長期署名フォーマット相互運用性実験報告書

平成18年３月

次世代電子商取引推進協議会

序文

2005 年 4 月に e 文書法が施行され、これまで民間企業において紙での保存が義務付けられていた文書をデジタルデータで保存することが可能となった。長期保存を考えると、電子署名を行う人（署名者）とそれを数十年後に検証する人（検証者）が同じシステムを利用するとは限らず、非標準のシステムの導入はひとつのベンダに制約されるだけでなく、そのベンダのサービス停止によって、保管していた文書データの利用が困難になる可能性がある。

ECOM では 2000 年度から文書や電子署名文書の保存技術に関するガイドラインの作成や各種調査研究を行ってきた。これらの活動を通じて幾つかのベンダが実際の製品を開発したり、特定のプロジェクトでこれらの技術を用いるようになってきた。

しかし、多くの実装はベンダ内に閉じており、相互運用性については今まで実証されてこなかった。そこで、ECOM に「長期署名フォーマット普及ワーキンググループ」を設置し、これまでの成果を整理し、長期署名フォーマットのプロファイルを策定し、このプロファイルに基づいた相互運用性テスト環境を構築し、製品の相互運用性テストを実施した。

この実験では広く参加を求めるため 4 回のプレス発表を行った。以下にタイトルを示す。

・長期署名フォーマット製品の相互運用性テスト実施に向けた活動を開始（5/13）
・長期署名フォーマットプロファイルを完成し意見募集を開始（8/10）
・相互運用性テストの参加企業の募集を開始（9/21）
・相互運用性テストの結果報告会を 12 月 16 日に開催（12/14）

試験においては、テストデータの作成や試験用タイムスタンプ局の無償提供など多くの関係者に尽力頂いた。この場を借りて再度お礼を申し上げたい。

今年度の活動は、会員各位の多大なご協力のもとに、長期署名フォーマット実装製品の相互運用性確保に大いに寄与することができた。本報告が広く国内外から参照され、長期署名フォーマットの普及促進の一助になれば幸いである。

平成 18 年 3 月

次世代電子商取引推進協議会

目　次

まえがき

e-文書法の施行にともない、電子文書の署名を長期にわたって検証可能な長期署名フォーマット対応の製品が出揃ってきた。しかしながら、CAdES（CMS Advanced Electronic Signatures）やXAdES（XML Advanced Electronic Signatures）と呼ばれるベース標準は汎用的な標準にありがちな多数のオプションと曖昧性を含んでいることから、製品間の相互運用性の問題が顕在化することは想像に難くない。電子署名の検証が出来ないリスクを放置することは、しいては電子署名文書の普及を阻害する要因にもなりかねない。

本年度は、この電子署名の相互運用性の問題をとりあげ、長期署名フォーマットのプロファイルの開発と、このプロファイルを実装した製品間の相互運用性試験の実施、および試験結果のプロファイルへのフィードバックに取り組んだ。

長期署名フォーマットのプロファイルは、ベース標準のオプションに対してアーカイブタイムスタンプの利用を想定してその要否を明示するとともに、ベース標準の曖昧な表現に一意の解釈を与えた。製品実装においてこのプロファイルを参照することにより、ミニマムの実装かつ相互運用性確保が期待できる。

プロファイルへの準拠性を確認する相互運用性試験にはベンダ 14 社の参加があった。

今回の実証実験では、特別な実証実験センターのような施設は持たず、証明書失効リストを配布するだけのための HTTP サーバーと、株式会社 PFU より提供頂いたテスト用タイムスタンプサービスをホスティングしたのみとした。相互運用性試験により、潜在する多くの問題をあぶり出し、これらをプロファイルにフィードバックできたことは大きな成果である。本報告の巻末に最新のプロファイルを添付した。

相互運用性試験以外のトピックスとしては、2005 年 10 月に国際標準化された PDF/A への長期署名フォーマットの適用と、ここ数年来調査を続けている長期保存目的の記録媒体の新たな動向がある。前者については、TC171 の国内委員会のメンバと連携した活動を始めており、後者については磁気テープ媒体で進捗がみられた。

本報告書の構成は 2 部構成とした。

第 1 部は、長期署名フォーマットの相互運用性テストについて報告する。

第 1 章「長期署名フォーマットプロファイルについて」では、プロファイル策定にあたっての基本方針について解説する。

第 2 章「長期署名フォーマット相互運用性実験経過」では、今回の実験の実験内容、実験方法、実験環境及び実験結果を紹介する。

第 3 章「課題と考察」では、今回の実験を通じてあぶり出された問題について考察を加えた。

なお、今回作成した「長期署名フォーマットプロファイル」、および実験で用いた「テストケース設計書」については、CAdES 版、XAdES 版ともに付録として巻末に掲載する。

第 2 部「電子文書長期保存に係わる課題」では、文書フォーマットや保存媒体に関して最新の情報を提供する。

第１章「PDF/A 文書に対する長期署名の標準化」では、PDF/A 文書の国際標準化活動に対する、長期署名との整合性についての提言活動を紹介する。

第２章「長期保存に使用時の磁気テープの課題」では、近年性能が上がってきたといわれる磁気テープについて、文書を長期に保存する場合の媒体として利用する場合の分析を多面的に行い、その分析結果を述べる。

第1部　長期署名フォーマット相互運用性実験

1．長期署名フォーマットプロファイルについて

1.1　基本方針

長期署名フォーマットの最大の特長の一つがそのポータビリティである。つまり、仕様が公開されており、誰でも（TTP（Trusted Third Party）でないエンティティであっても）構築や検証が可能なばかりか、長期署名の構築処理を途中で他者へ継承することが可能である、という性質を持つ。ところが、CAdES[1][2][3][4][5][6][7][8][9]、XAdES[10][11][12]ともに仕様が複雑であり、我々が目的とする『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』（電子署名文書長期保存に関するガイドライン[13]より）にとって、冗長な定義も含んでいる。逆に必要と思われる定義が不足していたり、また、解釈に曖昧性を生じたりする部分もある。

そこで、これらの問題を解決するために、プロファイルを定義する。プロファイルを定義するにあたっての方針は次の通りである。

(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』を満足する。このとき、長期署名フォーマットによって運用する際に、署名者証明書やTSA証明書に対するルートCAの証明書については、CA等のTTPが長期にわたって保管することを想定する。

(2) ベースとする標準は可能な限り最新版を採用する。

(3) 相互運用性の確保を可能とする定義を行なう。つまり、解釈に曖昧性が生じる部分については明確化し、必要と思われるが明記されていない部分については補足する。明確化や補足においては、必然性及び妥当性を持つと思われる内容とする。

(4) 実装の負担軽減を目的として、冗長と思われる定義上の選択肢を削減し、実装を必須とする範囲を縮小する。

1.2　CAdES長期署名プロファイル策定の具体的な説明

(1) 準拠する標準仕様（ベース標準）について

本プロファイルのベース標準は、ETSI TS 101 733 V1.5.1（2003-12）, "Electronic Signature Formats"[8]とする。ただし、後述するように、本標準仕様はアーカイブタイムスタンプの生成方法に曖昧な定義が多く、プロファイルとして定義し切れなかった。つまり、アーカイブタイムスタンプ対象データの妥当と思われる捉え方が複数通り考えられ、必然的に誰もがこう考えるであろうという１つの方法に絞り込むことができなかった。そのため、アーカイブタイムスタンプについては、RFC3126[1],"Electronic Signature Formats for long term electronic signatures"（ETSI TS 101 733 V.1.2.2（2000-12）[4]）に準拠することとした。

(2) 署名フォーマットについて

ベース標準では、SignedData[14]のバージョンは3でなければならないとされている。ところが実際に利用されている署名アプリケーションにはバージョン3をサポートしていないものが多いと思われる。従って、本プロファイルではSignedDataのバージョンを3に限定しない。ただし、署名者の証明書が署名対象に含まれる必要があるため、署名属性に署名者証明書（のハッシュ値）を含むことを必須とする。

(3) 複数署名のサポートについて

複数署名には並列署名（Independent Signatures）と直列署名（Embedded Signatures）がある[14]。ベース標準で並列署名に長期署名を適用することは問題なく可能である。ところが、直列署名についてはベース標準にも詳細な記述がなく、また、単純に長期署名を適用することはできない。従って、本プロファイルにおいては、並列署名についてのみ対象に含めることとし、直列署名については今後の課題として保留することとする。

(4) ES-Tの署名タイムスタンプの検証情報について

署名タイムスタンプの検証情報（タイムスタンプの証明書からそのルートCAに至るまでの証明書パス及びそれぞれの証明書の失効情報）の格納場所について、次の3通りが考えられる（ベース標準では2)のみに言及されている）。

1) タイムスタンプトークン[15]内のcertificatesとcrls
2) ESの検証情報と同じ場所（Complete validation reference dataとExtended validation data）
3) タイムスタンプトークン内のunsigned attribute（Extended validation data形式）

検証情報の場所を対象の署名データと密接に管理できるようにすることと、構築における実装の容易性を考慮し、長期署名の構築時は、1)を推奨することとする。また、検証時は標準仕様で示唆されている2)、及びSignedData形式であるタイムスタンプトークンに長期署名フォーマットを適用した3)にも対応できるように、1)～3)全てに対応することを必須とする。

(5) ES-Xについて

本プロファイルの目的は、『デジタル署名の長期的な有効性の維持』である。そのためにはアーカイブタイムスタンプは必須である。従って、本プロファイルではアーカイブタイムスタンプが必ず付与されることを前提とする。

この場合、ESの検証情報やそのリファレンスもアーカイブタイムスタンプの付与対象であるため、ESの検証情報に付与されるタイムスタンプやそのリファレンスに付与されるタイムスタンプは冗長となる（ただし、検証情報が有効性を失う前にアーカイブタイムスタンプを付与する必要がある）。つまり、ES-X Type 1やES-X Type 2をサポートする必要はなく、ES-X Longのみをサポートすればよい。

(6) 失効情報について

ベース標準では、検証情報に含むことのできる失効情報として、CRL、OCSP レスポンス、その他の失効情報（OtherRevRefs、OtherRevVals）を可能としている。本プロファイルでは実質的に利用されていないその他の失効情報を選択不可とする。(ただし実証実験では、CRL のみを利用した)

(7) ES-A のアーカイブタイムスタンプの対象について

ベース標準 ETSI TS 101 733 V1.5.1 (2003-12) [8]によると、アーカイブタイムスタンプの対象は、

- The encapContentInfo element of the SignedData sequence;
- When present, the Certificates and crls elements of the SignedData sequence;
- Together with all data elements in the SignerInfo sequence including all signed and unsigned attributes.

を結合した値である。ところが、それぞれのデータの取り扱い方法が明示されていないため曖昧性が残り、次の 4 通りの方法が考えられてしまう。(詳細は 3.3 を参照)

[方法 1]

・encapContentInfo

eContentType、eContent を DER 符号化する。

・unsignedAttrs

各要素を IMPLICIT [0]のままで、DER 符号化する。SET OF の各要素についてはソートのみ行い、構成要素となる Attribute は、そのままバイナリデータとして扱う。

・他エレメント

そのままのバイナリデータとして扱う。

[方法 2]

・encapContentInfo

eContentType、eContent を DER 符号化する。

・unsignedAttrs

・DER 符号化

RFC 3852 の「5.4 Message Digest Calculation Process」における signedAttrs の DER 化を参考にして、IMPLICIT [1] tag ではなく、EXPLICIT SET OF tag として、DER 符号化する。構成要素となる Attribute は、そのままバイナリデータとして扱う。

・他エレメント

そのままのバイナリデータとして扱う。

[方法 3]

・全てのエレメントを DER 符号化する。ただし、IMPLICIT [0]、IMPLICIT [1]、などは、そのま

まの表現にする

[方法 4]
・全てのエレメントを DER 符号化する。方法 3 をベースに、IMPLICIT [0]、IMPLICIT [1]、などを EXPLICIT tag に直す。

一方、旧標準仕様 RFC3126[1],"Electronic Signature Formats for long term electronic signatures"（ETSI TS 101 733 V.1.2.2（2000-12））[4]におけるアーカイブタイムスタンプの対象の定義には、曖昧性がない。従って、本プロファイルでは、旧標準仕様 RFC3126,"Electronic Signature Formats for long term electronic signatures"（ETSI TS 101 733 V.1.2.2（2000-12））に準拠することとする。

(8) ES-A のアーカイブタイムスタンプの検証情報について

アーカイブタイムスタンプの検証情報（タイムスタンプの証明書からそのルート CA に至るまでの証明書パス及びそれぞれの証明書の失効情報）の格納場所について、次の 2 通りが考えられる（標準仕様では言及されていない）。

1) タイムスタンプトークン内の certificates と crls
2) タイムスタンプトークン内の unsigned attribute（Extended validation data 形式）

本プロファイルでは、検証情報の場所を対象の署名データと密接に管理できるようにすることと、構築における実装の容易性を考慮し、構築時には 1)に格納することを推奨し、検証時には 1),2)を処理できることを必須とする。

(9) ハッシュ対象の正規化について

ASN.1 データ型は、データ型、長さ、および値を表すバイトによる可変長データ構造をあらわすことが可能なものであり、様々な通信プロトコルや公開鍵証明書、署名データなどで利用されている。

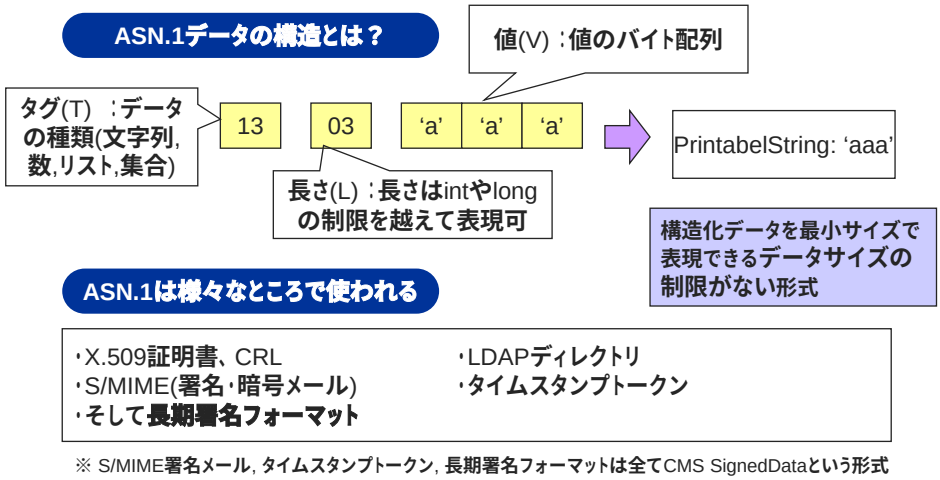


図 1.1-1 ASN.1 構造と用途

長期署名フォーマット（CAdES 版）は、一般的な電子署名データで使われる CMS SignedData（RFC 3369）[16]を拡張したもので、基本的に BER エンコーディングが用いられるが、署名やハッシュ対象は DER エンコーディングの規則が適用される。

- CMS SignedData の「SignedAttributes」
  RFC 3852[14]の「5.4. Message Digest Calculation Process」において、EXPLICIT SET OF tag で DER エンコードすることが規定されており、SET OF 型の構成要素は下記ルールでソートする必要がある。

（X.690[17] 11 Restrictions on BER employed by both CER and DER より）

11.6 Set-of components
The encodings of the component values of a set-of value shall appear in ascending order, the encodings being compared as octet strings with the shorter components being padded at their trailing end with 0-octets.

NOTE - The padding octets are for comparison purposes only and Do not appear in the encodings.

DER の「SET OF」型のソートの規定

このように署名やハッシュ生成時には BER→DER 正規化が必要となる（図 1.1-2 参照）。

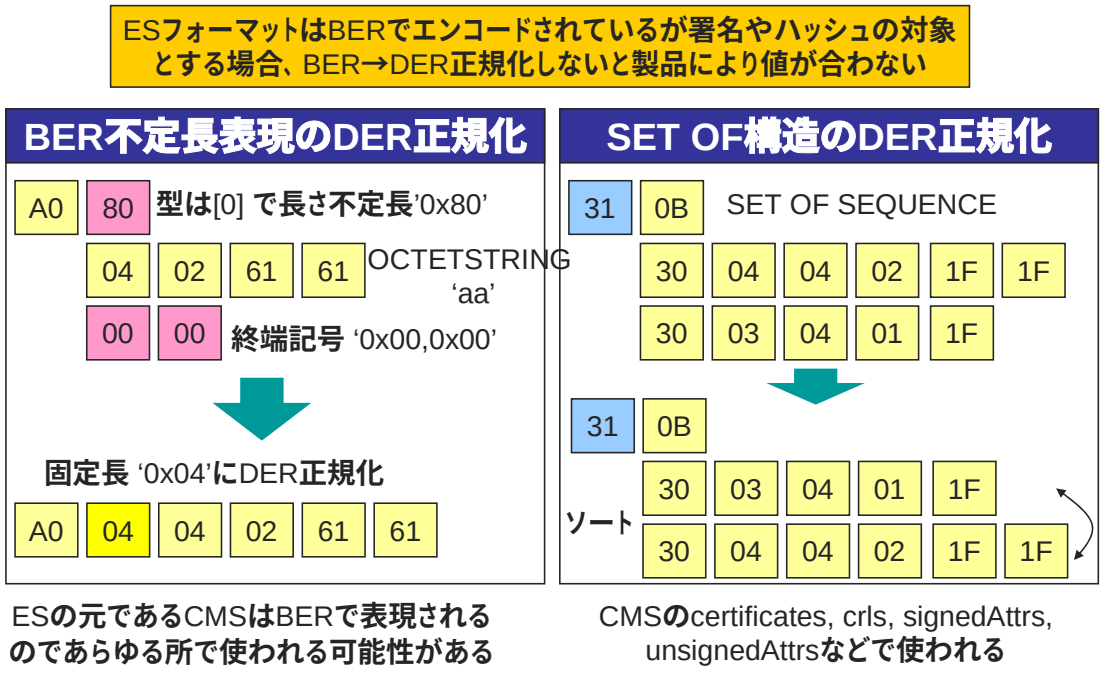


図 1.1-2 主要な DER 正規化操作

ここで ECOM プロファイルでは、署名タイムスタンプやアーカイブタイムスタンプの検証情報（CertificateSet, RevocationInfos）を CMS SignedData の certificates および crls フィールドに格納し BER でエンコーディングすることとしているが、アーカイブタイムスタンプ生成時には署名タイムスタンプ、アーカイブタイムスタンプをアーカイブハッシュ対象とするため、これも BER→DER（SET OF のソートも含む）正規化をする必要がある。このとき、例えばタイムスタンプトークンの中身まで正規化するかどうかが問題になる（図 1.1-3 参照）。signedAttr や encapContentInfo など、正規化対象は種々存在するため、それぞれ、できるだけ統一的にルールを定めておく必要がある。

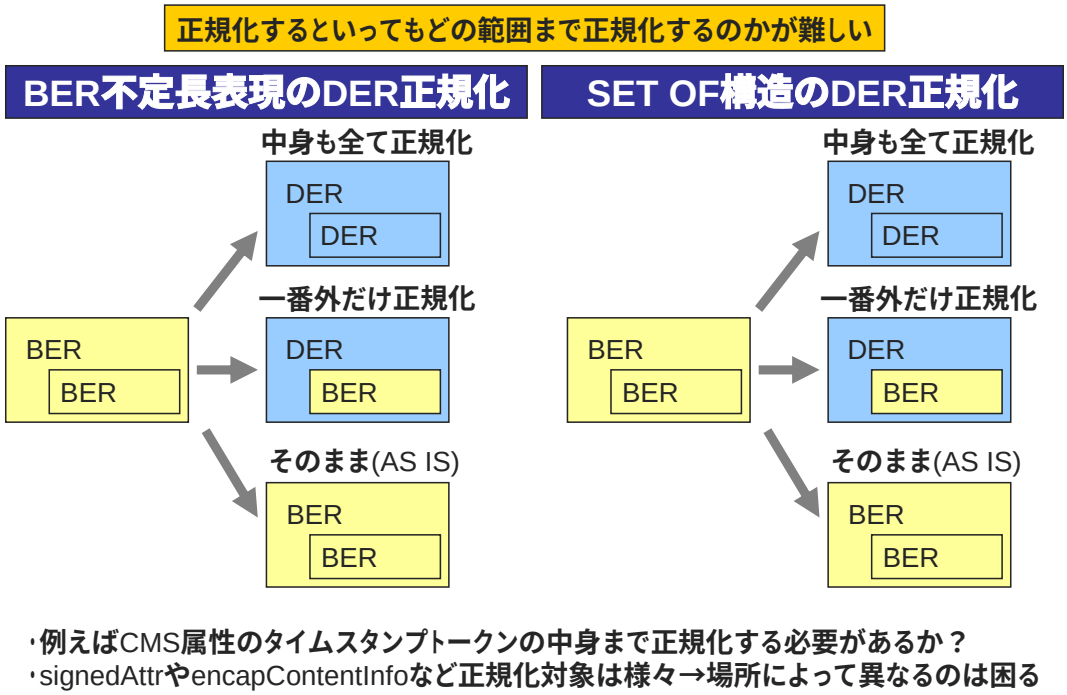


図 1.1-3 DER 正規化の範囲

本実験では「ハッシュ値生成前に行う DER SET 正規化の要件（注：データ出力の要件ではない）として、アーカイブタイムスタンプのハッシュ対象に含まれる SET の扱い」を以下とした。

- RFC 3369[16]により SignedAttributes … DER 正規化必要（SET 昇順）
- その他 … DER 正規化なし。読み込んだデータそのものとして処理。
- ハッシュに加える順序はプロファイルのリストで示した順序。

1.3 XAdES 長期署名プロファイル策定の具体的な説明

(1) ベース標準について

本プロファイルは、ETSI TS 101 903 V1.3.1 (2005-05), "XML Advanced Electronic Signatures (XAdES)"[12]に準拠するものとする。ただし、実験開始時点では国内ベンダーでどの程度

XAdES の実装者が存在するか不明であったので、準拠する仕様が他のバージョンの場合についても許容する。

また、同様の理由から 1.1(3)の方針に従った本標準仕様に対する仕様の追記については、踏み込んだプロファイルを策定できなかった。この点に関しては、本節以降および課題の節で後述する。

(2) 複数署名について

複数署名には並列署名（Independent Signatures）と直列署名（Countersignatures）の形式が考えられるが、V1.3.1 の仕様においても並列署名に長期署名を適用することについては問題ない。一方、直列署名に関しては XAdES の仕様においては、Countersignature が付与された最初の署名に対してアーカイブタイムスタンプを付与することは想定されているが、それ以外のパターンについて考慮されていない。したがって、プロファイルでは今後の課題として保留とする。

(3) ES-C、X、X-Long、X-Long Type1、X-Long Type2 について

本プロファイルの目的は、『デジタル署名の長期的な有効性の維持』である。そのためにはアーカイブタイムスタンプは必須である。従って、本プロファイルではアーカイブタイムスタンプが必ず付与されることを前提とする。

この場合、XAdES-BES や XAdES-BPES の検証情報やそのリファレンスもアーカイブタイムスタンプの付与対象であるため、XAdES-BES や XAdES-BPES の検証情報に付与されるタイムスタンプやそのリファレンスに付与されるタイムスタンプは冗長となる（ただし、検証情報が有効性を失う前にアーカイブタイムスタンプを付与する必要がある）。つまり、XAdES-X Type 1 や XAdES-X Type 2 は必要ない。次に、XAdES-X-L であるが、XAdES の仕様では XAdES-X-L を構築する際に XAdES-X Type 1 や XAdES-X Type 2 が必須となっている。タイムスタンプのコスト面や 1.1.(2)の基本方針を考慮し、XAdES-X-L についてもプロファイルとしては必須としない。

(4) XAdES-T の署名タイムスタンプの証拠情報について

署名タイムスタンプの検証情報（タイムスタンプの証明書からそのルート CA に至るまでの証明書パス及びそれぞれの証明書の失効情報）の格納場所について、次の 3 通りが考えられる（標準仕様では 2)のみに言及している）。

1) タイムスタンプトークン内の certificates と crls

2) 署名者証明書の検証情報と同じ場所（Complete validation reference data と Extended validation data）

3) タイムスタンプトークン内の unsigned attribute（Extended validation data 形式）

実験開始時点では国内ベンダーでどの程度 XAdES の実装者が存在するか不明であったこと、および XAdES の仕様でも厳密に規定されていないことからプロファイルとしても特に規定しない。

(5) ES-A のアーカイブタイムスタンプの証拠情報について

アーカイブタイムスタンプの検証情報（タイムスタンプの証明書からそのルート CA に至るまでの証明書パスおよびそれぞれの証明書の失効情報）の格納方法について、大きく分けて以下の 2 通りが考えられる。

① タイムスタンプトークンに証拠情報を埋め込む方法

② タイムスタンプトークンを別途保管する方法

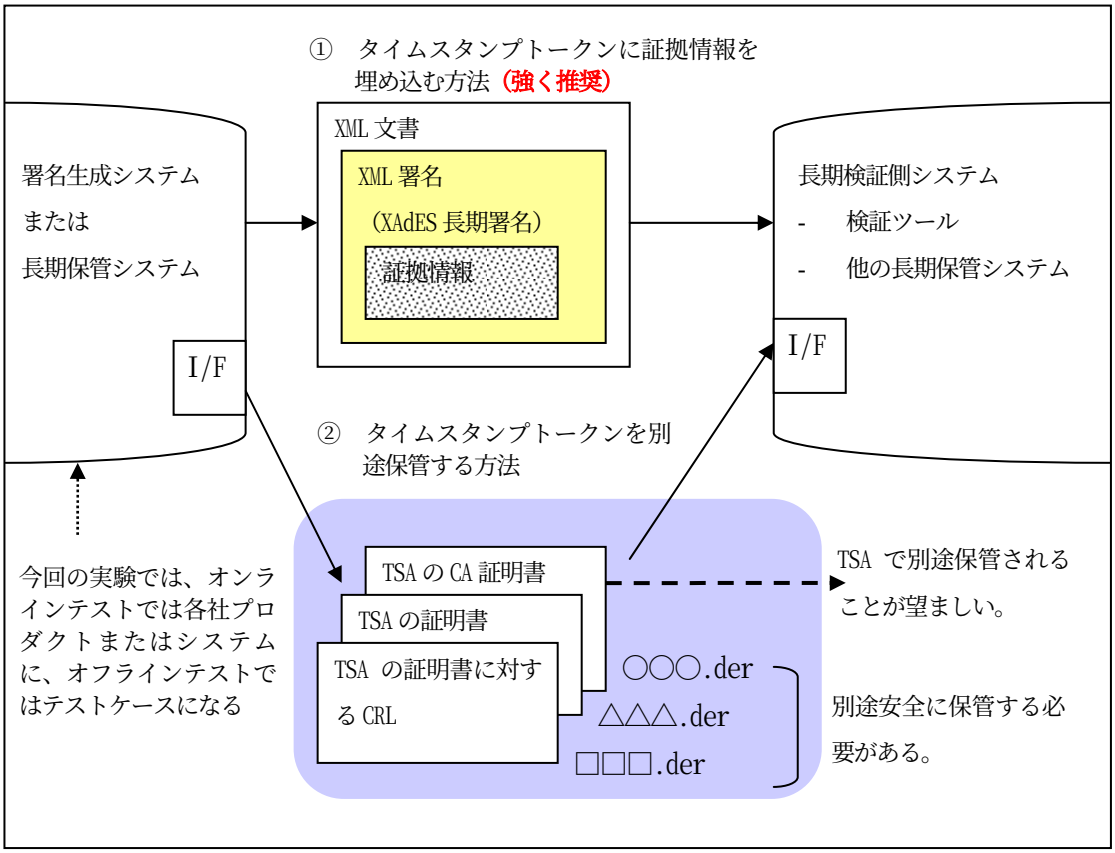


図 1.1-4 タイムスタンプトークンの証拠情報の扱いについて：タイムスタンプトークンの証拠情報の扱い方については①タイムスタンプトークン自身に埋め込む方法と②別途安全に保管する方法の二通りが考えられるが、①を強く推奨する。

相互互換性の観点から考えると、検証方法を明確に定義できるのでプロファイルとしては①の方法を強く推奨する。また、①の方法を選択した場合の証拠情報の長期署名フォーマットへの格納方法は次の 2 通り（両方ともベース標準では言及されていない）考えられるが、構築時には 1) に格納することを推奨し、検証時には 1),2) を処理できることを推奨する。

1) タイムスタンプトークン内の certificates と crls

2) タイムスタンプトークン内の unsigned attribute（Extended validation data 形式）

一方、②の方法は相互互換性の観点から考えると、別途保管したタイムスタンプの証拠情報を用いて検証者がタイムスタンプを検証できる必要があるため、検証者の検証プログラムがタイス

タンプの証拠情報を読み込む機能とそれらを使った長期署名フォーマットの検証を実行できる必要がある。

なお、課題と考察の節で後述するように、相互互換性試験に関しては①と②のどちらの方法でも良いようにプロファイルを定義して実験を行った。しかし、結果的に課題が生じたため、最終的なプロファイルは上記のように定義することとする。

1.4 参考文献

[1] RFC 3126 Electronic Signature Formats for long term electronic signatures, Sep 2001, D.Pinkas et. al, http://www.ietf.org/rfc/rfc3126.txt

[2] Internet Draft CMS Advanced Electronic Signatures (CAdES), J.Ross, et al.,Dec 2005, http://www.ietf.org/internet-drafts/draft-ietf-smime-cades-01.txt

[3] ETSI TS 101 733 V1.2.1 (2000-09) Electronic Signatures and Infrastructures (ESI); Electronic Signature Formats, Sep 2000, ETSI

[4] ETSI TS 101 733 V1.2.2 (2000-12) Electronic Signatures and Infrastructures (ESI); Electronic Signature Formats, Dec 2000, ETSI

[5] ETSI TS 101 733 V1.2.3 (2001-12) Electronic Signatures and Infrastructures (ESI); Electronic Signature Formats, Dec 2001, ETSI

[6] ETSI TS 101 733 V1.3.1 (2002-02) Electronic Signatures and Infrastructures (ESI); Electronic Signature Formats, Feb 2002, ETSI

[7] ETSI TS 101 733 V1.4.0 (2002-09) Electronic Signatures and Infrastructures (ESI); Electronic Signature Formats, Sep 2002, ETSI

[8] ETSI TS 101 733 V1.5.1 (2003-12) Electronic Signatures and Infrastructures (ESI); Electronic Signature Formats, Dec 2003, ETSI

[9] ETSI TS 101 733 V1.6.3 (2005-09) Electronic Signatures and Infrastructures (ESI); CMS Advanced Electronic Signatures (CAdES), Sep 2005, ETSI

[10] ETSI TS 101 903 V1.1.1 (2002-02) XML Advanced Electronic Signatures (XAdES), Feb 2002, ETSI

[11] ETSI TS 101 903 V1.2.2 (2004-04) XML Advanced Electronic Signatures (XAdES), Apr 2004, ETSI

[12] ETSI TS 101 903 V1.3.1 (DRAFT) XML Advanced Electronic Signatures (XAdES), DRAFT, ETSI

[13]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・公証ワーキンググループ

[14] RFC 3852 Cryptographic Message Syntax (CMS), Jul 2004, R.Housley, http://www.ietf.org/rfc/rfc3852.txt

[15] RFC 3161 Internet X.509 Public Key Infrastructure Time-Stamp Protocol (TSP), Aug 2001, C.Adams et al, http://www.ietf.org/rfc/rfc3126.txt

[16] RFC 3369 Cryptographic Message Syntax (CMS), Aug 2002, R.Housley, http://www.ietf.org/rfc/rfc3369.txt

[17] ITU-T X.690 Information technology - ASN.1 encoding rules: Specification of Basic Encoding Rules (BER), Canonical Encoding Rules (CER) and Distinguished Encoding Rules (DER), Jul 2002, ITU-T

2．長期署名フォーマットプロファイルの相互運用性テスト

次世代電子商取引推進協議会（ECOM）では 2000 年より電子文書の長期保存に関する調査、普及啓蒙活動を行ってきた[1][2][3][4][5][6][7][8][9]。2005 年度の ECOM の活動として、6 月に CAdES および XAdES に基づく長期署名フォーマットプロファイルを策定し、秋に本プロファイルに対する製品および製品の準拠性および、製品間の相互運用性を確認するための実証実験を実施した。この実証実験には日本国内の IT ベンダー14 社が参加した。本 2 節では実証実験の概要、実験方法および実験結果について報告する。また、第 3 節では実験により得られた相互運用上の課題を述べる。

2.1　実験計画

e-文書法（通称）が施行され、電子署名文書の長期保存技術が実際に利用され始めている。しかしながら、相互運用性のない署名文書保存システムの導入は利用者に多くのリスクを強いる恐れがあるばかりでなく、電子署名文書の保存技術の普及を阻害する要因になる。

次世代電子商取引推進協議会（ECOM）では「長期署名保存フォーマット普及ワーキンググループ」を設置し、RFC 3126 や XAdES などの標準に基づく長期署名フォーマットを普及させるために、データ構造や処理手順の必要条件をまとめた「長期署名フォーマットのプロファイル」を策定した。このプロファイルに基づいたテスト仕様を作成し、参加各社の製品の相互運用性テストを行ない、テストに合格した製品を公表した。本年度実証実験の概要を以下の図で示す。

目的	**以下を目的としてテストを設計･実施する** **･ECOM長期署名フォーマットプロファイルへの準拠性の確認** **･各組織の製品が生成するデータの相互運用性を確認**
期間	2005年10月～12月を予定
募集	2005年9月
テスト参加資格(テスト公表結果に記載される)	**･原則、ECOM会員（＋ 参加を呼びかけた海外を含む組織等）** **･CAdESもしくはXAdESのES-T、ES-C、ES-X Long、ES-Aのいずれかのフォーマットの生成、検証できるソフトを持つ組織(TA設定必須。TSP設定)** **･文書管理ソフトでもライブラリでも可（テスト内容を工夫）** **･製品、プロトタイプの別は問わない**
内容	**･オフライン共通データ検証テスト** **･製品マトリックス相互生成･検証テスト**
結果公表	**･テスト結果は2005年冬頃、公表予定** **･個々のテスト項目に対する結果の公表は組織に委ねる**

図 1.2-1 実証実験概要

実証実験は以下に示すスケジュールで実施された。

08/10 ECOM プロファイル
09/09 ECOM プロファイル意見募集締切
09/21 実証実験プロジェクト実施公示
10/05 実証実験プロジェクト説明会
10/20 PFU タイムスタンプサービス利用 NDA の回収
10/20 オンラインテスト用証明書の配布
10/30 実証実験参加申し込み、担当者提出締切
11/08 CRL 取得猶予期間対応の CRL 発行に修正
11/10 オンラインテスト用データ（署名対象、証明書）配布
11/10 オフラインテストケースドラフト配布
11/28 テストケース fix
12/12 オンラインテストデータ回収
12/15 オンライン/オフライン実験結果回収
12/16 実証実験結果報告セミナー

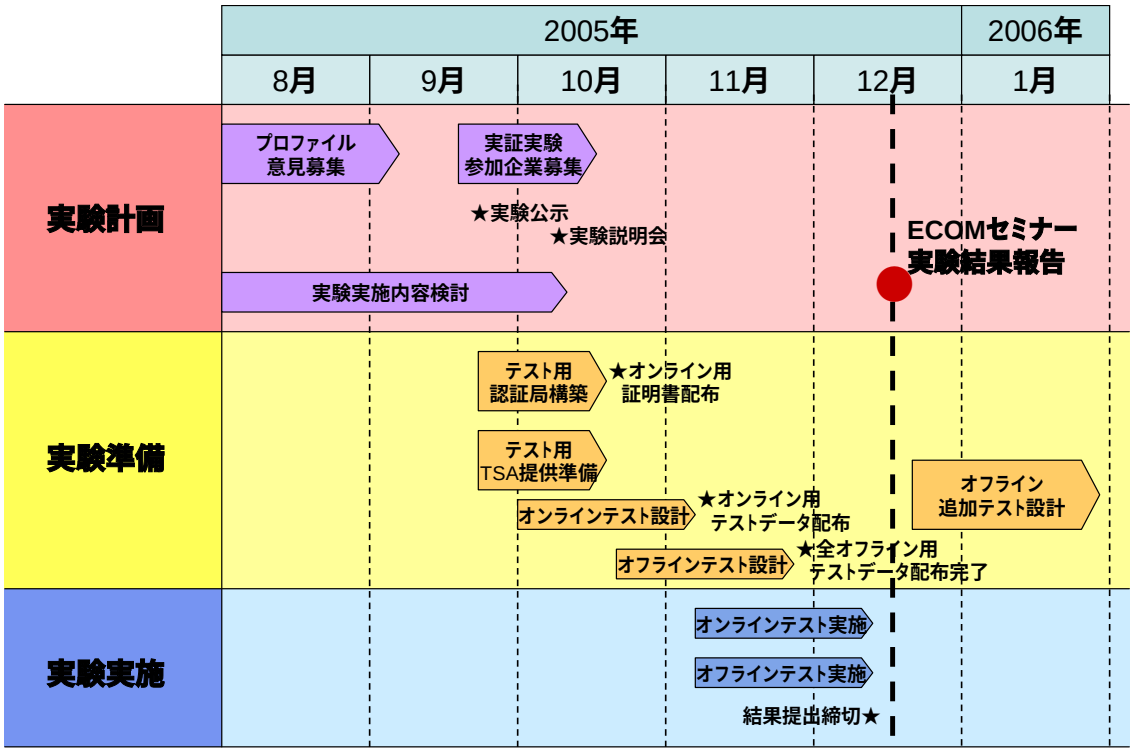


図 1.2-2 実証実験スケジュール

ETSI および中国、韓国、チャイニーズ台北などの関係機関への呼びかけを行ったが、残念ながら参加可能な実装が無いことやスケジュールの面で調整できず海外からの参加は無かった。しか

しながら、実証実験の結果について期待を寄せていた。

2.2 実験方法

今回の実証実験では、特別な実証実験センターのような施設は持たず、証明書失効リストを配布するだけのためのHTTPサーバーと、株式会社PFUよりご提供頂いた、テスト用タイムスタンプサービスをホスティングしたのみとした。

実験参加企業は、配布されたテストケース設計書および証明書、私有鍵、署名対象データ、長期署名フォーマットデータや各種設定ファイルで構成されるテストデータをメーリングリストなどの方法で取得し、インターネットに接続可能な環境でそれぞれ実証実験を実施した。実験担当者間の連絡、プロファイルの解釈、問題の指摘などのディスカッション基本的にはメーリングリストベースで進められたが、対面での議論が必要であると判断された場合に幾度か会合を持った。

本実験では、長期署名フォーマット ECOM プロファイルもしくは、国際標準に基づく長期書名フォーマットのデータを扱う製品の相互運用性を検証する。検証の際に使用するデータは今回の実験期間内にとどまらず、国内外の実装を持つ利用者が広く利用できるようテストデータを設計し公開することとする。

テストデータは以下の点に配慮しながら設計されている。

- テスト対象となる実装が文書管理アプリケーションであっても開発ツールキットであってもできる。
- 製品の利用者であってもテストを実施できるようにブラックボックステストとする。
- テスト実施期間後であっても、国内外の実装を持つ開発者やユーザがテストできるように、テストデータを配布可能とし、テストの実行時刻に依存せず、また、オフラインでも実施できるようなものとする。
- テストデータのプロファイルは基本的にはECOMプロファイルに則ったものとする。

一般的にPKIや署名を扱う製品の相互運用性テストを行う際に問題となるのが、各製品が生成もしくは出力できるデータは「正しい」とされるデータのみであり、意図的に「正しくない」データを生成することは困難である。しかしながら、「正しい」とされるデータによるテストだけでは、十分な検証機能を備えているかを判断することはできない。そこで、今回の実証実験では、参加企業の実装が生成したデータを、他の全ての参加企業が互いに検証するテストと、「成功系・失敗系」を含む検証機能を確認するためのテストの2種類を行うこととする。前者を「オンラインマトリックス生成・検証テスト」、後者を「オフライン検証テスト」と呼ぶことにする。

2.2.1 オンラインマトリックス生成・検証テスト

ある実装が生成した有効な長期署名フォーマットのデータが相互に読み込みおよび検証がで

きることを確認するためのテストを行う。あらかじめ指定された署名対象データ、証明書、CRL、タイムスタンプサービスを用いて参加企業全ての製品により長期署名フォーマットデータ(ES-T, ES-X Long, ES-A）を生成する。参加企業の各製品において、他社製品の生成したデータが有効であることを検証する。CRL およびタイムスタンプトークンはオンラインで取得する。

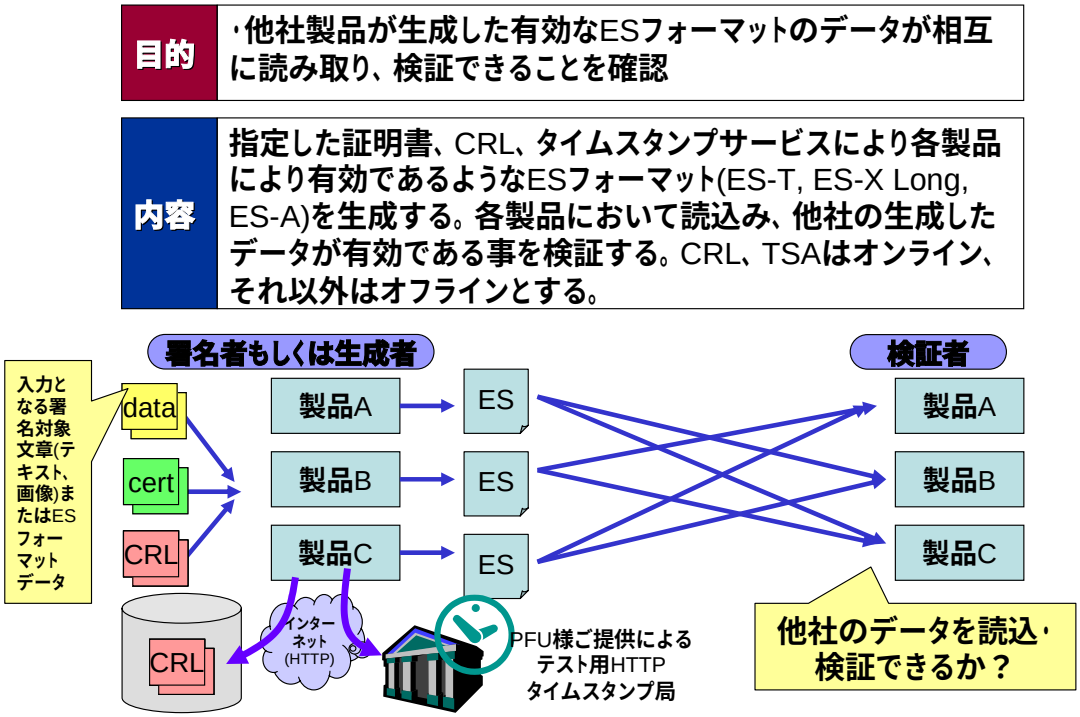


図 1.2-3 オンラインマトリックス生成・検証テスト

テストケースは 10 項目ある。オプショナルテストとして、ETSI TS 101 733 v1.5.1 に基づく新しいアーカイブタイムスタンプのハッシュ値の計算方法を用いたテストもある。

参加企業の実装では、ES-C フォーマットを出力できる製品が少ないために ES-C フォーマットを実験対象から外した。また、文書管理製品で ES-X Long の状態で ES-A とは異なる別のセキュアなアーカイビング方法を採用して保存する製品もあるため、ES-X Long はテスト対象に加えた。

2.2.2 オフライン検証テスト

ECOM プロファイルに基づく共通の ES フォーマットデータを用いて正しく検証することができるかを確認する。テストツールにより生成された ES フォーマットのデータ（ES, ES-T, ES-C, ES-X long, ES-A)、証明書、CRL、署名対象データをもとに、検証結果が期待値と一致するかどうかを確認する。

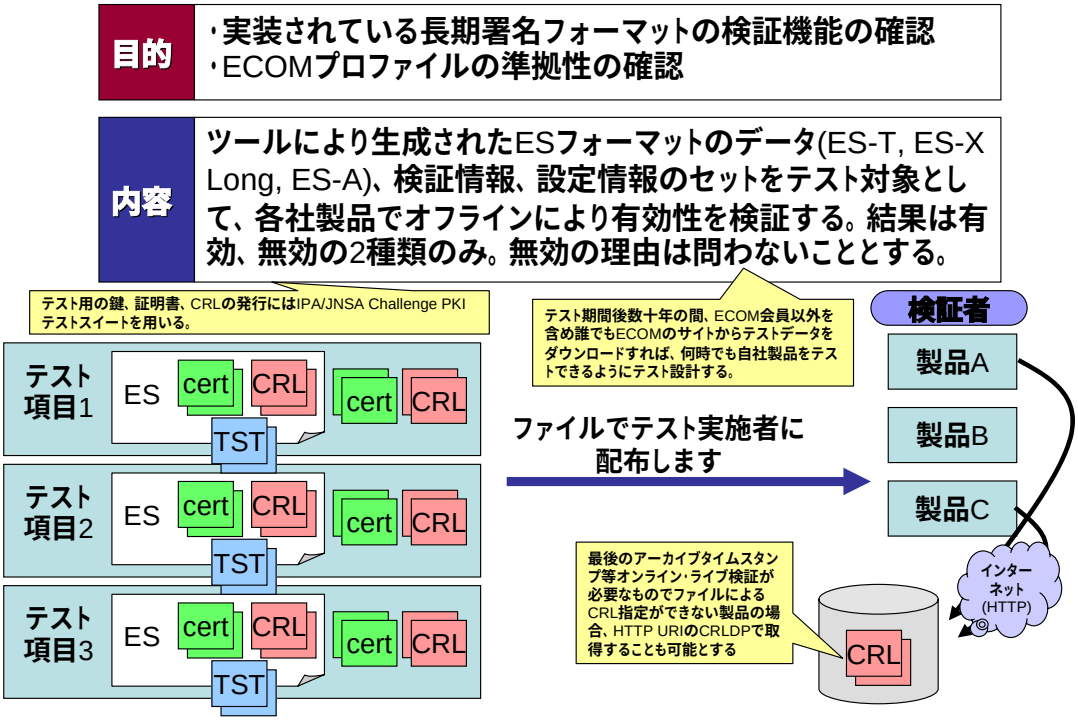


図 1.2-4 オフライン検証テスト

テスト項目は以下の内容を含む全 30 テスト項目となっている。

- ES-T，ES-C，ES-X Long，ES-A フォーマットの検証
- 内包署名と分離署名
- ハッシュアルゴリズム（SHA-1，SHA-256，SHA-512）
- BES と EPES
- RFC3126 と ETSI TS 101733 v1.5.1 以降のアーカイブハッシュ
- SigningTime および SignatureTimeStamp の時刻を考慮した失効、期限切れ検証
- 各種ハッシュ値の改竄の検証
- コンテンツタイムスタンプ
- 並列署名（＝独立署名）
- 署名ポリシーファイルを考慮した検証

