商取引推進協議会　主席研究員

紙田　政典　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

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

木村　道弘　日本電気株式会社

宮崎　一哉　三菱電機株式会社

TF5 メンバー（編集メンバー）

| 氏名 | 会社名 |
|---|---|
| 藤川　真樹 | 綜合警備保障株式会社 |
| 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 |
| 鈴木　邦康 | 株式会社ＮＴＴデータ |
| 鈴木　優一 | エントラストジャパン株式会社 |
| 宍倉　勝仁 | シヤチハタ株式会社 |
| 松山　科子 | ソニー株式会社 |
| 橋本　正一 | 日本電信電話株式会社 |
| 松永　和男 | 株式会社日立製作所 |
| 小村　昌弘 | 富士通株式会社 |

SWG3 メンバー（参加メンバー）

| 氏名 | 会社名 |
|---|---|
| 森田　純生 | 株式会社イーアイティー |
| 市来　丈彦 | 株式会社エヌジェーケー |
| 丹羽　圭二 ＊ | 株式会社エヌジェーケー |
| 太田　高広 ＊ | 株式会社エヌジェーケー |
| 風間　博之 | 株式会社ＮＴＴデータ |
| 河田　悦生 | ＮＴＴドコモ株式会社 |
| 関野　公彦 ＊ | ＮＴＴドコモ株式会社 |
| 西川　一紀 | 沖電気工業株式会社 |
| 長谷川　亮 | 株式会社オリエントコーポレーション |
| 保倉　　豊 | グローバルフレンドシップ株式会社 |
| 鈴木　良信 | コンピュータ・アソシエイツ株式会社 |
| 森　　宣彦 ＊ | コンピュータ・アソシエイツ株式会社 |
| 雨宮　隆征 | セイコーインスツルメンツ株式会社 |
| 岩崎　善徳 ＊ | セイコーインスツルメンツ株式会社 |
| 岡　　誠 | ソニー株式会社 |
| 星野　　理 | 株式会社帝国データバンク |
| 中原　　康 | 株式会社東芝 |
| 藤岡　直美 | 日本アビオニクス株式会社 |
| 渡部　　明 | 日本コムシス株式会社 |
| 小暮　貢次郎 | 日本信販株式会社 |
| 酒井　雅啓 | 日本電気株式会社 |
| 野口　雄治 | 日本認証サービス株式会社 |
| 高橋　正一 | 日本ボルチモアテクノロジーズ |
| 大沼　保夫 | 日本ユニシス株式会社 |
| 高山　聡一郎 | 株式会社日立製作所 |
| 船越　　亘 | 株式会社富士通総研 |
| 大木　直人 | 三菱マテリアル株式会社 |

（注）＊はオブザーバー

平成 14 年 3 月発行
発行：電子商取引推進協議会
東京都港区芝公園 3-5-8
機械振興会館 3F
Tel 03-3436-7500
e-mail info@ecom.jp

この資料は再生紙を使用しています。