2.2.3　合否判定基準

今回の実証実験の合否判定基準は以下のように定めた。

オンラインマトリックス生成・検証テスト合否判定基準
ある参加企業の実装が指定されたフォーマット（ES-T, ES-X Long, ES-A）を生成し、これを他の参加企業の実装により検証を行い、これらの企業より 8 割を超えるデータ無効の報告があがらなければ、その実装は正しい生成機能を具備しているとし、合格とする。
オフライン検証テスト合否判定基準
参加企業の実装が、オフライン検証テストの各フォーマット（ES-T, ES-C, ES-X Long, ES-A）の標準テストを実施し、テストケースの全てが成功、即ち、テストケースを構成する全てのテスト項目の期待値が一致しているならば、その実装は各フォーマットの検証機能を具備しているとし、合格とする。オプショナルテストケースの結果は合否には影響しない。

2.3　実験環境

2.3.1　実証実験用認証局

オンラインマトリックス生成・検証においては、成功系のテストしか行わないために、CRL の発行周期さえ配慮すれば、どのような認証局が発行する証明書でも構わないが、オフライン検証については一般の認証局ソフトウェア、もしくはサービスでは問題がある。長期署名フォーマットのテストでは、署名やタイムスタンプを行う時刻が検証における重要なファクターであるために、例えばテストの目的により、過去や未来の時点において存在していたはずの証明書を発行したり、極端に長い有効期限の証明書を発行したり、タイムスタンプ局のための証明書を発行する必要がある。

また、過去に様々な組織により行われた PKI に関する実証実験の経験から、証明書や鍵ペアは複数のテストケースで利用することを避け、テストケース毎に証明書の主体者名を意味のある名称、例えばテストケース名で分ける方が望ましい。これにより、テストデータ中の証明書を一瞥しただけで、テストデータがどのテストケースのためのデータであり、テストの期待値はどのようなものかを知ることができる。証明書失効リストについては、テスト期間中実験データの誤りが発覚し、入れ替えを行っても実装がキャッシュを使うかどうかにより実験結果が異なる場合がある。このような問題を切り分けるためにサブ CA もまたテストケース毎にわける方が望ましい。

このような要件を満足する証明書および証明書失効リストを発行するためのツールとして NPO 日本ネットワークセキュリティ協会で開発されたオープンソースのツールである Challenge PKI Test Suite を利用することとした。CGI によるインタフェースにより、証明書の各フィールドを詳細に渡り指定することができ、一部のみが異なるような証明書を効率的に発行することができる。

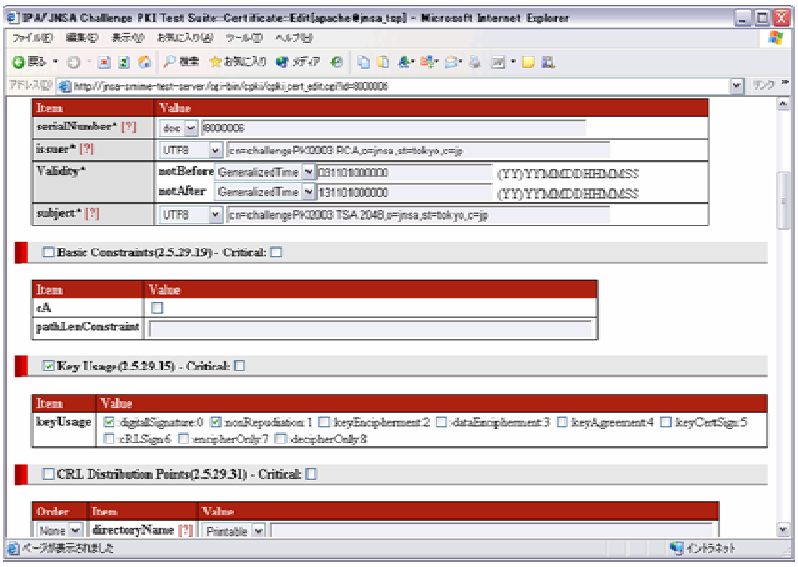


図 1.2-5 Challenge PKI Test Suite の証明書発行画面

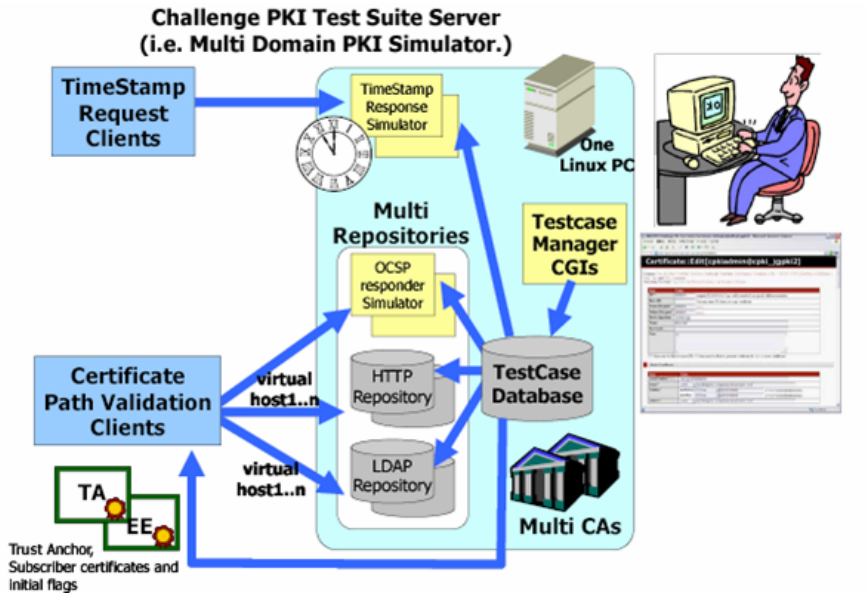


図 1.2-6 Challenge PKI Test Suite

ツールには、署名用とタイムスタンプ局用のサブ CA およびエンドエンティティ、および証明書発行リストを一括登録するための機能、CRL を毎日発行するための機能を追加している。

2.3.2 実証実験用タイムスタンプ局

オフラインテストで用いられるタイムスタンプトークンは、ツールにより生成されたトークンを用い、実証実験参加者がトークンを取得する必要は無いが、オンラインマトリックス生成・検証テストにおいては、参加者に対し RFC 3161 に基づくプロトコルでタイムスタンプトークンを取得できるような環境を提供する必要がある。

今回の実験では株式会社 PFU 様の協力により、実証実験参加企業に対し、機密保持契約の締結の下で商用サービスとほぼ同等のテスト用タイムスタンプ局サービスを無償で提供頂いた。

- タイムスタンプ局利用アカウント
- タイムスタンプトークン発行ライセンス
- サービス利用のための RFC 3161 対応の Java もしくは C 言語の SDK
- サービス利用 API のインタフェース仕様の開示

2.3.3 実証実験における信頼モデル

今回の実証実験における認証局の信頼モデルを以下に示す。

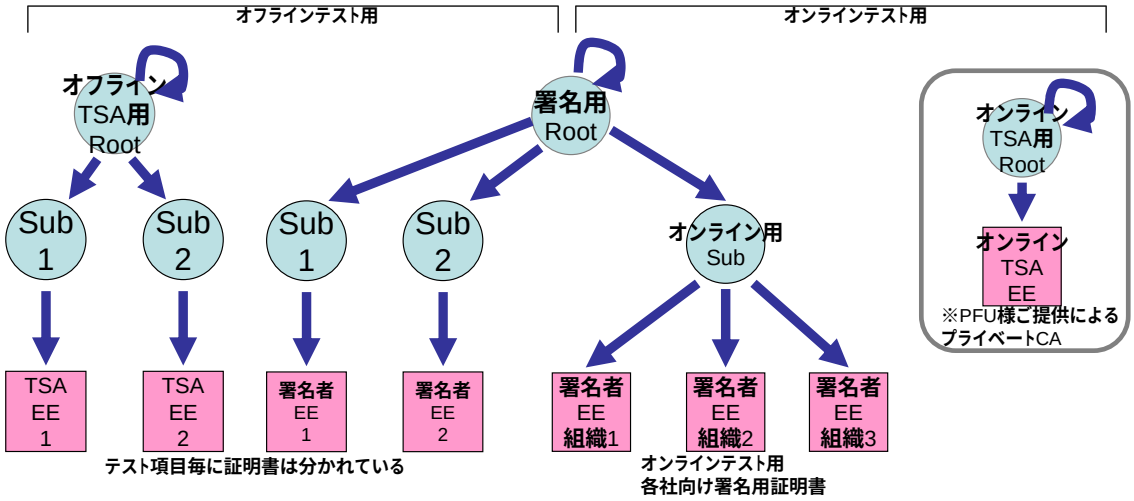


図 1.2-7 実証実験用認証局の信頼モデル

日本における特定認証業務の認証局と商用のタイムスタンプ局のトラストアンカは別々であることから、署名用認証局とタイムスタンプ局のトラストアンカは分けることとした。また、テストケース毎にサブ CA を分けている。テストケースにおいてトラストアンカを固定とすることにより、バッチによるテスト実施が可能な実装の場合には効率的にテストが行える。

2.3.4 テストデータ

オンラインマトリックス生成・検証テストおよび、オフライン検証テストのテストケース設計はエントラストジャパン株式会社が行った。テストデータの作成については、CAdES テストケースのデータはエントラストジャパン株式会社が、XAdES テストケースのデータについては日本電気株式会社が作成を行った。

2.4 実験参加企業

本年度の ECOM 長期署名フォーマット相互運用性実証実験には、CAdES 実証実験に 10 社、XAdES 実証実験に 3 社、実験協力に 1 社、合わせて 14 社が実証実験に参加した。

- CAdES 実証実験参加企業（五十音順）
  - RSA セキュリティ株式会社
  - 株式会社 NTT データ
  - 株式会社システムコンサルタント（株式会社日本電子公証機構）
  - 株式会社ハイパーギア
  - 株式会社 PFU
  - 株式会社日立製作所（システム開発研究所）
  - セコム株式会社
  - 日本電信電話株式会社（情報流通プラットフォーム研究所）
  - 三菱電機インフォメーションシステムズ株式会社
  - 三菱電機株式会社
- XAdES 実証実験参加企業（五十音順）
  - 関電システムソリューションズ株式会社
  - 日本電気株式会社
  - 富士ゼロックス株式会社
- 実験協力（テスト環境構築、テストケース設計、テストデータ提供）
  - エントラストジャパン株式会社

実証実験に使用した実装は既存製品のバージョンアップ版、新製品、文書管理アプリケーション、開発ライブラリ、プロトタイプなど多様な実装で実験が行われた。各実装の名称、バージョン、製品/プロトタイプ、リリース予定時期とまとめたのが以下の表である。

	企業名(略称)	種別	製品名	Ver	リリース予定
CAdES	RSAセキュリティ	既存製品	RSA BSAFE e文書法対応ライブラリ	V1.1(β1)	‘06.02
	NTTデータ	試作品	長期署名対応プラットフォーム(プロトタイプ)	0.1	―
	セコム	既存製品	セコム長期署名ライブラリ	1.3	‘06.01
	日本電子公証機構	既存製品	JN++ 電子署名タイムスタンプSDKキット	2.0	2006
	NTT	試作品	CYNOS-L(プロトタイプ)	1	―
	ハイパーギア	既存製品	HG/PscanServ Pro	4.0	‘06.02
	PFU	試作品	PFU長期署名ライブラリ(プロトタイプ)	0.1	―
	日立製作所	試作品	長期署名フォーマットライブラリ(プロトタイプ)	0.1	―
	三菱電機	試作品	―	―	―
	MDIS	既存製品	三菱署名有効性延長システムMistyGuard<EVERSIGN>	2.0	‘06春
XAdES	関電システム	新規製品	XAdES長期署名ライブラリ for .NET	1.3	‘06.01
	NEC	試作品	PKIサーバ/Carassuite原本保管サーバ	3.0プロトタイプ	―
	富士ゼロックス	新規製品	ArcSuite	―	―

図 1.2-8 実証実験に参加した実装一覧

2.5 実験結果

実証実験結果に基づくテスト合否結果をまとめたのが下表となる。

	参加企業	種別	オフライン検証テスト					オンラインマトリックス生成･検証テスト							
								データ生成				データ検証			
			ES-T	ES-C	ES-X	ES-A	ES-A 1.5.1	ES-T	ES-X	ES-A	ES-A 1.5.1	ES-T	ES-X	ES-A	ES-A 1.5.1
CAdES	RSAセキュリティ	製品版	✓	-	✓	✓	-	✓	✓	✓	-	✓	✓	✓	-
	NTTデータ	試作版	✓	-	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	セコム	製品版	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	日本電子公証機構	製品版	✓	✓	✓	✓	-	✓	✓	✓	-	✓	✓	✓	-
	NTT	試作版	✓	✓	✓	✓	-	✓	✓	✓	-	✓	✓	✓	-
	ハイパーギア	製品版	✓	-	✓	✓	-	✓	✓	✓	-	✓	✓	✓	-
	PFU	試作版	-	-	-	-	-	✓	✓	✓	-	-	-	-	-
	日立製作所	試作版	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	三菱電機	試作版	✓	-	✓	✓	-	✓	✓	✓	-	✓	✓	✓	-
	MDIS	製品版	✓	-	-	✓	-	✓	-	✓	-	✓	✓	✓	-
XAdES	関電システム	製品版	✓	-	-	✓	-	✓	-	✓	-	✓	-	✓	-
	NEC	試作版	✓	-	-	✓	-	✓	-	✓	-	✓	-	✓	-
	富士ゼロックス	製品版	✓	-	-	✓	-	✓	-	✓	-	✓	-	✓	-

凡例: “✓” 合格 “-” 非サポート

図 1.2-9 実証実験合否結果

2.5.1　オンラインマトリックステスト実験結果

本節では実験参加企業による各実装により生成されたテストデータを、他の実装がエラー無く読み込み正しく検証できるかどうかを確認するオンラインマトリックス生成・検証実験の結果を示す。今回の各実験結果を集計したものが下図となる。

	データ生成企業名(略称)	種別	ES-T	ES-XL	ES-A	備考
CADES	RSAセキュリティ	既存製品	○	○	○	
	NTTデータ	試作品	○	○	○	
	セコム	既存製品	○	○	○	
	日本電子公証機構	既存製品	○	○	○	
	NTT	試作品	○	○	○	
	ハイパーギア	既存製品	○	○	○	
	PFU	試作品	○	○	○	
	日立製作所	試作品	○	○	○	
	三菱電機	試作品	○	○	○	
	MDIS	既存製品	○	－	○	
XADES	関電システム	新規製品	○	－	○	
	NEC	試作品	○	－	○	
	富士ゼロックス	新規製品	○	－	○	

凡例：
○：サポート(合格)
×：不合格
－：製品非サポート

図 1.2-10 オンラインテスト集計結果

フォーマットのサポートを表明している参加企業 13 社のそれぞれの実装において、生成されたデータの相互運用性に問題が無いことが確認された。

以降の詳細結果において、結果を表す記号は以下の通りである。

“ ○ ”: 検証成功
“ × ”: 検証失敗
“ － ”: データ無しのため実験未実施
“ 無印 ”: サポートせず等、検証側の理由により実験未実施

2.5.1.1 CAdES オンラインマトリックステスト実験結果

① ON-T-1 実験結果

生成＼検証	RSAセキュリティ	NTTデータ	セコム	日本電子公証機構	NTT	HYPERGEAR	PFU	日立	三菱電機	MDIS	備考
RSAセキュリティ		○	○	○	○	○		○	○	○	
NTTデータ	○		○	○	○	○		○	○	○	
セコム	○	○		○	○	○		○	○	○	
日本電子公証機構	○	○	○		○	○		○	○	○	
NTT	○	○	○	○		○		○	○	○	
HYPERGEAR	○	○	○	○	○			○	○	○	
PFU	○	○	○	○	○	○		○	○	○	
日立	○	○	○	○	○	○			○	○	
三菱電機	○	○	○	○	○	○		○		○	
MDIS	○	○	○	○	○	○		○	○		

図 1.2-11 ON-T-1 内包署名 ES-T テスト結果

② ON-T-2 実験結果

企業名	RSAセキュリティ	NTTデータ	セコム	日本電子公証機構	NTT	HYPERGEAR	PFU	日立	三菱電機	MDIS	備考
RSAセキュリティ		―	―	―	―			―	―	―	
NTTデータ			○	○	○			○	○	○	
セコム		○		○	○			○	○	○	
日本電子公証機構		○	○		○			○	○	○	
NTT		―	―	―				―	―	―	
HYPERGEAR		―	―	―	―			―	―	―	
PFU		○	○	○	○			○	○	○	
日立		○	○	○	○				○	○	
三菱電機		○	○	○	○			○		○	
MDIS		○	○	○	○			○	○		

図 1.2-12 ON-T-2 分離署名 ES-T 実験結果

③　ON-X-1 内包署名 ES-X Long 実験結果

生成＼検証	RSAセキュリティ	NTTデータ	セコム	日本電子公証機構	NTT	HYPERGEAR	PFU	日立	三菱電機	MDIS	備考
RSAセキュリティ		△	○	○	○	○		○	○		
NTTデータ	○		○	○	○	○		○	○		
セコム	○	△		○	○	○		○	○		
日本電子公証機構	○	△	○		○	○		○	○		
NTT	○	(○)	○	○		○		○	○		
HYPERGEAR	○	△	○	○	○			○	○		
PFU	○	△	○	○	○	○		○	○		
日立	○	△	○	○	○	○			○		
三菱電機	○	△	○	○	○	○		○			
MDIS	―	―	―	―	―	―		―	―		

図 1.2-13 ON-X-1 内包署名 ES-X Long 実験結果

④　ON-X-2 分離署名 ES-X Long 実験結果

企業名	RSAセキュリティ	NTTデータ	セコム	日本電子公証機構	NTT	HYPERGEAR	PFU	日立	三菱電機	MDIS	備考
RSAセキュリティ		―	―	―	―			―	―		
NTTデータ		△	○	○	○			○	○		
セコム		○		○	○			○	○		
日本電子公証機構		△	○		○			○	○		
NTT		―	―	―				―	―		
HYPERGEAR		―	―	―	―			―	―		
PFU		△	○	○	○			○	○		
日立		△	○	○	○				○		
三菱電機		△	○	○	○			○			
MDIS	―	―	―	―	―			―	―		

図 1.2-14 ON-X-2 分離署名 ES-X Long 実験結果

⑤　ON-A1-1 内包署名第一世代 ES-A 実験結果

生成 ＼ 検証	RSAセキュリティ	NTTデータ	セコム	日本電子公証機構	NTT	HYPERGEAR	PFU	日立	三菱電機	MDIS	備考
RSAセキュリティ		○	○	○	○	○		○	○	○	
NTTデータ	○		○	○	○	○		○	○	○	
セコム	○	○		○	○	○		○	○	○	
日本電子公証機構	○	○	○		○	○		○	○	○	
NTT	○	○	○	○		○		○	○	○	
HYPERGEAR	○	○	○	○	○			○	○	○	
PFU	○	○	○	○	○	○		○	○	○	
日立	○	○	○	○	○	○			○	○	
三菱電機	○	○	○	○	○	○		○		○	
MDIS	○	○	○	○	○	○		○	○		

図 1.2-15 ON-A1-1 内包署名第一世代 ES-A 実験結果

⑥　ON-A2-1 内包署名第二世代 ES-A 実験結果

企業名	RSAセキュリティ	NTTデータ	セコム	日本電子公証機構	NTT	HYPERGEAR	PFU	日立	三菱電機	MDIS	備考
RSAセキュリティ		○	○	○	○	○		○	○	○	
NTTデータ	○		○	○	○	○		○	○	○	
セコム	○	○		○	○	○		○	○	○	
日本電子公証機構	○	○	○		○	○		○	○	○	
NTT	○	○	○	○		○		○	○	○	
HYPERGEAR	○	○	○	○	○			○	○	○	
PFU	○	○	○	○	○	○		○	○	○	
日立	○	○	○	○	○	○			○	○	
三菱電機	○	○	○	○	○	○		○		○	
MDIS	○	○	○	○	○	○		○	○		

図 1.2-16 ON-A2-1 内包署名第二世代 ES-A 実験結果

⑦　ON-A1-2 分離署名第一世代 ES-A 実験結果

生成 ＼ 検証	RSAセキュリティ	NTTデータ	セコム	日本電子公証機構	NTT	HYPERGEAR	PFU	日立	三菱電機	MDIS	備考
RSAセキュリティ		―	―	―	―			―	―	―	
NTTデータ			○	○	○			○	○	○	
セコム		○		○	○			○	○	○	
日本電子公証機構		○	○		○			○	○	○	
NTT		―	―	―				―	―	―	
HYPERGEAR		―	―	―	―			―	―	―	
PFU		○	○	○	○			○	○	○	
日立		○	○	○	○				○	○	
三菱電機		○	○	○	○			○		○	
MDIS		○	○	○	○			○	○		

図 1.2-17 ON-A1-2 分離署名第一世代 ES-A 実験結果

⑧　ON-A2-2 分離署名第二世代 ES-A 実験結果

企業名	RSAセキュリティ	NTTデータ	セコム	日本電子公証機構	NTT	HYPERGEAR	PFU	日立	三菱電機	MDIS	備考
RSAセキュリティ		―	―	―	―			―	―	―	
NTTデータ			○	○	○			○	○	○	
セコム		○		○	○			○	○	○	
日本電子公証機構		○	○		○			○	○	○	
NTT		―	―	―				―	―	―	
HYPERGEAR		―	―	―	―			―	―	―	
PFU		○	○	○	○			○	○	○	
日立		○	○	○	○				○	○	
三菱電機		○	○	○	○			○		○	
MDIS		○	○	○	○			○	○		

図 1.2-18 ON-A2-2 分離署名第二世代 ES-A 実験結果

⑨　ON-A1-3 アーカイブハッシュ v1.5.1 方式内包署名第一世代 ES-A 実験結果

生成 ＼ 検証	RSAセキュリティ	NTTデータ	セコム	日本電子公証機構	NTT	HYPERGEAR	PFU	日立	三菱電機	MDIS	備考
RSAセキュリティ		―	―					―			
NTTデータ			○					○			
セコム		○						○			
日本電子公証機構		―	―					―			
NTT		―	―					―			
HYPERGEAR		―	―					―			
PFU		―	―					―			
日立		○	○								
三菱電機		―	―					―			
MDIS		―	―					―			

図 1.2-19 ON-A1-3 アーカイブハッシュ v1.5.1 方式内包署名第一世代 ES-A 実験結果

⑩　ON-A2-3 アーカイブハッシュ v1.5.1 方式内包署名第二世代 ES-A 実験結果

企業名	RSAセキュリティ	NTTデータ	セコム	日本電子公証機構	NTT	HYPERGEAR	PFU	日立	三菱電機	MDIS	備考
RSAセキュリティ		―	―					―			
NTTデータ			○					○			
セコム		○						○			
日本電子公証機構		―	―					―			
NTT		―	―					―			
HYPERGEAR		―	―					―			
PFU		―	―					―			
日立		○	○								
三菱電機		―	―					―			
MDIS		―	―					―			

図 1.2-20 ON-A2-3 アーカイブハッシュ v1.5.1 方式内包署名第二世代 ES-A 実験結果

2.5.1.2 XAdES オンラインマトリックステスト実験結果

① XAdES ON-T-1 実験結果

生成＼検証	関電システム	NEC	富士ゼロックス	備考
関電システム		○	○	
NEC	○		○	
富士ゼロックス	ー	ー		

図 1.2-21 XAdES ON-T-1 内包署名 ES-T テスト結果

② ON-T-2 実験結果

生成＼検証	関電システム	NEC	富士ゼロックス	備考
関電システム		○	○	
NEC	○		○	
富士ゼロックス	○	○		

図 1.2-22 XAdES ON-T-2 分離署名 ES-T 実験結果

③　ON-A1-1 内包署名第一世代 XAdES-A 実験結果

生成＼検証	関電システム	NEC	富士ゼロックス	備考
関電システム			○	
NEC	○		○	
富士ゼロックス	ー	ー		

図 1.2-23 XAdES ON-A1-1 内包署名第一世代 ES-A 実験結果

④　ON-A2-1 内包署名第二世代 XAdES-A 実験結果

生成＼検証	関電システム	NEC	富士ゼロックス	備考
関電システム			×	※3 第一世代アーカイブタイムスタンプのTSA検証情報が第二世代ES-A内に含まれていない場合の検証はサポートしていない。
NEC	○		○	
富士ゼロックス	ー	ー		

図 1.2-24 XAdES ON-A2-1 内包署名第二世代 XAdES-A 実験結果

⑤　ON-A1-2 分離署名第一世代 XAdES-A 実験結果

生成＼検証	関電システム	NEC	富士ゼロックス	備考
関電システム			○	
NEC	○		○	
富士ゼロックス	○	○		

図 1.2-25 XAdES ON-A1-2 分離署名第一世代 XAdES-A 実験結果

⑥　ON-A2-2 分離署名第二世代 ES-A 実験結果

生成＼検証	関電システム	NEC	富士ゼロックス	備考
関電システム			×	※3 第一世代アーカイブタイムスタンプのTSA検証情報が第二世代ES-A内に含まれていない場合の検証はサポートしていない。（プロファイル的には正しい）
NEC	○		○	
富士ゼロックス	○	○		

図 1.2-26 XAdES ON-A2-2 分離署名第二世代 XAdES-A 実験結果

2.5.2　オフライン共通データ検証テスト実験結果

本節では実験参加企業による実装を用いたオフライン共通データ検証テストの実験結果を示す。ES-T, ES-C, ES-X Long, ES-A をサポートしていると表明している参加企業の実装において、必須テスト項目は全てテストが成功していることが確認された。

	データ生成企業名(略称)	種別	ES-T	ES-XL	ES-A	備考
CAdES	RSAセキュリティ	既存製品	○	○	○	
	NTTデータ	試作品	○	○	○	
	セコム	既存製品	○	○	○	
	日本電子公証機構	既存製品	○	○	○	
	NTT	試作品	○	○	○	
	ハイパーギア	既存製品	○	○	○	
	PFU	試作品	ー	ー	ー	
	日立製作所	試作品	○	○	○	
	三菱電機	試作品	○	○	○	
	MDIS	既存製品	○	ー	○	
XAdES	関電システム	新規製品	○	ー	○	
	NEC	試作品	○	ー	○	
	富士ゼロックス	新規製品	○	ー	○	

凡例：
○：サポート(合格)
×：不合格
ー：製品非サポート

図 1.2-27 オフライン検証テストの必須テスト項目合格集計結果

実験参加企業から報告された配布されたテストデータに対する実験結果を集計したものは以下の通りであった。

2.5.2.1 CAdES オフライン共通データ検証テスト実験結果

テスト項目番号	期待値	RSAセキュリティ	セコム	日立	NTT	PFU	日本電子公証機構	HYPERGEAR	MDIS	NTTデータ	三菱電機	テスト項目名
10001	有効	o	o	o	o	-	o	o	o	o	o	EST-ATTACH-NORMAL-OK
10002	無効	o	o	o	o	-	o	o	o	o	o	EST-ATTACH-EXPIRED-NG
10003	無効	o	o	o	o	-	o	o	o	o	o	EST-ATTACH-REVOKED-NG
10004	有効	o	o	o	o	-	o	o	o	o	o	EST-ATTACH-SIGTIME-REVOKED-OK
10005	無効	o	o	o	o	-	o	o	o	o	o	EST-ATTACH-SIGTS-REVOKED-NG
10006	無効	o	o	o	o	-	o	o	o	o	o	EST-ATTACH-ES-SIG-FORGED-NG
10007	無効	o	o	o	o	-	o	o	o	o	o	EST-ATTACH-SIGTS-SIG-FORGED-NG
10008	無効	o	o	o	o	-	o	o	o	o	o	EST-ATTACH-ES-MESSAGEDIGEST-FORGED-NG
10009	無効	o	o	o	o	-	o	o	o	o	o	EST-ATTACH-SIGTSTST-MESSAGEDIGEST-FORGED-NG
10010	有効	ー	o	o	o	-	o	-	o	o	o	EST-DETACH-NORMAL-OK
20001	有効	ー	o	-	-	-	o	-	o	o	o	EST-OTHERCERT-SHA256-OK
20002	有効	ー	o	-	-	-	o	-	o	o	o	EST-SIGTS-SHA256-OK
20003	有効	ー	o	-	-	-	o	-	o	o	o	EST-SIGTS-SHA512-OK
20004	有効	-	-	-	-	-	-	-	-	-	-	EST-CONTENT-TIMESTAMP-OK
20005	有効	-	-	-	-	-	-	-	-	-	-	EST-INDEPENDENT-SIGNATURES-OK
20006	有効	-	-	-	-	-	-	-	-	-	-	EST-EPES-WITHOUT-HASHCHECK-OK
20007	有効	-	-	-	-	-	-	-	-	-	-	EST-EPES-NORMAL-OK
20008	無効	-	-	-	-	-	-	-	-	-	-	EST-EPES-POLICY-HASH-NOT-MATCHING-NG
20009	無効	-	-	-	-	-	-	-	-	-	-	EST-EPES-NOT-BEFORE-VIOLATION-NG
20010	無効	-	-	-	-	-	-	-	-	-	-	EST-EPES-MANDATED-SIGNEDATTRS-VIOLATION-NG
20011	無効	-	-	-	-	-	-	-	-	-	-	EST-EPES-EXTERNAL-SIGNEDDATA-VIOLATION-NG
40001	有効	ー	o	o	o	-	o	-	-	-	-	ESC-ATTACH-NORMAL-OK
40002	有効	ー	o	o	o	-	o	-	-	-	-	ESC-DETACH-NORMAL-OK
50001	有効	o	o	o	o	-	o	o	-	o	o	ESXL-ATTACH-NORMAL-OK
50002	有効	ー	o	o	o	-	o	-	-	o	o	ESXL-DETACH-NORMAL-OK
60001	有効	o	o	o	o	-	o	o	-	o	o	ESXL-ATTACH-SIGTS-VALIDATIONINFO-NOT-INCLUDED-OK
70001	有効	o	o	o	o	-	o	o	o	o	o	ESA1-ATTACH-NORMAL-OK
70002	有効	ー	o	o	o	-	o	-	o	o	o	ESA1-DETACH-NORMAL-OK
80001	有効	-	o	o	-	-	-	-	-	o	-	ESA1-ATTACH-ETSI151-OK
80002	有効	-	o	o	-	-	-	-	-	o	-	ESA1-DETACH-ETSI151-OK

凡例
必須テスト項目
オプショナルテスト項目

図 1.2-28 CAdES オフラインテスト テスト項目検証結果

結果記号の意味は以下の通り。

“○”: 期待値と一致。期待値無効のテスト項目の場合には、テスト項目の説明にある無効理由と同等の警告を表示した場合も無効とみなしてよい。

“×”: 期待値と不一致。またはテスト項目と無関係の支障の無い警告が出る場合。

“ー”: 実装では対応せず。

テスト結果より以下のことが確認された。

- 必須テスト項目については、実装済みとしている全ての製品において期待値が一致している。
- 全実装が内包署名に対応しているが、分離署名の実装は半数程度。
- 多くの実装が ES-T、ES-A のみを扱う実装となっている。

2.5.2.2 XAdES オフライン共通データ検証テスト実験結果

本実験参加企業から報告された配布されたテストデータに対する実験結果を集計したものは以下の通りであった。

テスト項目番号	期待値	富士ゼロックス	関電ソリューションズ	NEC	テスト項目名
10001	有効	○	○	○	XAdEST-ATTACH-NORMAL-OK
10002	無効	○	○	○	XAdEST-ATTACH-EXPIRED-NG
10003	無効	○	○	○	XAdEST-ATTACH-REVOKED-NG
10004	有効	○	○	○	XAdEST-ATTACH-SIGTIME-REVOKED-OK
10005	無効	○	○	○	XAdEST-ATTACH-SIGTS-REVOKED-NG
10006	無効	○	○	○	XAdEST-ATTACH-ES-SIG-FORGED-NG
10007	無効	○	○	○	XAdEST-ATTACH-SIGTS-SIG-FORGED-NG
10008	無効	○	○	○	XAdEST-ATTACH-ES-MESSAGEDIGEST-FORGED-NG
10009	無効	○	○	○	XAdEST-ATTACH-SIGTSTST-MESSAGEDIGEST-FORGED-NG
10010	有効	○	○	○	XAdEST-DETACH-NORMAL-OK
70001	有効	○	○	○	XAdESA1-ATTACH-NORMAL-OK
70002	有効	○	○	○	XAdESA1-DETACH-NORMAL-OK

図 1.2-29 XAdES オフラインテスト テスト項目検証結果

結果記号の意味は以下の通り。

“○”: 期待値と一致。期待値無効のテスト項目の場合には、テスト項目の説明にある無効理由と同等の警告を表示した場合も無効とみなしてよい。
“×”: 期待値と不一致。またはテスト項目と無関係の支障の無い警告が出る場合。
“－”: 実装では対応せず。

テスト結果より以下のことが確認された。

- 各テスト項目については、実装済みとしている全ての製品において期待値が一致している。
- 全実装が分離署名に対応しているが、内包署名については実装されていないものある。
- 検証については複数バージョンを扱えるものもある。

2.6 テストデータおよびテストケースの公開

テストケース設計書や全てのテストデータを含む本年度実証実験のテストスイートは ECOM のサイト（http://www.ecom.jp）よりダウンロード可能であり、製品や試作品を開発した企業や、その利用者が自由にテストすることができる。全ての認証局の証明書および鍵ペアもまた含まれているので、認証局の鍵を用いて新たなテストケースを容易に追加することができる。

ただし、実証実験の期間後では、株式会社 PFU より提供頂いたテストサイトが使えないこと。そして、CRL を配布する HTTP リポジトリのサービスが停止してしまうので、ファイル指定で失効情報を指定するか、自ら HTTP リポジトリとなるウェブサーバーを立ち上げ、そのホスト名を実証実験で用いたものと同じにしてテストするという方法がある。

2.7 参考文献

[1] 平成 16 年度 EC 技術基盤の相互運用性に関する調査研究（電子署名生成・検証システムのセキュリティ環境の国際標準化等の調査)電子文書の長期保存と見読性に関するガイドライン,平成 17 年 2 月,電子商取引推進協議会・財団法人日本情報処理開発協会電子商取引推進センター
[2] 平成 15 年度 EC 技術基盤の相互運用性に関する調査研究（電子署名生成・検証システムのセキュリティ環境の国際標準化等の調査)署名ポリシー調査報告書,平成 16 年 3 月,電子商取引推進協議会・財団法人日本情報処理開発協会電子商取引推進センター
[3] 平成 15 年度 EC 技術基盤の相互運用性に関する調査研究（電子署名生成・検証システムのセキュリティ環境の国際標準化等の調査）電子署名文書長期保存に関する実用化動向調査報告書,平成 16 年 3 月,電子商取引推進協議会・財団法人日本情報処理開発協会電子商取引推進センター
[4] 平成 15 年度 EC 技術基盤の相互運用性に関する調査研究（電子署名生成・検証システムのセキュリティ環境の国際標準化等の調査）電子文書の長期保存と見読性に関する調査報告書,平成 16 年 3 月,電子商取引推進協議会・財団法人日本情報処理開発協会電子商取引推進センター
[5] 平成 14 年度 EC 技術基盤の相互運用性に関する調査研究（電子署名生成・検証システムのセキュリティ環境の国際標準化等の調査)タイムスタンプサービス調査報告書,平成 15 年 3 月,電子商取引推進協議会・財団法人日本情報処理開発協会電子商取引推進センター
[6] 平成 14 年度 EC 技術基盤の相互運用性に関する調査研究（電子署名生成・検証システムのセキュリティ環境の国際標準化等の調査）タイムスタンプサービス利用ガイドライン,平成 15 年 3 月,電子商取引推進協議会・財団法人日本情報処理開発協会電子商取引推進センター
[7] 平成 14 年度 EC 技術基盤の相互運用性に関する調査研究（電子署名生成・検証システムのセキュリティ環境の国際標準化等の調査）タイムスタンプサービス運用ガイドライン,平成 15 年 3

月,電子商取引推進協議会・財団法人日本情報処理開発協会電子商取引推進センター
[8] 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・公証ワーキンググループ
[9] 電子署名文書長期保存に関する中間報告,平成 13 年 3 月,電子商取引推進協議会 認証・公証ワーキンググループ

3．課題と考察

本章では、プロファイル作成ならびに相互運用実証実験より得られた、長期署名フォーマットの相互運用上の課題を考察する。

3.1　共通の課題

CAdES フォーマットと XAdES フォーマットに共通の問題として ES-T，ES-C，ES-X long，各世代の ES-A フォーマットの含まれる署名者やタイムスタンプ局の証明書を検証する際、何時の時点での有効性を検証する必要があるのか、解釈のずれによる検証結果の不整合があった。

- ES-T，ES-C，ES-X long，ES-A フォーマットのそれぞれにおいて、署名者証明書および TSA 証明書を何時の時点での有効性を確認すればよいのか、誤った解釈をしていた製品があった。

- 失効情報を取得するための猶予期間（Grace Period）を厳密に考慮する実装と、そうでない実装があり、厳密であるが故に ES データを生成できないという問題があった。

各フォーマットに含まれる証明書の検証時刻を図 1.3-1 にまとめる。

フォーマット	証明書の種類	有効性を確認すべき日時
ES-T	署名者証明書	署名タイムスタンプ属性のトークンの時刻
	署名タイムスタンプTSA証明書	現在時刻
ES-C ES-XL	署名者証明書	署名タイムスタンプ属性のトークンの時刻
	署名タイムスタンプTSA証明書	現在時刻またはセキュアアーカイブされた時刻
ES-A	署名者証明書	署名タイムスタンプ属性のトークンの時刻
	署名タイムスタンプTSA証明書	第1世代アーカイブタイムスタンプのトークンの時刻
	第1世代アーカイブタイムスタンプTSA証明書	第2世代アーカイブタイムスタンプのトークンの時刻
	第2世代アーカイブタイムスタンプTSA証明書	第3世代アーカイブタイムスタンプのトークンの時刻
	:	:
	第n-1世代アーカイブタイムスタンプTSA証明書	第n世代アーカイブタイムスタンプのトークンの時刻
	第n世代アーカイブタイムスタンプTSA証明書	現在時刻

図 1.3-1 各証明書の検証時刻

3.2　CAdES に関する課題

3.2.1　新しいアーカイブタイムスタンプのハッシュ計算法に関する問題

3.2.1.1　ハッシュ計算法の正規化に関する問題と実証実験における対策

ETSI TS 101 733 v1.5.1[1]以降もしくは IETF に提案されている新しい CAdES のインターネッ

トドラフトに基づくアーカイブタイムスタンプの計算方法を用いたテストをオンラインテスト、オフラインテスト共に 2 社が参加してテストを行った。この新しいハッシュ計算方法のテストを実施するにあたり、標準仕様ではハッシュ対象の正規化方法についての記述が殆ど無いため我々は数多くの前提条件を定めなければならなかった。

- 全てのハッシュ対象要素は ASN.1 構造のタグ、長さ、値のバイト配列を含むこととし、値のみではない。
- unsignedAttrs フィールドの SET OFF 構造を DER で正規化するが､内部構造は正規化しない。
- certificates、crls、signedAttr および unsignedAttrs フィールドなどの Implicit Context-specific タグを Explicit 形式に戻すような正規化を行わず、そのままハッシュ対象に加える。

このことは現行の標準仕様は古いハッシュ計算方法の使用よりも相互運用性を欠いた実装が数多く現れる可能性があることを意味している。ETSI や IETF などの標準化団体と連携しながら今回の実験より得られた知見を反映させる必要があると考えられる。

3.2.1.2 ハッシュ計算法に関する考察

実験後に、新しいアーカイブタイムスタンプのハッシュ計算法に関する考え方を整理し、実験で設定された正規化方法を含むいくつかの実現方法案を検討した。

- アーカイブタイムスタンプのハッシュ計算方法の考え方

(1) 署名データのライフサイクルモデル

ETSI TS 101 733 v1.5.1[1]で記載されたアーカイブタイムスタンプハッシュ対象データを素直に解釈すると、想定される署名データのライフサイクルモデルは、以下の通りとなる。

・ アーカイブタイムスタンプ適用以降は、署名データは、アーカイブ化
・ アーカイブ化された署名データは、基本的に、ビット列は変更されない
・ 例外は、アーカイブ以降に、適切なタイミングで繰り返し行われるアーカイブタイムスタンプ適用

(2) 正規化が必須なデータ要素

前述した署名データのライフサイクルを踏まえると、正規化ルールの適用対象は、少なくともデータの再構築が要求されるデータ要素であると思われる。具体的には、以下の通りとなる。

・ 分離署名を扱うことを踏まえ、encapContentInfo エレメント
・ アーカイブタイムスタンプ適用/検証時に、再構築される unsignedAttrs エレメント

- アーカイブタイムスタンプのハッシュ計算方法における正規化方法案

アーカイブタイムスタンプのハッシュ方法の正規化で重要なのは、ハッシュ対象データの連結要素となるデータエレメントの正規化方法である。実験後、検討された実現方法案を表 1.3-1 に示す。「As Is」と示したものは、ASN.1 データとして処理するのではなく、バイナリデータ列そのままとして扱うことを示す。なお、実験では、案1が該当する。

表 1.3-1 データエレメント正規化方法案

データエレメント	案1	案2	案3	案4	案5	案6
encapContentInfo	DER化	DER化	DER化	DER化	DER化 IMPLICIT [0] などをそのまま。 ※5	DER化 IMPLICIT [0] などを EXPLICIT に変更する。 ※5
certificates	As Is	As Is	外側のみ DER化	外側のみ DER化		
crls	As Is	As Is	外側のみ DER化	外側のみ DER化		
version	As Is	As Is	外側のみ DER化	外側のみ DER化		
sid	As Is	As Is	外側のみ DER化	外側のみ DER化		
digestAlgorithm	As Is	As Is	外側のみ DER化	外側のみ DER化		
signedAttrs	As Is ※1	As Is ※1	外側のみ DER化 ※1	外側のみ DER化 ※1		
signatureAlgorithm	As Is	As Is	外側のみ DER化	外側のみ DER化		
signature	As Is	As Is	外側のみ DER化	外側のみ DER化		
unsignedAttrs	IMPLICIT [0] のままで DER化。 ※2 ※3	EXPLICIT SET OF tag として DER化。 ※3 ※4	IMPLICIT [0] のままで DER化。 ※2 ※3	EXPLICIT SET OF tag として DER化。 ※3 ※4		

※1：RFC 3852[2]準拠の CMS データであれば、DER であることが期待できる

※2：SET OF ソートの DER として扱う

※3：Attribute の attrValues における AttributeValue に関しては、As Is

※4：RFC 3852 の「5.4 Message Digest Calculation Process」における signedAttrs の DER 化

を参考にした

※5：各エレメントを最小構成要素まで分解し、コンテキストを踏まえつつ DER 化

案 1 におけるアーカイブタイムスタンプ対象のデータ作成方法を図 1.3-2 及び 1.3-3 に示す。それぞれ、アーカイブタイムスタンプ適用時、アーカイブタイムスタンプ検証時におけるハッシュ対象データ作成手順となる。

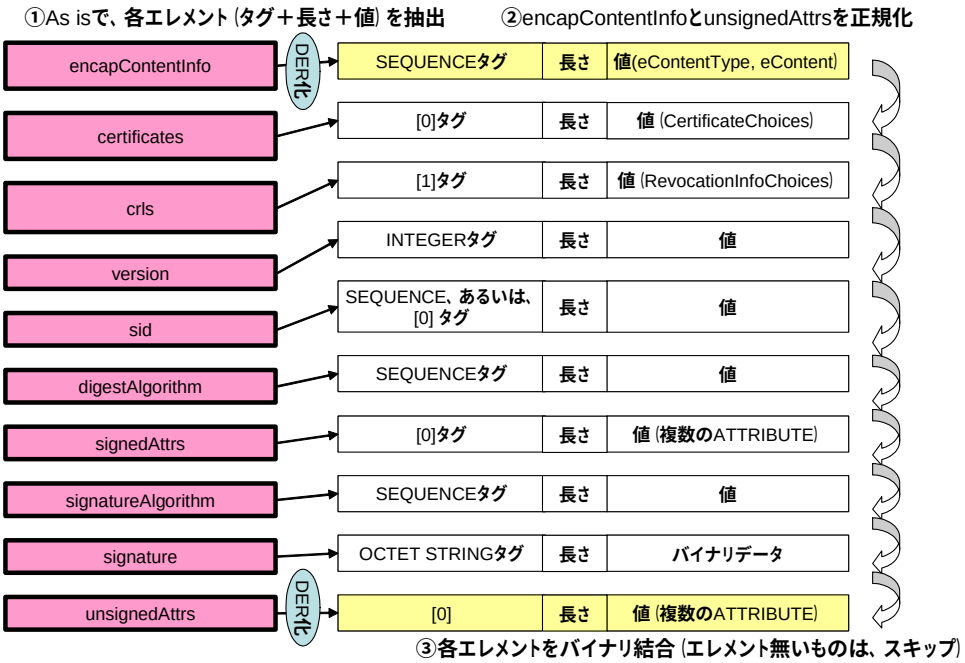


図 1.3-2 アーカイブタイムスタンプ適用時のハッシュ対象データの作成手順

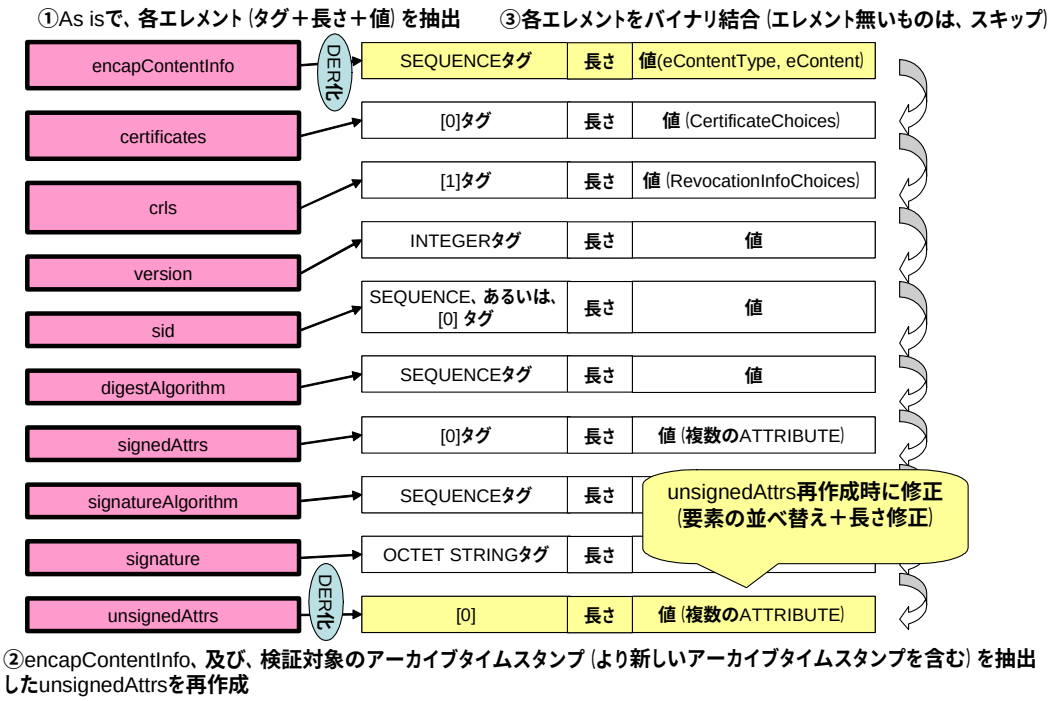


図 1.3-3 アーカイブタイムスタンプ検証時のハッシュ対象データの作成手順

現状では、これらの方法案に関する詳細な比較検討は行っていない状況である。ETSI へのフィードバックも視野に見据え、正規化方法の詳細検討は今後の課題である。なお、検討の観点としては、例えば、以下のものが考えられる。

- 市場に流通する ASN.1 処理ライブラリ系で比較的容易に対応可能か？
- 処理性能の観点から問題がないか？

3.3 XAdES に関する課題

3.3.1 タイムスタンプの証拠情報の課題について

3.3.1.1 概要

本節では、XAdES 長期署名フォーマットの相互運用性実験で課題となった、タイムスタンプの証拠情報の格納方法の課題について考察する。はじめに、タイムスタンプの証拠情報について概説し、ETSI XAdES および ECOM プロファイルでの扱いを述べる。それを元に今回の実験の課題について説明し、解決方法について考察する。

長期署名フォーマットでは、電子署名文書の真正性を長期に保証するために署名者の証明書の証拠情報を署名文書内に保存し、署名対象文書や署名値および証拠情報を含めてタイムスタンプを付与する。付与するタイムスタンプの形式には幾つかの方式が実用化または提案されており、XAdES 形式の長期署名フォーマットでは複数の方式をサポートできるが、今回の実験では現在の普及度合いなどを考慮し RFC 3161 形式のタイムスタンプを利用している。

RFC 3161 形式のタイムスタンプを利用した場合、タイムスタンプ自体も PKI を基盤とする電子

署名を利用する。したがって、タイムスタンプの有効期限が切れる前にタイムスタンプ再付与し続けることで、電子署名文書の真正性を長期間確保する必要がある。このとき、長期経過後も各タイムスタンプの有効性を確認できるためには、タイムスタンプ自身の証拠情報も何らかの方法で安全に保管する必要がある。

3.3.1.2　実験プロファイル

ETSI TS 101 903（XAdES）では、バージョンによらずタイムスタンプの証拠情報の格納方法や格納場所について言及していない。したがって本実験では、CMS 形式の長期署名フォーマット、上述のタイムスタンプの証拠情報の保存の必要性および相互運用性を考慮し、以下のようなプロファイルとして実験を行った。

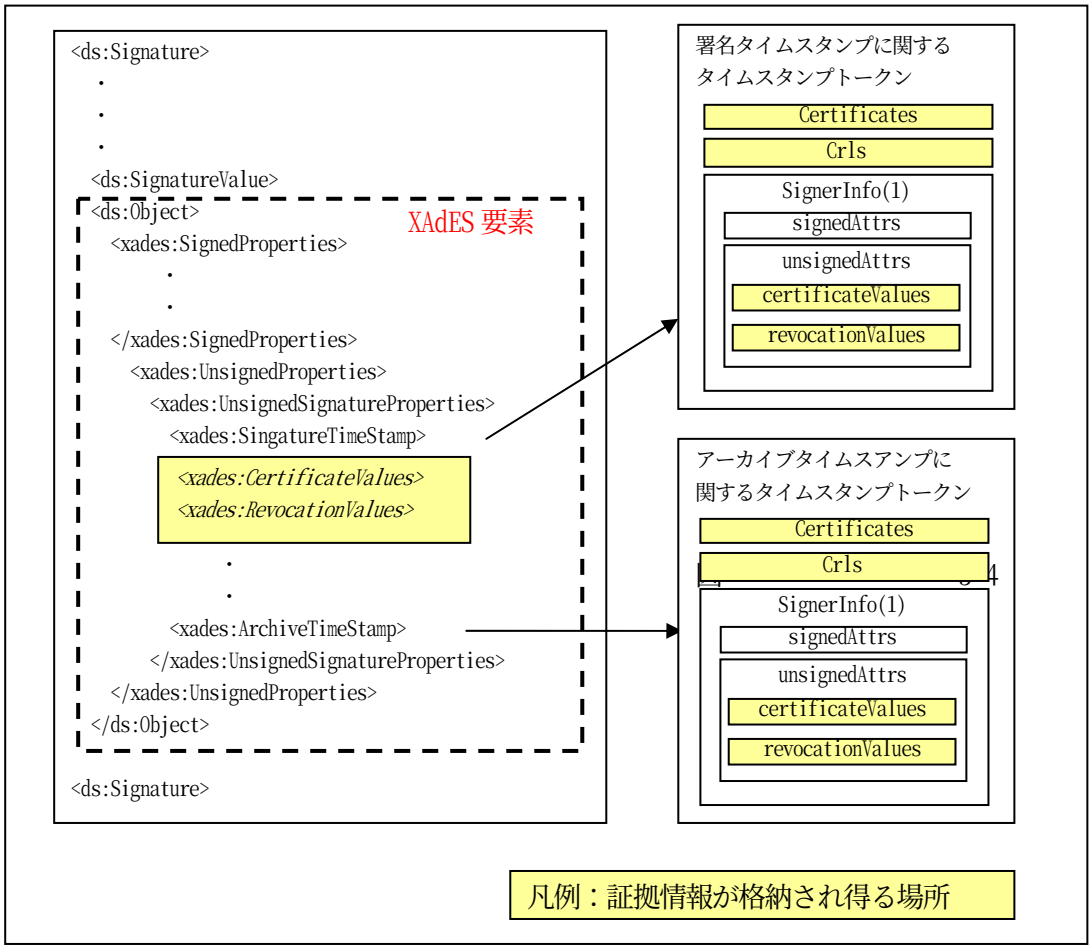
署名タイムスタンプの証拠情報の格納場所

タイムスタンプトークン自体の検証情報（認証パス及び失効情報）は、次のいずれかに格納する。

1) タイムスタンプトークン内に含める
2) 署名者証明書の検証情報と同じ場所（Complete CertificateRefs, Complete RevocationRefs, CertificateValues, RevocationValues）

アーカイブタイムスタンプの証拠情報の格納方法

アーカイブタイムスタンプ自身の検証情報（認証パス及び失効情報）はアーカイブタイムスタンプに含めるか、もしくは別途安全に保管しておく必要がある。



3.3.2　実験の結果と課題

上述のようなプロファイルに基づき実験を行った結果、証拠情報の格納方法は製品やシステム毎に異なっており相互運用性に課題があることがわかった。具体的には、証拠情報を長期署名フォーマット内に含める製品・システムと、証拠情報を長期署名フォーマットに含めず個別に管理する製品・システムがあった。第一世代のアーカイブタイムスタンプを付与した長期署名フォーマットデータについては、アーカイブタイムスタンプの証拠情報をタイムスタンプトークンにその時点では格納していないので相互運用性に問題は発生しない。しかし、第二世代目のアーカイブタイムスタンプを付与した段階で、第一世代目のアーカイブタイムスタンプの格納方法に違い発生した。製品・システムを修正するなどの対応で解決したものや、実験後も相互互換性の観点で制限があるものが残った。

3.3.3　考察と解決策

タイムスタンプの証拠情報の格納方法について、プロファイルで厳密に規定し切れなかったことが今回のような結果となった要因の一つと考えられる。

タイムスタンプの証拠情報の格納方法は、今回のプロファイルでは大きく分けると長期署名フォーマットに格納する方法とタイムスタンプトークンの証拠情報を別途安全に保管する方法の二

通りある。長期署名フォーマットに格納する方法は、タイムスタンプトークンへの格納方法を規定できれば、高いポータビリティと相互運用性を実現できる。一方、後者の方法は相互運用性の観点から考えると、別途保管したタイムスタンプの証拠情報を用いて検証者がタイムスタンプを検証できる必要があるため、検証者の検証プログラムがタイスタンプの証拠情報を読み込む機能とそれら使った長期署名フォーマットの検証を実行できる必要がある。

ただし、今回の実験で利用したプロファイルを実システムに適用した場合、以下のようなことが言えるので、今回の実験に適用したプロファイルは変更しないものとする。

- 署名タイムスタンプの証拠情報が必要となる第一世代 ES-A はプロファイル通りで相互運用性が確認できている。
- 課題のあった第二世代 ES-A が実際に作られるのは通常運用では数年先で、対策を打つ時間的余裕はある。
- 保存期間によって、第二世代 ES-A が不要な場合もある。

一方、今後については、再度実験などを通してプロファイルをより厳密に定義することが期待される。また、今後は XAdES の標準を策定している ETSI などに対してタイムスタンプの証拠情報についても仕様で厳密に言及するよう提案していくことが考えられる。

3.4 参考文献

[1] ETSI TS 101 733 V1.5.1 (2003-12) Electronic Signatures and Infrastructures (ESI); Electronic Signature Formats, Sep 2002, ETSI

[2] RFC 3852 Cryptographic Message Syntax (CMS), Jul 2004, R.Housley, http://www.ietf.org/rfc/rfc3852.txt

第2部　電子文書長期保存に関わる課題

1．PDF/A 文書に対する長期署名の標準化

PDF1.4 ベースの PDF/A は 2005 年 10 月に ISO19005-1 として国際標準化された。続いて PDF1.6 ベースの PDF の国際標準化作業が始まった。PDF/A の標準化方針は、ベースとなる PDF 仕様（PDF1.4 や PDF1.6 など）の完全なサブセットとし、新たに機能を加えることはしない。PDF1.4 ベースの PDF/A は、署名処理に制限はなく、長期署名が可能である。しかしながら、PDF1.6 では規定が追加され、PKCS#1 と PKCS#7 に限定されるため、このまま標準化されると長期署名を付与できなくなる。

このため、標準化作業中の PDF1.6 ベースの PDF/A も長期署名を付与できるよう，TC171 国内委員会を通して働きかけてきた。

以下に、その具体的な要求内容を紹介する。

1.1　前提事項

e-文書関連の法律への対応の観点から、最低限の要求として以下の 2 つの前提を置く。また、アプリケーション構造として図 2.1-1 の構造を想定する。

(a) 署名形式（注 1）は少なくとも Envelope 型をサポートする

(b) ダイジェスト方式（注 2）は少なくとも ByteRange ダイジェストをサポートする

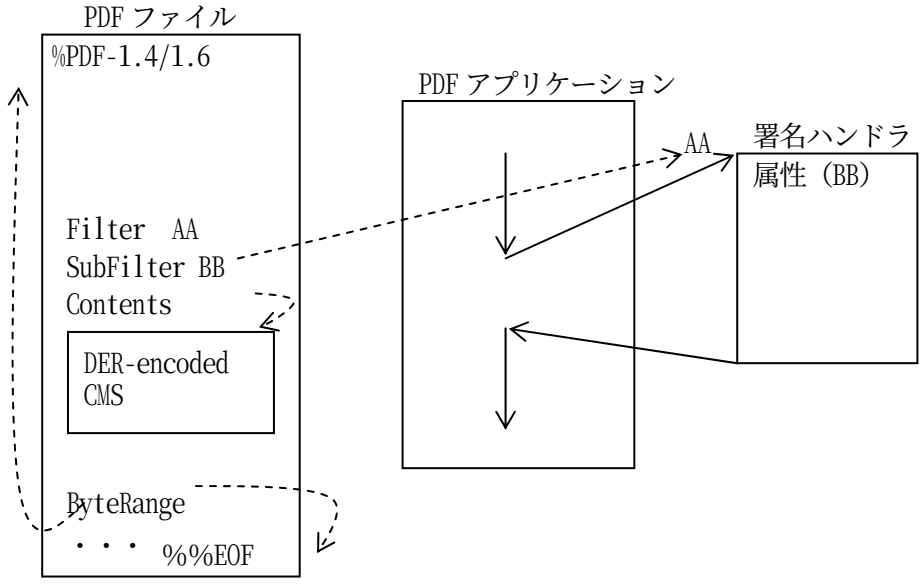


図 2.1-1 PDF ファイル、PDF アプリケーション、署名ハンドラの関係

図において、PDF アプリケーションに対して署名付与が指示されると、特定の署名ハンドラが呼び出され、署名が生成され、PDF 文書に署名の符号化名（SubFilter 名）と署名値（Contents）が埋め込まれる。署名検証は、検証が指示されると、PDF アプリケーションは、SubFilter 名を検索し、対応する署名ハンドラを呼び出す。

注1：署名形式の解説

署名形式には、Detached 型（分離形式）、Enveloping 型（内包形式）、Enveloped 型（包含形式）の 3 つがある。Enveloped 型は、署名付き文書を 1 つの文書ファイルとして管理できる。Detached 型は署名を別ファイルとして管理する必要があるが、文書ファイルには署名の影響を及ぼさない。Enveloping 型は、文書をバイナリオブジェクトと看做すので、文書の属性を別に管理しなければならない。

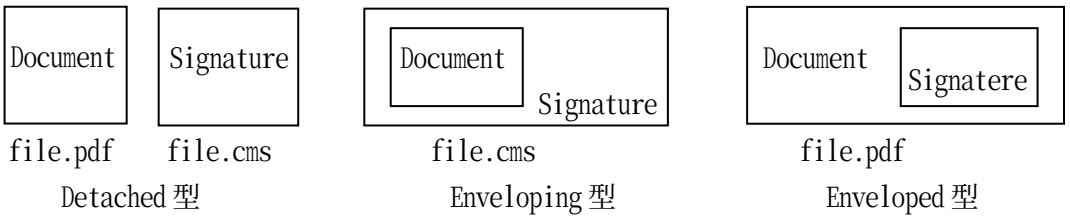


図 2.1-2 署名形式

注2：PDF のダイジェスト方式の解説

PDF の enveloped 型のダイジェスト方式には、ByteRange digest と Object digest がある。前者は、ファイル上のバイト列の並びに対してダイジェスト処理を行う。後者は、文書の構造のトラバーサルに従ってダイジェスト処理を行う。ダイジェストの詳細な方法は（PDF や XML などの）文書の構造に依存するが、文書を単にバイナリオブジェクトとして扱う場合は、バイト列のダイジェストとなる。

1.2 論点と考察

“PDF/A 文書に対する長期署名”の論点として次が挙げられる。

(a) 長期署名とのインタフェース確保
(b) SubFilter 名の一意性確保
(c) アーカイブタイムスタンプを重ねることによる“Contents”の増大への対処

以下ではそれぞれの論点について考察を述べる。

1.2.1 長期署名とのインタフェース確保

PDF1.4 ベースの PDF/A 仕様は、Signature dictionary 内の署名ハンドラの名前（Filter 名）、署名の符号化方法を示す名前（SubFilter 名）や署名値（Contents）に何も制約がないことから、CMS などの標準に従った長期署名が可能である。

しかしながら、現在標準化が予定されている、PDF1.6 ベースの PDF/A 仕様が、PDF1.6 仕様をそのまま継承すると、以下の理由で、CMS などの標準に従った長期署名ができない。また、PDF1.4 ベースの PDF/A 仕様との後方互換性も損なわれる。

(1) PDF1.6 では、SubFilter 名および Contents は、PKCS#1 と PKCS#7 のための限定された名前や符号化方式のみ定義されており（注 4）、CMS や、今後出現するであろう新たな方式への対

応は、考慮されていない。本来は、ここに、例えば“CMS-長期署名”が定義されるべきである。SubFilter 名 は optional なので、使わなければ PDF1.6 仕様には抵触しないが、そうすると“CMS-長期署名”を適切に判断する術がない。

(2) PDF1.6 でも、オプションとして、RFC3161 に基づいた署名タイムスタンプは可能である。失効情報の格納に関しても ASN.1 定義されている（注 5)。しかしながら、アーカイブタイムスタンプなどの長期署名との整合性がない。

注 4：PDF1.6、表 8.98 における定義

SubFilter name

"Defined values for public-key cryptographic signatures are adbe.x509.rsa_sha1, adbe.pkcs7.detached, and adbe.pkcs7.sha1 (see Section 8.7.2, "Signature Interoperability")."

Contents

"For public-key signatures, Contents is commonly either a DER-encoded PKCS#1 binary data object or a DER-encoded PKCS#7 binary data object.

なお、一般論と解釈すると CMS も仕様の範囲内と解釈できないこともない。

注 5：PDF1.6 における定義

```
adbe-revocationInfoArchival OBJECT IDENTIFIER ::=
        { adbe(1.2.840.113583) acrobat(1) security(1) 8 }
RevocationInfoArchival ::= SEQUENCE {
        crl [0] EXPLICIT SEQUENCE of CRLs, OPTIONAL
        ocsp [1] EXPLICIT SEQUENCE of OCSP Responses, OPTIONAL
        otherRevInfo [2] EXPLICIT SEQUENCE of OtherRevInfo, OPTIONAL}
OtherRevInfo ::= SEQUENCE {
        Type OBJECT IDENTIFIER
        Value OCTET STRING}
```

PDF/A と長期署名の関係については、“PDF/A の仕様と署名の仕様は切り離されるべき”であるというのが基本的なスタンスである。。署名技術は今後も PDF/A とは独立に進化し続けて行くと考えられ、相互に干渉することは避けなければならない。これは、何も定義しないという意味ではない。PDF/A に長期署名を組み込むインタフェースを用意し、長期署名を組み込めるようにしておくことが前提にある。

また、署名とタイムスタンプに必要なダイジェストの計算方法は、文書構造に依存することから、PDF/A の仕様として定義する必要がある（現在の仕様を残す)。

具体的には、PDF1.6 に記述されているインタフェースの阻害要因を削除する。例えば、表 8.98

における Subfilter 名や Contents に関する以下のような排他的な記述を削除する。

(a) "Defined values for public-key cryptographic signatures are adbe.x509.rsa_sha1, adbe.pkcs7.detached, and adbe.pkcs7.sha1 (see Section 8.7.2, "Signature Interoperability")."
(b) "For public-key signatures, Contents is commonly either a DER-encoded PKCS#1 binary data object or a DER-encoded PKCS#7 binary data object."

この措置により、PDF1.6 ベースの PDF/A は、PDF1.4 ベースの PDF/A と互換性を保った長期署名が可能になる（表 2.1-1 参照)。

表 2.1-1 署名ディクショナリ（Signature dictionary）のエントリ

No	Key	値	PDF1.4	PDF1.6
1	Type	Sig	Op	Op
2	Filter	証明ハンドラ名	M	M
3	SubFilter	署名アルゴリズム等を示す名前	Op	Op
4	Contents	署名値	M	M
5	Cert	証明書（PKCS#1 の場合のみ）	－	Op
6	ByteRange	署名対象（バイトオフセット＋長さ）	M	M
7	Reference	ダイジェスト算出方法など	－	Op
8	Changes	前署名と現署名の変更箇所	－	Op
9	Name	署名者名	Op	Op
10	M	署名日時	Op	Op
11	Location	署名場所	Op	Op
12	Reason	署名理由	Op	Op
13	ContactInfo	連絡先（電話番号等）	－	Op
14	R	署名時の署名ハンドラのバージョン	－	Op
15	V	Signature dictionary の版番号（default=0）	－	Op
16	Prop_Build	署名ハンドラが使う環境変数	－	Op
17	Prop_AuthTime	署名者が最後に認証された時からの経過時間	－	Op
18	Prop_AuthType	署名者認証方法（指紋等）	－	Op

M：必須、OP：オプション、－：対象外

1.2.2 SubFilter 名の一意性確保

PDF および PDF/A では、“Contents”の構造と署名ハンドラの振る舞いには、ユニークな名前（SubFilter 名）が付与される。PDF/A としての SubFilter 名をユニークに保つ方法には、ISO/IEC で定義されたオブジェクト識別子（OID）を利用する方法と、信頼できる第三者機関が登録機関に

なる方法とがある。Filter 名も同様である。

この扱いに関しては、OID の採用が最も現実的であると考えられるが、決まるまでは自己宣言した名前を使わざるを得ない。衝突を回避するための方法として、次のような SubFilter 名の記法（naming convention）が考えられる。

案 1　（標準仕様）（版番号）（構文識別）（署名形式）
例　ASN1 . ecom . 01 . ES-T . detached

案 2　要素間を空白またはドットで区切ったオブジクト識別子
例　iso.memberbody.jisc.x.y.z

1.2.3　アーカイブタイムスタンプを重ねることによる“Contents”の増大への対処

署名やタイムスタンプが 1 回だけの処理であれば、署名値（Contents）を含む文書の長さは固定である。しかし、その後に長期保存のためにアーカイブタイムスタンプを追加したり、タイムスタンプの期限が切れる前にアーカイブタイムスタンプを重ねたりすると、署名値（Contents）を含む文書の長さが可変になる。ダイジェストの範囲を文書全体に設定すると、ByteRange の値が変わってしまい（ByteRange もダイジェスト対象なので）、正しく検証できない。

“Contents”の増大への対処方法として、次の 4 案が考えられる（これがすべてではない）。案 3 と案 4 は PDF 仕様に追加定義することになり、PDF の設計思想に合うかどうか設計者への確認も含め、今後更なる調整を要する。現時点では、案 2 が現実解である。

（案 1）アーカイブタイムスタンプの追加を PDF/A 文書の更新として捉え、追加のタイムスタンプを更新されたボティとして加える。PDF 仕様に従って、新たな相互参照セクション、更新されたトレーラが追加される。

（案 2）アーカイブタイムスタンプの追加は長期署名に閉じた問題として捉え、PDF/A のスコープ外とする。タイムスタンプの追加による、実際の長さと ByteRange の不整合を避けるために、予め余分に Contents 領域を確保し、使わない部分はパディングする。

（案 3）案 2 同様 PDF/A のスコープ外として扱う。可変長 Contents の影響を回避するために正規化してダイジェストを算出する。正規化の対象は相互参照表と ByteRange である（これ以外にも影響するものがあるかもしれない）。

（案 4）案 3 を簡略化するために、署名値（Contents）をトレーラ直前に置く。ByteRange は、ダイジェストの対象から除外するか、元の ByteRange を Contents の適当な位置に退避する。

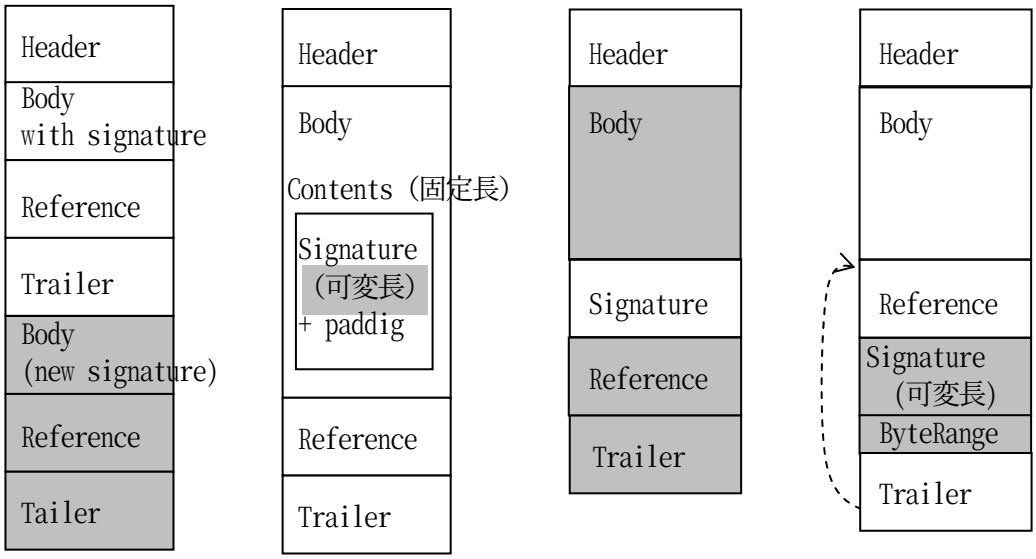


図 2.1-3“Contents”の増大への対処案

2．長期保存に使用時の磁気テープの課題

2.1　磁気テープの特徴

磁気テープには以下の特長があり、これまで、データバックアップとしての実績があります。

①大容量、②低ビットコスト、③高速データ転送

一方、磁気テープはその方式上、接触記録であることから、以下の 4 つの課題があります。

① ヘッド、テープ（媒体）が接触記録である。【接触記録】
② テープ（媒体）とテープ装置の走行系が接触している。【接触走行】
③ テープ（媒体）同士が重なりあって巻かれ、媒体同士の接触、巻き圧存在。【巻き保管】
④ テープ（媒体）をテープ装置の走行系に位置づける動作がある。【ロード/アンロード動作】

磁気テープは、大容量、高速データ転送速度、低ビットコスト、シーケンシャルアクセスの面から大容量、低頻度アクセスの電子データの長期保存への利用が期待されるものの電子署名文書のような価値の高い文書のオリジナルデータの保存媒体としての利用には環境条件や運用要件を十分確認しておく必要がある。

2.2　磁気テープの耐久性

磁気テープの長期保存性についての論議の前に、その前提条件となる磁気テープの耐久性の概要について説明する。一般に公開されている項目としては以下がある。

① パス数　② ロード／アンロード回数

「パス数」は 1.1 節①【接触記録】②【接触走行】に起因する制限事項であり、ロード／アンロード回数は④【ロード/アンロード動作】に起因する制限事項である。

これらは同じ規格のテープであっても、その製造方法により変わるものである。

現状、テープはバックアップ用途が主であり、「パス数」仕様を公開しているものは多いが、「ロード／アンロード回数」を仕様公開しているものは少ない。また、重要な事項として、「パス数」が、データの読出し回数を示すものではないことに留意しなければいけない。この原因はテープの内部には複数のトラックが設定されており、特定のデータを読出すには、このデータに行き着くまで、テープを何度か端から端まで走行させる必要があることによる。このような事情により、最近では、全部のトラックを走行させる「フルパス回数」を仕様として公開するものもある。

HP 社のホームページ【1】によれば、通常の「パス数」と「フルパス回数」の関係を下表のように算出している。尚、各テクノジーは進歩しているのでこの値は参考として捉えて頂きたい。

表 2.2-1 パス回数とフルパス回数の関係

テクノロジー	耐久性	
	パス回数	フルパス回数
Ultrium	1,000,000 回	262 回
SDLT	1,000,000 回	100 回
DLTTape IV	1,000,000 回	100 回
AIT	20,000 回	140 回
DDS	2,000 回	100 回

2.3 使用環境、保存環境上の注意事項

テープの種別毎に各メーカは、使用時、保管時の温度／湿度／最大湿球温度は規定しているが、この温湿度条件以外に使用上、保管上の注意事項として、参考文献【2】、【3】、【4】あがってものについて以下に紹介する。

① 長期間保管時は垂直に立てる。
② 直射日光のあたる場所に放置しない。
③ 塵埃の多い場所に放置しない。
④ テープ装置に入れぱなしにしない。
⑤ 保管時はケースに格納する。
⑥ カートリッジを 1m 以上の高さから落下させてしまった場合、速やかにデータを他のテープにコピーし、落下させたテープは使用しない。
⑦ 1年以上使用しなかったテープは可能であれば、ライトする前にリテンション（巻き戻し）をすることを推奨する。長期間テープを巻いたままにの状態にしておくとストレスが発生する可能性がある。
⑧ 電源ケーブル、モータ、電源など近づけない。
放射磁界により記録データが破壊される恐れがある。
⑨ 外部から持ち込まれたテープを使用するときは急激な環境の変化を避ける。
⑩ 粘着性のゴミ（テープから発生）
テープのバインダに含まれる粘着性の成分が湧出し、これが、磁気ヘッドおよびクリーナによって粘着性ゴミとしてかき集められる。このゴミはテープのコーティング面に転移したり、磁気ヘッドに付着する。この粘着性のゴミはテープの使用回数に関係なくバインダの中から湧出してくりため、磁気ヘッドクリーニングしても解決にならず、テープを廃棄する以外に解決策はなく、温度、湿度が上がると発生し易くなる。
⑪ 粉ゴミ
テープとテープガイド及び磁気ヘッドがしゅう動するため、ゴミの発生は避けらない。
テープの走行に伴いテープエッジおよびコーディング面が削れる。
ゴミの塊をリールに巻き込むとテープに凹凸が生じ、ヘッドタッチが悪くなり、リード／ライトエラーも発生する。テープガイドや磁気ヘッドの清掃を適時行うことが必要である。

2.4　磁気テープの長期保存性

温度、湿度だけでみれば、表 2.2-2 のように DLT,AIT/SAIT、LTO は約 30 年も期待寿命をもつ、DDS については約 10 年の期待寿命を有している。しかしながら、テープの場合は、1.2 の耐久性、1.3 の使用上の注意事項があるので、使用にあたっては十分な検討が必要である。

また、長期保存時にはさらに狭い温度、湿度環境を要請されていることもあり注意を必要とします。

表 2.2-2 磁気テープの期寿命

No	大分類	種別	保存　環境		再生時　使用環境		期待
			温度	湿度	温度	湿度	寿命
1	磁気	DDS	5～45℃	20～80%	5～32℃	20～60%	約 10 年
2	テープ	DLT	18～26℃	40～60%	10～40℃	20～80%	約 30 年
3		AIT/SAIT	17～23℃	20～50%	5～45℃	20～80%	約 30 年
4		LTO Ultrium1/2	16～32℃	20～80%	5～55℃	10～80%	約 30 年

2.5　磁気テープの課題の解決に向けて

磁気テープの技術も 2.2 耐久性、2.3 使用上の課題はあるものの技術革新が進んでおり、その条件緩和がなされてきている。しかしながら、これらの情報は一般企業が正しく収集、理解するには、はまだまだ困難な状況にある。このような現状を鑑み、磁気テープ、磁気テープ装置などの専門家である JEITA 磁気記録媒体標準化委員会も、長期保存に向けてのテープ装置/テープ選定、保管環境、運用のガイドラインづくりの検討を進める予定であり、成果を待ちたい。磁気テープも光ディスクなどの他の可換型の記録装置と同じく、テープ（媒体）のみが長期に保管できてもそれを読出すドライブ装置の提供または保守が担保されていない状況にあることは課題として残っている。

参考資料 1
HP StorageWorks Ultrium 960 テープ・ドライブテクニカル・ホワイトペーパ
http://www.compaq.co.jp/products/storage/whitepaper/
参考文献 2　富士通コワーコホームページ
LTO　Ultrium カートリッジテープ説明書（第 2 版)、
CT-210/210E カートリッジテープ説明書（第 2 版）
http://jp.fujitsu.com/group/coworco/support/
参考文献 3　日立マクセルホームページ　コンピュータテープの適切な取り扱い
http://www.maxell.co.jp/products/industrial/computer_tape/lto3.html
参考文献 4　イメーションホームページ　よくあるご質問
http://www.imation.co.jp/products/pc_tape/dlt/index.html

付　　録

Ⅰ．CMS 長期署名プロファイル（Version 1.0）

2006 年 3 月

次世代電子商取引推進協議会（ECOM）

目　次

はじめに

長期署名フォーマットの仕様が、次のドキュメントで定義されている。

・ETSI TS 101 733 V1.6.3 (2005-09), "Electronic Signature Formats"

・IETF RFC3126, "Electronic Signature Formats for long term electronic signatures"

上記標準は多くの選択的な定義を含んでおり、電子署名文書の長期有効性保証を可能とするためには、そのサブセットを実装すれば十分である。電子商取引実証推進協議会（ECOM）報告書「電子署名文書長期保存に関するガイドライン（H14 年 3 月）」では、電子署名文書の長期有効性保証のために必要なサブセットを定義した推奨プロファイル案の概要を紹介している。

今回のプロファイル Ver 1.0 では、前回定義したプロファイル案を更に詳細化かつ具体化し、電子署名文書の長期保証を可能とするシステムを実装する際に必要であり、できるだけ冗長性のない定義を示す。1 章ではプロファイル Ver 1.0 の詳細を説明し、2 章にプロファイル表を示す。

1．CMS 長期署名プロファイル詳細

本プロファイル Ver 1.0 では、最新の“CMS Advanced Electronic Signatures（CAdES）”の仕様をベースとし、効果的（真正性の長期保証が可能）かつ効率的（なるべく必要とするデータやタイムスタンプを限定）なプロファイルを示す。

1.1　署名フォーマット

基本となる電子署名文書の形式として、図 付 1.1-1 に示す BES（Basic Electronic Signature）と図 付 1.1-2 に示す EPES（Explicit Policy Electronic Signature）の 2 通りを利用可能とする。

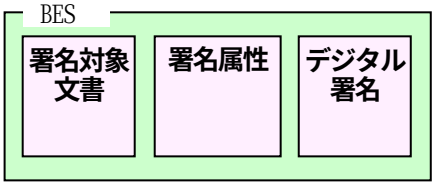


図 付 1.1-1 BES

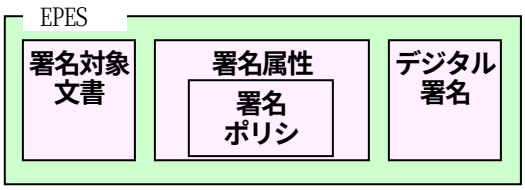


図 付 1.1-2 EPES

両者の違いは、署名ポリシの有無である。署名ポリシとは、署名の生成と検証に関して、署名

者と検証者がデジタル署名を有効とみなすための一連の規則を定めるものであり、"ETSI TR 102 272 V1.1.1 (2003.12)：Electronic Signatures and Infrastructures (ESI) ;ASN.1 format for signature policies"あるいは"RFC3125 : Electronic Signature"に規定されている。

電子署名文書の形式は次の仕様に準拠する。

- Cryptographic Message Syntax (CMS：RFC3852)
- Enhanced Security Services (ESS：RFC2634)

(1) General syntax

電子署名文書形式の General syntax は、CMS (RFC3852) に定義されているとおりである。

```
ContentInfo:
    ContentInfo ::= SEQUENCE {
                contentType  ContentType,
                content      [0] EXPLICIT ANY DEFINED BY contentType }

    ContentType ::= OBJECT IDENTIFIER
```

(2) Data content type

Data content type は、CMS (RFC3852) に定義されているとおりである。

```
    id-data OBJECT IDENTIFIER ::= { iso(1) member-body(2)
                                    us(840) rsadsi(113549) pkcs(1) pkcs7(7) 1 }
```

(3) Signed-data content type

Signed-data content type は、CMS (RFC3852) に定義されているとおりである。

```
    id-signedData OBJECT IDENTIFIER ::= { iso(1) member-body(2)
                                          us(840) rsadsi(113549) pkcs(1) pkcs7(7) 2 }
```

(4) SingedData type

SignedData の構文は、CMS (RFC3852) に定義されているとおりである。

```
    SignedData ::= SEQUENCE {
              Version          CMSVersion,
              digestAlgorithms DigestAlgorithmIdentifiers,
              encapContentInfo EncapsulatedContentInfo,
              certificates     [0] IMPLICIT CertificateSet OPTIONAL,
              crls             [1] IMPLICIT CertificateRevocationLists OPTIONAL,
              signerInfos      SignerInfos }

    DigestAlgorithmIdentifiers ::= SET OF DigestAlgorithmIdentifier

    SignerInfos ::= SET OF SignerInfo
```

- SignedData のバージョンは**3 である必要はない。**
  - 次のいずれかを満たす場合はバージョン 3 である。
    - certificates 属性がありバージョン 1 の属性証明書があり、バージョン 2 の属性証明書がない

- ✧ encapsulated content type が id-data 以外
- ✧ いずれかの SignerInfo がバージョン 3

➢ ただし、certificates 属性があって、かつ異なるタイプの証明書が存在するか、crls 属性があって、かつ異なるタイプの CRL が存在する場合を除く。

(5) EncapusulatedContentInfo type

EncapusulatedContentInfo type は、CMS（RFC3852）に定義されているとおりである。

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

ContentType ::= OBJECT IDENTIFIER
```

- 長期保存のためには、eContent を SignedData に含めておくか、別途保存・管理しておくことを推奨する。

(6) SingerInfo type

SignerInfo type は、CMS（RFC 3852）に定義されているとおりである。

```
SignerInfo ::= SEQUENCE {
                version            CMSVersion,
                sid                SignerIdentifier,
                digestAlgorithm    DigestAlgorithmIdentifier,
                signedAttrs        [0] IMPLICIT SignedAttributes OPTIONAL,
                signatureAlgorithm SignatureAlgorithmIdentifier,
                signature          SignatureValue,
                unsignedAttrs      [1] IMPLICIT UnsignedAttributes OPTIONAL }

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

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

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

Attribute ::= SEQUENCE {
                attrType        OBJECT IDENTIFIER,
                attrValues      SET OF AttributeValue }

AttributeValue ::= ANY

SignatureValue ::= OCTET STRING
```

- 署名者が延べ 1 名につき、ひとつの SignerInfo が対応し、署名が並列に複数添付される場合には、複数の SingerInfo が作られる。
- SignerInfo の**<u>バージョンは問わない</u>**
  - ➢ SignerIdentifier が issuerAndSerialNumber ならば 1
  - ➢ SignerIdentifier が subjectKeyIdentifier ならば 3

- 長期署名では、少なくとも SignedAttributes に次の値を格納していなければならない。
  - ContentType
  - MessageDigest
  - SigningCertificate

① Message digest の算出プロセス

CMS（RFC3852）に定義されているとおりである。

② Message signature の生成プロセス

CMS（RFC3852）に定義されているとおりである。

③ Message signature の検証プロセス

CMS（RFC3852）に定義されているものを本書で拡張したものである。

署名検証プロセスでは、必ず ESS Signing Certificate 属性、あるいは Other Signing Certificate 属性を利用して正しいことが確認された署名者公開鍵を用いる。

(7) 必須の CMS 属性

次の属性は署名データの signed attribute 中に存在しなければならない。

① Content type

構文は、CMS（RFC3852）に定義されているとおりである。

```
id-contentType OBJECT IDENTIFIER ::= { iso(1) member-body(2) us(840)
                                        rsadsi(113549) pkcs(1) pkcs9(9) 3 }

ContentType ::= OBJECT IDENTIFIER
```

- ContentType 属性は、signed attribute 内にただ１つだけ存在しなければならない。

② Message digest

構文は、CMS（RFC3852）に定義されているとおりである。

```
Id-messageDigest OBJECT IDENTIFIER ::= { iso(1) member-body(2) us(840)
                                          rsadsi(113549) pkcs(1) pkcs9(9) 4 }

MessageDigest ::= OCTET STRING
```

- MessageDigest 属性の値は、SignerInfo の DigestAlgorithm を利用して encapContentInfo eContent OCTET STRING の ASN.1 TLV の V から求めたものである。
  MessageDigest 属性は、signed attribute 内にただ１つだけ存在しなければならない。

③ Signing certificate 属性

signed-data には、ESS signing certificate あるいは Other signing certificate のどちらか一方のみの signing certificate 属性を含めなければならない。この属性値は、

“simple substitution 攻撃”と“re-issue 攻撃”を防ぐために用いる。

A) ESS signing certificate 属性の定義

ESS signing certificate 属性は ESS（RFC2634）に基づく。ESS signing certificate は signed attribute でなければならない。

signing certificate 属性または次項の Other signing certificate 属性は必ず存在しなければならず、属性値が空であってはならない。署名検証のための証明書はこの属性値から得なければならない。署名検証ポリシ（Signature Validation Policy）でここに他の証明書が存在することを規定していれば、signing certificate 属性に信頼点までのすべての証明書を含むこともある。ESSCertID は issuerSerial フィールドを含まなければならない。

SignerInfo の issuerAndSerialNumber は issuerSerial フィールドと整合が取れていなければならない。署名検証は ESSCertID で特定された証明書を利用して行う必要がある。もしも証明書のハッシュ値が署名検証用の証明書とマッチしないものであった場合は、署名は無効であるとみなさねばならない。

また、policy information フィールドは利用しない。

```
SigningCertificate ::=  SEQUENCE {
                certs        SEQUENCE OF ESSCertID,
                policies     SEQUENCE OF PolicyInformation OPTIONAL  --利用しない
}

id-aa-signingCertificate OBJECT IDENTIFIER ::= { iso(1)member-body(2) us(840)
                                rsadsi(113549) pkcs(1) pkcs9(9) smime(16) id-aa(2) 12 }

ESSCertID ::=  SEQUENCE {
                certHash        Hash,
                issuerSerial    IssuerSerial OPTIONAL
}

Hash ::= OCTET STRING -- SHA1 hash of entire certificate

IssuerSerial ::= SEQUENCE {
                Issuer          GeneralNames,
                serialNumber    CertificateSerialNumber
   }
```

B) Other signing certificate 属性の定義

SHA-1 以外のハッシュアルゴリズムを利用できることを除いて、ESS SigningCertificate の定義と同じである。

```
id-aa-ets-otherSigCert OBJECT IDENTIFIER ::= { iso(1) member-body(2) us(840)
                                    rsadsi(113549) pkcs(1) pkcs9(9) smime(16) id-aa(2) 19 }

OtherSigningCertificate ::=  SEQUENCE {
            certs       SEQUENCE OF OtherCertID,
            policies    SEQUENCE OF PolicyInformation OPTIONAL  --利用しない
    }

OtherCertID ::= SEQUENCE {
            otherCertHash           OtherHash,
            issuerSerial            IssuerSerial OPTIONAL
    }

OtherHash ::= CHOICE {
            sha1Hash  OtherHashValue,  -- SHA-1の場合、ここに格納
            otherHash  OtherHashAlgAndValue
    }
OtherHashValue ::= OCTET STRING

OtherHashAlgAndValue ::= SEQUENCE {
            hashAlgorithm  AlgorithmIdentifier,
            hashValue      OtherHashValue
   }
```

※ESS SigningCertificate 属性、Other signing certificate 属性、いずれも検証時には処理できなければならない。

(8) EPES に対する必須属性

① Signature policy identifier

EPES は署名ポリシに対するリファレンスを持たなければならない。signature policy identifier は signed attribute でなければならない。

```
id-aa-ets-sigPolicyId OBJECT IDENTIFIER ::= { iso(1) member-body(2) us(840)
                                rsadsi(113549) pkcs(1) pkcs9(9) smime(16) id-aa(2) 15 }

SignaturePolicyIdentifier ::= CHOICE{
            SignaturePolicyId          SignaturePolicyId,
            SignaturePolicyImplied     SignaturePolicyImplied }

SignaturePolicyId ::= SEQUENCE {
            sigPolicyIdentifier   SigPolicyId,
            sigPolicyHash         SigPolicyHash,
            sigPolicyQualifiers   SEQUENCE SIZE (1..MAX) OF
            SigPolicyQualifierInfo      OPTIONAL
}

SignaturePolicyImplied ::= NULL

SigPolicyId ::= OBJECT IDENTIFIER

SigPolicyHash ::= OtherHashAlgAndValue

SigPolicyQualifierInfo ::= SEQUENCE {
```

```
            sigPolicyQualifierId  SigPolicyQualifierId,
            sigQualifier          ANY DEFINED BY sigPolicyQualifierId
}

SigPolicyQualifierId ::=  OBJECT IDENTIFIER
id-spq-ets-uri OBJECT IDENTIFIER ::= { iso(1) member-body(2) us(840)
                                    rsadsi(113549) pkcs(1) pkcs9(9) smime(16) id-spq(5) 1 }

SPuri ::= IA5String

id-spq-ets-unotice OBJECT IDENTIFIER ::= { iso(1) member-body(2) us(840)
                                    rsadsi(113549) pkcs(1) pkcs9(9) smime(16) id-spq(5) 2 }

SPUserNotice ::= SEQUENCE {
            noticeRef       NoticeReference OPTIONAL,
            explicitText    DisplayText OPTIONAL
}

NoticeReference ::= SEQUENCE {
            organization    DisplayText,
            noticeNumbers   SEQUENCE OF INTEGER
}

DisplayText ::= CHOICE {
            visibleString   VisibleString  (SIZE (1..200)),
            bmpString       BMPString      (SIZE (1..200)),
            utf8String      UTF8String     (SIZE (1..200))
}
```

(9) オプションの CMS 属性

次にあげる属性は、本書で定義する署名データに出現しても良い属性であり、本属性が存在することを理由として、構築時や検証時にエラーとしてはならない。

① Signing time

構文は、CMS（RFC3852）に定義されているとおりだが、長期署名では UTCTime でなく GeneralizedTime（YYYYMMDDHHMMSSZ と表記。秒の端数は含めない）の使用を推奨する。

```
id-signingTime OBJECT IDENTIFIER ::= { iso(1) member-body(2) us(840)
                                                    rsadsi(113549) pkcs(1) pkcs9(9) 5 }

SigningTime ::= Time

Time ::= CHOICE {
            utcTime         UTCTime,
            generalizedTime  GeneralizedTime }
```

- SigningTime 属性は、signed attribute 内に複数存在してはならない。

② Countersignature

countersignature は unsigned attribute でなければならない。

```
id-countersignature OBJECT IDENTIFIER ::= { iso(1) member-body(2) us(840)
                                            rsadsi(113549) pkcs(1) pkcs9(9) 6 }

Countersignature ::= SignerInfo
```

※unsigned attribute に含まれる countersignature は、生成時期に制約がない（つまり ES-A 生成後の署名に対しても countersignature を添付可能と考えられる）。また長期署名フォーマットの適用も可能と考えられる。countersignature の対象となっている署名における archiveTimeStamp の対象については注意を要する。

(10) オプションの ESS 属性

次にあげる属性は、本書で定義する署名データに含めても良い属性であり、本属性が存在することを理由として、構築時や検証時にエラーとしてはならない。

① content reference 属性

content reference 属性は signedAttribute。

```
id-aa-contentReference   OBJECT IDENTIFIER ::= { iso(1) member-body(2)
                           us(840) rsadsi(113549) pkcs(1) pkcs-9(9) smime(16) id-aa(2) 10 }

ContentReference ::= SEQUENCE {
            contentType              ContentType,
            signedContentIdentifier   ContentIdentifier,
            originatorSignatureValue  OCTET STRING }
```

② Content Identifier 属性

Contentidentifier は signed attribute。

```
id-aa-contentReference   OBJECT IDENTIFIER ::= { iso(1) member-body(2) us(840)
                                    rsadsi(113549) pkcs(1) pkcs-9(9) smime(16) id-aa(2) 10 }

ContentReference ::= SEQUENCE {
            contentType ContentType,
            signedContentIdentifier ContentIdentifier,
            originatorSignatureValue OCTET STRING }
```

③ Content Hints 属性

```
ContentHints ::= SEQUENCE {
            contentDescription UTF8String (SIZE (1..MAX)) OPTIONAL,
             contentType ContentType }

id-aa-contentHint OBJECT IDENTIFIER ::= { iso(1) member-body(2) us(840)
            rsadsi(113549) pkcs(1) pkcs-9(9) smime(16) id-aa(2) 4}
```

(11) 他のオプション属性

次にあげる属性は、本書で定義する署名データに含めても良い属性であり、本属性が存在することを理由として、構築や検証時にエラーとしてはならない。

① Commitment Type Indication 属性

commitmentTypeIndication は signedAttribute である。

```
id-aa-ets-commitmentType OBJECT IDENTIFIER ::= { iso(1) member-body(2)
                        us(840) rsadsi(113549) pkcs(1) pkcs-9(9) smime(16) id-aa(2) 16}

CommitmentTypeIndication ::= SEQUENCE {
        commitmentTypeId            CommitmentTypeIdentifier,
        commitmentTypeQualifier     SEQUENCE SIZE (1..MAX) OF
                                            CommitmentTypeQualifier      OPTIONAL
}

CommitmentTypeIdentifier ::= OBJECT IDENTIFIER

CommitmentTypeQualifier ::= SEQUENCE {
        commitmentTypeIdentifier   CommitmentTypeIdentifier,
        qualifier                  ANY DEFINED BY  commitmentTypeIdentifier
}
```

```
id-cti-ets-proofOfOrigin OBJECT IDENTIFIER ::= { iso(1) member-body(2) us(840)
                                    rsadsi(113549) pkcs(1) pkcs-9(9) smime(16) cti(6) 1}
id-cti-ets-proofOfReceipt OBJECT IDENTIFIER ::= { iso(1) member-body(2) us(840)
                                    rsadsi(113549) pkcs(1) pkcs-9(9) smime(16) cti(6) 2}
id-cti-ets-proofOfDelivery OBJECT IDENTIFIER ::= { iso(1) member-body(2) us(840)
                                    rsadsi(113549) pkcs(1) pkcs-9(9) smime(16) cti(6) 3}
id-cti-ets-proofOfSender OBJECT IDENTIFIER ::= { iso(1) member-body(2) us(840)
                                    rsadsi(113549) pkcs(1) pkcs-9(9) smime(16) cti(6) 4}
id-cti-ets-proofOfApproval OBJECT IDENTIFIER ::= { iso(1) member-body(2) us(840)
                                    rsadsi(113549) pkcs(1) pkcs-9(9) smime(16) cti(6) 5}
id-cti-ets-proofOfCreation OBJECT IDENTIFIER ::= { iso(1) member-body(2) us(840)
                                    rsadsi(113549) pkcs(1) pkcs-9(9) smime(16) cti(6) 6}
```

Proof of origin	署名者がその文書を生成したこと、承認したこと、そして送信したことを示す。
Proof of receipt	署名者がその文書を受け取ったことを示す。
Proof of delivery	TSP（信頼できるサービスプロバイダ）がその文書を受信者がアクセスできる状態でローカルなストアにおいたことを提示したことを示す。
Proof of sender	その提示をしたエンティティがその文書を送信したことを示す。（生成したのではなくても良い）
Proof of approval	署名者がその文書を承認したことを示す。
Proof of creation	署名者がその文書を生成したことを示す。（承認したり送信したりする必要はない）

② Signer Location 属性

Signer Location は signedAttribute である。

```
id-aa-ets-signerLocation OBJECT IDENTIFIER ::= { iso(1) member-body(2) us(840)
                                    rsadsi(113549) pkcs(1) pkcs-9(9) smime(16) id-aa(2) 17}

SignerLocation ::= SEQUENCE {
        -- 少なくとも次のどれか１つが必
    countryName         [0] DirectoryString     OPTIONAL,
        -- X.500 の Coutry 名
    localityName        [1] DirectoryString     OPTIONAL,
        -- X.500 の locality 名
    postalAdddress      [2] PostalAddress       OPTIONAL
}

PostalAddress ::= SEQUENCE SIZE(1..6) OF DirectoryString
```

③ Signer Attributes 属性

signer-attributes は signed attribute である。

```
id-aa-ets-signerAttr OBJECT IDENTIFIER ::= { iso(1) member-body(2) us(840)
                                    rsadsi(113549) pkcs(1) pkcs-9(9) smime(16) id-aa(2) 18}

SignerAttribute ::= SEQUENCE OF CHOICE {
    claimedAttributes       [0]  ClaimedAttributes,
    certifiedAttributes     [1]  CertifiedAttributes
}

ClaimedAttributes ::= SEQUENCE OF Attribute

CertifiedAttributes ::= AttributeCertificate
```

④ Content Time-Stamp 属性

content time-stamp は signed attribute である。

```
id-aa-ets-contentTimestamp OBJECT IDENTIFIER ::= { iso(1) member-body(2) us(840)
                                    rsadsi(113549) pkcs(1) pkcs-9(9) smime(16) id-aa(2) 20}

ContentTimestamp:= TimeStampToken
```

(12) 複数署名のサポート

① 並列署名（Independent Signatures）

並列署名は、同一の文書に対して並行して複数人による複数の署名を添付する場合に用いる。個々の署名は独立であり、これを実現するためには複数の SignerInfo を利用する。SignerInfo 毎に独立して長期署名フォーマットを適用できる。

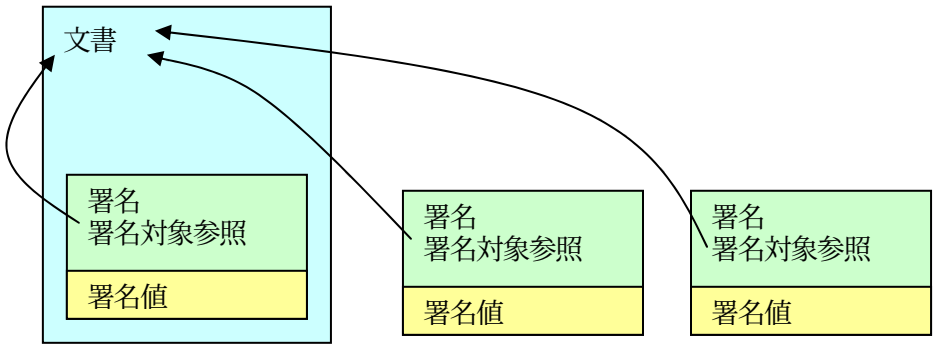


図 付1.1-3 並列署名

② 直列署名（Embedded Signatures）

直列署名は、署名に対して署名を重ねていく場合に用いる。これを実現するためには、counterSignature 属性を利用する。countersignature は、生成時期に制約がない（つまりES-A 生成後の署名に対しても countersignature を添付可能と考えられる)。また長期署名フォーマットの適用も可能と考えられる。countersignature の対象となっている署名におけるarcheveTimeStamp の対象については注意を要する。

countersignature に対する長期署名フォーマットの適用については、署名対象が本文（図中の文書）ではなく、そのハッシュ値のみを含む署名値であるため、その有効性について注意深く検討する必要がある。本プロファイルでは countersignature の長期署名フォーマットの適用については規定しない。

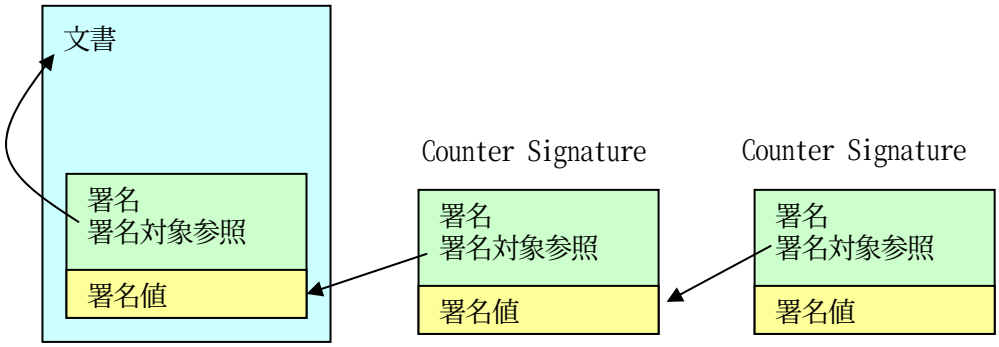


図 付1.1-4 直列署名

1.2 ES-T（Electronic Signature Time-stamped）

ES-T は、デジタル署名の存在時刻を確定するために、電子署名文書中の署名値（CMS の SignerInfo の SignatureValue)に対して TSA から取得したタイムスタンプトークンを追加したものである。署名値は電子文書のハッシュ値をもとに計算されるため、署名値から生成したタイムスタンプトークンは、署名の存在時刻とともに、電子データの存在時刻も証明することとなる。

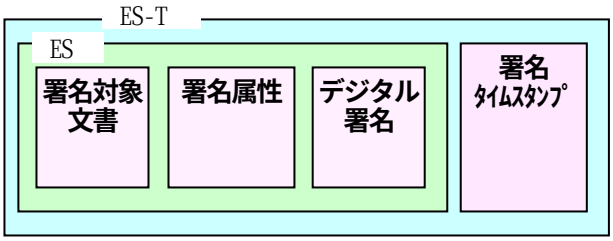


図 付1.1-5 ES-T

ひとつの署名に対していくつかの異なったTSAからタイムスタンプトークンを取得して格納しても良い。複数の署名が添付されている場合、個々の署名値に対してそれぞれタイムスタンプトークンを取得してもよいし、ある署名についてのみタイムスタンプトークンを取得してもよい。

Signature Timestamp属性のOIDは次の値である。

```
id-aa-signatureTimeStampToken OBJECT IDENTIFIER ::= { iso(1) member-body(2) us(840)
                                    rsadsi(113549) pkcs(1) pkcs-9(9) smime(16) id-aa(2) 14}
```

Signature Timestamp属性の値は、ASN.1形式のSignatureTimeStampTokenである。

```
SignatureTimeStampToken ::= TimeStampToken
```

TimeStampTokenのmessageImprintフィールドの値は、signedDataのSignerInfo内のsignatureフィールドの値（タイプと長さを除いた値の部分のみ）である。

TimeStampTokenは、RFC3161で定義される。

SignatureTimestampの検証情報（認証パス及び失効情報、ESの検証情報に準じる）は次のいずれかに格納する。

1) タイムスタンプトークン内のcertificatesとcrls
2) ESの検証情報と同じ場所（Complete validation reference dataとExtended validation data）
3) タイムスタンプトークン内のunsigned attribute(Extended validation data形式)

※構築時は、1)または3)を推奨、検証時は1)～3)全てに対応することを必須とする。

1.3 ES-C（Complete validation reference data）

ES-Cは、ES-Tに対してデジタル署名の検証の際に利用する認証パス上の全ての公開鍵証明書（ただし署名者の公開鍵証明書を除く）とそれぞれの公開鍵証明書のCRLやOCSPレスポンスなどの失効情報（署名者の公開鍵証明書の失効情報を含む）に対するリファレンス情報を追加したものである。

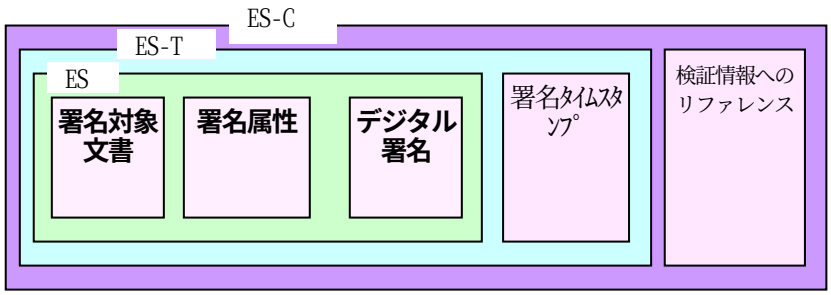


図 付1.1-6 ES-C

complete validation reference data 付の電子署名は、署名の検証に必要なすべてのデータ（証明書及び失効情報）を備える電子署名文書である。

Complete validation reference data の最小構成は次のとおり：

- Signature Timestamp 属性
- Complete Certificate Refs
- Complete Revocation Refs

Complete validation reference data は次の情報を含む X-Long validation data を構成してもよい（将来、検証プロセスがこれらのデータにアクセスできなくなることに備えるため）：

- Complete Certificate Values **←必須とする。**
- Complete Revocation Values **←必須とする。**

Complete validation reference data はまた次の情報を含む Extended validation data を構成してもよい（将来の CA の危殆化に備えることと、検証データの完全性を確保するため）：

- ES-C Timestamp（ES-X Type1 の場合に存在） **←利用しない（無視してかまわない）。**
- Time-Stamped Certificates and CRLs references（ES-X Type2 の場合に存在） **←利用しない（無視してかまわない）。**

① Complete Certificate Refs 属性の定義

Complete Certificate Refs 属性は unsigned attribute である。Complete Certificate Refs 属性は ES の検証に用いる署名者の証明書に至るすべての CA の証明書を参照する。（ただし署名者の証明書への参照は含まない）

この属性は 1 署名につき一つだけ含む。

注記 1：署名者の証明書は signing certificate 属性で参照される。

注記 2：署名タイムスタンプの認証パスを含んでも良い。

Complete Certificate Refs 属性の OID は次のとおりである。

```
id-aa-ets-certificateRefs OBJECT IDENTIFIER ::= { iso(1) member-body(2) us(840)
                                    rsadsi(113549) pkcs(1) pkcs-9(9) smime(16) id-aa(2) 21}
```

Complete Certificate Refs 属性は ASN.1 構文の CompleteCertificateRefs を値として持つ。

```
CompleteCertificateRefs ::= SEQUENCE OF OtherCertID
```

OtherCertID には IssuerSerial を含まねばならない。certHash は参照される証明書のハッシュ値とマッチしなければならない。

② Complete Revocation Refs 属性の定義

Complete Revocation Refs 属性は unsigned attribute である。この属性は 1 署名に対して一つだけ存在する。この属性は、ES-C を検証するために必要な署名者及び CA の証明書に対する CRL あるいは OCSP レスポンスのすべてを参照する。

注記：署名タイムスタンプの認証パスに対する失効情報を格納しても良い。

Complete Revocation Refs 属性の OID は次のとおりである。

```
id-aa-ets-revocationRefs OBJECT IDENTIFIER ::= { iso(1) member-body(2) us(840)
                                rsadsi(113549) pkcs(1) pkcs-9(9) smime(16) id-aa(2) 22}
```

Complete Revocation Refs 属性は ASN.1 構文の CompleteRevocationRefs を値として持つ。

```
CompleteRevocationRefs ::= SEQUENCE OF CrlOcspRef

CrlOcspRef ::= SEQUENCE {
        crlids          [0] CRLListID       OPTIONAL,
        ocspids         [1] OcspListID      OPTIONAL,
        otherRev        [2] OtherRevRefs    OPTIONAL
  }
```

CompleteRevocationRefs は signing certificate に対する CrlOcspRef を必ず 1 つ持たなければならず、CompleteCertificateRefs 属性の中の各 OtherCertID に対して 1 つずつ CrlOcspRef を持たなければならない。2 番目以降の CrlOcspRef の順番は、対応する OtherCertID の順番と同じでなければならない。信頼している CA の証明書を除く証明書パス上のすべての証明書に対して、CRLListID、OcspListID、OtherRevRefs のうち、少なくとも一つを含めなければならない。CRL あるいは OCSP レスポンス以外の失効情報は利用しない。

```
CRLListID ::=  SEQUENCE {
        crls        SEQUENCE OF CrlValidatedID}

CrlValidatedID ::=  SEQUENCE {
        crlHash                 OtherHash,
        crlIdentifier           CrlIdentifier OPTIONAL}

CrlIdentifier ::= SEQUENCE {
        crlissuer               Name,
        crlIssuedTime           UTCTime,
        crlNumber               INTEGER OPTIONAL
    }
```

```
OcspListID ::=  SEQUENCE {
        ocspResponses        SEQUENCE OF OcspResponsesID}

OcspResponsesID ::=  SEQUENCE {
        ocspIdentifier              OcspIdentifier,
        ocspRepHash                 OtherHash    OPTIONAL
    }

OcspIdentifier ::= SEQUENCE {
        ocspResponderID    ResponderID,
          -- As in OCSP response data
        producedAt      GeneralizedTime
          -- As in OCSP response data
    }
```

crlValidatedID を作成する際、crlHash は、署名を含む CRL の完全な DER エンコードされたデータに対して計算する。crlIdentifier は、通常、他の情報によって CRL が推測できないときに存在する。

crlIdentifier は、CRL を特定するためのものであり、発行者名と発行時刻（CRL が含む“thisUpdate”が示す時刻）を利用している。

crlListID 属性は、unsigned attribute である。CRL が Delta CRL であれば、complete revocation list には CRL の集合に対する参照が含まれねばならない

OcspIdentifier は OSCP レスポンスを特定するもので、発行者名と発行時刻（OCSP レスポンスに含まれる“producedAt”が示す時刻）を用いる。同時刻に発行された OCSP を区別するには、OcspResponceID に含まれるレスポンスのハッシュ値を用いる。

注記１：署名タイムスタンプの失効情報を含めてよい。

```
OtherRevRefs ::= SEQUENCE {
        otherRevRefType      OtherRevRefType
        otherRevRefs         ANY DEFINED BY otherRevRefType
    }

OtherRevRefType ::= OBJECT IDENTIFIER
```

1.4 Extended Validation Data（ES-X）

ES-X は、将来の CA の危殆化に備えたり、検証データの完全性を確保したり入手困難になることに備えるために、ES-C を拡張するものである。

ES-X には、ES-X Long（図 付1.1-7）、ES-X Type1（図 付1.1-8）、ES-X Type2（図 付1.1-9）の３通りのフォーマットが用意される。本プロファイルでは、ES-X Long の使用のみを認める。

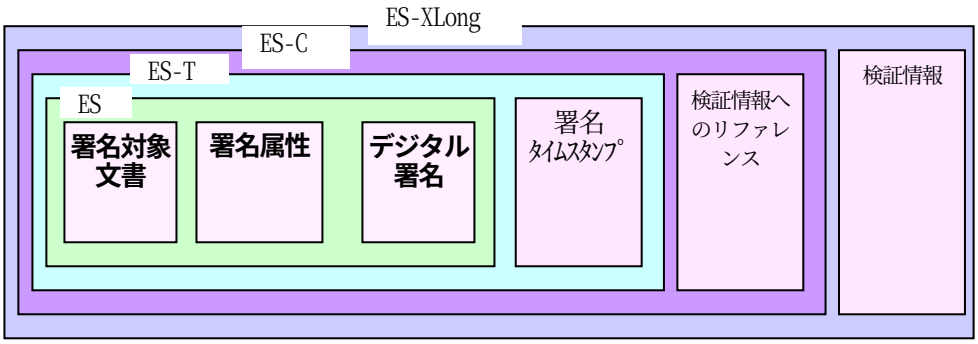


図 付1.1-7 ES-X Long

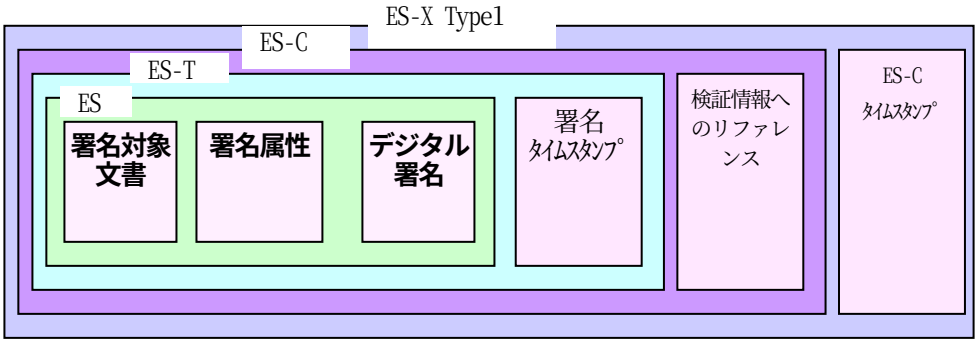


図 付1.1-8 ES-X Type1（使用しない）

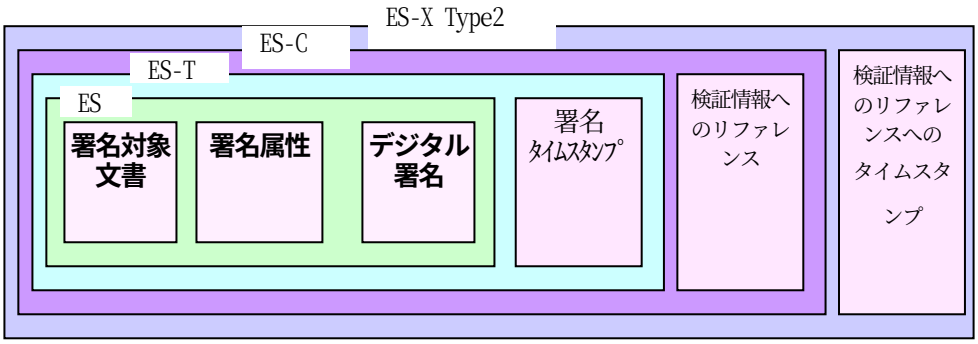


図 付1.1-9 ES-X Type2（使用しない）

ES-X Long は検証情報そのものを電子署名文書内に抱え込むフォーマットである。ES-X Type1 は、ES-C 全体に対するタイムスタンプを取得して追加するもの、ES-X Type2 は、検証情報へのリファレンスのみに対するタイムスタンプを取得して追加するものである。

検証情報のリファレンスには検証情報のハッシュ値が含まれるが、そのとき用いるハッシュ関数が脆弱化するケースを想定すると、リファレンスをタイムスタンプの対象とするのでは、リファレンスと検証情報そのものとの対応関係を証明することができなくなる。更に、検証情報そのもの（特に中間のサブ CA の公開鍵証明書や失効情報など）の消失に備えるには、検証情報そのものを保持しておく必要がある。

長期保存のためには、電子文書、デジタル署名、タイムスタンプ、検証情報全体をタイムスタンプや耐タンパな仕組みで保護する必要がある。長期署名フォーマットでは、この後に述べるアーカイブタイムスタンプによって保護する。つまり、適切な時期にアーカイブタイムスタンプを追加することによって、ES-C タイムスタンプも検証情報リファレンスへのタイムスタンプも必要なくなり、重要なのは検証情報そのものを確保しておくことである。

ES-X Long は保護対象となる全てのデータを格納するフォーマットに相当する。電子署名文書の長期保存を可能とするシステムを実装するためには、ES-X Long のみをサポートすればよい。

① Certificate Values 属性の定義

Certificate Values 属性は unsigned attribute である。この属性は 1 署名につき 1 つだけ存在する。この属性により、CompleteCertificationRefs が参照する証明書および署名者の証明書を保持する。（署名者証明書をここに含めるのは、格納必須である場所が他に指定されていないため、SingedData の Certificates などではアーカイブタイムスタンプの対象とはならず、保護されないため）

注意：Attribute Certificate が利用されるときは、この構造が用いられるのではなく、signer-attributes 属性が用いられる。

Certificate Values 属性を示す OID は次のとおりである。

```
id-aa-ets-certValues OBJECT IDENTIFIER ::= { iso(1) member-body(2) us(840)
                                    rsadsi(113549) pkcs(1) pkcs-9(9) smime(16) id-aa(2) 23}
```

Certificate Values 属性は値として次の ASN.1 構文で表される CertificateValues を取る。

```
CertificateValues ::=  SEQUENCE OF Certificate
```

Certificate の定義は RFC3280 と ITU-T Recommendation X.509 を参照のこと。

② Revocation Values 属性の定義

Revocation Values 属性は unsigned attribute である。この属性は 1 署名につき 1 つだけ存在する。この属性は、CompleteRevocationRefs 属性で参照される CRL と OCSP レスポンスの値を保持する。

Revocation Values 属性の OID は次のとおりである。

```
id-aa-ets-revocationValues OBJECT IDENTIFIER ::= { iso(1) member-body(2) us(840)
                                    rsadsi(113549) pkcs(1) pkcs-9(9) smime(16) id-aa(2) 24}
```

Revocation Values 属性は値として次の ASN.1 構文で表される RevocationValues を取る。

```
RevocationValues ::=  SEQUENCE {
            crlVals          [0] SEQUENCE OF CertificateList     OPTIONAL,
            ocspVals         [1] SEQUENCE OF BasicOCSPResponse   OPTIONAL,
            otherRevVals     [2] OtherRevVals                    OPTIONAL  ←利用しない
 }
```

```
OtherRevVals ::= SEQUENCE {
        otherRevValType     OtherRevValType,
        otherRevVals        ANY DEFINED BY otherRevValType
}

OtherRevValType ::= OBJECT IDENTIFIER
```

Other revocation values **←利用しない。**
CertificateList の定義は、RFC3280 と ITU-T Recommendation X.509 を参照のこと。
BasicOCSPResponse の定義は、RFC2560 を参照のこと。

③ ES-C Time-Stamp 属性の定義 ←利用しない。無視してかまわない。

④ Time-Stamped Certificates and CRLs 属性の定義 **←利用しない。無視してかまわない。**

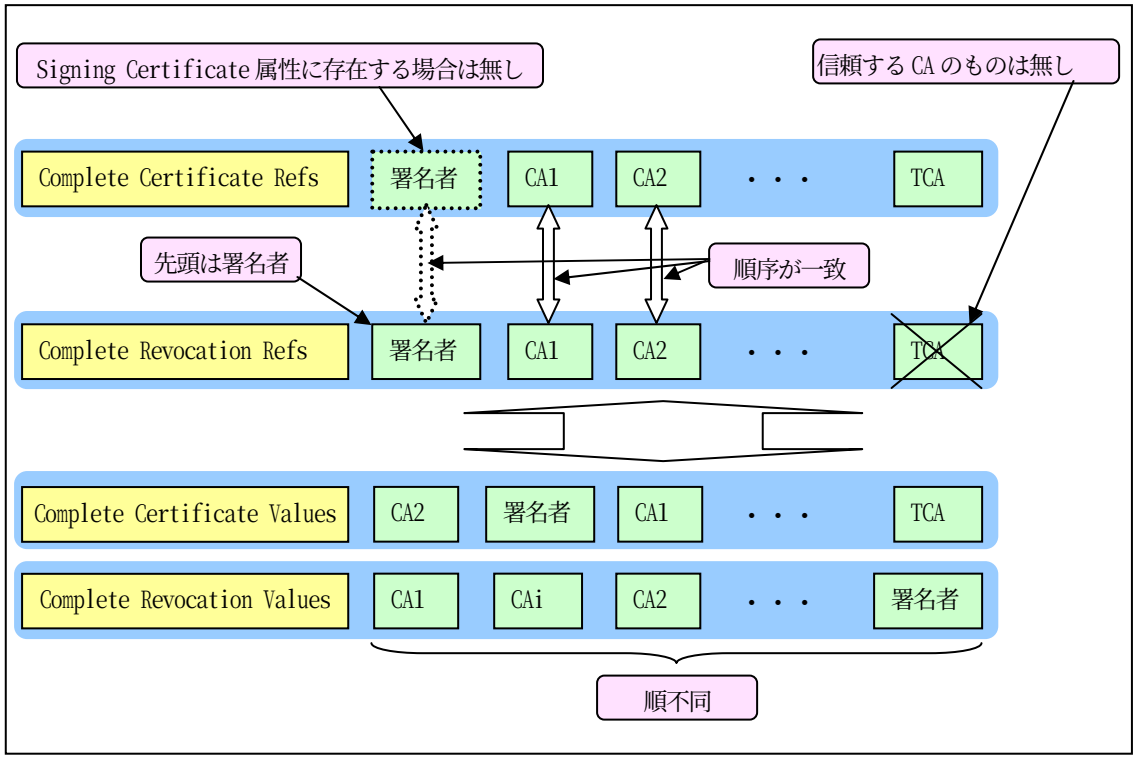


図 付1.1-10 Complete Validation Reference Data と Validation Values との関係

1.5 ES-A（Archive Validation Data）

電子署名の検証可能期間を極めて長くしようとしたとき、タイムスタンプの署名の危殆化や TSA の証明書の有効期限切れが発生しうるため、タイムスタンプの署名を複数回重ねることが要求されることがある。このとき、archive time-stamp 属性が用いられる。このタイムスタンプは

期間をおいて繰り返し付与される。

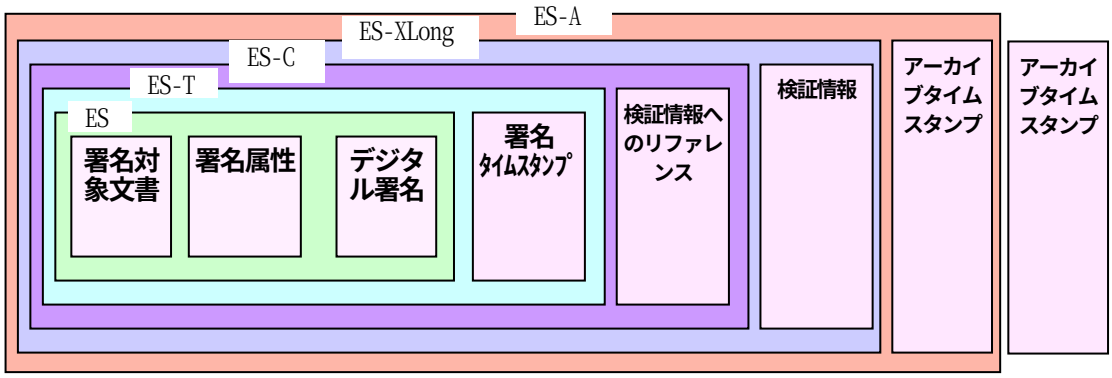


図 付1.1-11 ES-A

① Archive Time-Stamp属性の定義

Archive Time-Stamp は署名対象文書と署名全体に対するタイムスタンプである。Certificate Values と Revocation Values 属性がない場合、タイムスタンプをとる前にこれらの属性を加えなければならない。Archive Time-Stamp 属性は、unsigned attribute である。この属性は、1 署名に対して、時間の経過や複数の TSA から得ることにより複数添付できる。

Archive Time-Stamp属性の OID は次のとおりである。

```
id-aa-ets-archiveTimestamp OBJECT IDENTIFIER ::= { iso(1) member-body(2) us(840)
                                    rsadsi(113549) pkcs(1) pkcs-9(9) smime(16) id-aa(2) 27}
```

Archive Time-Stamp属性の値として、次の ASN.1 構文の ArchiveTimeStampToken が入る。

```
ArchiveTimeStampToken ::= TimeStampToken
```

TimeStampToken の messageImprint の値は、次のデータをこの順に結合した値（ただし、タイプと長さを除いたもの）のハッシュ値である。（図 付1.1-12 参照）

- encapContentInfo eContent OCTET STRING;
- signedAttributes;
- signature field within SignerInfo;
- SignatureTimeStampToken attribute;
- CompleteCertificateRefs attribute;
- CompleteRevocationRefs attribute;
- CertificateValues attribute
  （まだこの値を確保していなければ、ES-A を作る際に確保しなければならない。）
- RevocationValues attribute
  （まだこの値を確保していなければ、ES-A を作る際に確保しなければならない。）

- ESCTimeStampToken attribute if present; **←利用しない（対象に含めない）。**
- TimestampedCertsCRLs attribute if present; **←利用しない（対象に含めない）。**
- any previous ArchiveTimeStampToken attributes.

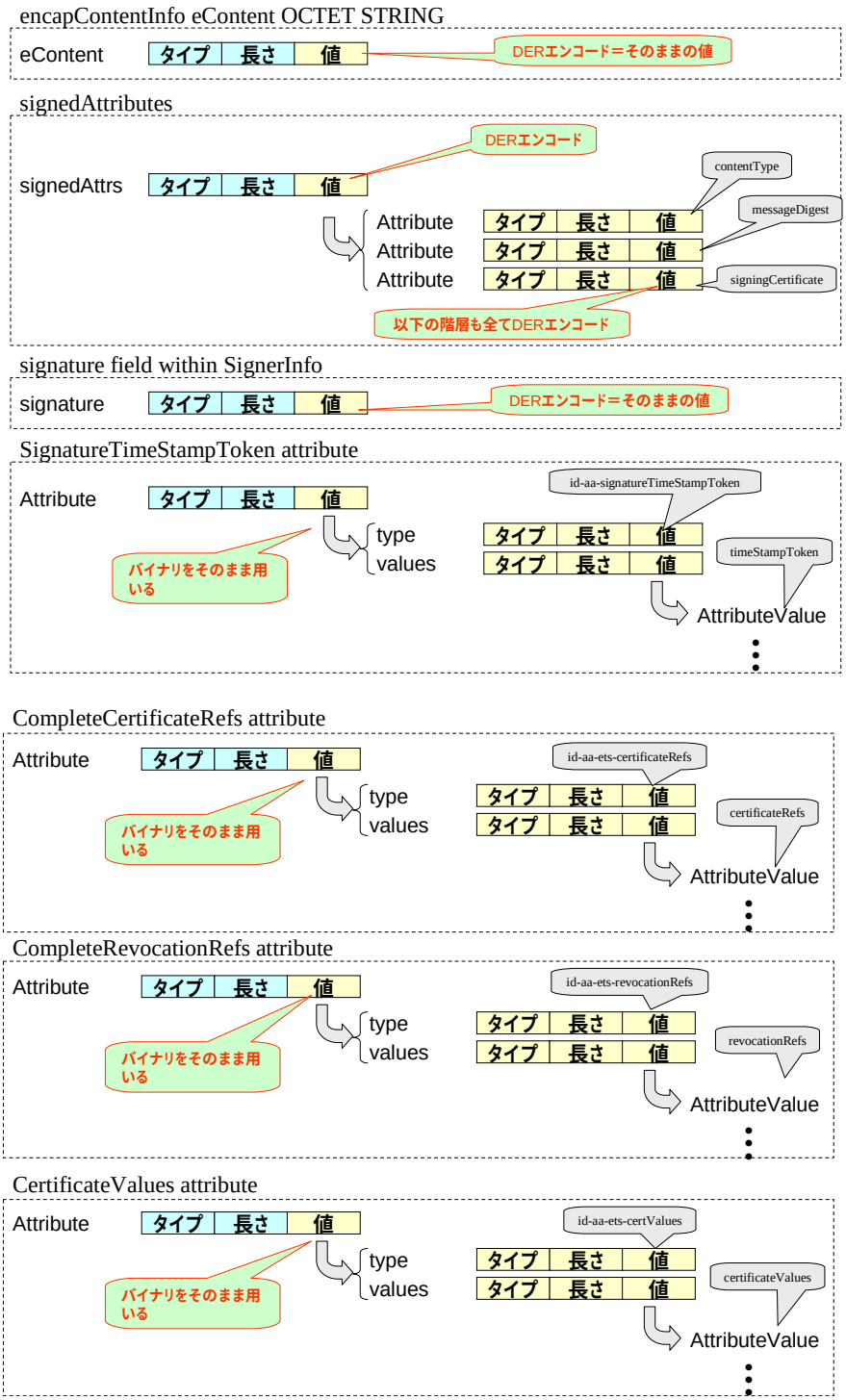


図 付1.1-12 アーカイブタイムスタンプの対象（その1）

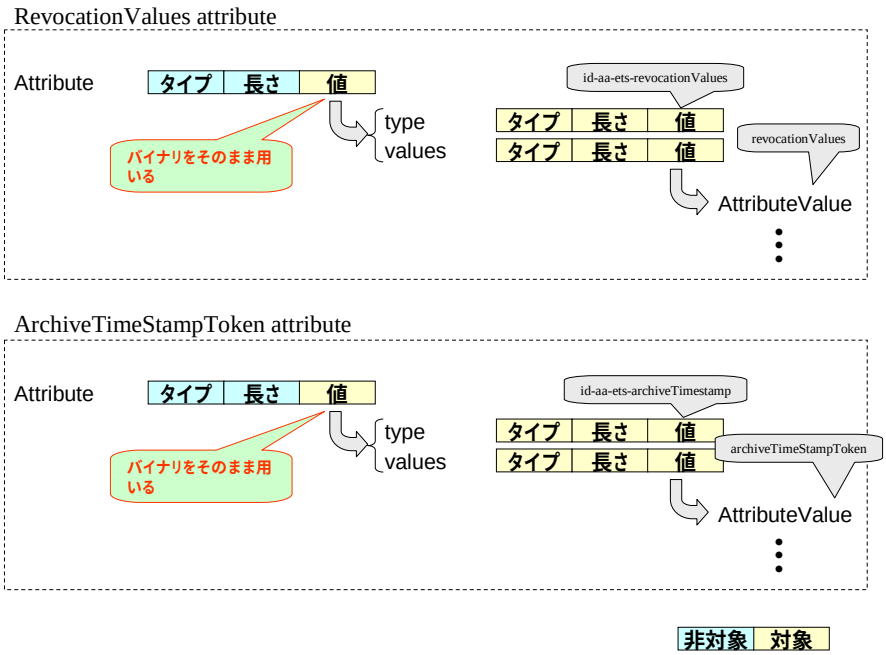


図 付 1.1-13 アーカイブタイムスタンプの対象（その 2）

TimeStampToken に関しては、RFC3161 を参照のこと。

タイムスタンプは、オリジナルの署名よりも強いアルゴリズム（あるいは長い鍵）を利用するのが望ましい。

アーカイブタイムスタンプの検証情報は次のいずれかに格納することが考えられる。

1) タイムスタンプトークン内の certificates と crls

2) タイムスタンプトークン内の unsigned attribute（Extended validation data 形式）

本プロファイルでは、構築時には 1)に格納することを推奨し、検証時には 1),2)を処理できることを必須とする。

なお、draft-pinkas-smime-cades-00.txt や ETSI TS 101 733 V1.5.1 では、アーカイブタイムスタンプの取得対象が異なる。つまり、TimeStampToken の messageImprint の値は、次のデータをこの順に結合した値（ただし、タイプと長さを除いたもの）のハッシュ値である。

- SignedData 内の encapContentInfo
- もしも存在した場合は、SignedData 内の Certificates と crls
- すべての署名属性、非署名属性を含む SignerInfo の全てのデータ

ところが、最後の項目を対象とするとした場合、各アーカイブタイムスタンプの検証時に、そのアーカイブタイムスタンプが対象とした情報が確定できない場合がある。例えば、contersignature が後から添付された場合、countersignature にアーカイブタイムスタンプが付

与された場合などがそれに当たる。従って、本プロファイルでは、タイムスタンプ対象を明確に定める旧仕様（RFC3126 や TSI TS 101 733 V1.4.0 以前）に基づいた仕様を採用することとし、新版の仕様は対象外とする。

1.6 長期署名の構築

長期署名を構築する際に利用する各種データの取得時期を次表に示す。

フォーマット	データの種類	取得すべき日時
ES-T	署名タイムスタンプ	署名者証明書の有効期限及び失効以前。署名生成あるいは入手後速やかに取得することが望ましい。
	署名タイムスタンプの検証情報―認証パス上の全ての証明書	第 1 世代アーカイブタイムスタンプ取得以前
	署名タイムスタンプの検証情報―上記の失効情報	第 1 世代アーカイブタイムスタンプ取得直前
ES-C ES-XL	署名者の検証情報―認証パス上の全ての証明書	第 1 世代アーカイブタイムスタンプ取得以前
	署名者の検証情報―上記の失効情報	署名タイムスタンプ取得時点から猶予期間が経過した後
ES-A	第 1 世代アーカイブタイムスタンプ	署名タイムスタンプ及び署名者の検証情報が無効となる以前。（署名タイムスタンプ証明書の有効期限及び失効以前で、かつ署名者証明書及び署名者検証情報の発行者（CA）の証明書の有効期限及び失効以前。）
	第 1 世代アーカイブタイムスタンプの検証情報―認証パス上の全ての証明書	第 2 世代アーカイブタイムスタンプ取得以前
	第 1 世代アーカイブタイムスタンプの検証情報―上記の失効情報	第 2 世代アーカイブタイムスタンプ取得直前
	第 2 世代アーカイブタイムスタンプ	第 1 世代アーカイブタイムスタンプの有効期限及び失効以前
	：	：
	第 n 世代アーカイブタイムスタンプ	第 n-1 世代アーカイブタイムスタンプのトークンの有効期限及び失効以前
	第 n 世代アーカイブタイムスタンプの検証情報―認証パス上の全ての証明書	第 n+1 世代アーカイブタイムスタンプ取得以前
	第 n 世代アーカイブタイムスタンプの検証情報―上記の失効情報	第 n+1 世代アーカイブタイムスタンプ取得直前

なお失効情報取得の際の「猶予期間」は、失効要求が発生してから実際に失効情報に反映するまでの期間である（図 付 1.1-14 参照）。実際にどのくらいの期間を猶予期間とするかは、署名者

証明書を発行する認証局のポリシに依存して決める。

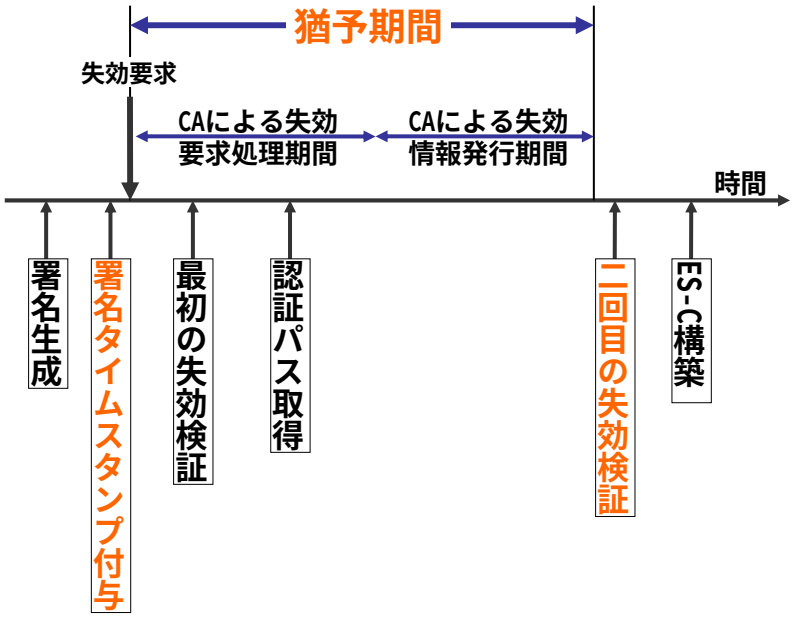


図 付 1.1-14 猶予期間

1.7 長期署名の検証

証明書の検証時刻は次表のとおりである。

フォーマット	証明書の種類	有効性を確認すべき日時
ES-T	署名者証明書	署名タイムスタンプ属性のトークンの時刻
	署名タイムスタンプ TSA 証明書	現在時刻
ES-C ES-XL	署名者証明書	署名タイムスタンプ属性のトークンの時刻
	署名タイムスタンプ TSA 証明書	現在時刻またはセキュアアーカイブされた時刻
ES-A	署名者証明書	署名タイムスタンプ属性のトークンの時刻
	署名タイムスタンプ TSA 証明書	第 1 世代アーカイブタイムスタンプのトークンの時刻
	第 1 世代アーカイブタイムスタンプ TSA 証明書	第 2 世代アーカイブタイムスタンプのトークンの時刻
	第 2 世代アーカイブタイムスタンプ TSA 証明書	第 3 世代アーカイブタイムスタンプのトークンの時刻
	：	：
	第 n-1 世代アーカイブタイムスタンプ TSA 証明書	第 n 世代アーカイブタイムスタンプのトークンの時刻
	第 n 世代アーカイブタイムスタンプ TSA 証明書	現在時刻

2．CMS 長期署名プロファイルまとめ

	CAdES BES	CAdES EPES	CAdES ES-T	CAdES ES-C	CAdES ES-X Long	CAdES ES-A
SignedAttributes	○	○	○	○	○	○
ContentType	○	○	○	○	○	○
MessageDigest	○	○	○	○	○	○
SigningTime	△	△	△	△	△	△
SigningCertificate	○	○	○	○	○	○
SignaturePolicyIdentifier	×	○	△2	△2	△2	△2
ContentReference	△	△	△	△	△	△
ContentIdentifier	△	△	△	△	△	△
ContentHints	△	△	△	△	△	△
CommitmentTypeIndication	△	△	△	△	△	△
SignerLocation	△	△	△	△	△	△
SignerAttribute	△	△	△	△	△	△
ContentTimeStamp	△	△	△	△	△	△
UnsignedAttribute	△	△	○	○	○	○
CounterSignature	△	△	△	△	△	△
SignatureTimeStamp	×	×	○	○	○	○
CompleteCertificateRefs	×	×	×	○	○	○
CompleteRevocationRefs	×	×	×	○	○	○
AttributeCertificateRefs	×	×	×	△	△	△
AttributeRevocationRefs	×	×	×	△	△	△
CertificateValues	×	×	×	×	○	○
RevocationValues	×	×	×	×	○	○
ES-C TimeStamp	×	×	×	×	×	×
TimeStampedCertsAndCrls	×	×	×	×	×	×
ArchiveTimeStamp	×	×	×	×	×	○

○：必須要素
△：オプション要素
△2：EPES をベースとする場合、SignaturePolicyIdentifier 要素は必須
×：不要（あってはならない要素）
網掛け：標準仕様にそのまま従うもの

	ETSI TS 101 733 V 1.4.0 以前 (RFC 3126)	ETSI TS 101 733 V 1.5.1
アーカイブタイムスタンプの計算対象	○	― ※1

※1：計算方法に不確定な要素があり、現バージョンのプロファイルでは対象外とする。

Ⅱ．XAdES 長期署名プロファイル（Version 1.0）

2006 年 3 月

次世代電子商取引推進協議会（ECOM）

目　次

1．XML 形式の長期署名フォーマット

本稿では、XML 署名に適用する長期署名フォーマットのプロファイルを示す。ここで示す長期署名フォーマットのプロファイルは、ETSI[1] TS 101 903 V1.3.1（2005-05），"XML Advanced Electronic Signatures（XAdES）"に準拠するものであり、CMS 長期署名フォーマットである ETSI TS101 703 V1.5.1（2003-12）"Electronic Signature Formats"とほぼ同等の内容を XML 署名に適用するものである。今回同時に策定する CMS 長期署名フォーマットのプロファイルでは、ETSI TS101 703 V1.5.1（2003-12）"Electronic Signature Formats"から必要なものを抽出したサブセットとして定義したが、XAdES は ETSI TS101 703 V1.5.1（2003-12） "Electronic Signature Formats"と内容的に同等なことから、本稿で示す署名フォーマットのプロファイルも XAdES のサブセットとし、内容的には CMS 長期署名フォーマットのプロファイルに沿った形で定義する。

なお、XAdES には W3C の Note[2]としても公開されている V1.1.1（2002-02）、既に策定され ETSI 公開されている ETSI TS 101 903 V1.2.2（2004-4）、および現在策定中でありドラフト版である ETSI TS 101 903 V1.3.1（2005-05）、の 3 つのバージョンが存在する。国内の動向を見ると未だそれほど実装が進んでいないが、現在策定中の最新バージョンは V1.1.1（2002-02）と同様に W3C の Note として公開される予定もあり、今後の実装や普及が見込まれる。そこで本稿では、現在の最新ドラフトである V1.3.1（2005-05）に準拠する形で示す。

2．署名フォーマット

基本となる電子署名文書の形式として、署名ポリシの有無により"Basic Electronic Signature"（XAdES-BES）、"Explicit Policy based Electronic Signature"（XAdES-EPES）の 2 通りを利用できる。署名ポリシとは、署名者と検証者がデジタル署名を有効とみなすための、署名の生成と検証に関する一連の規則を定めるものである。署名ポリシについては、ETSI TR 102 038 V1.1.1（2002-04），"XML format for signature policies"にコンピュータ処理可能な XML 形式のフォーマットを規定しており、記述内容は CMS 長期署名フォーマット同様 RFC3125, "Electronic Signature Policies" とまったく同じものである。

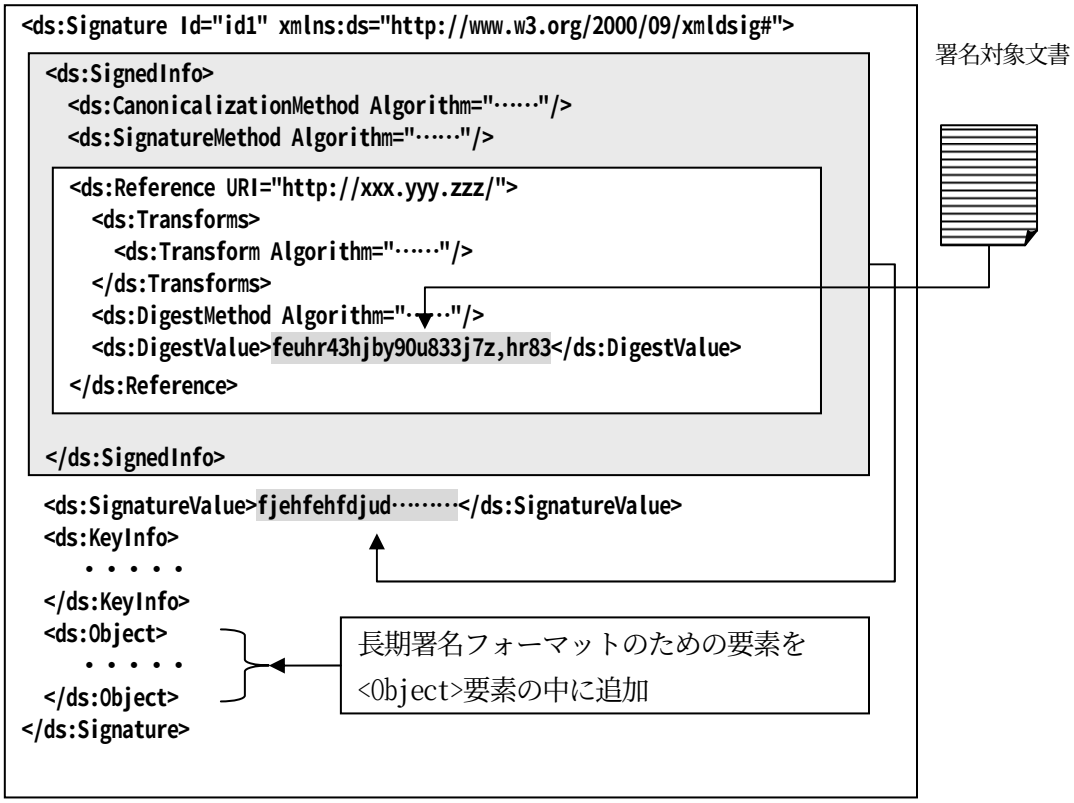
電子署名文書の形式は、W3C/IETF Recommendation（February 2002）:"XML-Signature Syntax and Processing"（XMLDSIG）の仕様に基づいている。

2.1 XML 署名の基本フォーマット

XML 形式の電子署名文書の基本フォーマットは、XMLDSIG に定義されている通りである。リスト 1 に基本的な XML 署名フォーマットの例を示す。

[1] http://www.etsi.org/
[2] http://www.w3.org/TR/XAdES/

リスト１：基本的な XML 署名フォーマットの例



XML 署名では、署名対象を XML 署名文書の<ds:Reference>要素の URI 属性で指定する。署名対象は複数指定することができ、同じ XML 文書内の<ds:Signature>要素より上位の要素（Enveloped 形式）、<ds:Object>要素の中に含まれる要素（Enveloping 形式）、<ds:Signature>要素と親子関係のない要素、および XML 署名文書とは別ファイルの任意のフォーマットの文書（Detached 形式）を指定することができる。XML 署名文書と別ファイルを署名対象文書とする Detached 形式の XML 署名文書を長期保存する場合、署名対象文書は別途保存しておくことを推奨する。

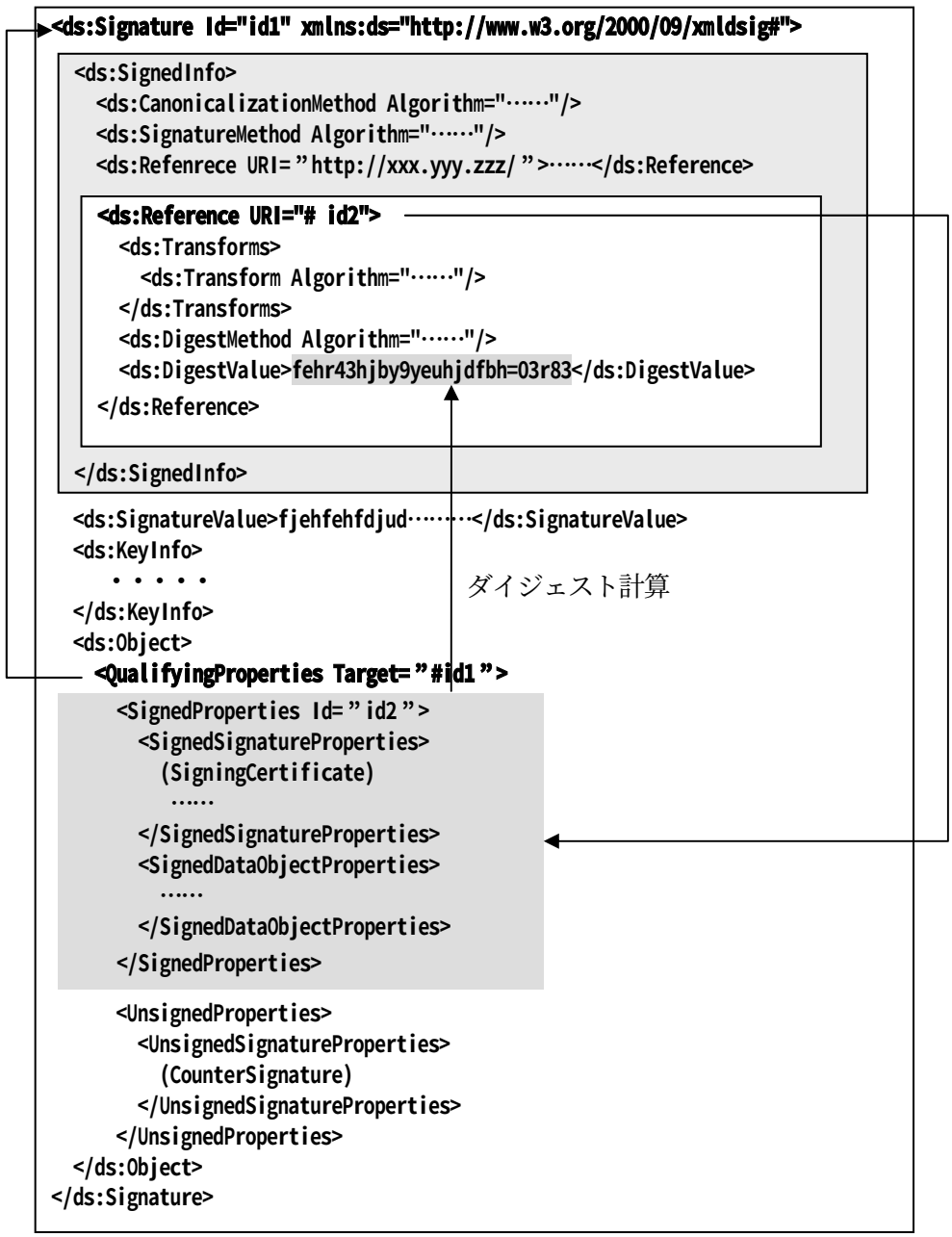
2.2 長期署名フォーマット

XAdES（V1.3.1）では、基本となる署名形式として"Basic Electronic Signature"（XAdES-BES）、"Explicit Policy based Electronic Signature"（XAdES-EPES）、Electronic Signature with Time（XAdES-T）および Electronic Signature with Complete Validation Data References（XAdES-C）の４つが定義されている。

2.2.1 XAdES の基本フォーマット

XAdES の基本となる署名フォーマットは、XMLDSIG の<ds:Object>要素内に必要な情報が追加された形をとる。リスト 2 に XAdES の基本フォーマットを示し、リスト中の要素について説明する。

リスト 2：XAdES-BES のフォーマット



(1) QualifyingProperties 要素

この要素は、<ds:Object>要素内に含められ長期保存に必要な要素を格納するコンテナの役割を果たす。リスト 3 に QualifyingProperties 要素の XMLSchema 定義を示す。

リスト 3：QualifyingProperties 要素の XMLSchema 定義

```
<xsd:element name="QualifyingProperties" type="QualifyingPropertiesType"/>
<xsd:complexType name="QualifyingPropertiesType">
  <xsd:sequence>
    <xsd:element name="SignedProperties"
                 type="SignedPropertiesType"   minOccurs="0"/>
    <xsd:element name="UnsignedProperties"
                 type="UnsignedPropertiesType" minOccurs="0"/>
  </xsd:sequence>
  <xsd:attribute name="Target" type="xsd:anyURI" use="required"/>
  <xsd:attribute name="Id"     type="xsd:ID"     use="optional"/>
</xsd:complexType>
```

QualifyingProperties 要素には、署名値の計算対象となる SignedProperties 要素と、署名値の計算対象にはならない UnsignedProperties 要素の二つの要素が含まれる。また、SignedProperties 要素は必ずひとつ存在しなくてはならない。Target 属性は必須属性であり、ds:Signature 要素の Id 属性を参照する必要がある。

(2) SignedProperties 要素

この要素は、XMLDSIG の署名値の計算対象となるよう、ds:Reference タグで参照される。この要素には署名計算の時に一つだけ SignedSignatureProperties 要素を含まなくてはならず、SignedDataObjectProperties 要素を含む場合もある。リスト 4 に SignedProperties 要素の XMLSchema 定義を示す。

リスト 4：SignedProperties 要素の XMLSchema 定義

```
<xsd:element name="SignedProperties" type="SignedPropertiesType" />
<xsd:complexType name="SignedPropertiesType">
  <xsd:sequence>
    <xsd:element name="SignedSignatureProperties"
                 type="SignedSignaturePropertiesType"/>
    <xsd:element name="SignedDataObjectProperties"
                 type="SignedDataObjectPropertiesType" minOccurs="0"/>
  </xsd:sequence>
  <xsd:attribute name="Id" type="xsd:ID" use="optional"/>
</xsd:complexType>
```

また、SignedProperties 要素を参照している ds:Reference 要素の Type 属性に以下の値をセットしなければならない。

http://uri.etsi.org/01903/V1.3.1#SignedProperties

(3) UnsignedProperties 要素

この要素は、署名されない特性を持つ。リスト 5 に UnsignedProperties 要素の XMLSchema 定義を示す。

リスト 5：UnsignedProperties 要素の XMLSchema 定義

```
<xsd:element name="UnsignedProperties" type="UnsignedPropertiesType" />
<xsd:complexType name="UnsignedPropertiesType">
  <xsd:sequence>
    <xsd:element name="UnsignedSignatureProperties"
                 type="UnsignedSignaturePropertiesType" minOccurs="0"/>
    <xsd:element name="UnsignedDataObjectProperties"
                 type="UnsignedDataObjectPropertiesType" minOccurs="0"/>
  </xsd:sequence>
  <xsd:attribute name="Id" type="xsd:ID" use="optional"/>
</xsd:complexType>
```

(4) SignedSignatureProperties 要素

この要素は、QualifyingProperties 要素の Target 属性で指定された XML 署名を限定する特性を子要素に持ち、XML 署名の署名対象として署名計算に含められる。リスト 6 に SignedSignatureProperties 要素の XMLSchema 定義を示す。

リスト 6：SignedSignatureProperties 要素の XMLSchema 定義

```
<xsd:element name="SignedSignatureProperties"
             type="SignedSignaturePropertiesType" />

<xsd:complexType name="SignedSignaturePropertiesType">
  <xsd:sequence>
    <xsd:element name="SigningTime"
                 type="xsd:dateTime" minOccurs="0"/>
    <xsd:element name="SigningCertificate"
                 type="CertIDListType" minOccurs="0"/>
    <xsd:element name="SignaturePolicyIdentifer"
                 type="SignaturePolicyIdentifierType" minOccurs="0"/>
    <xsd:element name="SignatureProductionPlace"
                 type="SignatureProductionPlaceType"  minOccurs="0"/>
    <xsd:element name="SignerRole"
                 type="SignerRoleType" minOccurs="0"/>
  </xsd:sequence>
  <xsd:attribute name="Id" type="xsd:ID" use="optional"/>
</xsd:complexType>
```

(5) UnsignedSignatureProperties 要素

この要素は、QualifyingProperties 要素の Target 属性で指定された XML 署名を限定する特性を子要素に持ち、XML 署名の署名対象として署名計算に含められない。リスト 7 に UnsignedSignatureProperties 要素の XMLSchema を示す。

リスト 7：UnsignedSignatureProperties 要素の XMLSchema 定義

```
<xsd:element name="UnsignedSignatureProperties"
             type="UnsignedSignaturePropertiesType"/>

<xsd:complexType name="UnsignedSignaturePropertiesType">
  <xsd:choice maxOccurs="unbounded">
    <xsd:element name="CounterSignature"
                 type="CounterSignatureType" />
    <xsd:element name="SignatureTimeStamp"
                 type="XAdESTimeStampType/>
    <xsd:element name="CompleteCertificateRefs"
                 type="CompleteCertificateRefsType"/>
    <xsd:element name="CompleteRevocationRefs"
                 type="CompleteRevocationRefsType"/>
    <xsd:element name="AttributeCertificateRefs"
                 type="CompleteCertificateRefsType"/>
    <xsd:element name="AttributeRevocationRefs"
                 type="CompleteRevocationRefsType"/>
    <xsd:element name="SigAndRefsTimeStamp"
                 type="XAdESTimeStampType"/>
    <xsd:element name="RefsOnlyTimeStamp"
                 type="XAdESTimeStampType"/>
    <xsd:element name="CertificateValues"
                 type="CertificateValuesType"/>
    <xsd:element name="RevocationValues"
                 type="RevocationValuesType"/>
    <xsd:element name="ArchiveTimeStamp"
                 type="XAdESTimeStampType"/>
  </xsd:choice>
  <xsd:attribute name="Id" type="xsd:ID" use="optional"/>
</xsd:complexType>
```

この要素には、以降で説明する様々な XAdES の形式で使われる長期保存に必要な要素が含められる。

(6) SignedDataObjectProperties 要素

この要素は、幾つかの署名されたデータオブジェクトを限定するような特性を含み、署名値の計算で計算対象とされる要素である。リスト 8 に SignedDataObjectProperties 要素の XMLSchema 定義を示す。

リスト 8：SignedDataObjectProperties 要素の XMLSchema 定義

```
<xsd:element name="SignedDataObjectProperties"
             type="SignedDataObjectPropertiesType"/>

<xsd:complexType name="SignedDataObjectPropertiesType">
  <xsd:sequence>
    <xsd:element name="DataObjectFormat"
                 type="DataObjectFormatType"
                 minOccurs="0" maxOccurs="unbounded"/>
    <xsd:element name="CommitmentTypeIndication"
                 type="CommitmentTypeIndicationType"
                 minOccurs="0" maxOccurs="unbounded"/>
    <xsd:element name="AllDataObjectsTimeStamp"
                 type="XAdESTimeStampType"
                 minOccurs="0" maxOccurs="unbounded"/>
    <xsd:element name="IndividualDataObjectsTimeStamp"
                 type="XAdESTimeStampType"
                 minOccurs="0" maxOccurs="unbounded"/>
  </xsd:sequence>
  <xsd:attribute name="Id" type="xsd:ID" use="optional"/>
</xsd:complexType>
```

(7) UnsignedDataObjectProperties 要素

この要素は、幾つかの署名されたデータオブジェクトを限定するような特性を含み、署名値の計算計算対象とされない要素である。リスト 9 に UnsignedDataObjectProperties 要素の XMLSchema 定義を示す。UnsignedDataObjectProperties については、ETSI TS101 703 V1.5.1（2003-12）では記述されていないが、将来の拡張性と完全性のブレを吸収するために定義する。

リスト 9：UnsignedDataObjectProperties 要素の XMLSchema 定義

```
<xsd:element name="UnsignedDataObjectProperties"
             type="UnsignedDataObjectPropertiesType" />

<xsd:complexType name="UnsignedDataObjectPropertiesType">
  <xsd:sequence>
    <xsd:element name="UnsignedDataObjectProperty"
                 type="AnyType" minOccurs="0" maxOccurs="unbounded"/>
  </xsd:sequence>
  <xsd:attribute name="Id" type="xsd:ID" use="optional"/>
</xsd:complexType>
```

2.2.2 データ型定義

(1) AnyType データ型

このデータ型は、要素の内容を特に規定したくない場合に用いる。このデータ型の要素の要素内容には、任意の要素やテキストなどを保持できる。また、任意の属性を制限なく追加する

ことが出来る。リスト 10 に AnyType データ型の XMLShema 定義を示す。

リスト 10：AnyType データ型の XMLSchema 定義

```
<xsd:complexType name="AnyType" mixed="true">
  <xsd:sequence minOccurs="0" maxOccurs="unbounded">
    <xsd:any namespace="##any" processContents="lax"/>
  </xsd:sequence>
  <xsd:anyAttribute namespace="##any"/>
</xsd:complexType>
```

(2) ObjectIdentifierType データ型

このデータ型は、オブジェクト識別子（OID）を格納するためのデータ型である。リスト 11 に ObjectIdentifierType データ型の XMLSchema 定義を示す。Identifier 要素では、ASN.1 におけるオブジェクトを識別する OID と XML のリソースを識別する URI の両方を指定することが出来る。

- XML リソースを指定する場合、Identifier 要素内に URI を記述する。Qualifier 属性は利用しない。
- ASN.1 で利用される OID でリソースを指定する場合は、URN の形式または URI としてエンコードした形で指定する。Qualifier 属性はどちらのエンコードが使われているかを示すために使われ、OIDAsURN、OIDAsURI のどちらかの値を取る。

Description 要素はオプションの要素で、オブジェクト識別子に関する説明文を格納する。また、DocumentationReferences 要素もオプションの要素で、オブジェクト識別子の追加説明への任意の個数の参照が含まれる。

リスト 11：ObjectIdentifier データ型の XMLSchema 定義

```
<xsd:complexType name="ObjectIdentifierType">
  <xsd:sequence>
    <xsd:element name="Identifier" type="IdentifierType"/>
    <xsd:element name="Description" type="xsd:string" minOccurs="0"/>
    <xsd:element name="DocumentationReferences"
                 type="DocumentationReferencesType" minOccurs="0"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="IdentifierType">
  <xsd:simpleContent>
    <xsd:extension base="xsd:anyURI">
      <xsd:attribute name="Qualifier" type="QualifierType"  use="optional"/>
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>
<xsd:simpleType name="QualifierType">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="OIDAsURI"/>
    <xsd:enumeration value="OIDAsURN"/>
  </xsd:restriction>
</xsd:simpleType>

<xsd:complexType name="DocumentationReferencesType">
  <xsd:sequence maxOccurs="unbounded">
    <xsd:element name="DocumentationReference" type="xsd:anyURI"/>
  </xsd:sequence>
</xsd:complexType>
```

(3) EncapsulatedPKIDataType データ型

このデータ型は、ASN.1 でエンコードされたデータを XML 文書に格納するために使われる。例えば、X.509 証明書や、失効リスト、属性証明書やタイムスタンプのデータを格納するために使われる。格納時は、これらのデータを base64 でエンコードして格納する。Encoding 属性には、ASN.1 データのエンコード方法を URI で記述する。記述できる URI を表 付 2.2-1 に示す。リスト 12 に EncapsulatedPKIDataType データ型の XMLSchema 定義を示す。

表 付 2.2-1 ASN.1 データのエンコード方法に関する URI

エンコード方法	URI
DER	http://uri.etsi.org/01903#DER
BER	http://uri.etsi.org/01903#BER
CER	http://uri.etsi.org/01903#CER
PER	http://uri.etsi.org/01903#PER
XER	http://uri.etsi.org/01903#XER

リスト 12：EncapsulatedPKIDataType データ型の XMLSchema 定義

```
<xsd:complexType name="EncapsulatedPKIDataType">
  <xsd:simpleContent>
    <xsd:extension base="xsd:base64Binary">
      <xsd:attribute name="Id" type="xsd:ID" use="optional"/>
      <xsd:attribute name="Encoding" type="xsd:anyURI"
        use="optional"/>
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>
```

(4) XAdESTimeStampType データ型

タイムスタンプを格納するデータ XAdESTimeStampType データ型は GenericTimeStampType データ型の派生として定義される。GenericTimeStampType データ型からは、XAdESTimeStampType データ型と OtherTimeStampType データ型が派生して定義されるが、OtherTimeStampType データ型は非推奨である。リスト 13 に GenericTimeStampType データ型の XMLSchema 定義を示す。また、それから派生される XAdESTimeStampType データ型の XMLSchema 定義をリスト 14 に示す。

リスト 13：GenericTimeStampType データ型の XMLSchema 定義

```
<xsd:element name=”Include” type=”IncludeType”>
<xsd:complexType name="IncludeType">
  <xsd:attribute name="URI" type="xsd:anyURI" use="required"/>
  <xsd:attribute name="referencedData" type="xsd:boolean" use="optional"/>
</xsd:complexType>

<xsd:element name="ReferenceInfo" type="ReferenceInfoType"/>
<xsd:complexType name="ReferenceInfoType">
  <xsd:sequence>
    <xsd:element ref="ds:DigestMethod"/>
    <xsd:element ref="ds:DigestValue"/>
  </xsd:sequence>
  <xsd:attribute name="Id" type="xsd:ID" use="optional"/>
  <xsd:attribute name="URI" type="xsd:anyURI" use="optional"/>
</xsd:complexType>

<xsd:complexType name="GenericTimeStampType" abstract="true">
  <xsd:sequence>
    <xsd:choice minOccurs="0">
      <xsd:element ref="Include" maxOccurs="unbounded"/>
      <xsd:element ref="ReferenceInfo" maxOccurs="unbounded"/>
    </xsd:choice>
    <xsd:element ref="ds:CanonicalizationMethod" minOccurs="0"/>
    <xsd:choice maxOccurs="unbounded">
      <xsd:element name="EncapsulatedTimeStamp"
                   type="EncapsulatedPKIDataType"/>
      <xsd:element name="XMLTimeStamp" type="AnyType"/>
    </xsd:choice>
  </xsd:sequence>
  <xsd:attribute name="Id" type="xsd:ID" use="optional"/>
</xsd:complexType>
```

リスト 14：XAdESTimeStampType データ型の XMLSchema 定義

```
<xsd:element name="XAdESTimeStamp" type="XAdESTimeStampType"/>

<xsd:complexType name="XAdESTimeStampType">
  <xsd:complexContent>
    <xsd:restriction base="GenericTimeStampType">
      <xsd:sequence>
        <xsd:element ref="Include"  minOccurs="0" maxOccurs="unbounded"/>
        <xsd:element ref="ds:CanonicalizationMethod" minOccurs="0"/>
        <xsd:choice maxOccurs="unbounded">
          <xsd:element name="EncapsulatedTimeStamp"
                       type="EncapsulatedPKIDataType"/>
          <xsd:element name="XMLTimeStamp" type="AnyType"/>
        </xsd:choice>
      </xsd:sequence>
      <xsd:attribute name="Id" type="xsd:ID" use="optional"/>
    </xsd:restriction>
  </xsd:complexContent>
</xsd:complexType>
```

XAdESTimeStampType データ型の要素にタイムスタンプトークンが含められることになるが、TSA へ送るダイジェスト値の算出方法に関しては 2 通りの方法を定義する。

- Implicit Mode

  XAdESTimeStampType の要素の中に Include 要素がない場合、TSA へ送付するダイジェスト値の計算対象となる XAdES 要素の指定は暗黙的に実行される。処理手順は以下のようになる。

  Signed Properties の場合

  1. 各 XAdESTimeStampType 型の要素の仕様として指示されている要素や署名された要素内容を取り出す。
  2. 取り出した要素や署名された要素内容に対して、もし対象としている XAdESTimeStampType 型の要素内に ds:Canonicalization 要素が存在すれば、それに従った正規化を行う。ds:Canonicalization 要素が存在しなければ、XMLDSIG で標準的な正規化手法で正規化する。
  3. 処理された各データを連結する。

  Unsigned Properties の場合

  1. タイムスタンプトークンを含むプロパティより前に現れる、UnsignedSignatureProperties の子要素を全て取り出す。
  2. 取り出した要素のそれぞれについて、もし対象としている XAdESTimeStampType 型の要素内に ds:Canonicalization 要素が存在すれば、それに従った正規化を行う。ds:Canonicalization 要素が存在しなければ、XMLDSIG で標準的な正規化手法で正規化する。
  3. 処理された各データを連結する。

どちらの場合も連結されたデータをもとにダイジェスト値を計算し、TSA に送付する。
また、以下の要素はこのモードで計算される。

- SignatureTimeStamp 要素
- RefsOnlyTimeStamp 要素
- SigAndRefsTimeStamp 要素

- Include Mode

  XAdESTimeStampType の要素の中に Include 要素がある場合、TSA へ送付するダイジェスト値の計算対象となる XAdES 要素の指定は明示的に処理される。Include 要素の URI 属性は、TSA へ送付するダイジェスト計算の計算対象を参照する。ds:Reference 要素自体を参照する場合は、XAdESTimeStampType の要素に referencedData 属性が存在する場合がある。その値が ture の場合は、XMLDSIG の処理モデル従って ds:Reference 要素を処理した結果を元にタイムスタンプを計算することになる。もし、referencedData 属性が存在しないか、値が false である場合は、ds:Refernece 要素自体を計算対象とする。各 Include 要素は以下のような手順に従って処理される。

  1. URI 属性で参照されているデータを取り出す。
  2. 取り出したデータが ds:Reference 要素で referneceData 属性が true の場合、取り出された ds:Reference 要素を XMLDSIG の処理モデルに従って処理しその結果を得る。referencedData 属性が存在しない場合、または値が false である場合は ds:Reference 要素自体を計算対象として取り出す。
  3. 取り出した結果データが XML 形式の場合は、ds:Canonicalization 要素に従って正規化を行う。ds:Canonicalization 要素がない場合は、XMLDSIG で使われる標準的な正規化手法が使われる。
  4. 計算結果を処理済の Include 要素の計算結果につなぎ合わせる。

  連結されたデータをもとにダイジェスト値を計算し、TSA に送付する。また、以下の要素はこのモードで計算される。

  - AllDataObjectsTimeStamp 要素
  - IndividualDataObjectsTimeStamp 要素

なお、ArchiveTimeStamp 要素については両方のモードが利用される。

2.2.3 Basic electronic signature（XAdES-BES）

XAdES-BES では、以下のどちらか一方が必須となる。

- <SignedSignatureProperties>要素の中に<SigningCertificate>要素を含み、署名値計算の対象として署名計算に含める。
- <ds:Signature>要素に<ds:KeyInfo>を含み、署名値計算の対象として署名計算に含める。

以降、それぞれについて説明する。

(1) SigningCertificate 要素

この要素には、署名者証明書のダイジェスト値と証明書への参照を含めなければならない。また、信頼点までの証明書チェーンを構成する証明書のダイジェスト値とシリアル番号を含むことも出来る。ただし、署名ポリシで規定されている場合には、信頼点までの証明書チェーンを構成する証明書のダイジェスト値と発行者のシリアル番号を含まなければならない。リスト 15 に SigningCertificate 要素の XMLSchema 定義を示す。なお、ここで言う証明書の参照とは、証明書を発行した認証局の DN と証明書のシリアル番号を含む ds:X509IssuerSerialType 型の要素<IssuerSerial>で表現される。

この要素は、SigndProperties の含まれる他の要素とともに署名値の計算対象として署名計算に含められる。

これらの要素は、Simple sustitution 攻撃を防ぐために用いる。

リスト 15：SigningCertificate 要素の XMLSchema 定義

```
<xsd:element name="SigningCertificate" type="CertIDListType"/>

<xsd:complexType name="CertIDListType">
  <xsd:sequence>
    <xsd:element name="Cert" type="CertIDType" maxOccurs="unbounded"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="CertIDType">
  <xsd:sequence>
    <xsd:element name="CertDigest" type="DigestAlgAndValueType"/>
    <xsd:element name="IssuerSerial" type="ds:X509IssuerSerialType"/>
  </xsd:sequence>
  <xsd:attribute name="URI" type="xsd:anyURI" use="optional"/>
</xsd:complexType>

<xsd:complexType name="DigestAlgAndValueType">
  <xsd:sequence>
    <xsd:element ref="ds:DigestMethod"/>
    <xsd:element ref="ds:DigestValue"/>
  </xsd:sequence>
</xsd:complexType>
```

(2) ds:KeyInfo 要素

SigningCertificage 要素が存在しない、もしくは署名計算の対象となっていない場合は、ds:KeyInfo 要素が必須であり、以下の条件を満たさなければならない。

- ds:KeyInfo は、署名者証明書を含む ds:X509Data 要素を含まなければならない
- ds:KeyInfo は、信頼点までの証明書チェーンを構成する証明書も含む場合がある。
- ds:SignedInfo 要素の ds:Reference 要素で ds:KeyInfo を参照することにより、署名の計算対象として署名値の計算に含まれなければならない。

2.2.4 Explicit policy electronic signatures（XAdES-EPES）

XAdES-EPES は、XAdES で基本となる形式の 1 つで、XAdES-BES に署名ポリシに関する要素である SignaturePolicyIdentifier 要素を追加したものである。SignaturePolicyIdentifier 要素は、SignedSignatureProperties 要素に追加され XAdES-EPES では必須要素である（リスト 6)。この属性は、作成者の署名で保護される。

(1) SignaturePolicyIdentifier 要素

リスト 16 に SignaturePolicyIdentifier 要素の XMLSchema 定義を示す。

リスト 16：SignaturePolicyIdentifier 要素の XMLSchema 定義

```
<xsd:element name="SignaturePolicyIdentifier"
               type="SignaturePolicyIdentifierType"/>

<xsd:complexType name="SignaturePolicyIdentifierType">
  <xsd:choice>
    <xsd:element name="SignaturePolicyId"
                 type="SignaturePolicyIdType"/>
    <xsd:element name="SignaturePolicyImplied"/>
  </xsd:choice>
</xsd:complexType>

<xsd:complexType name="SignaturePolicyIdType">
  <xsd:sequence>
    <xsd:element name="SigPolicyId"
                 type="ObjectIdentifierType"/>
    <xsd:element ref="ds:Transforms" minOccurs="0"/>
    <xsd:element name="SigPolicyHash"
                 type="DigestAlgAndValueType"/>
    <xsd:element name="SigPolicyQualifiers"
                 type="SigPolicyQualifiersListType"
                 minOccurs="0"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="SigPolicyQualifiersListType">
  <xsd:sequence>
    <xsd:element name="SigPolicyQualifier"
                 type="AnyType"
                 maxOccurs="unbounded"/>
  </xsd:sequence>
</xsd:complexType>
```

2.2.5 XAdES-BES でオプショナルな要素

XAdES-BES では、SingedSigantureProperties 要素に以下の要素を含めることが出来る。

- SigningTime
- SignatureProductionPlace
- SignerRole

SignedDataObjectProperties 要素に以下の要素を含めることが出来る。

- DataObjectFormat
- CommitmentTypeIndication
- AllDataObjectsTimeStamp
- IndividualDataObjectsTimeStamp

UnsignedSignatureProperties 要素に以下の要素を含めることができる。

- CounterSignature

以降、それぞれを簡単に説明する。

(1) SigningTime 要素

この要素は、署名者が署名を実行した時刻を表す要素である。内部のデータ型は、W3CRecommendation "XML Schema Part2:Datatypes"で定義される xsd:dateTime 型となる。この要素は、ひとつの署名文書のたかだか 1 つしか含まれない。リスト 17 に SigningTime の XMLSchema 定義を示す。

リスト 17：SigningTime 要素の XMLSchema 定義

```
<xsd:element name="SigningTime" type="xsd:dateTime"/>
```

(2) SignatureProductionPlace 要素

この要素は、署名が生成された場所を示す。リスト 18 に SignatureProductionPlace 要素の XMLSchema 定義を示す。

リスト 18：SignatureProductionPlace 要素の XMLSchema 定義

```
<xsd:element name="SignatureProductionPlace"
             type="SignatureProductionPlaceType"/>

<xsd:complexType name="SignatureProductionPlaceType">
  <xsd:sequence>
    <xsd:element name="City" type="xsd:string" minOccurs="0"/>
    <xsd:element name="StateOrProvince" type="xsd:string" minOccurs="0"/>
    <xsd:element name="PostalCode" type="xsd:string" minOccurs="0"/>
    <xsd:element name="CountryName" type="xsd:string" minOccurs="0"/>
  </xsd:sequence>
</xsd:complexType>
```

(3) SignerRole 要素

契約によっては、ある特定のポジションの人によって署名されたかどうかのみが重要である場合がある。この要素は、そのような場合に対応するために署名者の役割を表す要素である。リスト 19 に SignerRole 要素のの XMLSchema 定義を示す。また、署名者の役割を記述するために以下の 2 つの方法を定義する。

- 署名者自らが主張する役割名
  ClaimedRoles 要素に役割名を含める

- 認められた役割を含む属性証明書
  CertifiedRolses 要素に属性証明書を格納する

リスト 19：SignerRole 要素の XMLSchema 定義

```
<xsd:element name="SignerRole" type="SignerRoleType"/>
<xsd:complexType name="SignerRoleType">
  <xsd:sequence>
    <xsd:element name="ClaimedRoles"
                 type="ClaimedRolesListType" minOccurs="0"/>
    <xsd:element name="CertifiedRoles"
                 type="CertifiedRolesListType" minOccurs="0"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="ClaimedRolesListType">
  <xsd:sequence>
    <xsd:element name="ClaimedRole"
                 type="AnyType" maxOccurs="unbounded"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="CertifiedRolesListType">
  <xsd:sequence>
    <xsd:element name="CertifiedRole"
                 type="EncapsulatedPKIDataType" maxOccurs="unbounded"/>
  </xsd:sequence>
</xsd:complexType>
```

(4) DataObjectFormat 要素

この要素は、署名されたデータオブジェクトのデータフォーマットを記述する要素である。この要素は、もし署名されたデータ内に暗黙のうちにデータフォーマットが含まれておらず、署名されたデータを検証のため人間のユーザに提供する場合は必須である。この要素は、署名されたデータオブジェクト毎に追加することができる。リスト 20 に DataObjectFormat 要素の XMLShema 定義を示す。

リスト 20：DataObjectFormat 要素の XMLSchema 定義

```
<xsd:element name="DataObjectFormat" type="DataObjectFormatType"/>

<xsd:complexType name="DataObjectFormatType">
  <xsd:sequence>
    <xsd:element name="Description"
                 type="xsd:string" minOccurs="0"/>
    <xsd:element name="ObjectIdentifier"
                 type="ObjectIdentifierType" minOccurs="0"/>
    <xsd:element name="MimeType"
                 type="xsd:string" minOccurs="0"/>
    <xsd:element name="Encoding"
                 type="xsd:anyURI" minOccurs="0"/>
  </xsd:sequence>
  <xsd:attribute name="ObjectReference" type="xsd:anyURI" use="required"/>
</xsd:complexType>
```

ObjectReference 属性は必須属性であり、特定したいデータオブジェクトの対応する ds:Signature 要素内の ds:Reference 要素を参照する。その他に以下のような情報を伝達することが出来る。

- Description 要素で、署名されたデータオブジェクトに関するテキスト情報を記述できる。
- ObjectIdentifier 要素で、署名されたデータオブジェクトのタイプを指定できる。
- MimeType 要素で、署名されたデータオブジェクトの MIME type を指定できる。
- Encoding 要素で、署名されたデータオブジェクトの enoding フォーマットを指定できる。

Description 要素、ObjectIdentifier 要素および MimeType 要素のうち少なくとも 1 つが DataObjectFomat 要素中に存在しなくてはいけない。

(5) CommitmentTypeIndication 要素

リスト 21 に CommitmentTypeIndication 要素の XMLSchema 定義を示す。表 付 2.2-2 にコミットメントとその内容の一覧を示す。

リスト 21：CommitmentTypeIndication 要素の XMLSchema 定義

```
<xsd:element name="CommitmentTypeIndication"
             type="CommitmentTypeIndicationType"/>

<xsd:complexType name="CommitmentTypeIndicationType">
  <xsd:sequence>
    <xsd:element name="CommitmentTypeId"
                 type="ObjectIdentifierType"/>
    <xsd:choice>
      <xsd:element name="ObjectReference"
                   type="xsd:anyURI" maxOccurs="unbounded"/>
      <xsd:element name="AllSignedDataObjects"/>
    </xsd:choice>
    <xsd:element name="CommitmentTypeQualifiers"
                 type="CommitmentTypeQualifiersListType" minOccurs="0"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="CommitmentTypeQualifiersListType">
  <xsd:sequence>
    <xsd:element name="CommitmentTypeQualifier"
                 type="AnyType" minOccurs="0" maxOccurs="unbounded"/>
  </xsd:sequence>
</xsd:complexType>
```

表 付2.2-2 コミットメントの種別とその内容の一覧

コミットメント	内容
Proof of origin	署名者がその文書を生成したこと、承認したこと、送信したことを示す
Proof of receipt	署名者がその文書を受け取ったことを示す
Proof of delivery	TSP（信頼できるサービスプロバイダがその文書をアクセスできる状態でローカルなストアに置いたことを提示したことを示す
Proof of sender	その提示をしたエンティティがその文書を送信したことを示す（生成したものでなくてもよい）
Proof of approval	署名者がその文書を承認したことを示す
Proof of creation	署名者がその文書を生成したことを示す（承認したり送信したりする必要はない）

(6) AllDataObjectsTimeStamp 要素

この要素は、署名計算の実行前に署名対象データに付与されたタイムスタンプを格納する。リスト 22 に AllDataObjectsTimeStamp 要素の XMLSchema 定義を示す。

リスト 22：AllDataObjectsTimeStamp 要素の XMLSchema 定義

```
<xsd:element name="AllDataObjectsTimeStamp" type="XAdESTimeStampType"/>
```

ds:SignedInfo に含まれていて SignedProperties 要素以外に署名者が署名したい全ての ds:Referenece を元にタイムスタンプが計算される。この要素を生成するアプリケーションは、SignedPropeties 要素で参照しているものを除く全ての ds:Refernece 要素について Include 要素を生成しなければならない。また、各 Include 要素の referencedData 属性は ture でなくてはならない。

(7) IndividualDataObjectsTimeStamp 要素

この要素は、署名計算の実行前に個別のデータに付与されたタイムスタンプを格納する。リスト 23 に IndividualDataObjectsTimeStamp 要素の XMLSchema 定義を示す。

リスト 23：IndividualDataObjectsTimeStamp 要素の XMLSchema 定義

```
<xsd:element name="IndividualDataObjectsTimeStamp" type="XAdESTimeStampType"/>
```

ds:SignedInfo に含まれている任意の ds:Reference 要素を元にタイムスタンプが計算される。ただし、SignedProperties を参照している ds:Reference 要素を対象として含むことは出来ない。アプリケーションは、SignedPropeties 要素で参照しているものを除く ds:Refernece 要素で対象とするものついては Include 要素を生成しなければならない。また、各 Include 要素の referencedData 属性は ture でなくてはならない。ひとつの XAdES の文書の中にこの要素を複数含めることが出来る。

(8) CounterSignature 要素

CounterSignature 要素は、UnsignedSingatureProperties 要素内に格納される。リスト 24 に CounterSignature 要素の XMLSchema 定義を示す。

リスト 24：CounterSignater 要素の XMLSchema 定義

```
<xsd:element name="CounterSignature" type="CounterSignatureType" />

<xsd:complexType name="CounterSignatureType">
  <xsd:sequence>
    <xsd:element ref="ds:Signature"/>
  </xsd:sequence>
</xsd:complexType>
```

2.2.6 Electronic signature with time（XAdES-T）

XAdES-T は、デジタル署名の存在時刻を確定するために電子署名文書中の署名値(ds:Signature 要素内の ds:SignedInfo 要素内の SignatureValue 要素)に対して TSA から取得したタイムスタンプトークンを追加したものである。署名値から生成したタイムスタンプトークンは、署名の存在

時刻とともに、電子データの存在時刻も証明することとなる。XAdES-T の形式はは、XAdES-BES または XAdES-EPES の UnSignedSignatureProperties 要素に SignatureTimeStamp 要素を追加した形式である（UnsignedSignatureProperties 要素の XMLShema 定義はリスト 7 を参照のこと）。

(1) SignatureTimeStamp 要素

この要素は、ds:SignatureValue 要素に対して付与したタイムスタンプを格納する。1 つの署名に対して複数の異なった TSA から取得したタイムスタンプ格納するために、1 つの XAdES 文書に複数の SignatureTiemStamp 要素を格納することが出来る。リスト 25 に SignatureTimeStamp 要素の XMLSchema 定義を示す。

リスト 25：SignatureTimeStamp 要素の XMLSchema 定義

```
<xsd:element name="SignatureTimeStamp" type="XAdESTimeStampType"/>
```

この要素に含まれるタイムスタンプトークンの計算は、Implicit Mode で実行される。具体的には、ds:SignatureValue 要素が、TSA に送付されるダイジェスト値の計算の入力となる。

タイムスタンプトークン自体の検証情報（認証パス及び失効情報）は、次のいずれかに格納する。

1) タイムスタンプトークン内に含める
2) 署名者証明書の検証情報と同じ場所（Complete Certificate Refs, Complete RevocationRefs, CertificateValues, RevocationValues）

2.2.7 Electronic signature with complete validation data references（XAdES-C）

XAdES-C は、XAdES-T に CompleteCertificateRefs 要素と CompleteRevocationRefs 要素を加えたものとなる。CompleteCertificateRefs 要素は、署名者証明書の検証に使われる証明書チェーン上の全ての証明書（署名者証明書を除く）への参照をもつ。CompleteRevocationRefs 要素は、署名者証明書および CA 証明書の検証に必要な全ての失効情報への参照が格納される。

なお、後述する XAdES-A を構成するときは、CompleteCertificateRefs, CompleteRevocationRefs はオプション要素となる。

(1) CompleteCertificateRefs 要素

この要素は、XAdES 文書中に高々ひとつだけ含めることが出来る。リスト 26 に CompleteCertificateRefs 要素の XMLSchema 定義を示す。

リスト 26：CompleteCertificateRefs 要素の XMLSchema 定義

```
<xsd:element name="CompleteCertificateRefs"
             type="CompleteCertificateRefsType"/>

<xsd:complexType name="CompleteCertificateRefsType">
  <xsd:sequence>
    <xsd:element name="CertRefs" type="CertIDListType" />
  </xsd:sequence>
  <xsd:attribute name="Id" type="xsd:ID" use="optional"/>
</xsd:complexType>
```

(2) CompleteRevocationRefs 要素

リスト 27 に CompleteRevocationRefs 要素の XMLSchema 定義を示す。

リスト 27：CompleteRevocationRefs 要素の XMLSchema 定義

```
<xsd:element name="CompleteRevocationRefs" type="CompleteRevocationRefsType"/>

<xsd:complexType name="CompleteRevocationRefsType">
  <xsd:sequence>
    <xsd:element name="CRLRefs" type="CRLRefsType" minOccurs="0"/>
    <xsd:element name="OCSPRefs" type="OCSPRefsType" minOccurs="0"/>
    <xsd:element name="OtherRefs" type="OtherCertStatusRefsType"
                 minOccurs="0"/>
  </xsd:sequence>
  <xsd:attribute name="Id" type="xsd:ID" use="optional"/>
</xsd:complexType>

<xsd:complexType name="CRLRefsType">
  <xsd:sequence>
    <xsd:element name="CRLRef" type="CRLRefType" maxOccurs="unbounded"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="CRLRefType">
  <xsd:sequence>
    <xsd:element name="DigestAlgAndValue" type="DigestAlgAndValueType"/>
    <xsd:element name="CRLIdentifier" type="CRLIdentifierType" minOccurs="0"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="CRLIdentifierType">
  <xsd:sequence>
    <xsd:element name="Issuer" type="xsd:string"/>
    <xsd:element name="IssueTime" type="xsd:dateTime" />
    <xsd:element name="Number" type="xsd:integer"  minOccurs="0"/>
  </xsd:sequence>
  <xsd:attribute name="URI" type="xsd:anyURI" use="optional"/>
</xsd:complexType>

<xsd:complexType name="OCSPRefsType">
  <xsd:sequence>
    <xsd:element name="OCSPRef" type="OCSPRefType" maxOccurs="unbounded"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="OCSPRefType">
  <xsd:sequence>
    <xsd:element name="OCSPIdentifier" type="OCSPIdentifierType"/>
    <xsd:element name="DigestAlgAndValue"
      type="DigestAlgAndValueType"
      minOccurs="0"/>
  </xsd:sequence>
</xsd:complexType>
```

```
<xsd:complexType name="OCSPIdentifierType">
  <xsd:sequence>
    <xsd:element name="ResponderID" type="xsd:string"/>
    <xsd:element name="ProducedAt" type="xsd:dateTime"/>
  </xsd:sequence>
  <xsd:attribute name="URI" type="xsd:anyURI" use="optional"/>
</xsd:complexType>

<xsd:complexType name="OtherCertStatusRefsType">
  <xsd:sequence>
    <xsd:element name="OtherRef" type="AnyType"
      maxOccurs="unbounded"/>
  </xsd:sequence>
</xsd:complexType>
```

ETSI TS 101 903 V1.3.1（2005-05）では、署名の中に属性証明書が含まれる場合は、AttributeCertifcateRefs 要素と AttributeRevocationRefs 要素も加えることが出来るように定義されている。しかし、本書の効果的かつ効率的（必要最低限の）プロファイルを提示するという方針に従い、AttributeCertificate には言及しないこととする。

2.2.8 Extended signatures with time forms（XAdES-X）

XAdES-X は、将来の CA の危殆化に備えたり、検証データの完全性を確保したり検証データが入手困難になることに備えるため、XAdES-C を拡張したものである。拡張の仕方により 2 種類プロファイルを定義する。

なお、後述する XAdES-A 形式を構成する場合、XAdES-X として追加される SigAndRefsTimeStamp 要素および RefsOnlyTimeStamp 要素はオプション要素である。

- XAdES-X type1
  XAdES-C 全体に対するタイムスタンプを取得して追加するもので、具体的には UnsingedSignatureProperties 要素に SigAndRefsTimeStamp 要素を追加したものである（UnsignedSignatureProperties 要素の XMLSchema 定義はリスト 7 を参照のこと）。複数の TSP からタイムスタンプを取得する考慮して、複数の SigAndRefsTimeStamp 要素を追加することができるよう定義する。タイムスタンプを取得するために TSP に送信するハッシュ値は、SignatureValue 要素、SignatureTimeStamp 要素、CompleteCertificateRefs 要素および CompleteRevocationRefs 要素を元に計算する。

- XAdES-X type2
  検証情報のリファレンスのみに対するタイムスタンプを取得して追加するもので、具体的には UnsignedSignatureProperties 要素に RefsOnlyTimeStamp 要素を追加したものである（UnsignedSignatureProperties 要素の XMLSchema 定義はリスト 7 を参照のこと）。複数の TSP からタイムスタンプを取得する考慮して、複数の RefsOnlyTimeStamp 要素を追加する

ことができるよう定義する。タイムスタンプを取得するために TSP に送信するハッシュ値は、CompleteCertificateRefs 要素および CompleteRevocationRefs 要素を元に計算する。

(1) SigAndRefsTimeStamp 要素

この要素に含まれるタイムスタンプトークンの計算は、Implicit Mode で実行される。具体的には、ds:SignatureValue 要素およびすべての SignatureTimeStamp 要素を正規化して連結する。次に、CompleteCertificateRefs 要素、CompleteRevocationRefs 要素をそれぞれ正規化し、XAdES 文書内の出現順序に従って連結する。連結した結果が TSA に送付されるダイジェスト値の計算の入力となる。リスト 28 に SigAndRefsTimeStamp 要素の XMLSchema 定義を示す。

リスト 28：SigAndRefsTimeStamp 要素の XMLSchema 定義

```
<xsd:element name="SigAndRefsTimeStamp" type="XAdESTimeStampType"/>
```

(2) RefsOnlyTimeStamp 要素

この要素には、CompleteCertificateRefs 要素、CompleteRevocationRefs 要素を連結したデータを下に計算したタイムスタンプが格納される。この要素に含まれるタイムスタンプトークンの計算は、Implicit Mode で実行される。具体的には、CompleteCertificateRefs 要素、CompleteRevocationRefs 要素をそれぞれ正規化し、XAdES 文書内の出現順序に従って連結する。連結した結果が TSA に送付されるダイジェスト値の計算の入力となる。リスト 29 に RefsOnlyTimeStamp 要素の XMLSchema 定義を示す。

リスト 29：RefsOnlyTimeStamp 要素の XMLSchema 定義

```
<xsd:element name="RefsOnlyTimeStamp" type="XAdESTimeStampType"/>
```

2.2.9 Extended long electronic signatures with time（XAdES-X-L）

XAdES-X-L は、XAdES-X type1 か type2 のいずれかに対して、UnsingeSignatureProperties 要素に CertificateValues 要素と RevocationValues 要素を加えたものである（UnsignedSignatureProperties 要素の XMLSchema 定義はリスト 7 を参照のこと）。

なお、後述する XAdES-A 形式を構成する場合、XAdES-X type1 や type2 に対して XAdES-X-L を構成する必要はなく、CeritificateValues 要素と RevocationValues 要素があれば良い。

(1) CertificateValues 要素

この要素は、署名者証明書および CompleteCertificateRefs 要素で参照される証明書チェーンを含まなければならない。ただし、ds:Signature 要素内の ds:KeyInfo 要素に既に含まれている証明書は CertificateValues 要素内に含む必要はない。リスト 30 に CertifiCateValues 要素の XMLSchema 定義を示す。

リスト 30：CertificateValues 要素の XMLSchema 定義

```
<xsd:element name="CertificateValues" type="CertificateValuesType"/>

<xsd:complexType name="CertificateValuesType">
    <xsd:choice minOccurs="0" maxOccurs="unbounded">
        <xsd:element name="EncapsulatedX509Certificate"
                     type="EncapsulatedPKIDataType"/>
        <xsd:element name="OtherCertificate" type="AnyType"/>
    </xsd:choice>
    <xsd:attribute name="Id" type="xsd:ID" use="optional"/>
</xsd:complexType>
```

(2) RevocationValues 要素

この要素には、署名検証に必要な証明書の証拠情報が格納される。この要素は一つの XAdES 署名文書について高々一つしか存在しない。リスト 31 に RevocationValues 要素の XMLSchema 定義を示す。

リスト 31：RevocationValues 要素の XMLSchema 定義

```
<xsd:element name="RevocationValues" type="RevocationValuesType"/>

<xsd:complexType name="RevocationValuesType">
  <xsd:sequence>
    <xsd:element name="CRLValues" type="CRLValuesType" minOccurs="0"/>
    <xsd:element name="OCSPValues" type="OCSPValuesType" minOccurs="0"/>
    <xsd:element name="OtherValues" type="OtherCertStatusValuesType"
                 minOccurs="0"/>
  </xsd:sequence>
  <xsd:attribute name="Id" type="xsd:ID"  use="optional"/>
</xsd:complexType>

<xsd:complexType name="CRLValuesType">
  <xsd:sequence>
     <xsd:element name="EncapsulatedCRLValue"
                  type="EncapsulatedPKIDataType"
                  maxOccurs="unbounded"/>
  </xsd:sequence>
</xsd:complexType>


<xsd:complexType name="OCSPValuesType">
  <xsd:sequence>
      <xsd:element name="EncapsulatedOCSPValue"
                   type="EncapsulatedPKIDataType" maxOccurs="unbounded"/>
  </xsd:sequence>
</xsd:complexType>
```

2.2.10 Archival electronic signatures（XAdES-A）

電子署名の長期保存のためには、署名対象文書、デジタル署名、タイムスタンプおよび検証情報全体をタイムスタンプや耐タンパな仕組みで保護する必要がある。長期署名フォーマットでは、アーカイブタイムスタンプによりこれを実現する。このとき、XAdES-C や XAdES-X で利用した検証情報へのリファレンスや検証情報のリファレンスへのタイムスタンプは必要なくなり、検証情報そのものを CertificateValues や RevocationValues で確保し、アーカイブタイムスタンプを付与すればよい。

また、電子署名の検証可能期間を極めて長くしようとしたとき、タイムスタンプの署名の危殆化や TSA 証明書の有効期限切れが発生しうるため、タイムスタンプの署名を複数回重ねることが要求されることがある。このとき、ArchiveTimeStamp 要素が用いられる。このタイムスタンプは期間を置いて繰り返される。

(1) ArchiveTimeStamp 要素

この要素は、アーカイブタイムスタンプが格納される。この要素に含まれるタイムスタンプの含まれるハッシュ値の計算方法は以下のようになる。

1. ArchiveTimeStamp 要素内に、ds:SignedInfo 要素内の ds:Reference 毎に Include 要素を生成する。Inculde 要素の URI 属性では、それぞれ対応する ds:Reference 要素を参照する。また、referneceData 属性は ture でなければならない。
2. 以下の要素を取り出す
   - ds:SignedInfo
   - ds:SignatureValue
   - ds:KeyInfo
3. XAdES 文書中の出現順序に従い以下の要素を取り出す
   - XAdES 文書中にある SignatureTimeStamp 要素
   - 存在した場合は、CounterSignatureProperties 要素
   - 存在した場合は、CompleteCertificateRefs 要素
   - 存在した場合は、CompleteRevocationRefs 要素
   - CertificateValues 要素。存在しない場合は、追加しなくてはならない
   - RevocationValues 要素。存在しない場合は、追加しなくてはならない
   - SigAndRefsTimeStamp 要素
   - RefsOnlyTimeStamp 要素
   - 既に存在する ArchiveTimeStamp 要素
   - QualifyingProperties を含まず、ds:Reference で参照されていないような ds:Object 要素
4. 取り出したすべての要素について正規化を行い、その結果を連結してハッシュ計算の入力とする。

リスト 32 に ArchiveTimeStamp 要素の XMLScheme 定義を示す。同時に複数の TSA へタイムス

タンプのリクエストを送信した場合、得られる複数のタイムスタンプトークンは、一つの ArchiveTimeStamp 要素内に格納しなければならない。アーカイブタイムスタンプ自身の検証情報（タイムスタンプの証明書からそのルート CA に至るまでの証明書パスおよびそれぞれの証明書の失効情報）の格納方法について、大きく分けて以下の 2 通りが考えられる。

① タイムスタンプトークンに証拠情報を埋め込む方法

② タイムスタンプトークンを別途保管する方法

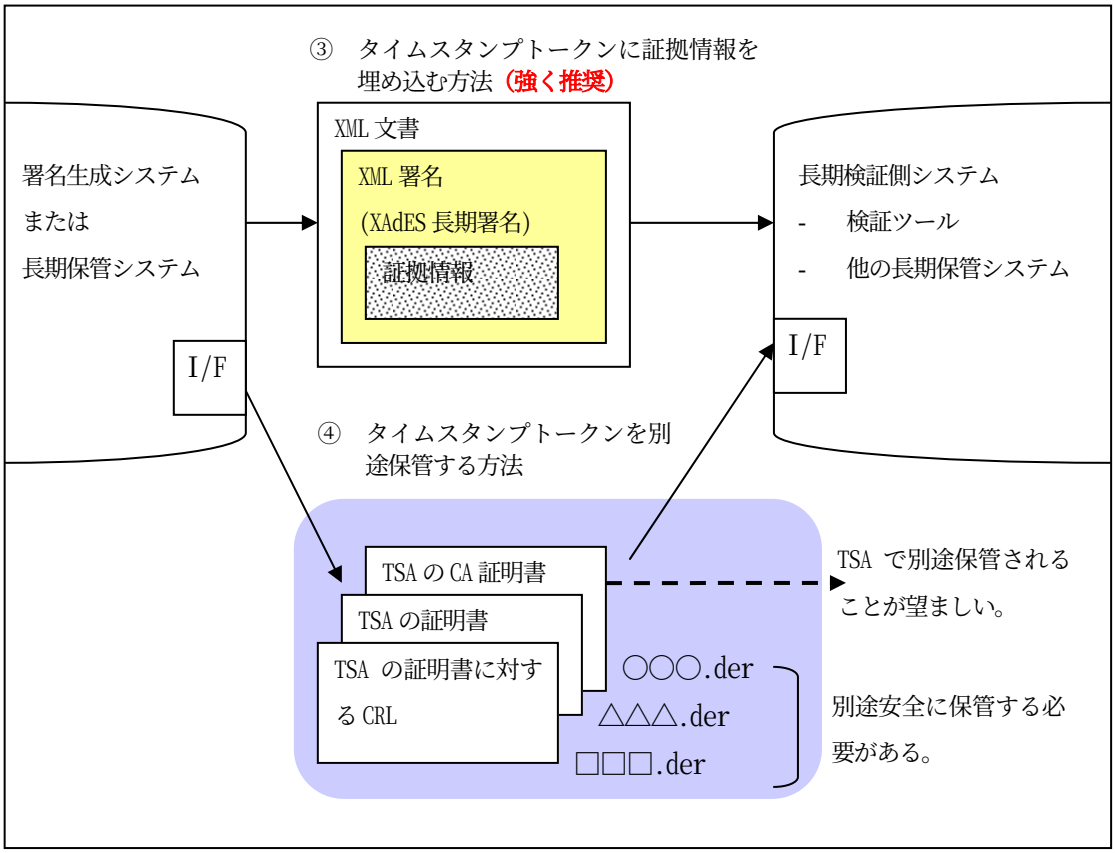


図 付 1.2-1 タイムスタンプトークンの証拠情報の扱いについて：タイムスタンプトークンの証拠情報の扱い方については①タイムスタンプトークン自身に埋め込む方法と②別途安全に保管する方法の二通りが考えられるが、①を強く推奨する。

相互互換性の観点から考えると、検証方法を明確に定義できるのでプロファイルとしては①の方法を強く推奨する。また、①の方法を選択した場合の証拠情報の長期署名フォーマットへの格納方法は次の 2 通り（両方とも標準仕様では言及されていない）考えられるが、構築時には 1)に格納することを推奨し、検証時には 1),2)を処理できることを推奨する。

1) タイムスタンプトークン内の certificates と crls

2) タイムスタンプトークン内の unsigned attribute（Extended validation data 形式）

一方、②の方法は相互運用性の観点から考えると、別途保管したタイムスタンプの証拠情報を用いて検証者がタイムスタンプを検証できる必要があるため、検証者の検証プログラムがタイス

タンプの証拠情報を読み込む機能とそれら使った長期署名フォーマットの検証を実行できる必要がある。

リスト 32：ArchiveTimeStamp 要素の XMLSchema 定義

```
<xsd:element name="ArchiveTimeStamp" type="XAdESTimeStampType"/>
```

2.3 XAdES 長期署名フォーマットにおける必須要素

本稿では、何種類かの形式の XAdES 署名を定義したが、表 付 2.2-3 では XAdES の各形式における必須要素、オプション要素の分類を示す。

表 付 2.2-3 XAdES の形式と要素の対応

	XAdES-BES	XAdES-EPES	XAdES-T	XAdES-A	
QualifyingProperties	○	○	○	○	
SignedProperties	○	○	○	○	
SignedSignatureProperties	○	○	○	○	
SigningTime	△	△	△	△	XAdES（V1.1.1）では必須
SigningCertificate	△1	△1	△1	△1	XAdES（V1.1.1）では必須
SignaturePolicyIdentifier	×	○	△2	△2	XAdES（V1.1.1）では必須
SignatureProductionPlace	△	△	△	△	
SignerRole	△	△	△	△	
SignedDataObjectProperties	△	△	△	△	
DataObjectFormat	△	△	△	△	
CommitmentTypeIndication	△	△	△	△	
AllDataObjectsTimeStamp	△	△	△	△	
IndividualDataObjectsTimeStamp	△	△	△	△	
UnSignedProperties	△	△	○	○	
UnSignedSignatureProperties	△	△	○	○	
CounterSignature	△	△	△	△※	
SignatureTimeStamp	×	×	○	○	
CompleteCertificateRefs	×	×	×	△※	
CompleteRevocationRefs	×	×	×	△※	
AttributeCertificateRefs	×	×	×	△3※	XAdES（V1.1.1）では未定義
AttributeRevocationRefs	×	×	×	△3※	XAdES（V1.1.1）では未定義

				SigAndRefsTimeStamp	×	×	×	△3※
				RefsOnlyTimeStamp	×	×	×	△3※
				CertificateValues	×	×	×	○
				RevocationValues	×	×	×	○
				ArchiveTimeStamp	×	×	×	○

○：必須要素

△：オプション要素（本属性が存在することを理由として、構築時や検証時にエラーとしてはならない。）

△1：ds:KeyInfo に署名者証明書が格納されていて、以下の条件を満たす場合はこの要素はなくても良い。

- ds:KeyInfo は、署名者証明書を含む ds:X509Data 要素を含まなければならない
- ds:KeyInfo は、信頼点までの証明書チェーンを構成する証明書も含む場合がある。
- ds:SignedInfo 要素の ds:Reference 要素で ds:KeyInfo を参照することにより、署名の計算対象として署名値の計算に含まれなければならない。

△2：XAdES-EPES を元に XAdES-T や XAdES-A を構成した場合、SignaturePolicyIdentifier 要素は必須となる。

△3：存在は任意だが推奨しない。

※：存在する場合は、ArchiveTimeStamp の計算の対象に加える必要がある

×：不要（あってはならない要素）

また、準拠する ETSI 101 903 "XML Advanced Electronic Signatures（XAdES）"のバージョンは、表 付 2.2-4 に示すいずれかを選択する。

表 付 2.2-4 準拠する ETSI 101 903 "XML Advanced Electronic Signatures（XAdES）"のバージョン

	ETSI TS 101 903 V1.1.1 （2002-02）	ETSI TS 101 903 V1.2.2（2004-4）	ETSI TS 101 903 V1.3.1（2005-05）
準拠するバージョン	△	△	△

III．長期署名フォーマット

ECOM 相互運用実証実験

CAdES テストケース設計書

2006 年 3 月

次世代電子商取引推進協議会（ECOM）

セキュリティ WG

長期署名フォーマット普及 SWG

長期署名フォーマット相互運用性実証実験プロジェクト

目　次

1．はじめに

本書では、長期署名フォーマットおよび ECOM プロファイルへの準拠性を確認するためのテストの内容を示したテストケース設計書である。

1.1　本書における表記

本仕様書では以下の表記を用いることとする。

表記	説明
<…>	テスト項目
<…-OK>	検証結果の期待値が有効のテスト項目
<…-NG>	検証結果の期待値が無効のテスト項目
<… 00000>	テスト項目名の末尾の 5 桁の数字はテスト項目番号
[…]	参考文献

1.2　テストの構成

・テストカテゴリ（今回のテストではオフラインテストカテゴリとオンラインテストカテゴリに大別される）

・テストケース（個々のテストケースであり機能評価判定の単位となる。複数のテスト項目を含む）

・テスト項目（テストの最小単位であり、検証の結果として期待値通りかそうでないかを成功・失敗として表現する。）

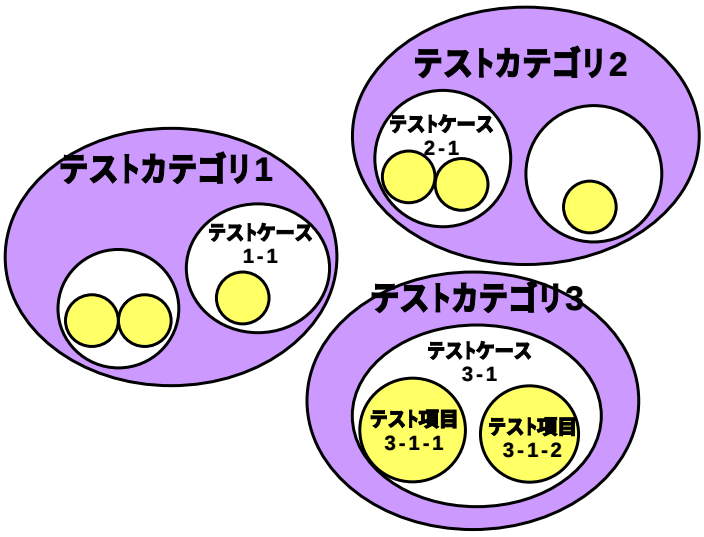


図 付 3.1-1 テストの構成

2. オフライン 共通データ検証テストカテゴリ

ECOM プロファイルに基づく共通の ES フォーマットデータを用いて正しく検証することができるかを確認する。テストツールにより生成された ES フォーマットのデータ（ES, ES-T, ES-C, ES-X long, ES-A）、証明書、CRL、署名対象データをもとに、検証結果が期待値と一致するかどうかを確認する。

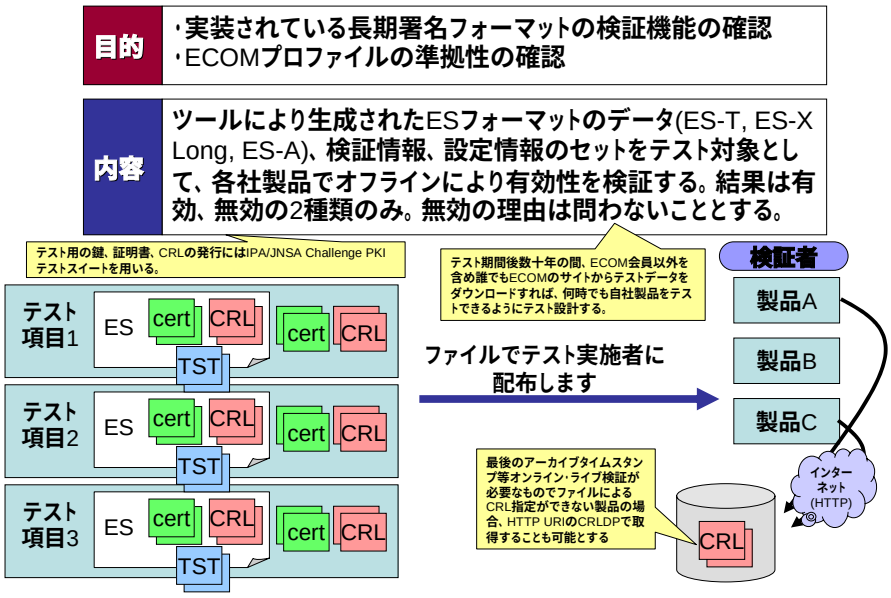


図 付 3.2-1 オフライン検証テスト

2.1 テストの準備

- **CRL のための設定**

オンラインで CRL を取得する場合には、検証環境におけるインターネット接続環境の準備。実験期間終了後にはホスト名を同じくする HTTP リポジトリの立ち上げと設定。もしくは、ファイルによる CRL の設定。

- **トラストアンカの設定**

テストスイートで配布されるオフラインテスト用の署名者用ルート証明書、TSA 用ルート証明書をトラストアンカとして設定する。

2.2 テストの実施

- **署名対象データの設定**

内包署名の場合にはファイル'TARGET_AAA.txt'（ファイルの内容は"aaa"という 3 文字 3 バイトの文字列のみ）、分離署名の場合には'TARGET_BBB.bin'（ファイルの内容は 0x01-0x09,0x00 の繰り返し 1024000 バイトのバイナリファイル）を設定する。

- **検証時刻の設定**

検証時刻はフォーマット毎に異なる。フォーマットに応じ検証時刻の設定を行う。検証可能

な現在時刻の範囲はUTC2002年1月1日0時0分0秒よりUTC2035年12月31日23時59分59秒までとし、各証明書、CRLもまたこれが可能なように設定されている。

- **検証対象の長期署名フォーマットデータの設定**

テストスイートにおいて長期署名フォーマットの検証対象テストデータは'data.der'というファイル名となっており、各テスト項目毎に別々のディレクトリに保存されている。

- **検証の実施**

これを実施すべき全てのテスト項目について実施する。

署名対象データのSHA1ハッシュ値は以下の通り。

- **TARGET_AAA.txt**

SHA-1: 7e240de74fb1ed08fa08d38063f6a6a91462a815

- **TARGET_BBB.bin**

SHA-1: 82918e6b4c2ba314491b2797c3bb4715bae0b713

2.3 テストデータに共通の情報

- **有効期限の時刻は例外ケースを除き00:00:00から23:59:59に統一**
- **署名時刻、タイムスタンプ時刻は例外ケースを除き12:00:00に統一**
- **時刻の表記は特に断りの無い限り、UTC時刻であるとする。**

2.4 オフライン検証テストのテスト項目の概要

オフライン検証テストのテストケースは以下の内容を含んでいる。

- **ES-T，ES-C，ES-X Long，ES-Aフォーマットの検証**
- **内包署名と分離署名**
- **ハッシュアルゴリズム（SHA-1，SHA-256，SHA-512）**
- **BESとEPES**
- **RFC3126とETSI TS 101733 v1.5.1以降のアーカイブハッシュ**
- **SigningTime，SignatureTimeStamp時刻による失効、期限切れ検証**
- **各種ハッシュ値の改竄の検証**
- **コンテンツタイムスタンプ**
- **並列署名（＝独立署名）**
- **署名ポリシーファイルを考慮した検証**

全30テスト項目のリストを以下に示す。

番号	テスト項目名	期待値
10001	EST-ATTACH-NORMAL-OK	有効
BES 内包署名による ES-T フォーマットのデータが有効となることを検証する。		
10002	EST-ATTACH-EXPIRED-NG	無効
ES-T フォーマットで署名者証明書が期限切れの場合、無効となることを検証する。		
10003	EST-ATTACH-REVOKED-NG	無効
期限切れではないが署名タイムスタンプの genTime の値よりも前に署名者証明書が失効している場合に ES-T データが無効であることを検証する。		
10004	EST-ATTACH-SIGTIME-REVOKED-OK	有効
SigingTime 属性の値の時点では失効しているが、署名タイムスタンプの時点では失効していない場合に、SigningTime 属性の値に関わらず ES-T データが有効であることを検証する。		
10005	EST-ATTACH-SIGTS-REVOKED-NG	無効
SigningTime 属性の値の時点では失効してないが、署名タイムスタンプの時点で失効している場合に、署名タイムスタンプを考慮して ES-T データが無効であることを検証する。		
10006	EST-ATTACH-ES-SIG-FORGED-NG	無効
signerInfo の signature フィールドが改竄されている場合に ES-T データが無効であることを検証する。		
10007	EST-ATTACH-ES-SIGTS-SIG-FORGED-NG	無効
署名タイムスタンプのタイムスタンプトークンの signature フィールドが改竄されている場合に、ES-T データが無効であることを検証する。		
10008	EST-ATTACH-ES-MESSAGEDIGEST-FORGED-NG	無効
signedAttrs フィールド中の MessageDigest CMS 属性の値が改竄されている場合に、ES-T データが無効であることを検証する。		
10009	EST-ATTACH-SIGTSTST-MESSAGEDIGEST-FORGED-NG	無効
署名タイムスタンプのタイムスタンプトークンの MessageDigest CMS 属性が改竄されている場合に、ES-T データが無効であることを検証する。		
10010	EST-DETACH-NORMAL-OK	有効
BES 分離署名による ES-T フォーマットのデータが有効であることを検証する。		
20001	EST-OTHERCERT-SHA256-OK	有効
SHA-256 アルゴリズムによる OtherSigningCertificate CMS 属性がある場合に、ES-T フォーマットのデータが有効であることを検証する。		
20002	EST-SIGTS-SHA256-OK	有効
TSTInfo の MessageImprint および SignerInfo の DigestAlgorithm フィールドが SHA-256 アルゴリズムであり、signatureAlgorithm が SHA256withRSA であるようなタイムスタンプトークンの署名タイムスタンプである場合に、ES-T フォーマットデータが有効であることを検証する。		
20003	EST-SIGTS-SHA512-OK	有効
TSTInfo の MessageImprint および SignerInfo の DigestAlgorithm フィールドが SHA-512 アルゴリズムであり、signatureAlgorithm が SHA512withRSA であるようなタイムスタンプトークンの署名タイムスタンプである場合に、ES-T フォーマットデータが有効であることを検証する。		
20004	EST-CONTENT-TIMESTAMP-OK	有効
signedAttributes フィールドに ContentTimeStamp CMS 属性がある場合に、ES-T フォーマットデータが有効であることを検証する。		
20005	EST-INDEPENDENT-SIGNATURES-OK	有効
2 つの signerInfo を持つような並列署名（独立署名）である ES-T フォーマットデータが有効であることを検証する。		
20006	EST-EPES-WITHOUT-HASHCHECK-OK	有効
signaturePolicyIdentifier CMS 属性があるような EPES に基づく ES-T フォーマットデータが有効であることを検証する。		
20007	EST-EPES-NORMAL-OK	有効
signaturePolicyIdentifier CMS 属性がある EPES に基づく ES-T フォーマットデータにおいて、署名ポリシーファイルを参照しながら有効であることを検証する。		
20008	EST-EPES-POLICY-HASH-NOT-MATCH-NG	無効
signaturePolicyIdentifier CMS 属性のハッシュ値が署名ポリシーと一致しない場合に ES-T フォーマットが無効であることを検証する。		

20009	EST-EPES-NOT-BEFORE-VIOLATION-NG	無効
署名ポリシーファイルの signingPeriod の notBefore フィールドの時刻が遠い将来であり、まだ有効期間内に無い場合、署名ポリシーが無効であるために ES-T フォーマットが現時点で無効であることを検証する。		
20010	EST-EPES-MANDATED-SIGNEDATTRS-VIOLATION-NG	無効
署名ポリシーファイルの mandatedSignedAttr フィールドで必須とされている SigningTime 属性が無い場合に、ES-T フォーマットデータが無効であることを検証する。		
20011	EST-EPES-EXTERNAL-SIGNEDDATA-VIOLATION-NG	無効
署名ポリシーファイルにおいて externalSignedData が TRUE、即ち分離署名を要求しているにも関らず、内包署名である場合に、ES-T フォーマットデータが無効であることを検証する。		
40001	ESC-ATTACH-NORMAL-OK	有効
内包署名の BES に基づく ES-C フォーマットデータが有効であることを検証する。		
40002	ESC-DETACH-NORMAL-OK	有効
分離署名の BES に基づく ES-C フォーマットデータが有効であることを検証する。		
50001	ESXL-ATTACH-NORMAL-OK	有効
内包署名の BES に基づく ES-X Long フォーマットデータが有効であることを検証する。		
50002	ESXL-DETACH-NORMAL-OK	有効
分離署名の BES に基づく ES-X Long フォーマットデータが有効であることを検証する		
60001	ESXL-ATTACH-SIGTS-VALIDATIONINFO-NOT-INCLUDED-OK	有効
署名タイムスタンプの TSA 証明書のための懸賞情報がトークンに含まれず、ファイルなどの別の方法で検証情報が提供される場合に、ES-X Long フォーマットのデータが有効であることを検証する。		
70001	ESA1-ATTACH-NORMAL-OK	有効
内包署名による第一世代、即ち 1 つしかアーカイブタイムスタンプ CMS 属性が無い ES-A フォーマットデータが有効であることを検証する。		
70002	ESA1-DETACH-NORMAL-OK	有効
分離署名による第一世代、即ち 1 つしかアーカイブタイムスタンプ CMS 属性が無い ES-A フォーマットデータが有効であることを検証する。		
80001	ESA1-ATTACH-ETSI151-OK	有効
ESTI TS 101 733 v1.5.1 以降のアーカイブハッシュ計算方法による内包署名による第一世代の ES-A フォーマットデータが有効であることを検証する。		
80002	ESA1-DETACH-ETSI151-OK	有効
ESTI TS 101 733 v1.5.1 以降のアーカイブハッシュ計算方法による分離署名による第一世代の ES-A フォーマットデータが有効であることを検証する。		

2.5 ES-T フォーマット標準テスト項目

2.5.1 <EST-ATTACH-NORMAL-OK 10001>

署名者証明書、署名タイムスタンプの TSA 証明書が有効期間内にあり、共に失効しない場合。ES-T データが有効であることを検証する。本テストケースは ES-T フォーマットの標準テストである。

期待値	有効（valid）
署名を実施したとする時刻	2001.1.1 12:00:00
サイニングタイム属性の時刻	属性無し
署名タイムスタンプの時刻	2001.1.1 12:00:00
署名者証明書	2001.1.1～2035.12.31
署名者証明書の検証 CRL	2001.1.2 00:00:00～2001.1.2 23:59:59

署名タイムスタンプ TSA 証明書	2001.1.1～2035.12.31
署名 TS TSA 証明書検証 CRL	2001.1.1 00:00:00～2035.12.31 23:59:59

2.5.2 <EST-ATTACH-EXPIRED-NG 10002>

署名タイムスタンプの TSA 証明書は有効であるが、署名証明書が期限切れの時点で署名タイムスタンプを付した場合､署名者証明書を検証する CRL に記載されていないとき ES-T データが無効であることを検証する。

期待値	無効（invalid）
署名を実施したとする時刻	2001.1.3 12:00:00
サイニングタイム属性の時刻	属性無し
署名タイムスタンプの時刻	2001.1.3 12:00:00
署名者証明書	2001.1.1 00:00:00～2001.1.1 23:5959
署名者証明書の検証 CRL	2001.1.2 00:00:00～2001.1.2 23:59:59
署名タイムスタンプ TSA 証明書	2001.1.1～2035.12.31
署名 TS TSA 証明書検証 CRL	2001.1.1 00:00:00～2035.12.31 23:59:59

2.5.3 <EST-ATTACH-REVOKED-NG 10003>

署名者証明書、署名タイムスタンプの TSA 証明書が有効期間内にあり、サイニングタイム属性の時刻および署名タイムスタンプ時刻において、署名者証明書が失効して CRL に記載されている場合、ES-T データが無効であることを検証する。

期待値	無効（invalid）
署名を実施したとする時刻	2001.1.2 12:00:00
サイニングタイム属性の時刻	2001.1.2 12:00:00
署名タイムスタンプの時刻	2001.1.2 12:00:00
署名者証明書	2001.1.1～2035.12.31
署名者証明書の検証 CRL	2001.1.4 00:00:00-2001.1.4 23:59:59
署名者証明書 CRL 中失効日時	2005.1.1 12:00
署名タイムスタンプ TSA 証明書	2001.1.1～2035.12.31
署名 TS TSA 証明書検証 CRL	2001.1.4 00:00:00-2001.1.4 23:59:59

2.5.4 <EST-ATTACH-SIGTIME-REVOKED-OK 10004>

署名者証明書、署名タイムスタンプの TSA 証明書が有効期間内にあり、署名タイムスタンプ時刻では失効していないが、サイニング属性の時刻において署名者証明書が失効して CRL に記載されている場合、署名時刻は無視し、署名タイムスタンプ時刻での証明書有効性を検証することにより、ES-T データが有効であることを検証する。

期待値	有効（valid）
署名を実施したとする時刻	2001.1.1 12:00:00（=SignatureTS）
サイニングタイム属性の時刻	2001.1.4 12:00:00
署名タイムスタンプの時刻	2001.1.1 12:00:00
署名者証明書	2001.1.1～2035.12.31
署名者証明書の検証 CRL	2001.1.3 00:00:00-2001.1.3 23:59:59
署名者証明書 CRL 中失効日時	2005.1.2 12:00:00
署名タイムスタンプ TSA 証明書	2001.1.1～2035.12.31
署名 TS TSA 証明書検証 CRL	2001.1.3 00:00:00-2001.1.3 23:59:59

2.5.5 <EST-ATTACH-SIGTS-REVOKED-NG 10005>

署名者証明書、署名タイムスタンプの TSA 証明書が有効期間内にあり、SigningTime 属性の時刻では失効していないが、署名タイムスタンプ時刻において署名者証明書が失効して CRL に記載されている場合、署名時刻は無視し、署名タイムスタンプ時刻での証明書有効性を検証することにより、ES-T データが無効であることを検証する。

期待値	無効（invalid）
署名を実施したとする時刻	2001.1.1 12:00:00
サイニングタイム属性の時刻	2001.1.1 12:00:00
署名タイムスタンプの時刻	2001.1.3 12:00:00
署名者証明書	2001.1.1～2035.12.31
署名者証明書の検証 CRL	2001.1.4 00:00:00-2001.1.4 23:59:59
署名者証明書 CRL 中失効日時	2005.1.2 12:00:00
署名タイムスタンプ TSA 証明書	2001.1.1～2035.12.31
署名 TS TSA 証明書検証 CRL	2001.1.4 00:00:00-2001.1.4 23:59:59

2.5.6 <EST-ATTACH-ES-SIG-FORGED-NG 10006>

ES-T フォーマットの CMS SignedData の SignerInfo において signature フィールドにある署名値が改竄されていた場合に無効であることを検証する。

期待値	無効（invalid）
署名を実施したとする時刻	2001.1.1 12:00:00
サイニングタイム属性の時刻	属性無し
署名タイムスタンプの時刻	2001.1.1 12:00:00
署名者証明書	2001.1.1～2035.12.31
署名者証明書の検証 CRL	2001.1.2 00:00:00～2001.1.2 23:59:59
署名タイムスタンプ TSA 証明書	2001.1.1～2035.12.31

署名 TS TSA 証明書検証 CRL	2001.1.2 00:00:00～2001.1.2 23:59:59

2.5.7 <EST-ATTACH-SIGTS-SIG-FORGED-NG 10007>

ES-T フォーマットの SiginatureTimeStamp 属性中の TimeStampToken の CMS SignedData 構造の SignerInfo において signature フィールドにある署名値が改竄されていた場合に無効であることを検証する。

期待値	無効（invalid）
署名を実施したとする時刻	2001.1.1 12:00:00
サイニングタイム属性の時刻	属性無し
署名タイムスタンプの時刻	2001.1.1 12:00:00
署名者証明書	2001.1.1～2035.12.31
署名者証明書の検証 CRL	2001.1.2 00:00:00～2001.1.2 23:59:59
署名タイムスタンプ TSA 証明書	2001.1.1～2035.12.31
署名 TS TSA 証明書検証 CRL	2001.1.2 00:00:00～2001.1.2 23:59:59

2.5.8 <EST-ATTACH-ES-MESSAGEDIGEST-FORGED-NG 10008>

ES-T フォーマットの CMS SignedData の signedAttributes 中の MessageDigest 属性の値が改竄されていた場合に無効であることを検証する。

期待値	無効（invalid）
署名を実施したとする時刻	2001.1.1 12:00:00
サイニングタイム属性の時刻	属性無し
署名タイムスタンプの時刻	2001.1.1 12:00:00
署名者証明書	2001.1.1～2035.12.31
署名者証明書の検証 CRL	2001.1.2 00:00:00～2001.1.2 23:59:59
署名タイムスタンプ TSA 証明書	2001.1.1～2035.12.31
署名 TS TSA 証明書検証 CRL	2001.1.1 00:00:00～2035.12.31 23:59:59

2.5.9 <EST-ATTACH-SIGTSTST-MESSAGEDIGEST-FORGED-NG 10009>

ES-T フォーマットの SignatureTimeStamp 属性に含まれるタイムスタンプトークンの signedAttributes 中の MessageDigest 属性の値が改竄されていた場合に無効であることを検証する。

期待値	無効（invalid）
署名を実施したとする時刻	2001.1.1 12:00:00
サイニングタイム属性の時刻	属性無し

署名タイムスタンプの時刻	2001.1.1 12:00:00
署名者証明書	2001.1.1～2035.12.31
署名者証明書の検証 CRL	2001.1.2 00:00:00～2001.1.2 23:59:59
署名タイムスタンプ TSA 証明書	2001.1.1～2035.12.31
署名 TS TSA 証明書検証 CRL	2001.1.1 00:00:00～2035.12.31 23:59:59

2.5.10 <EST-DETACH-NORMAL-OK 10010>

署名対象文書に対して分離署名を行った ES-T フォーマットにおいてデータが有効であることを検証する。

期待値	有効（valid）
署名を実施したとする時刻	2001.1.1 12:00:00
サイニングタイム属性の時刻	属性無し
署名タイムスタンプの時刻	2001.1.1 12:00:00
署名者証明書	2001.1.1～2035.12.31
署名者証明書の検証 CRL	2001.1.2 00:00:00～2001.1.2 23:59:59
署名タイムスタンプ TSA 証明書	2001.1.1～2035.12.31
署名 TS TSA 証明書検証 CRL	2001.1.1 00:00:00～2035.12.31 23:59:59

2.6 ES-T フォーマットオプショナルテスト項目

2.6.1 <EST-OTHERCERT-SHA256-OK 20001>

テスト項目<EST-ATTACH-NORMAL-OK>と比較して、署名者証明書を特定するための情報として、ESSSigningCertificate 属性ではなく、ハッシュアルゴリズムに SHA256 を用いた場合で証明書のハッシュ値が一致している場合に、有効であることを検証する。

期待値	有効（valid）
署名を実施したとする時刻	2001.1.1 12:00:00
サイニングタイム属性の時刻	属性無し
署名タイムスタンプの時刻	2001.1.1 12:00:00
署名者証明書	2001.1.1～2035.12.31
署名者証明書の検証 CRL	2001.1.2 00:00:00～2001.1.2 23:59:59
署名タイムスタンプ TSA 証明書	2001.1.1～2035.12.31
署名 TS TSA 証明書検証 CRL	2001.1.1 00:00:00～2035.12.31 23:59:59

2.6.2 <EST-SIGTS-SHA256-OK 20002>

ES-T フォーマットの署名タイムスタンプのタイムスタンプトークンのハッシュアルゴリズム

に SHA256、署名アルゴリズムに SHA256withRSA が用いられた場合に有効であることを検証する。

- **TimeStampToken の TSTInfo の MessageImprint は SHA256**
- **TimeStampToken の SignerInfo の DigestAlgorithm は SHA256**
- **TimeStampToken の SignerInfo の SignatureAlgorithm は SHA256withRSA**

期待値	有効（valid）
署名を実施したとする時刻	2001.1.1 12:00:00
サイニングタイム属性の時刻	属性無し
署名タイムスタンプの時刻	2001.1.1 12:00:00
署名者証明書	2001.1.1～2035.12.31
署名者証明書の検証 CRL	2001.1.2 00:00:00～2001.1.2 23:59:59
署名タイムスタンプ TSA 証明書	2001.1.1～2035.12.31
署名 TS TSA 証明書検証 CRL	2001.1.1 00:00:00～2035.12.31 23:59:59

2.6.3 <EST-SIGTS-SHA512-OK 20003>

ES-T フォーマットの署名タイムスタンプのタイムスタンプトークンのハッシュアルゴリズムに SHA512、署名アルゴリズムに SHA512withRSA が用いられた場合に有効であることを検証する。

- **TimeStampToken の TSTInfo の MessageImprint は SHA512**
- **TimeStampToken の SignerInfo の DigestAlgorithm は SHA512**
- **TimeStampToken の SignerInfo の SignatureAlgorithm は SHA512withRSA**

期待値	有効（valid）
署名を実施したとする時刻	2001.1.1 12:00:00
サイニングタイム属性の時刻	属性無し
署名タイムスタンプの時刻	2001.1.1 12:00:00
署名者証明書	2001.1.1～2035.12.31
署名者証明書の検証 CRL	2001.1.2 00:00:00～2001.1.2 23:59:59
署名タイムスタンプ TSA 証明書	2001.1.1～2035.12.31
署名 TS TSA 証明書検証 CRL	2001.1.1 00:00:00～2035.12.31 23:59:59

2.6.4 <EST-CONTENT-TIMESTAMP-OK 20004>

ES-T フォーマットのデータの CMS 署名属性に有効なコンテンツタイムスタンプが含まれている場合に有効であることを検証する。

期待値	有効（valid）
署名を実施したとする時刻	2001.1.1 12:00:00
コンテンツタイムスタンプの時刻	2001.1.1 09:00:00
署名タイムスタンプの時刻	2001.1.1 12:00:00
署名者証明書	2001.1.1～2035.12.31
署名者証明書の検証 CRL	2001.1.2 00:00:00～2001.1.2 23:59:59
署名タイムスタンプ TSA 証明書	2001.1.1～2035.12.31
署名 TS TSA 証明書検証 CRL	2001.1.1 00:00:00～2035.12.31 23:59:59

2.6.5 <EST-INDEPENDENT-SIGNATURES-OK 20005>

二人の署名者による並列署名（独立署名とも言う）の双方に有効な署名タイムスタンプが付与された ES-T フォーマットのデータが有効であることを検証する。

二人の署名者用証明書は同一のサブ CA から発行されているとする。

期待値	有効（valid）
署名 1 を実施したとする時刻	2001.1.1 12:00:00
サイニングタイム属性 1 の時刻	属性無し
署名タイムスタンプ 1 の時刻	2001.1.1 12:00:00
署名者証明書 1 の有効期限	2001.1.1～2035.12.31
署名 2 を実施したとする時刻	2001.1.1 13:00:00
サイニングタイム属性 2 の時刻	属性無し
署名タイムスタンプ 2 の時刻	2001.1.1 13:00:00
署名者証明書 2 の有効期限	2001.1.1～2035.12.31
署名者証明書の検証 CRL	2001.1.2 00:00:00～2001.1.2 23:59:59
署名タイムスタンプ TSA 証明書	2001.1.1～2035.12.31
署名 TS TSA 証明書検証 CRL	2001.1.1 00:00:00～2035.12.31 23:59:59

2.6.6 <EST-EPES-WITHOUT-HASHCHECK-OK 20006>

signedAttributes フィールドに署名ポリシー識別子を明示的に持つ EPES（Explicit Policy Electronic Signatures）フォーマットに対し署名タイムスタンプを付与した ES-T データを読み込みエラーとならないことを検証する。

署名ポリシーを厳密に扱う実装では、テストデータとして配布される署名ポリシーファイルを共に検証に用いる。

期待値	有効（valid）
署名を実施したとする時刻	2001.1.1 12:00:00
サイニングタイム属性の時刻	属性無し

署名タイムスタンプの時刻	2001.1.1 12:00:00
署名者証明書	2001.1.1～2035.12.31
署名者証明書の検証 CRL	2001.1.2 00:00:00～2001.1.2 23:59:59
署名タイムスタンプ TSA 証明書	2001.1.1～2035.12.31
署名 TS TSA 証明書検証 CRL	2001.1.1 00:00:00～2035.12.31 23:59:59
署名ポリシーOID	1.2.3.4.5
署名ポリシーSHA1 ハッシュ値	af1d3ea7aef706a898191dd257218f5e9acafaa1

2.6.7 <EST-EPES-NORMAL-OK 20007>

signedAttributes フィールドに署名ポリシー識別子を明示的に持つ EPES フォーマットに対し署名タイムスタンプを付与した ES-T データおよび署名ポリシーファイルを読み込み EPES フォーマットより生成された ES-T データが有効であることを検証する。

期待値	有効（valid）
署名を実施したとする時刻	2001.1.1 12:00:00
サイニングタイム属性の時刻	属性無し
署名タイムスタンプの時刻	2001.1.1 12:00:00
署名者証明書	2001.1.1～2035.12.31
署名者証明書の検証 CRL	2001.1.2 00:00:00～2001.1.2 23:59:59
署名タイムスタンプ TSA 証明書	2001.1.1～2035.12.31
署名 TS TSA 証明書検証 CRL	2001.1.1 00:00:00～2035.12.31 23:59:59
署名ポリシーOID	1.2.3.4.5.20000
署名ポリシーSHA1 ハッシュ値	90490e8e411bd495415988298d07ab45922e8bff

2.6.8 <EST-EPES-POLICY-HASH-NOT-MATCH-NG 20008>

EPES より生成された ES-T フォーマットのデータにおいて、その SignaturePolicyIdentifier CMS 署名属性の持つハッシュ値が署名ポリシーファイルのハッシュ値と一致しない場合に、その ES-T データが無効であることを検証する。

期待値	無効（invalid）
署名を実施したとする時刻	2001.1.1 12:00:00
サイニングタイム属性の時刻	属性無し
署名タイムスタンプの時刻	2001.1.1 12:00:00
署名者証明書	2001.1.1～2035.12.31
署名者証明書の検証 CRL	2001.1.2 00:00:00～2001.1.2 23:59:59
署名タイムスタンプ TSA 証明書	2001.1.1～2035.12.31
署名 TS TSA 証明書検証 CRL	2001.1.1 00:00:00～2035.12.31 23:59:59

署名ポリシーOID	1.2.3.4.5.20000
署名ポリシーSHA1 ハッシュ値	90490e8e411bd495415988298d07ab45922e8bff
CMS 署名ポリシー識別子属性のハッシュ値	1f1f1f1f1f1f1f1f1f1f1f1f1f1f1f1f1f1f

2.6.9 <EST-EPES-NOT-BEFORE-VIOLATION-NG 20009>

EPES より生成された ES-T フォーマットのデータの使用する署名ポリシーデータにおいて、signingPeriod の notBefore フィールドの値が常に無効となるような値（2035 年 12 月 31 日）に設定されていた場合に、署名ポリシーが有効期間内に無いために、関連付けられた ES-T データが無効となることを検証する。

期待値	無効（invalid）
署名を実施したとする時刻	2001.1.1 12:00:00
サイニングタイム属性の時刻	属性無し
署名タイムスタンプの時刻	2001.1.1 12:00:00
署名者証明書	2001.1.1～2035.12.31
署名者証明書の検証 CRL	2001.1.2 00:00:00～2001.1.2 23:59:59
署名タイムスタンプ TSA 証明書	2001.1.1～2035.12.31
署名 TS TSA 証明書検証 CRL	2001.1.1 00:00:00～2035.12.31 23:59:59
署名ポリシーOID	1.2.3.4.5.20009
署名ポリシーSHA1 ハッシュ値	eab88babb6ffc05343fc8ef0ca6a7dd4920b7e02
署名ポリシーの sigingPeriod.notBefore フィールド	2035.12.31 23:59:59

2.6.10 <EST-EPES-MANDATED-SIGNEDATTRS-VIOLATION-NG 20010>

EPES より生成された ES-T フォーマットのデータに関連付けられた署名ポリシーデータにおいて、commonRules の signerAndVerifierValue の signerRules の mandatedSigndAttr に sigingTime の OID が含まれているにもかかわらず、ES-T フォーマットのデータには signingTime CMS 属性が含まれていない場合に、署名ポリシーに違反するため ES-T データが無効となることを検証する。

期待値	無効（invalid）
署名を実施したとする時刻	2001.1.1 12:00:00
サイニングタイム属性の時刻	属性無し
署名タイムスタンプの時刻	2001.1.1 12:00:00
署名者証明書	2001.1.1～2035.12.31
署名者証明書の検証 CRL	2001.1.2 00:00:00～2001.1.2 23:59:59
署名タイムスタンプ TSA 証明書	2001.1.1～2035.12.31

署名 TS TSA 証明書検証 CRL	2001.1.1 00:00:00～2035.12.31 23:59:59
署名ポリシーOID	1.2.3.4.5.20010
署名ポリシーSHA1 ハッシュ値	5a6c1d137ca139771adbd8d41c868d682ded8b20
mandatedSignedAttr	1.2.840.113549.1.9.4 1.2.840.113549.1.9.5（signingTime） 1.2.840.113549.1.9.16.2.15

2.6.11 <EST-EPES-EXTERNAL-SIGNEDDATA-VIOLATION-NG 20011>

EPES より生成された ES-T フォーマットのデータに関連付けられた署名ポリシーデータにおいて、commonRules の signerAndVerifierValue の signerRules の externalSignedData フィールドの値が TRUE、即ち署名ポリシーが分離署名であることを要求している場合に、ES-T フォーマットのデータが内包署名であったとき、署名ポリシーに違反するため ES-T データが無効となることを検証する。

期待値	無効（invalid）
署名を実施したとする時刻	2001.1.1 12:00:00
サイニングタイム属性の時刻	属性無し
署名タイムスタンプの時刻	2001.1.1 12:00:00
署名者証明書	2001.1.1～2035.12.31
署名者証明書の検証 CRL	2001.1.2 00:00:00～2001.1.2 23:59:59
署名タイムスタンプ TSA 証明書	2001.1.1～2035.12.31
署名 TS TSA 証明書検証 CRL	2001.1.1 00:00:00～2035.12.31 23:59:59
署名ポリシーOID	1.2.3.4.5.20011
署名ポリシーSHA1 ハッシュ値	b363f51a65438136d26ce87f3078657df52b5dc4
externalSignedData	TRUE

2.7 ES-C フォーマット標準テスト項目

ES-C フォーマットは ECOM プロファイルにおいてオプショナルであるため、標準テスト項目は無しとする。

2.8 ES-C フォーマットオプショナルテスト項目

2.8.1 <ESC-ATTACH-NORMAL-OK 40001>

署名者証明書、署名タイムスタンプの TSA 証明書が有効期間内にあり、共に失効しない場合。ES-C データが有効であることを検証する。本テストケースは ES-C フォーマットの標準テストである。

期待値	有効（valid）
署名を実施したとする時刻	2001.1.1 12:00:00
サイニングタイム属性の時刻	属性無し
署名タイムスタンプの時刻	2001.1.1 12:00:00
署名者証明書	2001.1.1～2035.12.31
署名者証明書の検証 CRL	2001.1.2 00:00:00～2001.1.2 23:59:59
署名タイムスタンプ TSA 証明書	2001.1.1～2035.12.31
署名 TS TSA 証明書検証 CRL	2001.1.2 00:00:00～2001.1.2 23:59:59

2.8.2 <ESC-DETACH-NORMAL-OK 40002>

署名対象文書に対して分離署名を行った ES-C フォーマットにおいてデータが有効であることを検証する。

期待値	有効（valid）
署名を実施したとする時刻	2001.1.1 12:00:00
サイニングタイム属性の時刻	属性無し
署名タイムスタンプの時刻	2001.1.1 12:00:00
署名者証明書	2001.1.1～2035.12.31
署名者証明書の検証 CRL	2001.1.2 00:00:00～2001.1.2 23:59:59
署名タイムスタンプ TSA 証明書	2001.1.1～2035.12.31
署名 TS TSA 証明書検証 CRL	2001.1.2 00:00:00～2001.1.2 23:59:59

2.9 ES-X Long フォーマット標準テスト項目

2.9.1 <ESXL-ATTACH-NORMAL-OK 50001>

署名者証明書、署名タイムスタンプの TSA 証明書が有効期間内にあり、共に失効しておらず、これらの検証情報を含む ES-X Long データが有効であることを検証する。本テストケースは ES-X Long フォーマットの標準テストである。

期待値	有効（valid）
署名を実施したとする時刻	2001.1.1 12:00:00
サイニングタイム属性の時刻	属性無し
署名タイムスタンプの時刻	2001.1.1 12:00:00
署名者証明書	2001.1.1～2035.12.31
署名者証明書の検証 CRL	2001.1.2 00:00:00～2001.1.2 23:59:59
署名タイムスタンプ TSA 証明書	2001.1.1～2035.12.31
署名 TS TSA 証明書検証 CRL	2001.1.2 00:00:00～2001.1.2 23:59:59

2.9.2 <ESXL-DETACH-NORMAL-OK 50002>

署名対象文書に対して分離署名を行った ES-X Long フォーマットにおいてデータが有効であることを検証する。

期待値	有効（valid）
署名を実施したとする時刻	2001.1.1 12:00:00
サイニングタイム属性の時刻	属性無し
署名タイムスタンプの時刻	2001.1.1 12:00:00
署名者証明書	2001.1.1～2035.12.31
署名者証明書の検証 CRL	2001.1.2 00:00:00～2001.1.2 23:59:59
署名タイムスタンプ TSA 証明書	2001.1.1～2035.12.31
署名 TS TSA 証明書検証 CRL	2001.1.2 00:00:00～2001.1.2 23:59:59

2.10 ES-X Long フォーマットオプショナルテスト項目

2.10.1 <ESXL-ATTACH-SIGTS-VALIDATIONINFO-NOT-INCLUDED-OK 60001>

ES-X Long フォーマットを検証する際、署名タイムスタンプ属性のタイムスタンプトークンの TSA 証明書の検証情報がトークン無いに含まれず、別の手段により渡される場合、この ES X-Long フォーマットが有効であることを検証する。

期待値	有効（valid）
署名を実施したとする時刻	2001.1.1 12:00:00
サイニングタイム属性の時刻	属性無し
署名タイムスタンプの時刻	2001.1.1 12:00:00
署名者証明書	2001.1.1～2035.12.31
署名者証明書の検証 CRL	2001.1.2 00:00:00～2001.1.2 23:59:59
署名タイムスタンプ TSA 証明書	2001.1.1～2035.12.31
署名 TS TSA 証明書検証 CRL	2001.1.2 00:00:00～2001.1.2 23:59:59

2.11 ES-A フォーマット標準テスト項目

2.11.1 <ESA1-ATTACH-NORMAL-OK 70001>

ECOM 長期署名フォーマットプロファイル 2005 で定めた RFC3126 に基づくアーカイブタイムスタンプのハッシュ計算方法により付与されたアーカイブタイムスタンプ属性を一つ含む ES-A フォーマットが有効であることを検証する。

期待値	有効（valid）
署名を実施したとする時刻	2001.1.1 12:00:00
署名タイムスタンプの時刻	2001.1.1 12:00:00
署名者証明書	2001.1.1～2035.12.31
署名者証明書の検証 CRL	2001.1.2 00:00:00～2001.1.2 23:59:59
署名タイムスタンプ TSA 証明書	2001.1.1～2035.12.31
署名 TS TSA 証明書検証 CRL	2001.1.3 00:00:00～2001.1.3 23:59:59
Archive タイムスタンプ 1 の時刻	2001.1.3 12:00:00
Archive タイムスタンプ 1 TSA 証明書	2001.1.1～2035.12.31
Archive TS TSA 証明書検証 CRL	2001.1.4 00:00:00～2035.12.31 23:59:59

2.11.2 <ESA1-DETACH-NORMAL-OK 70002>

署名対象文書に対して分離署名を行った第一世代の ArchiveTimeStamp のみを持つ ES-A フォーマットにおいてデータが有効であることを検証する。

期待値	有効（valid）
署名を実施したとする時刻	2001.1.1 12:00:00
署名タイムスタンプの時刻	2001.1.1 12:00:00
署名者証明書	2001.1.1～2035.12.31
署名者証明書の検証 CRL	2001.1.2 00:00:00～2001.1.2 23:59:59
署名タイムスタンプ TSA 証明書	2001.1.1～2035.12.31
署名 TS TSA 証明書検証 CRL	2001.1.3 00:00:00～2001.1.3 23:59:59
Archive タイムスタンプ 1 の時刻	2001.1.3 12:00:00
Archive タイムスタンプ 1 TSA 証明書	2001.1.1～2035.12.31
Archive TS TSA 証明書検証 CRL	2001.1.4 00:00:00～2035.12.31 23:59:59

2.12 ES-A フォーマットオプショナルテスト項目

2.12.1 <ESA1-ATTACH-ETSI151-OK 80001>

ECOM 長期署名フォーマットプロファイルの範囲外ではあるが、ETSI TS 101 733 v1.5.1 以降で定めている新しいアーカイブハッシュ計算方法に本設計書付録で示した正規化法を用いたアーカイブタイムスタンプ属性を一つ含む ES-A フォーマットが有効であることを検証する。

期待値	有効（valid）
署名を実施したとする時刻	2001.1.1 12:00:00
署名タイムスタンプの時刻	2001.1.1 12:00:00
署名者証明書	2001.1.1～2035.12.31

署名者証明書の検証 CRL	2001.1.2 00:00:00～2001.1.2 23:59:59
署名タイムスタンプ TSA 証明書	2001.1.1～2035.12.31
署名 TS TSA 証明書検証 CRL	2001.1.3 00:00:00～2001.1.3 23:59:59
Archive タイムスタンプ 1 の時刻	2001.1.3 12:00:00
Archive タイムスタンプ 1 TSA 証明書	2001.1.1～2035.12.31
Archive TS TSA 証明書検証 CRL	2001.1.4 00:00:00～2035.12.31 23:59:59

2.12.2 <ESA1-DETACH-ETSI151-OK 80002>

テスト項目<ESA1-ATTACH-ETSI151-OK>と同じ条件で分離署名であった場合に ES-A フォーマットが有効であることを検証する。最初のハッシュ対象 encapContentInfo はコンテンツを含め内部まで DER 正規化されていなければならない。署名対象データは<ESA1-ATTACH-ETSI151-OK>と同じデータとし、他の分離署名の署名対象とは異なる。

期待値	有効（valid）
署名を実施したとする時刻	2001.1.1 12:00:00
署名タイムスタンプの時刻	2001.1.1 12:00:00
署名者証明書	2001.1.1～2035.12.31
署名者証明書の検証 CRL	2001.1.2 00:00:00～2001.1.2 23:59:59
署名タイムスタンプ TSA 証明書	2001.1.1～2035.12.31
署名 TS TSA 証明書検証 CRL	2001.1.3 00:00:00～2001.1.3 23:59:59
Archive タイムスタンプ 1 の時刻	2001.1.3 12:00:00
Archive タイムスタンプ 1 TSA 証明書	2001.1.1～2035.12.31
Archive TS TSA 証明書検証 CRL	2001.1.4 00:00:00～2035.12.31 23:59:59

2.13 ES-T 標準テストケース

本節では ES-T フォーマットを扱う実装が満足すべきテストケースを示す。

2.13.1 <OFF-T-1>

テストケース名	OFF-T-1
一般的な内包署名の ES-T フォーマットを読み込むことができる	
成功条件：以下のテスト項目が全て期待値通り	
10001	EST-ATTACH-NORMAL-OK

2.13.2 <OFF-T-2>

テストケース名	OFF-T-2
ES-T フォーマットの署名者証明書の期限切れを扱える	
成功条件：以下のテスト項目が全て期待値通り	
10001	EST-ATTACH-NORMAL-OK
10002	EST-ATTACH-EXPIRED-NG

2.13.3 <OFF-T-3>

テストケース名	OFF-T-3
ES-T フォーマットの署名者証明書の失効を扱える	
成功条件：以下のテスト項目が全て期待値通り	
10001	EST-ATTACH-NORMAL-OK
10003	EST-ATTACH-REVOKED-NG

2.13.4 <OFF-T-4>

テストケース名	OFF-T-4
ES-T フォーマットの署名者証明書の認証パス検証を正しく行える	
成功条件：以下のテスト項目が全て期待値通り	
10001	EST-ATTACH-NORMAL-OK
10002	EST-ATTACH-EXPIRED-NG
10003	EST-ATTACH-REVOKED-NG

2.13.5 <OFF-T-5>

テストケース名	OFF-T-5
ES-T フォーマットでサイニングタイムに関係なく署名タイムスタンプの時刻により署名者証明書の失効検証ができる	
成功条件：以下のテスト項目が全て期待値通り	
10001	EST-ATTACH-NORMAL-OK
10003	EST-ATTACH-REVOKED-NG
10004	EST-ATTACH-SIGTIME-REVOKED-OK
10005	EST-ATTACH-SIGTS-REVOKED-NG

2.13.6 <OFF-T-6>

テストケース名	OFF-T-6
ES-T フォーマットの SignerInfo の署名値の改竄を検知できる	
成功条件：以下のテスト項目が全て期待値通り	
10001	EST-ATTACH-NORMAL-OK
10006	EST-ATTACH-ES-SIG-FORGED-NG

2.13.7 <OFF-T-7>

テストケース名	OFF-T-7
ES-T フォーマットの署名タイムスタンプのタイムスタンプトークンの SignerInfo の署名値の改竄を検知できる	
成功条件：以下のテスト項目が全て期待値通り	
10001	EST-ATTACH-NORMAL-OK
10007	EST-ATTACH-SIGTS-SIG-FORGED-NG

2.13.8 <OFF-T-8>

テストケース名	OFF-T-8
ES-T フォーマットの MessageDigest のハッシュ値の改竄を検知できる	
成功条件：以下のテスト項目が全て期待値通り	
10001	EST-ATTACH-NORMAL-OK
10008	EST-ATTACH-ES-MESSAGEDIGEST-FORGED-NG

2.13.9 <OFF-T-9>

テストケース名	OFF-T-8
ES-T フォーマットの署名タイムスタンプのタイムスタンプトークンの MessageDigest のハッシュ値の改竄を検知できる	
成功条件：以下のテスト項目が全て期待値通り	
10001	EST-ATTACH-NORMAL-OK
10009	EST-ATTACH-SIGTSTST-MESSAGEDIGEST-FORGED-NG

2.13.10 <OFF-T-10>

テストケース名	OFF-T-10
分離署名の ES-T フォーマットを扱える	
成功条件：以下のテスト項目が全て期待値通り	
10010	EST-DETACH-NORMAL-OK

2.14 ES-T オプショナルテストケース

本節では ES-T フォーマットを扱う実装の機能を確認するために行うことが可能はオプショナルテストケースを示す。

2.14.1 <OFF-T-OP-1>

テストケース名	OFF-T-OP-1
OtherSigningCeritificate 属性において SHA-256 である ES-T フォーマットを扱える	
成功条件：以下のテスト項目が全て期待値通り	
10001	EST-ATTACH-NORMAL-OK
20001	EST-OTHERCERT-SHA256-OK

2.14.2 <OFF-T-OP-2>

テストケース名	OFF-T-OP-2
署名タイムスタンプのタイムスタンプトークンのハッシュや署名に SHA-256 アルゴリズムが使われている ES-T フォーマットを扱える	
成功条件：以下のテスト項目が全て期待値通り	
10001	EST-ATTACH-NORMAL-OK
20002	EST-SIGTS-SHA256-OK

2.14.3 <OFF-T-OP-3>

テストケース名	OFF-T-OP-3
署名タイムスタンプのタイムスタンプトークンのハッシュや署名に SHA-512 アルゴリズムが使われている ES-T フォーマットを扱える	
成功条件：以下のテスト項目が全て期待値通り	
10001	EST-ATTACH-NORMAL-OK
20003	EST-SIGTS-SHA512-OK

2.14.4 <OFF-T-OP-4>

テストケース名	OFF-T-OP-4
CMS 署名属性にコンテンツタイムスタンプ属性が含まれる ES-T フォーマットを正しく検証できる	
成功条件：以下のテスト項目が全て期待値通り	
10001	EST-ATTACH-NORMAL-OK
20004	EST-CONTENT-TIMESTAMP-OK

2.14.5 <OFF-T-OP-5>

テストケース名	OFF-T-OP-5
独立署名（並列署名)、即ち signerInfo が 2 つあり、署名に用いたそれらの署名者証明書が同一の信頼点である ES-T フォーマットを正しく検証できる	
成功条件：以下のテスト項目が全て期待値通り	
10001	EST-ATTACH-NORMAL-OK
20005	EST-INDEPENDENT-SIGNATURES-OK

2.14.6 <OFF-T-OP-6>

テストケース名	OFF-T-OP-6
EPES フォーマットに基づく ES-T フォーマットにおいて読み込み時エラーとならないことを検証する。	
成功条件：以下のテスト項目が全て期待値通り	
10001	EST-ATTACH-NORMAL-OK
20006	EST-EPES-WITHOUT-HASHCHECK-OK

備考：署名ポリシーを正しく扱う実装は、テストデータに含まれる署名ポリシーを用いて検証を行う。署名ポリシーを処理しない実装はエラーが発生しないことのみを確認する。

2.14.7 <OFF-T-OP-7>

テストケース名	OFF-T-OP-7
EPES フォーマットに基づく ES-T フォーマットにおいて署名ポリシーのハッシュ値の一致確認を行い正しく署名ポリシーが扱えることを検証する。	
成功条件：以下のテスト項目が全て期待値通り	
10001	EST-ATTACH-NORMAL-OK
20007	EST-EPES-NORMAL-OK
20008	EST-EPES-POLICY-HASH-NOT-MATCH-NG

2.14.8 <OFF-T-OP-8>

テストケース名	OFF-T-OP-8
EPES に基づく ES-T フォーマットにおいて、署名ポリシーの notBefore を正しく扱うことができる。	
成功条件：以下のテスト項目が全て期待値通り	
10001	EST-ATTACH-NORMAL-OK
20007	EST-EPES-NORMAL-OK
20009	EST-EPES-NOT-BEFORE-VIOLATION-NG

2.14.9 <OFF-T-OP-9>

テストケース名	OFF-T-OP-9
EPES に基づく ES-T フォーマットにおいて、署名ポリシーの signerRules の mandatedSignedAttrs を正しく扱うことができる。	
成功条件：以下のテスト項目が全て期待値通り	
10001	EST-ATTACH-NORMAL-OK
20007	EST-EPES-NORMAL-OK
20010	EST-EPES-MANDATED-SIGNEDATTRS-VIOLATION-NG

2.14.10 <OFF-T-OP-11>

テストケース名	OFF-T-OP-10
EPES に基づく ES-T フォーマットにおいて、署名ポリシーへが externalSignedData が TRUE,即ち分離署名を要求しているのに内包署名であるような ES-T データを正しく扱うことができる。	
成功条件：以下のテスト項目が全て期待値通り	
10001	EST-ATTACH-NORMAL-OK
20007	EST-EPES-NORMAL-OK
20011	EST-EPES-EXTERNAL-SIGNEDDATA-VIOLATION-NG

2.15 ES-C オプショナルテストケース

2.15.1 <OFF-C-OP-1>

テストケース名	OFF-C-OP-1
一般的な内包署名の ES-C フォーマットを読み込むことができる	
成功条件：以下のテスト項目が全て期待値通り	
40001	ESC-ATTACH-NORMAL-OK

2.15.2 <OFF-C-OP-2>

テストケース名	OFF-C-OP-2
分離署名の ES-C フォーマットを扱える	
成功条件：以下のテスト項目が全て期待値通り	
40001	ESC-ATTACH-NORMAL-OK
40002	ESC-DETACH-NORMAL-OK

2.16 ES-X Long 標準テストケース

2.16.1 <OFF-X-1>

テストケース名	OFF-X-1
一般的な内包署名の ES-X Long フォーマットを読み込むことができる	
成功条件：以下のテスト項目が全て期待値通り	
50001	ESXL-ATTACH-NORMAL-OK

2.16.2 <OFF-X-2>

テストケース名	OFF-X-2
分離署名の ES-X Long フォーマットを扱える	
成功条件：以下のテスト項目が全て期待値通り	
50002	ESXL-DETACH-NORMAL-OK

2.17 ES-X Long オプショナルテストケース

2.17.1 <OFF-X-OP-1>

テストケース名	OFF-X-OP-1
ES-X Long フォーマットで署名タイムスタンプの検証情報がそのタイムスタンプトークン内に含まれていない場合に別途与えられる検証情報を元に検証できる	
成功条件：以下のテスト項目が全て期待値通り	
50001	ESXL-ATTACH-NORMAL-OK
60001	ESXL-ATTACH-SIGTS-VALIDATIONINFO-NOT-INCLUDED-OK

2.18 ES-A 標準テストケース

2.18.1 <OFF-A-1>

テストケース名	OFF-A-1
ECOM プロファイルに基づく内包署名の第一世代の ES-A フォーマットを扱うことができる	
成功条件：以下のテスト項目が全て期待値通り	
70001	ESA1-ATTACH-NORMAL-OK

2.18.2 <OFF-A-2>

テストケース名	OFF-A-2
ECOM プロファイルに基づく分離署名の第一世代の ES-A フォーマットを扱うことができる	
成功条件：以下のテスト項目が全て期待値通り	
70002	ESA1-DETACH-NORMAL-OK

2.19 ES-A オプショナルテストケース

2.19.1 <OFF-A-OP-1>

テストケース名	OFF-A-OP-1
ETSI TS 101 733 v1.5.1 以降のハッシュ計算法に基づく内包署名の第一世代の ES-A フォーマットを扱うことができる	
成功条件：以下のテスト項目が全て期待値通り	
80001	ESA1-ATTACH-ETSI151-OK

2.19.2 <OFF-A-OP-2>

テストケース名	OFF-A-OP-2
ETSI TS 101 733 v1.5.1 以降のハッシュ計算法に基づく分離署名の第一世代の ES-A フォーマットを扱うことができる	
成功条件：以下のテスト項目が全て期待値通り	
80002	ESA1-DETACH-ETSI151-OK

3．オンライン マトリックス生成・相互検証テストカテゴリ

ある実装が生成した有効な長期署名フォーマットのデータが相互に読み込みおよび検証ができることを確認するためのテストを行う。あらかじめ指定された署名対象データ、証明書、CRL、タイムスタンプサービスを用いて参加企業全ての製品により長期署名フォーマットデータ(ES-T, ES-X Long, ES-A）を生成する。参加企業の各製品において、他社製品の生成したデータが有効であることを検証する。CRL およびタイムスタンプトークンはオンラインで取得する。

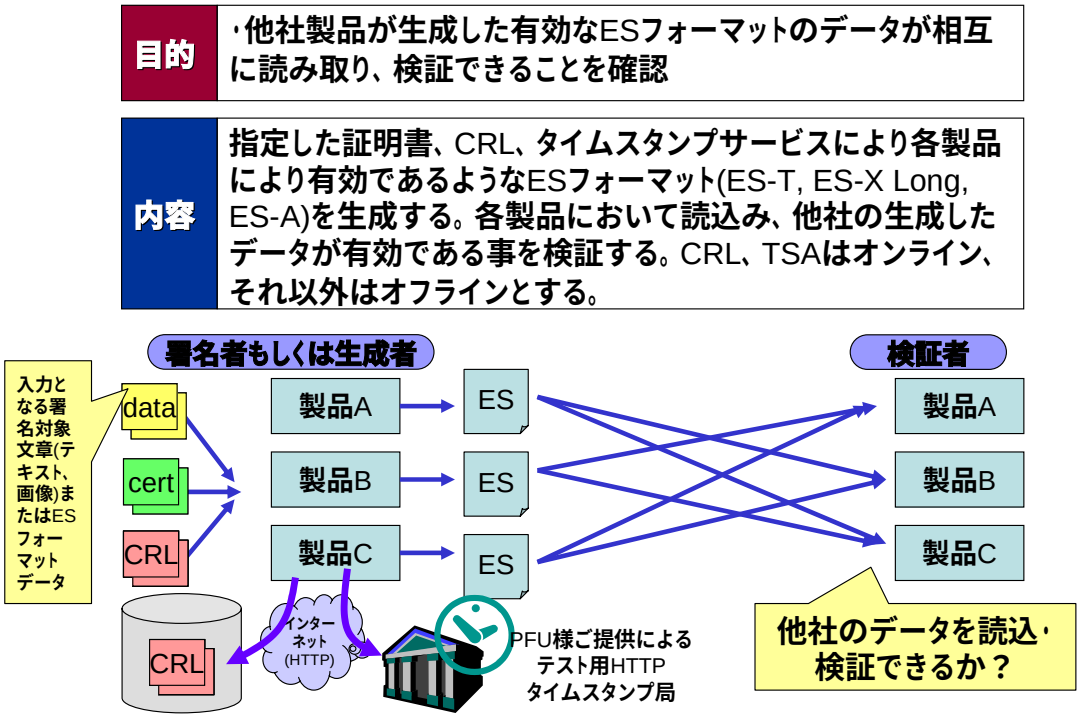


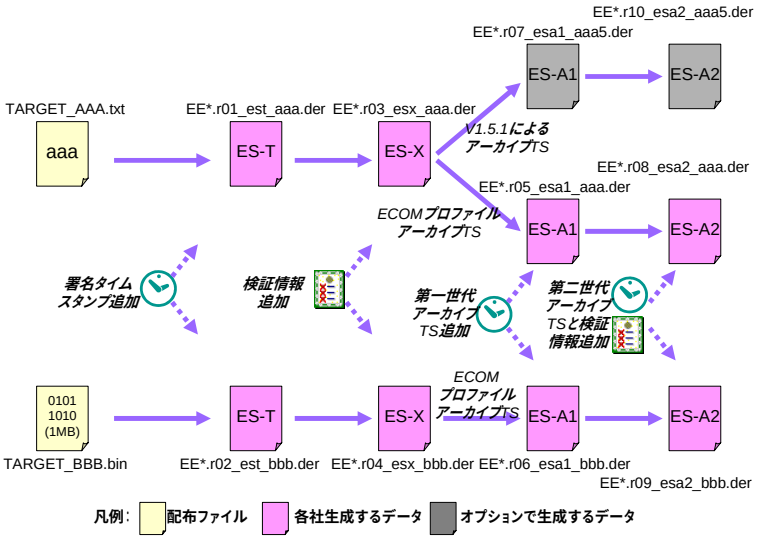
図 付3.3-1 オフライン検証テスト

テストケースは 10 項目ある。オプショナルテストとして、ETSI TS 101 733 v1.5.1 に基づく新しいアーカイブタイムスタンプのハッシュ値の計算方法を用いたテストもある。

参加企業の実装では、ES-C フォーマットを出力できる製品が少ないために ES-C フォーマットを実験対象から外した。また、文書管理製品で ES-X Long の状態で ES-A とは異なる別のセキュアなアーカイビング方法を採用して保存する製品もあるため、ES-X Long はテスト対象に加えた。

3.1 生成するデータ

小さいサイズのテキストデータを内包署名、1MB 程度のバイナリデータファイルを分離署名として、それぞれ ES-T フォーマット、ES-X Long フォーマット、第一世代とその次の世代の ES-A フォーマットを生成する。オプショナルテストとして ETSI TS 101 733 v1.5.1 以降のアーカイブタイムスタンプハッシュ対象計算方法（4.1 節参照）を用いたデータ生成を行ってよい。



タイムスタンプトークンの取得は今回のテスト用に提供されたタイムスタンプ局を使用することとする。失効情報の取得は証明書の cRLDistributionPoints 拡張に記載された URL より取得してもよいし、テストデータに含まれるファイルを使用してもよい。

3.2 テストの準備

- **CRL のための設定**

CRL を取得するために、検証環境におけるインターネット接続環境の準備を行う。署名用認証局とタイムスタンプ局用認証局の CRL 発行間隔は１日となっている。

- **トラストアンカの設定**

テストスイートで配布されるオフラインテスト用の署名者用ルート証明書、TSA 用ルート証明書をトラストアンカとして設定する。

3.3 テストの実施（生成）

- **署名対象データの設定**

内包署名の場合にはファイル'TARGET_AAA.txt'（ファイルの内容は“aaa”という 3 文字 3 バイトの文字列のみ）、分離署名の場合には'TARGET_BBB.bin'（ファイルの内容は 0x01-0x09,0x00 の繰り返し 1024000 バイトのバイナリファイル）を設定する。

- **データ生成の実施**
- **生成したデータを全てアーカイブし、参加企業の検証者に送信する**

3.4 テストの実施（検証）

- **署名対象データの設定**

内包署名の場合にはファイル'TARGET_AAA.txt'（ファイルの内容は“aaa”という 3 文字 3 バイトの文字列のみ）、分離署名の場合には'TARGET_BBB.bin'（ファイルの内容は 0x01-0x09,0x00 の繰り返し 1024000 バイトのバイナリファイル）を設定する。

- **検証時刻の設定**

検証時刻はフォーマット毎に異なる。フォーマットに応じ検証時刻の設定を行う。検証可能な現在時刻の範囲はUTC2002年1月1日0時0分0秒よりUTC2035年12月31日23時59分59秒までとし、各証明書、CRLもまたこれが可能なように設定されている。

- **検証対象の長期署名フォーマットデータの設定**
- **検証の実施**

3.5 テストケース

以下のレギュレーションに関する記述で「◎」印は必ず満足しなければならないルールとし、それ以外は可能ならば準拠しなければならないルールとする。

3.5.1 <ON-T-1> データ内包型ES-T生成・相互検証テストケース

以下のレギュレーションで各製品ES-Tデータを生成する。

- **◎署名対象が文字列“aaa”(x61 x61 x61)**
- **◎内包署名とする。(EncapContentInfoに“aaa”を含む)**
- **MessageDigestはSHA1**
- **署名アルゴリズムはSHA1withRSA**
- **BESフォーマットよりES-Tを生成**

各製品を用いこれが有効であることを検証する。

3.5.2 <ON-T-2> データ分離型ES-T生成・相互検証テストケース

テストケース<ON-T-1>をベースに以下のレギュレーションを加えたもので各製品ES-Tデータを生成する。

- **◎署名対象が1MB程度のデータファイル**
- **◎分離署名とする。(EncapContentInfoに署名対象を含まない)**

3.5.3 <ON-X-1> データ内包型ES-X Long生成・相互検証テストケース

以下のレギュレーションで各社ES-X Longデータを生成する。

- **◎<ON-T-1>で生成されたES-Tデータを対象とし生成**
- **<ON-T-1>のES-Tを生成後、48時間以降経過した後にES-X Longを生成する**
- **署名、および署名タイムスタンプの証明書検証情報の格納内容、方法はECOMプロファイルに基づく**

各製品を用いこれが有効であることを検証する。

3.5.4 <ON-X-2> データ分離型ES-X Long生成・相互検証テストケース

テストケース<ON-X-1>を基本に以下のレギュレーションを加えたもので各製品によりES-X Longデータを生成する。

- **◎<ON-T-2>で生成されたES-Tデータを対象とし生成**

- **◎署名対象が 1MB 程度のデータファイル**
- **◎分離署名とする。（EncapContentInfo に署名対象を含まない）**

各製品を用いこれが有効であることを検証する。

3.5.5 <ON-A1-1> データ内包型第一世代 ES-A 生成・相互検証テストケース

以下のレギュレーションで各製品により ES-A データを生成する。

- **◎<ON-X-1>で生成された ES-X Long データ、もしくは製品が未対応ならば、<ON-T-1>で生成された ES-T データを対象とし生成**
- **アーカイブタイムスタンプの生成・検証方法は ECOM プロファイル（RFC 3126 or ESTI TS 101 733 v1.4.0 以前の方法）に基づく**
- **署名、および署名タイムスタンプの証明書検証情報情報の格納内容、方法は ECOM プロファイルに基づく**

各製品を用いこれが有効であることを検証する。

3.5.6 <ON-A1-2> データ分離型第一世代 ES-A 生成・相互検証テストケース

<ON-A1-1>のレギュレーションを基本に以下のレギュレーションを加えたもので各製品により ES-A データを生成する。

- **◎<ON-X-2>で生成された ES-X Long データ、もしくはこれに製品が未対応ならば、<ON-T-2>で生成された ES-T データを対象とし生成**
- **◎署名対象が 1MB 程度のデータファイル**
- **◎分離署名とする。（EncapContentInfo に署名対象を含まない）**

各製品を用いこれが有効であることを検証する。

3.5.7 <ON-A1-3> データ内包型第一世代新方式 ES-A 生成・相互検証テストケース（OP）

本テストケースはオプショナルである。以下のレギュレーションで各製品により ES-A データを生成する。

- **◎<ON-X-1>で生成された ES-X Long データを対象とし生成**
- **◎アーカイブタイムスタンプの生成・検証方法は ESTI TS 101 733 v1.5.1 に基づき、ECOM 長期署名フォーマット SWG で合意を得た正規化方法を用いた計算方法を用いる。**
- **署名、および署名タイムスタンプの証明書検証情報の格納内容、方法は ECOM プロファイルに基づく**

各社製品を用いこれが有効であることを検証する。

3.5.8 <ON-A2-1> データ内包型第二世代 ES-A 生成・相互検証テストケース

以下のレギュレーションで各製品 ES-A データを生成する。

- **◎<ON-A1-1>で生成された ES-A を対象とし生成（署名延長）**
- **◎アーカイブタイムスタンプの生成・検証方法は ECOM プロファイル（RFC 3126 or ESTI TS 101 733 v1.4.0 以前の方法）に基づく**

- **署名、および署名およびアーカイブタイムスタンプ属性の証明書検証情報の格納内容、方法は ECOM プロファイルに基づく**

各製品を用いこれが有効であることを検証する。

3.5.9 <ON-A2-2> データ分離型第二世代 ES-A 生成・相互検証テストケース

<ON-A2-1>のレギュレーションを基本に以下のレギュレーションを加えたもので各製品 ES-A データを生成する。

- **◎<ON-A1-2>で生成された ES-A データを対象とし生成（署名延長）**
- **◎署名対象が 1MB 程度のデータファイル**
- **◎分離署名とする。（EncapContentInfo に署名対象を含まない）**

各社製品を用いこれが有効であることを検証する。

3.5.10 <ON-A2-3> データ内包型第二世代新方式 ES-A 生成・相互検証テストケース（OP）

本テストケースはオプショナルである。以下のレギュレーションで各社 ES-A データを生成する。

- **◎<ON-A1-3>で生成された ES-A データを対象とし生成**
- **◎アーカイブタイムスタンプの生成・検証方法は ESTI TS 101 733 v1.5.1 に基づき、ECOM 長期署名フォーマット SWG で合意を得た正規化方法を用いた計算方法を用いる。**
- **署名、および署名タイムスタンプの証明書検証情報の格納内容、方法は ECOM プロファイルに基づく**

各社製品を用いこれが有効であることを検証する。

4．参考資料

4.1 ECOM オプショナルテストで用いられる ETSI TS 101 733 v1.5.1 以降によるアーカイブハッシュ計算法

H17 年度に ECOM 長期署名フォーマット普及 SWG が策定した CAdES フォーマットプロファイルでは、RFC 3126 や ETSI TS 101 733 v1.4.2 で用いられていたアーカイブタイムスタンプのハッシュ対象計算方法を採用している。これは、ETSI TS 101 733 v1.5.1 で記載されている方法を相互運用させるためには、データの正規化方法や署名延長中に他の CMS 非署名属性が加わった場合の処理方法、直列署名（CounterSignature）の場合の処理などが記述されておらず、日本発のプロファイルとして ETSI と調整を必要としたことから従来方法を採用している。

今回の実証実験のために、現時点で妥当と思われる v1.5.1 ベースの計算方法を仮決めし、この方法を元にオプショナルテストとして v1.5.1 ベースのテストケースを作成した。当然の事ながら、今後、調整されるであろう v1.5.1 ベースの標準やプロファイルの改変に対して直接的に影響を与えるものではなく、単に議論のきっかけを与えたに過ぎない。

標準よりハッシュ対象は下図の通りである。

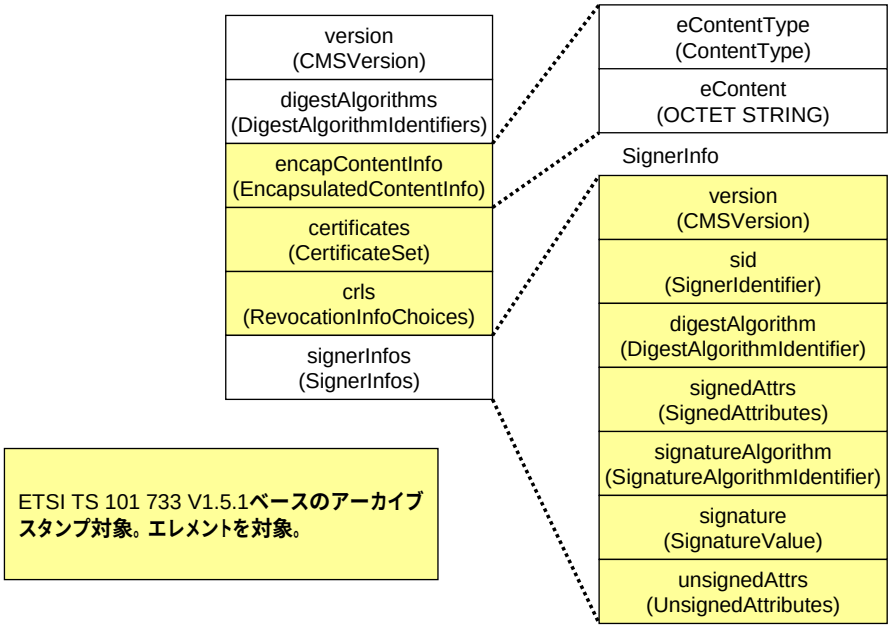


図 付 3.4-1 ETSI TS 101 733 v1.5.1 以降のアーカイブハッシュ対象

その計算方法は以下の通りとする。

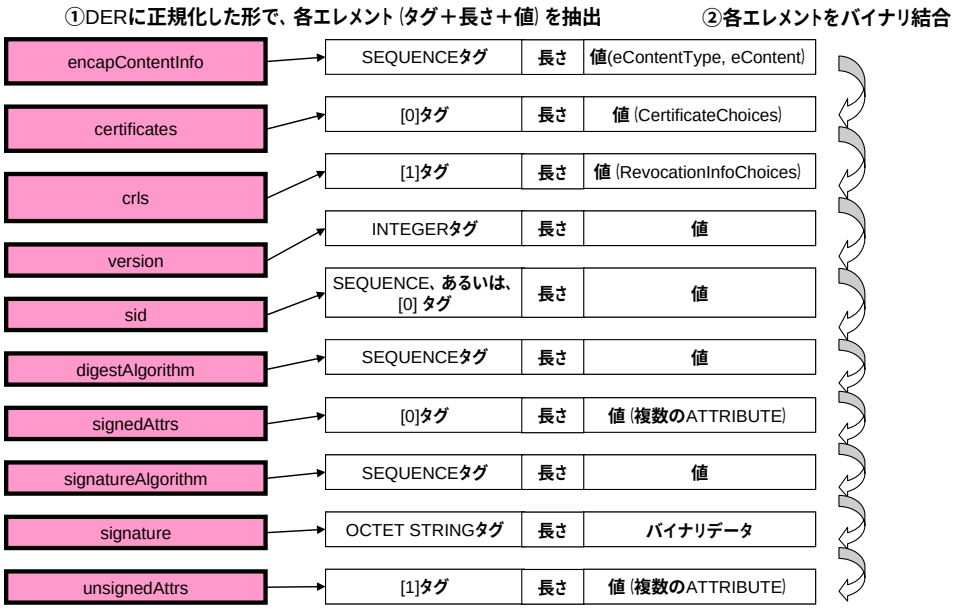
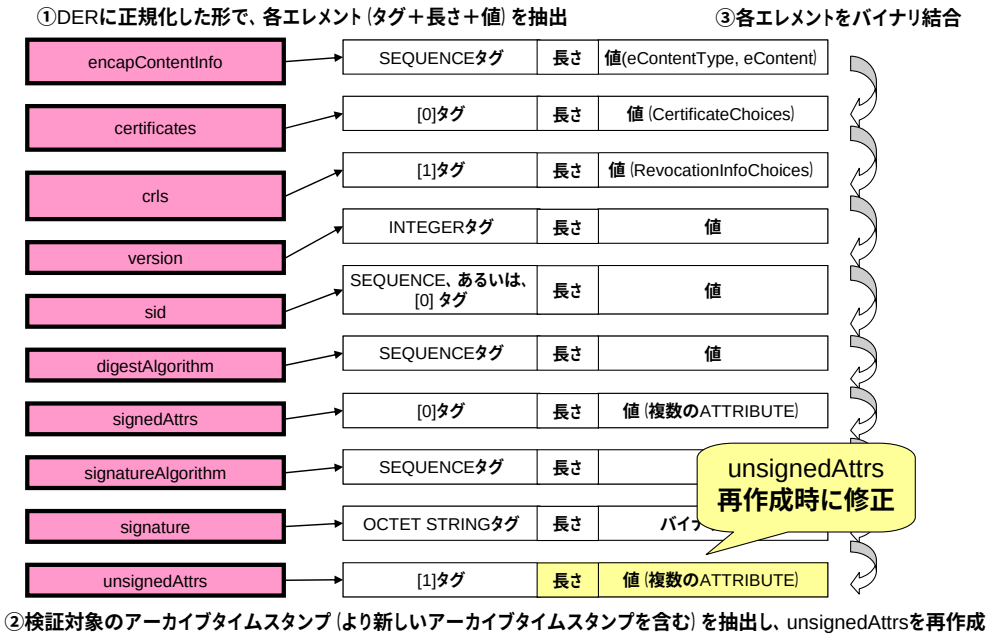


図 付 3.4-2 本実証実験のために定めた v1.5.1 ベースのハッシュ計算法

ハッシュ対象に ASN.1 TLV（タグ、長さ、値）構造のタグおよび長さ部分もハッシュ対象に含まれることに注意しなければならない。また、ContextSpecific タグ（[0]や[1]など）は、SET

や SEQUENCE に正規化することなく、そのままハッシュ対象として加えなければならない。

各アーカイブタイムスタンプの検証時には、下図の手順でハッシュ対象を生成する。特に注意しなければならないのは CMS 非署名属性の再計算についてである。例えば CMS 非署名属性中に n 個のアーカイブスタンプがあったとして、その途中の i 番目のアーカイブタイムスタンプを検証する際には、CMS 非署名属性において 1 番目から i 番目アーカイブタイムスタンプ属性の一つ手前の属性までを要素とする新しい CMS 非署名属性の ContextSpecific '[1]'の ASN.1 SET 構造を再生成し、その結果 ASN.1 TLV 構造の長さバイトも再計算を行い、これをハッシュ対象としなければならない。



5．付録：実験用データプロファイル

本節では実証実験で用いられるデータのプロファイルを示す。

5.1　実験用長期署名フォーマットデータのプロファイル

長期署名フォーマットのデータは全て CMS SignedData フォーマットに基づいており、その中で各フォーマット毎に signedAttributes フィールドおよび unsignedAttributes フィールドに必要となる CMS 属性が異なる。

5.1.1 BES（Basic Electronic Signature）

フィールド	値
version	v3(3)
digestAlgorithms	{ SHA1 }
encapContentInfo	内包/分離はテスト項目に依存
certificates	署名者証明書のみ
crls	無
signerInfos	有(要素数=1)
signerInfo	160bit
version	v3(3)
sid	署名者証明書のsubjectKeyIdentifier拡張の値と同じ
digestAlgorithm	SHA1
signedAttrs	有
contentInfo	=data(1.2.840.113549.1.7.1)
messageDigest	有
eSSSigningCertificate	有(発行者名、シリアル番号、SHA1フィンガープリント)
signatureAlgorithm	SHA1withRSA
signature	署名値
unsignedAttrs	無

5.1.2 EPES（Explicit Policy-based Electronic Signature）

フィールド	値
version	v3(3)
digestAlgorithms	{ SHA1 }
encapContentInfo	内包/分離はテスト項目に依存
certificates	署名者証明書のみ
crls	無
signerInfos	有(要素数=1)
signerInfo	160bit
version	v3(3)
sid	署名者証明書のsubjectKeyIdentifier拡張の値と同じ
digestAlgorithm	SHA1
signedAttrs	有
contentInfo	=data(1.2.840.113549.1.7.1)
messageDigest	有
sigPolicyId	有(SHA1フィンガープリント)
eSSSigningCertificate	有(発行者名、シリアル番号、SHA1フィンガープリント)
signatureAlgorithm	SHA1withRSA
signature	署名値
unsignedAttrs	無

5.1.3 ES-T

フィールド	値
version	v3(3)
digestAlgorithms	{ SHA1 }
encapContentInfo	内包/分離はテスト項目に依存
certificates	署名者証明書のみ
crls	無
signerInfos	有(要素数=1)
signerInfo	160bit
version	v3(3)
sid	署名者証明書のsubjectKeyIdentifier拡張の値と同じ
digestAlgorithm	SHA1
signedAttrs	有
contentInfo	=data(1.2.840.113549.1.7.1)
messageDigest	有
eSSSigningCertificate	有(発行者名、シリアル番号、SHA1フィンガープリント)
signatureAlgorithm	SHA1withRSA
signature	署名値
unsignedAttrs	有
signatureTimeStamp	トークンは実験用データプロファイルに従う

5.1.4 ES-X Long

フィールド	値
version	v3(3)
digestAlgorithms	{ SHA1 }
encapContentInfo	内包/分離はテスト項目に依存
certificates	署名者証明書のみ
crls	無
signerInfos	有(要素数=1)
signerInfo	160bit
version	v3(3)
sid	署名者証明書のsubjectKeyIdentifier拡張の値と同じ
digestAlgorithm	SHA1
signedAttrs	有
contentInfo	=data(1.2.840.113549.1.7.1)
messageDigest	有
eSSSigningCertificate	有(発行者名、シリアル番号、SHA1フィンガープリント)
signatureAlgorithm	SHA1withRSA
signature	署名値
unsignedAttrs	有
signatureTimeStamp	トークンは実験用データプロファイルに従う(検証情報を含む)
completeCertificateRefs	ECOMプロファイルに従う
completeRevocationRefs	ECOMプロファイルに従う
certificateValues	ECOMプロファイルに従う
revocationValues	ECOMプロファイルに従う

5.1.5 ES-A（第一世代）

フィールド	値
version	v3(3)
digestAlgorithms	{ SHA1 }
encapContentInfo	内包/分離はテスト項目に依存
certificates	署名者証明書のみ
crls	無
signerInfos	有(要素数=1)
signerInfo	160bit
version	v3(3)
sid	署名者証明書のsubjectKeyIdentifier拡張の値と同じ
digestAlgorithm	SHA1
signedAttrs	有
contentInfo	=data(1.2.840.113549.1.7.1)
messageDigest	有
eSSSigningCertificate	有(発行者名、シリアル番号、SHA1フィンガープリント)
signatureAlgorithm	SHA1withRSA
signature	署名値
unsignedAttrs	有
signatureTimeStamp	トークンは実験用データプロファイルに従う(検証情報を含む)
completeCertificateRefs	ECOMプロファイルに従う
completeRevocationRefs	ECOMプロファイルに従う
certificateValues	ECOMプロファイルに従う
revocationValues	ECOMプロファイルに従う
archiveTimeStamp	トークンは実験用データプロファイルに従う

5.1.6 ES-A（第二世代以降）

フィールド	値
version	v3(3)
digestAlgorithms	{ SHA1 }
encapContentInfo	内包/分離はテスト項目に依存
certificates	署名者証明書のみ
crls	無
signerInfos	有(要素数=1)
signerInfo	160bit
version	v3(3)
sid	署名者証明書のsubjectKeyIdentifier拡張の値と同じ
digestAlgorithm	SHA1
signedAttrs	有
contentInfo	=data(1.2.840.113549.1.7.1)
messageDigest	有
eSSSigningCertificate	有(発行者名、シリアル番号、SHA1フィンガープリント)
signatureAlgorithm	SHA1withRSA
signature	署名値
unsignedAttrs	有
signatureTimeStamp	トークンは実験用データプロファイルに従う(検証情報を含む)
completeCertificateRefs	ECOMプロファイルに従う
completeRevocationRefs	ECOMプロファイルに従う
certificateValues	ECOMプロファイルに従う
revocationValues	ECOMプロファイルに従う
archiveTimeStamp1	トークンは実験用データプロファイルに従う(検証情報を含む)
archiveTimeStamp2 ...	トークンは実験用データプロファイルに従う(署名延長)

5.2 実験用タイムスタンプトークンのプロファイル

5.2.1 TimeStampToken

TimeStampToken は CMS SignedData の構造となっている。ECOM プロファイルの ES-X Long, ES-A の検証情報の格納方法の定義に従い、certificates, crls フィールドに検証情報を持つ場合がある。

フィールド	値
version	v3(3)
digestAlgorithms	{ SHA1 }
encapContentInfo	前述TSTInfoプロファイルに従う
certificates	ECOMプロファイルにより検証情報としてTSA証明書およびパスを含みうる
crls	ECOMプロファイルにより検証情報として全てのCRLを含みうる
signerInfos	有(要素数=1)
signerInfo	160bit
version	v1(1)
sid	TSA証明書のIssuerAndSerialNumber
digestAlgorithm	SHA1
signedAttrs	有
contentInfo	=tSTInfo(1.2.840.113549.1.9.16.1.4)
messageDigest	有
eSSSigningCertificate	有(発行者名、シリアル番号、SHA1フィンガープリント)
signatureAlgorithm	SHA1withRSA
signature	署名値
unsignedAttrs	無

5.2.2 TSTInfo

フィールド	値
バージョン	v1(1)
policy	TSAPolicyId=0.1.2.3.4.5
messageImprint	有
hashAlgorithm	SHA1
hashedMessage	160bit
serialNumber	値はTSA証明書のシリアル番号と同じとする(※1)
genTime	GeneralizedTime(小数点以下最大3桁を含む)
accuracy	500ミリ秒
ordering	TRUE
nonce	0x1234567890(固定)
tsa	directoryName=TSA証明書の主体者名
extensions	無

※1：本来は該当 TSA より発行されたトークンのシリアル番号となるがテスト上 TSA からは 1 つのトークンしか発行されないので便宜上 TSA 証明書のシリアル番号と同じとし、テスト項目番号がすぐにわかるようにする。

5.3 実験用証明書のプロファイル

5.3.1 実験用証明書の共通のプロファイル

フィールド	値
バージョン	V3
シリアル番号	5バイトのASN.1 INTEGER(※1)
署名アルゴリズム	SHA1withRSA
発行者DN	PrintableString(全てのDNはPrintableStringとする)
有効期限	UTCTime(使用される時刻は2000/1/1 0:00:00～2035/12/31 23:59:59とする)
主体者DN	PrintableString
公開鍵情報	有
X.509拡張	有
keyUsage	有

5.3.2 RootCA 証明書のプロファイル

フィールド	値	クリティカル
バージョン	V3	
シリアル番号	有	
署名アルゴリズム	SHA1withRSA	
発行者DN	PrintableString	
有効期限	UTCTime	
主体者DN	PrintableString	
公開鍵情報	2048bit	
X.509拡張	有	
keyUsage	CertSign, CRLSign	TRUE
subjectKeyIdentifier	有 SHA1-160bit	FALSE
basicConstraints	有	TRUE
CAフラグ	TRUE	-

5.3.3 SubCA 証明書のプロファイル

フィールド	値	クリティカル
バージョン	V3	
シリアル番号	有	
署名アルゴリズム	SHA1withRSA	
発行者DN	PrintableString	
有効期限	UTCTime	
主体者DN	PrintableString	
公開鍵情報	1024bit	
X.509拡張	有	
keyUsage	CertSign, CRLSign	TRUE
subjectKeyIdentifier	有 SHA1-160bit	FALSE
authorityKeyIdentifier	有	FALSE
keyIdentifier	有 SHA1-160bit	-
basicConstraints	有	TRUE
CAフラグ	TRUE	-
cRLDistributionPoints	有	FALSE
DistPt.fullName.URI	http://配布ホスト/**/*.crl	-

5.3.4 署名者用 End Entity 証明書のプロファイル

フィールド	値	クリティカル
バージョン	V3	
シリアル番号	有	
署名アルゴリズム	SHA1withRSA	
発行者DN	PrintableString	
有効期限	UTCTime	
主体者DN	PrintableString	
公開鍵情報	1024bit	
X.509拡張	有	
keyUsage	digitalSignature, nonRepudiation	TRUE
basicConstraints	有(空シーケンス)	FALSE
CAフラグ	無	-
subjectKeyIdentifier	有 SHA1-160bit	FALSE
authorityKeyIdentifier	有	FALSE
keyIdentifier	有 SHA1-160bit	-
cRLDistributionPoints	有	FALSE
DistPt.fullName.URI	http://配布ホスト/**/*.crl	-

5.3.5 TSA 証明書のプロファイル

フィールド	値	クリティカル
バージョン	V3	
シリアル番号	有	
署名アルゴリズム	SHA1withRSA	
発行者DN	PrintableString	
有効期限	UTCTime	
主体者DN	PrintableString	
公開鍵情報	1024bit	
X.509拡張	有	
keyUsage	digitalSignature, nonRepudiation	TRUE
subjectKeyIdentifier	有 SHA1-160bit	FALSE
authorityKeyIdentifier	有	FALSE
keyIdentifier	有 SHA1-160bit	-
extKeyUsage	1.3.6.1.5.5.7.3.8(timeStamping)	TRUE
cRLDistributionPoints	有	FALSE
DistPt.fullName.URI	http://配布ホスト/**/*.crl	-

5.3.6 オンライン TSA 用 RootCA 証明書のプロファイル

フィールド	値	クリティカル
バージョン	V3	
シリアル番号	有	
署名アルゴリズム	SHA1withRSA	
発行者DN	PrintableString	
有効期限	UTCTime	
主体者DN	PrintableString	
公開鍵情報	2048bit	
X.509拡張	有	
keyUsage	CertSign, CRLSign	TRUE
subjectKeyIdentifier	有 SHA1-160bit	FALSE
authorityKeyIdentifier	有	FALSE
keyIdentifier	有 SHA1-160bit	-
authorityCertIssuer	directoryName(PrintableString)	-
authorityCertSerialNumber	(0x00)	-
basicConstraints	有	FALSE
CAフラグ	TRUE	-

5.3.7 オンライン TSA 証明書のプロファイル

フィールド	値	クリティカル
バージョン	V3	
シリアル番号	有	
署名アルゴリズム	SHA1withRSA	
発行者DN	PrintableString	
有効期限	UTCTime	
主体者DN	PrintableString	
公開鍵情報	1024bit	
X.509拡張	有	
keyUsage	digitalSignature, nonRepudiation	TRUE
subjectKeyIdentifier	有 SHA1-160bit	FALSE
authorityKeyIdentifier	有	FALSE
keyIdentifier	有 SHA1-160bit	-
authorityCertIssuer	directoryName(PrintableString)	-
authorityCertSerialNumber	(0x00)	-
basicConstraints	有(空シーケンス)	FALSE
CAフラグ	無	-
extKeyUsage	1.3.6.1.5.5.7.3.8(timeStamping)	TRUE
cRLDistributionPoints	有	FALSE
DistPt.fullName.URI	https://配布ホスト/**/*.crl	-

5.3.8 オンライン/オフライン/署名者/TSA 共通 CRL プロファイル

フィールド	値	クリティカル
バージョン	V2(1)	
署名アルゴリズム	SHA1withRSA	
発行者DN	PrintableString	
thisUpdate	UTCTime	
nextUpdate	UTCTime	
revokedCertificate		
userCertificate	失効する証明書のシリアル番号	
revocationDate	UTCTime	
crlEntryExtensions		
cRLReason		FALSE
X.509拡張	有	
cRLNumber		FALSE

5.4　実験用署名ポリシーのプロファイル

フィールド	値
signPolicyHashAlg	SHA1
signPolicyInfo	有
signPolicyIdentifier	1.2.3.4.5*
dateOfIssue	2001.01.01
policyIssuerName	ou=SIGNATURE-POLICY-AUTHORITY,o=ECOM,c=JP
fieldOfApplication	"for …" テスト用ポリシーとしてのメモ
signatureValidationPolicy	
signingPeriod	
notBefore	有
notAfter	無
commonRules	
signerAndVerifierRules[0]	
signerRules	
externalSignedData?	無
mandatedSignedAttr	messageDigest, sigPolicyId
mandatedUnsignedAttr	signatureTimeStamp
mandatedCertificateRef?	無
mandatedCertificateInfo?	無
signPolExtensions?	無
verifierRules	
mandatedUnsignedAttr	空シーケンス
signPolExtensions?	無
signingCertTrustCondition[1]	
signerTrustTrees	署名者用CA証明書
signerRevReq	EE=crlCheck(0), CA=crlCheck(0)
timeStampTrustCondition[2]	
ttsCertificateTrustTrees[0]?	TSA用CA証明書
ttsRevReq[1]?	EE=crlCheck(0), CA=crlCheck(0)
attributeTrustCondition[3]	無
algorithmConstraintSet[4]	無
commitmentRules	空シーケンス
signPolExtensions	無
signPolExtensions	無
signPolicyHash	無

Ⅳ．長期署名フォーマット
ECOM 相互運用実証実験
XAdES テストケース設計書

2006 年 3 月

次世代電子商取引推進協議会（ECOM）

セキュリティ WG

長期署名フォーマット普及 SWG

長期署名フォーマット相互運用性実証実験プロジェクト

目　次

1．はじめに

本仕様書は、ECOM セキュリティ WG 長期署名フォーマット普及 SWG の長期署名フォーマット相互運用性実証実験プロジェクトにおいて実施される実証実験の XAdES 長期署名フォーマットに関するテスト内容について記述したものである。

1.1 本書における表記

本仕様書では、以下の表記を用いることとする（表 付 4.1-1)。

表 付 4.1-1 ASN.1 データのエンコード方法に関する URI

表記	説明
<…>	テスト項目
<…OK>	検証結果の期待値が有効であるテスト項目
<…NG>	検証結果の期待値が無効であるテスト項目
[…]	参考文献

1.2 テストの構成

CAdES テストケース設計書に記述されたものと同様の構成とする。

2．オフライン共通データ検証テストカテゴリ

ECOM プロファイルに基づく共通の XAdES フォーマットデータを用いて、実験者のシステムや製品でそれらが正しく検証できるかどうかを確認する。テストツールより生成された XAdES フォーマットのデータ（XAdES-T、XAdES-A)、証明書、CRL、署名対象データをもとに、検証結果が期待値と一致するかどうかを確認する。

2.1 テストの準備

テスト実施する際は、以下の項目の準備が必要となる。

- CRL の設定
  証明書検証時にオンラインで CRL を取得する場合には、検証環境におけるインターネット接続環境の準備。実験期間終了後にはホスト名を同じくする HTTP リポジトリの立ち上げと設定。もしくは、ファイルによる CRL の設定。
- トラストアンカの設定
  テストスイートで配布されるオフラインテスト用の署名者用ルート証明書、TSA 用ルート証明書をトラストアンカとして設定する。

2.2　テストの実施

本節では、テストの実施時の設定や条件などを説明する。

・　署名対象データの設定

内包型署名の場合は、署名対象文字列を"aaa"とし XML 署名の形式として enveloping 形式で署名対象文字列を指定する。ただし、XML 署名の Object 要素として格納するため、base64 で encode された値（YWFh）で格納するもとする。内包型署名の場合の XML 署名文書の例を以下に示す（リスト 1）。

リスト 1：内包型署名の場合の XML 署名文書の例

```
<?xml version="1.0" encoding="UTF-8" ?>
<ds:Signature Id="Signature-ID1" xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
  <ds:SignedInfo> ……</ds:SignedInfo>
  <ds:SignatureValue>……</ds:SignatureValue>
  <ds:KeyInfo >……</ds:KeyInfo >
  <ds:Object Encoding="http://www.w3.org/2000/09/xmldsig#base64"
             Id="signdata">YWFh</ds:Object>
  <ds:Object xmlns:xa="http://uri.etsi.org/01903/v1.3.1#">……</ ds:Object >
</ds:Signature>
```

base64 でエンコードされたテキスト”aaa”

分離署名の場合には、'TARGET_BBB.bin ”（ファイルの内容は、0x01-0x09,0x00 の繰り返しで 1024000 バイトのバイナリファイル）を設定する。

・　検証時刻の設定

検証時刻はフォーマット毎に異なる。フォーマットに応じ検証時刻の設定を行う。検証可能な現在時刻の範囲は UTC2002 年 1 月 1 日 0 時 0 分 0 秒より UTC2035 年 12 月 31 日 23 時 59 分 59 秒までとし、各証明書や CRL もまたこれが可能なように設定されている。

・　検証対象の長期署名フォーマットデータの設定

テストスィートにおいて長期署名フォーマットの検証対象テストデータは、“<テストケース名>-V131.xml ”というファイル名となっており、各テスト項目ごとに別々のディレクトリに保存されている。

・　検証の実施

実施すべき全てのテスト項目について実施する。署名対象データのハッシュ値を base64 でエンコードしたものは以下の通り。

"aaa"　　：**fiQN5O+x7Qj6CNOAY/amqRRiqBU=**

TARGET_BBB.bin　：**gpGOaOwroxRJGyeXw7tHFbrgtxM=**

2.3　テストデータに共通の情報

・　有効期限の時刻は、例外ケースを除き 00:00:00 から 23:59:59 に統一する。

・　署名時刻、タイムスタンプ時刻は例外ケースを除き 12:00:00 に統一する

・ 時刻の表記は、特に断りのない限り、UTC 時刻とする。

2.4 XAdES-T フォーマット標準テスト

2.4.1 <XAdEST -ATTACH-NORMAL-OK 10001>

署名者証明書および署名タイムスタンプの TSA 証明書が有効期間内にあり共に失効しない場合、XAdES-T データが有効であることを検証する。表 付 4.2-1 にテスト結果の期待値とテスト時に利用する時刻や証明書、CRL に関するテストパラメータを示す。

表 付 4.2-1 <XAdEST -ATTACH-NORMAL-OK 10001>におけるテスト結果の期待値とテストパラメータ

期待値	有効（valid）
署名を実施したとする時刻	2001.1.1 12:00:00
サイニングタイム属性の時刻	属性なし
署名タイムスタンプの時刻	2001.1.1 12:00:00
署名者証明書の有効期限	2001.1.1～2035.12.31
署名者証明書の検証 CRL	2001.1.2 00:00:00～2001.1.3 23:59:59
署名タイムスタンプ TSA 証明書	2001.1.1～2035.12.31
署名TS TSA証明書検証CRL	2001.1.2 00:00:00～2001.1.3 23:59:59

2.4.2 <XAdEST -ATTACH-EXPIERED-NG 10002>

署名タイムスタンプの TSA 証明書は有効であるが、署名証明書が期限切れの時点で署名タイムスタンプを付した場合、署名者証明書を検証する CRL に記載されていないとき XAdES データが無効であることを検証する。表 3 にテスト結果の期待値とテスト時に利用する時刻や証明書、CRL に関するテストパラメータを示す。

表 付 4.2-2 <XAdEST -ATTACH-EXPIERED-NG 10002>におけるテスト結果の期待値とテストパラメータ

期待値	無効（invalid）
署名を実施したとする時刻	2001.1.3 12:00:00
サイニングタイム属性の時刻	属性なし
署名タイムスタンプの時刻	2001.1.3 12:00:00
署名者証明書の有効期限	2001.1.1 00:00:00～2001.1.1 23:59:59
署名者証明書の検証 CRL	2001.1.2 00:00:00～2000.1.2 23:59:59
署名タイムスタンプ TSA 証明書	2001.1.1～2035.12.31
署名TS TSA証明書検証CRL	2001.1.1 00:00:00～2035.12.31 23:59:59

2.4.3 <XAdEST -ATTACH-REVOKED-NG 10003>

署名者証明書、署名タイムスタンプの TSA 証明書が有効期間内にあり、サイニングタイム属性

の時刻および署名タイムスタンプ時刻において、署名者証明書が失効して CRL に記載されている場合、ES-T データが無効であることを検証する。表 4 にテスト結果の期待値とテスト時に利用する時刻や証明書、CRL に関するテストパラメータを示す。

表 付 4.2-3 <XAdEST -ATTACH-REVOKED-NG 10003>におけるテスト結果の期待値とテストパラメータ

期待値	無効（invalid）
署名を実施したとする時刻	2001.1.2 12:00:00
サイニングタイム属性の時刻	2001.1.2 12:00:00
署名タイムスタンプの時刻	2001.1.2 12:00:00
署名者証明書の有効期限	2001.1.1～2035.12.31
署名者証明書の検証 CRL	2001.1.4 00:00:00～2001.1.4 23:59:59
署名者証明書 CRL 中の失効日時	2005.1.1 12:00
署名タイムスタンプ TSA 証明書	2001.1.1～2035.12.31
署名TS TSA証明書検証CRL	2001.1.4 00:00:00～2001.1.4 23:59:59

2.4.4 <XAdEST -ATTACH-SIGTIME-REVOKED-OK 10004>

署名者証明書、署名タイムスタンプの TSA 証明書が有効期間内にあり、署名タイムスタンプ時刻では失効していないが、サイニング属性の時刻において署名者証明書が失効して CRL に記載されている場合、署名時刻は無視し、署名タイムスタンプ時刻での証明書有効性を検証することにより、ES-T データが有効であることを検証する。表 5 にテスト結果の期待値とテスト時に利用する時刻や証明書、CRL に関するテストパラメータを示す。

表 付 4.2-4 <XAdEST -ATTACH-SIGTIME-REVOKED-OK 10004>におけるテスト結果の期待値とテストパラメータ

期待値	有効（valid）
署名を実施したとする時刻	2001.1.1 12:00:00
サイニングタイム属性の時刻	2001.1.4 12:00:00
署名タイムスタンプの時刻	2001.1.1 12:00:00
署名者証明書の有効期限	2001.1.1～2035.12.31
署名者証明書の検証 CRL	2001.1.3 00:00:00～2001.1.3 23:59:59
署名者証明書 CRL 中失効日時	2005.1.2 12:00:00
署名タイムスタンプ TSA 証明書	2001.1.1～2035.12.31
署名TS TSA証明書検証CRL	2001.1.3 00:00:00～2001.1.3 23:59:59

2.4.5 <XAdEST -ATTACH-SIGTS-REVOKED-NG 10005>

署名者証明書、署名タイムスタンプの TSA 証明書が有効期間内にあり、SigningTime 属性の時刻では失効していないが、署名タイムスタンプ時刻において署名者証明書が失効して CRL に記載されている場合、署名時刻は無視し、署名タイムスタンプ時刻での証明書有効性を検証すること

により、ES-T データが無効であることを検証する。

表 付 4.2-5 <XAdEST -ATTACH-SIGTS-REVOKED-NG 10005>におけるテスト結果の期待値とテストパラメータ

期待値	有効（invalid）
署名を実施したとする時刻	2001.1.1 12:00:00
サイニングタイム属性の時刻	2001.1.1 12:00:00
署名タイムスタンプの時刻	2001.1.3 12:00:00
署名者証明書の有効期限	2001.1.1～2035.12.31
署名者証明書の検証 CRL	2001.1.4 00:00:00～2001.1.4 23:59:59
署名者証明書 CRL 中失効日時	2005.1.2 12:00:00
署名タイムスタンプ TSA 証明書	2001.1.1～2035.12.31
署名TS TSA証明書検証CRL	2001.1.4 00:00:00～2001.1.4 23:59:59

2.4.6 <XAdEST -ATTACH-ES-SIG-REVOKED-NG 10006>

ES-T フォーマットの CMS SignedData の SIgnerInfo において signature フィールドにある署名値が改ざんされていた場合に無効であることを検証する。

表 付 4.2-6 <XAdEST -ATTACH-EE-SIG-FORGED-NG 10006>におけるテスト結果の期待値とテストパラメータ

期待値	有効（invalid）
署名を実施したとする時刻	2001.1.1 12:00:00
サイニングタイム属性の時刻	属性なし
署名タイムスタンプの時刻	2001.1.1 12:00:00
署名者証明書の有効期限	2001.1.1～2035.12.31
署名者証明書の検証 CRL	2001.1.2 00:00:00～2002.1.4 23:59:59
署名タイムスタンプ TSA 証明書	2001.1.1～2035.12.31
署名TS TSA証明書検証CRL	2001.1.2 00:00:00～2001.1.2 23:59:59

2.4.7 <XAdEST -ATTACH-SIGTS-SIG-FORGED-NG 10007>

ES-T フォーマットの SignatureTimeStamp 属性中の TimeStampToken の CMS SignedData 構造の SignerInfo において signature フィールドにある署名値が改ざんされていた場合に無効であることを検証する。

表 付 4.2-7 <XAdEST -ATTACH-SIGTS-SIG-FORGED-NG 10007>におけるテスト結果の期待値とテストパラメータ

期待値	無効（invalid）
署名を実施したとする時刻	2001.1.1 12:00:00
サイニングタイム属性の時刻	属性なし
署名タイムスタンプの時刻	2001.1.1 12:00:00

署名者証明書の有効期限	2001.1.1～2035.12.31
署名者証明書の検証 CRL	2001.1.2 00:00:00～2001.1.2 23:59:59
署名タイムスタンプ TSA 証明書	2001.1.1～2035.12.31
署名TS TSA証明書検証CRL	2001.1.2 00:00:00～2001.1.2 23:59:59

2.4.8 <XAdEST -ATTACH-ES-MESSAGEDIGEST-FORGED-NG 10008>

ES-T フォーマットの CMS SignedData の signedAttributes の中の MessageDigest 属性の値が改ざんされていた場合に無効であることを検証する。

表 付 4.2-8 <XAdEST -ATTACH-ES-MESSAGEDIGEST-FORGED-NG 10008>におけるテスト結果の期待値とテストパラメータ

期待値	無効（invalid）
署名を実施したとする時刻	2001.1.1 12:00:00
サイニングタイム属性の時刻	属性なし
署名タイムスタンプの時刻	2001.1.1 12:00:00
署名者証明書の有効期限	2001.1.1～2035.12.31
署名者証明書の検証 CRL	2001.1.2 00:00:00～2001.1.2 23:59:59
署名タイムスタンプ TSA 証明書	2001.1.1～2035.12.31
署名TS TSA証明書検証CRL	2001.1.2 00:00:00～2001.1.2 23:59:59

2.4.9 <XAdEST -ATTACH-SIGTSTST-MESSAGEDIGEST-FORGED-NG 10009>

ES-T フォーマットの SignatureTimeStamp 属性に含まれるタイムスタンプトークンの signedAttributes の中の MessageDigest 属性の値が改ざんされていた場合に無効であることを検証する。

表 付 4.2-9 <XAdEST -ATTACH-SIGTSTST-MESSAGEDIGEST-FORGED-NG 10009>におけるテスト結果の期待値とテストパラメータ

期待値	無効（invalid）
署名を実施したとする時刻	2001.1.1 12:00:00
サイニングタイム属性の時刻	属性なし
署名タイムスタンプの時刻	2001.1.1 12:00:00
署名者証明書の有効期限	2001.1.1～2035.12.31
署名者証明書の検証 CRL	2001.1.2 00:00:00～2001.1.2 23:59:59
署名タイムスタンプ TSA 証明書	2001.1.1～2035.12.31
署名TS TSA証明書検証CRL	2001.1.2 00:00:00～2001.1.2 23:59:59

2.4.10 <XAdEST -DETACH-NORMAL-OK 10010>

署名対象文書に対して分離署名を行った ES-T フォーマットにおいてデータが有効であることを検証する。

表 付4.2-10 <XAdEST -DETACH-NORMAL-OK 10010>におけるテスト結果の期待値とテストパラメータ

期待値	有効（valid）
署名を実施したとする時刻	2001.1.1 12:00:00
サイニングタイム属性の時刻	属性なし
署名タイムスタンプの時刻	2001.1.1 12:00:00
署名者証明書の有効期限	2001.1.1～2035.12.31
署名者証明書の検証 CRL	2001.1.2 00:00:00～2001.1.2 23:59:59
署名タイムスタンプ TSA 証明書	2001.1.1～2035.12.31
署名TS TSA証明書検証CRL	2001.1.1 00:00:00～2001.1.2 23:59:59

2.5 ES-A フォーマット標準テスト

2.5.1 <XAdESA1-ATTACH-NORMAL-OK 70001>

ECOM XAdES 長期署名フォーマットプロファイルに基づくアーカイブタイムスタンプを一つ付与された ES-A フォーマットが有効であることを検証する。

表 付4.2-11 <XAdESA 1-ATTACH-NORMAL-OK 70001>におけるテスト結果の期待値とテストパラメータ

期待値	有効（valid）
署名を実施したとする時刻	2001.1.1 12:00:00
署名タイムスタンプの時刻	2001.1.1 12:00:00
署名者証明書の有効期限	2001.1.1～2035.12.31
署名者証明書の検証 CRL	2001.1.2 00:00:00～2001.1.2 23:59:59
署名タイムスタンプ TSA 証明書	2001.1.1～2035.12.31
署名TS TSA 証明書検証CRL	2001.1.3 00:00:00～2001.1.3 23:59:59
Archiveタイムスタンプ1の時刻	2001.1.3 12:00
Archiveタイムスタンプ1のTSA証明書	2001.1.1～2035.12.31
Archive TS TSA証明書検証CRL	2001.1.4 00:00:00～2035.12.31 23:59:59

2.5.2 <XAdESA1-DETACH-NORMAL-OK 70002>

DETACHED 形式の署名を行った XML 署名に対し ECOM XAdES 長期署名フォーマットプロファイルに基づくアーカイブタイムスタンプを一つ付与された ES-A フォーマットが有効であることを検証する。

表 付4.2-12 <XAdESA 1-DETACH-NORMAL-OK 70002>におけるテスト結果の期待値とテストパラメータ

期待値	有効（valid）
署名を実施したとする時刻	2001.1.1 12:00:00
署名タイムスタンプの時刻	2001.1.1 12:00:00

署名者証明書の有効期限	2001.1.1～2035.12.31
署名者証明書の検証 CRL	2001.1.2 00:00:00～2001.1.2 23:59:59
署名タイムスタンプ TSA 証明書	2001.1.1～2035.12.31
署名TS TSA証明書検証CRL	2001.1.3 00:00:00～2001.1.3 23:59:59
Archiveタイムスタンプ1の時刻	2001.1.3 12:00
Archiveタイムスタンプ1のTSA証明書	2001.1.1～2035.12.31
Archive TS TSA証明書検証CRL	2001.1.4 00:00:00～2035.12.31 23:59:59

2.6 XAdES-T 標準テストケース

本節では XAdES-T フォーマットを扱う実装が満足すべきテストケースを示す。

2.6.1 <OFF-T-1>

テストケース名	OFF-T-1
一般的な内包署名の ES-T フォーマットを読み込むことができる。	
成功条件：以下テスト項目が全て期待値通りになること。	
10001	XAdEST-ATTACH-NORMAL-OK

2.6.2 <OFF-T-2>

テストケース名	OFF-T-2
XAdES-T フォーマットの署名者証明書の期限切れを扱える	
成功条件：以下テスト項目が全て期待値通りになること。	
10001	XAdEST-ATTACH-NORMAL-OK
10002	XAdEST-ATTACH-EXPIRED-NG

2.6.3 <OFF-T-3>

テストケース名	OFF-T-3
XAdES-T フォーマットの署名者証明書の失効を扱える	
成功条件：以下テスト項目が全て期待値通りになること。	
10001	XAdEST-ATTACH-NORMAL-OK
10003	XAdEST-ATTACH-REVOKED-NG

2.6.4 <OFF-T-4>

テストケース名	OFF-T-4
XAdES-T フォーマットの署名者証明書の認証パス検証を正しく扱える	
成功条件：以下テスト項目が全て期待値通りになること。	
10001	XAdEST-ATTACH-NORMAL-OK
10002	XAdEST-ATTACH-EXPIRED-NG

10003	XAdEST-ATTACH-REVOKED-NG

2.6.5 <OFF-T-5>

テストケース名	OFF-T-5
XAdES-T フォーマットでサイニングタイムに関係なく署名タイムスタンプの時刻により署名者証明書の失効検証ができる。	
成功条件：以下テスト項目が全て期待値通りになること。	
10001	XAdEST-ATTACH-NORMAL-OK
10002	XAdEST-ATTACH-EXPIRED-NG
10003	XAdEST-ATTACH-REVOKED-NG
10004	XAdEST-ATTACH-SIGTIME-REVOKED-OK
10005	XAdEST-ATTACH-SIGTS-REVOKED-NG

2.6.6 <OFF-T-6>

テストケース名	OFF-T-6
XAdES-T フォーマットの Signature 要素内の署名値の改ざんを検知できる。	
成功条件：以下テスト項目が全て期待値通りになること。	
10001	XAdEST-ATTACH-NORMAL-OK
10006	XAdEST-ATTACH-ES-SIG-FORGED-NG

2.6.7 <OFF-T-7>

テストケース名	OFF-T-7
XAdES-T フォーマットの署名タイムスタンプトークンの SignerInfo の署名値の改ざんを検知できる。	
成功条件：以下テスト項目が全て期待値通りになること。	
10001	XAdEST-ATTACH-NORMAL-OK
10007	XAdEST-ATTACH-SIGTS-FORGED-NG

2.6.8 <OFF-T-8>

テストケース名	OFF-T-8
XAdES-T フォーマットの DigestValue 要素のハッシュ値の改ざんを検知できる。	
成功条件：以下テスト項目が全て期待値通りになること。	
10001	XAdEST-ATTACH-NORMAL-OK
10008	XAdEST-ATTACH-ES-MESSAGEDIGEST-FORGED-NG

2.6.9 <OFF-T-9>

テストケース名	OFF-T-9
XAdES-T フォーマットの署名タイムスタンプトークンの MessageDigest のハッシュ値の改ざんを検知できる。	
成功条件：以下テスト項目が全て期待値通りになること。	
10001	XAdEST-ATTACH-NORMAL-OK
10009	XAdEST-ATTACH-SIGTSTST-MESSAGEDIGEST-FORGED-NG

2.6.10 <OFF-T-10>

テストケース名	OFF-T-10
分離署名の XAdES-T のフォーマットを扱える	
成功条件：以下テスト項目が全て期待値通りになること。	
10010	XAdEST-DETACH-NORMAL-OK

2.7 XAdES-A 標準テストケース

2.7.1 <OFF-A-1>

テストケース名	OFF-A-1
ECOM プロファイルに基づく内包署名の第一世代の ES-A フォーマットを扱うことができる。	
成功条件：以下テスト項目が全て期待値通りになること。	
70001	XAdESA1-ATTACH-NORMAL-OK

2.7.2 <OFF-A-2>

テストケース名	OFF-A-2
ECOM プロファイルに基づく分離署名の第一世代の ES-A フォーマットを扱うことができる。	
成功条件：以下テスト項目が全て期待値通りになること。	
70001	XAdESA1-DETACH-NORMAL-OK

3．オンライン マトリックス生成・相互検証テストカテゴリ

長期署名フォーマットを扱う製品を持つ実証実験参加組織が、指定されたレギュレーションに基づく長期書名データファイルをそれぞれ生成し、このデータが有効であることを各製品が検証できることを確認するテストである。

3.1 生成するデータ

署名対象となるデータは小さいサイズのテキストデータを 1MB のバイナリデータを用意する。

小さいサイズのテキストデータを Enveloping 形式、1MB 程度のバイナリデータファイルを Detached 形式として XML 署名を生成し、それぞれ ES-T フォーマット、第一世代とその次の世代の ES-A フォーマットを生成する。

タイムスタンプトークンの取得は今回のテスト用に提供されたタイムスタンプ局を使用することとする。失効情報の取得は証明書の cRLDistributionPoints 拡張に記載された URL より取得してもよいし、テストデータに含まれるファイルを使用してもよい。

3.2 テストの準備

テスト実施する際は、以下の項目の準備が必要となる。

- CRL の設定
  CRL を取得するために、検証環境におけるインターネット接続環境の準備を行う。署名用認証局とタイムスタンプ局用認証局の CRL 発行間隔は 1 日となっている。
- トラストアンカの設定
  テストスイートで配布されるオフラインテスト用の署名者用ルート証明書、TSA 用ルート証明書をトラストアンカとして設定する。

3.3 テストの実施（生成）

本節では、テストの実施時の署名データ生成側の設定や条件などを説明する。

- 署名対象データの設定
  内包型署名の場合は、署名対象文字列を“aaa”とし XML 署名の形式として enveloping 形式で署名対象文字列を指定する。ただし、XML 署名の Object 要素として格納するため、base64 で encode された値（YWFh）で格納するもとする。内包型署名の場合の XML 署名文書の例を以下に示す（リスト 2）。

リスト 2：内包型署名の場合の XML 署名文書の例

```
<?xml version="1.0" encoding="UTF-8" ?>
<ds:Signature Id="Signature-ID1" xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
  <ds:SignedInfo> ……</ds:SignedInfo>
  <ds:SignatureValue>……</ds:SignatureValue>
  <ds:KeyInfo >……</ds:KeyInfo >
  <ds:Object Encoding="http://www.w3.org/2000/09/xmldsig#base64"
             Id="signdata">YWFh</ds:Object>
  <ds:Object xmlns:xa="http://uri.etsi.org/01903/v1.3.1#">……</ ds:Object >
</ds:Signature>
```

base64 でエンコードされたテキスト”aaa”

分離署名の場合には、'TARGET_BBB.bin”（ファイルの内容は、0x01-0x09,0x00 の繰り返しで 1024000 バイトのバイナリファイル）を設定する。

- データ生成の実施
- 生成したデータを全てアーカイブし、参加企業の検証者に送信する。

3.4 テストの実施（検証）

- 署名対象データの設定
  内包型署名の場合は、署名対象文字列を“aaa”としXML署名の形式としてenveloping形式で署名対象文字列を指定されている。したがって、XML 署名の仕様に従い検証できれば良い。分離署名の場合には、'TARGET_BBB.bin”（ファイルの内容は、0x01-0x09,0x00の繰り返しで1024000バイトのバイナリファイル）が署名対象となる。署名対象ファイルの指定方法として以下の2通りが考えられるので両社とも検証できる必要がある。
  - ✧ Reference要素のURI属性で明示的に署名対象要素を参照している場合
  - ✧ Reference 要素では、明示的に署名対象要素が指定されておらず、別途署名対象ファイルを指定する必要がある場合。

- 検証時刻の設定
  検証時刻はフォーマット毎に異なる。フォーマットに応じ検証時刻の設定を行う。検証可能な現在時刻の範囲はUTC2002年1月1日0時0分0秒よりUTC2035年12月31日23時59分59秒までとし、各証明書やCRLもまたこれが可能なように設定されている。
- 検証対象の長期署名フォーマットデータの設定
  データ生成より受け取った長期署名フォーマットの検証対象テストデータを検証する。
- 検証の実施
  実施すべき全てのテスト項目について実施する。署名対象データのハッシュ値を base64でエンコードしたものは以下の通り。
  "aaa" ：**fiQN5O+x7Qj6CNOAY/amqRRiqBU=**
  TARGET_BBB.bin ：**gpGOa0wroxRJGyeXw7tHFbrgtxM=**

3.5 テストケース

以下のレギュレーションに関する記述で「◎」印は必ず満足しなければならないルールとし、それ以外は可能ならば準拠しなければならないルールとする。

3.5.1 <ON-T-1>Enveloped形式 XAdES-T生成・相互検証テストケース

以下のレギュレーションで各製品やサービスのXAdES-Tデータを生成する。

- ◎署名対象文字列“aaa”
- ◎内包証明とする。(XML内部にaaaを含みそれを署名対象とする。)
- DigestMethodはSHA1
- 署名アルゴリズムはSHA1withRSA
- XAdES-BESフォーマットよりXAdES-Tを生成

各製品やシステムは、これらの条件で生成されたデータが有効であることを検証する。

3.5.2 <ON-T-2>Detached 形式 XAdES-T 生成・相互検証テストケース

テストケース<ON-T-1>をベースに以下のレギュレーションを加えたもので各製品やシステムで XAdES-T データを生成する。

- ◎署名対象が 1MB のデータファイル
- ◎分離署名とする。XML 署名文書に署名対象を含まない。

各製品やシステムは、これらの条件で生成されたデータが有効であることを検証する。

3.5.3 <ON-A1-1>Enveloped 形式 第 1 世代 XAdES-A 生成・相互検証テストケース

以下のレギュレーションで各製品やシステムで XAdES-A データを生成する。

- ◎<ON-T-1>で生成された XAdES-T データを対象として XAdES-A データを生成。
- 署名、およびアーカイブタイムスタンプ属性の証明書検証情報の格納内容、方法は ECOM プロファイルに基づく

各製品やシステムは、これらの条件で生成されたデータが有効であることを検証する。

3.5.4 <ON-A1-2>Detached 形式 第 1 世代 XAdES-A 生成・相互検証テストケース

以下のレギュレーションで各製品やシステムで XAdES-A データを生成する。

- ◎<ON-T-2>で生成された XAdES-T データを対象として XAdES-A を生成する。
- 署名、およびアーカイブタイムスタンプ属性の証明書検証情報の格納内容、方法は ECOM プロファイルに基づく

各製品やシステムは、これらの条件で生成されたデータが有効であることを検証する。

3.5.5 <ON-A2-1>Enveloped 形式 第 2 世代 XAdES-A 生成・相互検証テストケース

以下のレギュレーションで各製品やシステムで XAdES-A を生成する。

- ◎<ON-A1-1>で生成された XAdES-A を対象とし生成する（署名延長）
- 署名、およびアーカイブタイムスタンプ属性の証明書検証情報の格納内容、方法は ECOM プロファイルに基づく

各製品やシステムは、これらの条件で生成されたデータが有効であることを検証する。

3.5.6 <ON-A2-2>Detached 形式 第 2 世代 XAdES-A 生成・相互検証テストケース

以下のレギュレーションで各製品やシステムで XAdES-A を生成する。

- ◎<ON-A1-2>で生成された XAdES-A を対象とし生成する（署名延長）
- 署名、およびアーカイブタイムスタンプ属性の証明書検証情報の格納内容、方法は ECOM プロファイルに基づく

各製品やシステムは、これらの条件で生成されたデータが有効であることを検証する。

4．付録：実験データ用プロファイル

本節では実証実験で用いられるデータのプロファイルを示す。なお、証明書およびタイムスタンプトークンのプロファイルはCAdESの実験に利用したものを用いる。

4.1　実験用長期署名フォーマットデータプロファイル

長期署名フォーマットのデータは、全てXAdESの仕様に基づいている。

4.1.1　XAdES-BES

要素	内容
ds:Signature	
ds:SignedInfo	あり
ds:CanonicalizationMethod	Canonical XML(REC-xml-c14n-20010315)
ds:SignatureMethod	RSAwithSHA1(http://www.w3.org/2000/09/xmldsig#rsa-sha1)
ds:Reference	複数の場合も考慮する（署名形式はdetached形式とする）
ds:Transforms	署名対象がXMLの場合は、Canonical XMLを使用する。
ds:DigestMethod	http://www.w3.org/2000/09/xmldsig#sha1
ds:DigestValue	署名対象文書のダイジェスト値
ds:SignatureValue	署名値
ds:KeyInfo	ECOMプロファイルに従う。
ds:Object	有り(SingingCertificateの有無に依存する)
QualifyingProperties	有り(SingingCertificateの有無に依存する)
SignedProperties	有り(SingingCertificateの有無に依存する)
SignedSignatureProperties	有り(SingingCertificateの有無に依存する)
SigningCertificate	ECOMプロファイルに従う（発行者名、シリアル番号、SHA1フィンガープリント）

※：テスト項目により値は変化するがデータ共通の値はこの値となる。

4.1.2 XAdES-T

要素	内容
ds:Signature	
ds:SignedInfo	あり
ds:CanonicalizationMethod	Canonical XML(REC-xml-c14n-20010315)
ds:SignatureMethod	RSAwithSHA1(http://www.w3.org/2000/09/xmldsig#rsa-sha1)
ds:Reference	複数の場合も考慮する（署名形式はdetached形式とする）
ds:Transforms	署名対象文書のフォーマットに依存する。 署名対象がXMLの場合は、Canonical XMLを使用する。
ds:DigestMethod	http://www.w3.org/2000/09/xmldsig#sha1
ds:DigestValue	署名対象文書のダイジェスト値
ds:SignatureValue	署名値
ds:KeyInfo	ECOMプロファイルに従う。
ds:Object	有り
QualifyingProperties	有り
SignedProperties	有り(SingingCertificateの有無に依存する)
SignedSignatureProperties	有り(SingingCertificateの有無に依存する)
SigningCertificate	ECOMプロファイルに従う（発行者名、シリアル番号、SHA1フィンガープリント）
UnSignedProperties	有り
UnSignedSignatureProperties	有り
SignatureTimeStamp	トークンは実験用データプロファイルに従う

4.1.3 XAdES-A（第一世代）

要素						内容
ds:Signature						
	ds:SignedInfo					あり
		ds:CanonicalizationMethod				Canonical XML(REC-xml-c14n-20010315)
		ds:SignatureMethod				/xmldsig#rsa-sha1)
		ds:Reference				detached形式とする)
			ds:Transforms			署名対象がXMLの場合は、Canonical XMLを使用する。
			ds:DigestMethod			http://www.w3.org/2000/09/xmldsig#sha1
			ds:DigestValue			署名対象文書のダイジェスト値
	ds:SignatureValue					署名値
	ds:KeyInfo					ECOMプロファイルに従う。
	ds:Object					有り
		QualifyingProperties				有り
			SignedProperties			有り(SingingCertificateの有無に依存する)
				SignedSignatureProperties		有り(SingingCertificateの有無に依存する)
					SigningCertificate	ECOMプロファイルに従う（発行者名、シリアル番号、SHA1フィンガープリント)
			UnSignedProperties			有り
				UnSignedSignatureProperties		有り
					SignatureTimeStamp	トークンは実験用データプロファイルに従う
					CompleteCertificateRefs	ECOMプロファイルに従う
					s	ECOMプロファイルに従う
					CertificateValues	ECOMプロファイルに従う
					RevocationValues	ECOMプロファイルに従う
					ArchiveTimeStamp	トークンは実験用データプロファイルに従う

※：テスト項目により値は変化するがデータ共通の値はこの値となる。

4.1.4 XAdES-A（第２世代）

要素	内容
ds:Signature	
ds:SignedInfo	あり
ds:CanonicalizationMethod	Canonical XML(REC-xml-c14n-20010315)
ds:SignatureMethod	RSAwithSHA1(http://www.w3.org/2000/09/xmldsig#rsa-sha1)
ds:Reference	複数の場合も考慮する（署名形式はdetached形式とする）
ds:Transforms	署名対象文書のフォーマットに依存する。 署名対象がXMLの場合は、Canonical XMLを使用する。
ds:DigestMethod	http://www.w3.org/2000/09/xmldsig#sha1
ds:DigestValue	署名対象文書のダイジェスト値
ds:SignatureValue	署名値
ds:KeyInfo	ECOMプロファイルに従う。
ds:Object	有り
QualifyingProperties	有り
SignedProperties	有り
SignedSignatureProperties	有り
SigningCertificate	ECOMプロファイルに従う（発行者名、シリアル番号、SHA1フィンガープリント）
UnSignedProperties	有り
UnSignedSignatureProperties	有り
SignatureTimeStamp	トークンは実験用データプロファイルに従う
CompleteCertificateRefs	ECOMプロファイルに従う
CompleteRevocationRefs	ECOMプロファイルに従う
CertificateValues	ECOMプロファイルに従う
RevocationValues	ECOMプロファイルに従う
ArchiveTimeStamp	トークンは実験用データプロファイルに従う
ArchiveTimeStamp	トークンは実験用データプロファイルに従う

※：テスト項目により値は変化するがデータ共通の値はこの値となる。

メンバーリスト

事務局

前田　陽二	次世代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
米丸　恒治	神戸大学

リーダ

木村　道弘	日本電気株式会社
宮崎　一哉	三菱電機株式会社
（実験プロジェクトリーダ）	
漆嶌　賢二	エントラストジャパン株式会社

編集メンバー（上記リーダ以外）

氏名	所属
後藤　　淳	日本電気株式会社
政本　廣志	日本電信電話株式会社
谷川　嘉伸	株式会社日立製作所
溝上　卓也	日立ソフトウェアエンジニアリング株式会社

メンバー（上記以外）

氏名	所属
出本　　浩	株式会社エヌ・ティ・ティ・データ
保倉　　豊	グローバルフレンドシップ株式会社
佐藤　雅史	セコム株式会社
和田　宗樹	株式会社帝国データバンク
石原　達也	東芝ソリューション株式会社
大窪　伸幸	株式会社 PFU
時得　克司	富士ゼロックス（株）
小谷　誠剛	富士通株式会社
斎藤　幹男	富士電機ホールディングス株式会社
北島　郁夫	松下電器産業株式会社
木沢　　誠	松下電器産業株式会社
土手　祐典	三菱電機株式会社
西谷　研次	株式会社 UFJ 銀行
佐藤　孝一	（株）中電シーティーアイ

長期署名フォーマット相互運用性実験報告書

平成 18 年 3 月発行

発　行　　次世代電子商取引推進協議会

発行所　　財団法人　日本情報処理開発協会
　　電子商取引推進センター
　　東京都港区芝公園三丁目 5 番 8 号
　　機械振興会館　3 階
　　TEL：03(3436)7500

この資料は再生紙を使用しています。